

# Journal Of Computer And Information Sciences



SAKARYA UNIVERSITY

e-ISSN 2636-8129

VOLUME 5

ISSUE 3

DECEMBER 2022

*Prediction of Unknown Terrorist Group Names Responsible for Attacks in Turkey*  
*Using of Hierarchical Loglinear Model in Multiway Frequency Tables and an Application on Suicide Cases*  
*A Review of Recent Developments on Secure Authentication using RF Fingerprints Techniques*  
*FA-AODV: Flooding Attacks Detection Based Ad Hoc On-Demand Distance Vector Routing Protocol for VANET*  
*The Effect of Numerical Mapping Techniques on Performance in Genomic Research*  
*Process Mining in Manufacturing: A Literature Review*  
*Software Development for the Use of Generalized Parabolic Blending in Data Prediction Processes*



www.saucis.sakarya.edu.tr

*Using Multi-Label Classification Methods to Analyze Complaints Against Cargo Services During the COVID-19 Outbreak: Comparing Survey-Based and Word-Based Labeling*  
*Pseudo-Supervised Defect Detection Using Robust Deep Convolutional Autoencoders*  
*Simulation of Cargo Unloading Problem: A Case Study on Estimating the Optimal Number of Trucks and Cranes*  
*Sector-Based Stock Price Prediction with Machine Learning Models*  
*A Deep Transfer Learning-Based Comparative Study for Detection of Malaria Disease*  
*A Decision Support System For Detecting Stage In Hodgkin Lymphoma Patients Using Artificial Neural Network and Optimization Algorithms*  
*Automatic Classification of White Blood Cells Using Pre-Trained Deep Models*



# SAUCIS

**Sakarya University Journal of Computer and Information  
Sciences Volume: 5 – Issue No: 3 (December 2022)**  
<http://saucis.sakarya.edu.tr/issue/74792>

## Editor in Chief

Nejat Yumuşak, Sakarya University, nyumusak@sakarya.edu.tr

## Associate Editors

İhsan Hakan Selvi, Sakarya University, Turkey, ihselvi@sakarya.edu.tr

Muhammed Fatih Adak, Sakarya University, Turkey, fatihadak@sakarya.edu.tr

Mustafa Akpınar, Sakarya University, Turkey, akpınar@sakarya.edu.tr

Unal Cavusoglu, Sakarya University, Turkey, unalc@sakarya.edu.tr

Veysel Harun Sahin, Sakarya University, Turkey, vsahin@sakarya.edu.tr

## Editorial Assistants - Secretary

Deniz Balta, Sakarya University, Turkey, ddural@sakarya.edu.tr

Fatma Akalin, Sakarya University, Turkey, fatmaakalin@sakarya.edu.tr

Gozde Yolcu Oztel, Sakarya University, Turkey, gyolcu@sakarya.edu.tr

Ibrahim Delibasoglu, Sakarya University, Turkey, ibrahimdelibasoglu@sakarya.edu.tr

Muhammed Kotan, Sakarya University, Turkey, mkotan@sakarya.edu.tr

Sumeyye Kaynak, Sakarya University, Turkey, sumeyye@sakarya.edu.tr

Ahmet Erhan Tanyeri, Sakarya University, Turkey, tanyeri@sakarya.edu.tr

## Editorial Board

Ahmet Ozmen, Sakarya University, Turkey, ozmen@sakarya.edu.tr

Aref Yelghi, Istanbul Ayvansaray University, ar.yelqi@gmail.com

Ayhan Istanbulu, Balikesir University, Turkey, iayhan@balikesir.edu.tr

Aysegul Alaybeyoglu, Izmir Katip Celebi University, Turkey, alaybeyoglu@gmail.com

Bahadir Karasulu, Canakkale Onsekiz Mart University, bahadirkarasulu@comu.edu.tr

Celal Ceken, Sakarya University, Turkey, celalceken@sakarya.edu.tr

Cihan Karakuzu, Bilecik Seyh Edebali University, cihan.karakuzu@bilecik.edu.tr

Fahri Vatansever, Bursa Uludag University, fahriv@uludag.edu.tr

Ibrahim Turkoglu, Firat University, Turkey, iturkoglu@firat.edu.tr

Levent Alhan, Sakarya University, Turkey, leventalhan@sakarya.edu.tr

Kamal Z Zamli, Malaysia Pahang University, Malaysia, kamalz@ump.edu.my

Muhammed Fatih Adak, Sakarya University, Turkey, fatihadak@sakarya.edu.tr

Mustafa Akpınar, Sakarya University, Turkey, akpınar@sakarya.edu.tr



# SAUJCIS



## Editorial Board (Cont.)

Nuri Yilmazer, Texas A&M University, US, [nuri.yilmazer@tamuk.edu](mailto:nuri.yilmazer@tamuk.edu)

Nejat Yumuşak, Sakarya University, [nyumusak@sakarya.edu.tr](mailto:nyumusak@sakarya.edu.tr)

Orhan Er, Bozok University, Turkey, [orhan.er@bozok.edu.tr](mailto:orhan.er@bozok.edu.tr)

Priyadip Ray, Lawrence Livermore National Laboratory, [priyadipr@gmail.com](mailto:priyadipr@gmail.com)

Resul Das, Firat University, Turkey, [rdas@firat.edu.tr](mailto:rdas@firat.edu.tr)

Veysel Harun Sahin, Sakarya University, Turkey, [vsahin@sakarya.edu.tr](mailto:vsahin@sakarya.edu.tr)



# SAUCIS

Sakarya University Journal of Computer and Information Sciences  
Volume: 5 – Issue No: 3 (December 2022)  
<http://saucis.sakarya.edu.tr/issue/74792>

## Contents

Author(s), Paper Title	Pages
<i>Ibrahim A. Fadel, Cemil Öz</i> Prediction of Unknown Terrorist Group Names Responsible for Attacks in Turkey	257-268
<i>Fatih Üçkardeş</i> Using of Hierarchical Loglinear Model in Multiway Frequency Tables and an Application on Suicide Cases	269-277
<i>Hüseyin Parmaksız, Cihan Karakuzu</i> A Review of Recent Developments on Secure Authentication using RF Fingerprints Techniques	278-303
<i>Bugra Alp Tosunoglu, Cemal Kocak</i> FA-AODV: Flooding Attacks Detection Based Ad Hoc On-Demand Distance Vector Routing Protocol for VANET	304-314
<i>Seda Nur Gülocak, Bihter Daş</i> The Effect of Numerical Mapping Techniques on Performance in Genomic Research	315-340
<i>Yüksel Yurtay</i> Process Mining in Manufacturing: A Literature Review	341-355
<i>Hakan Üstünel</i> Software Development for the Use of Generalized Parabolic Blending in Data Prediction Processes	356-370
<i>Tolga Kuyucuk, Levent Çallı</i> Using Multi-Label Classification Methods to Analyze Complaints Against Cargo Services During the COVID-19 Outbreak: Comparing Survey-Based and Word-Based Labeling	371-384
<i>Mahmut Nedim Alpdemir</i> Pseudo-Supervised Defect Detection Using Robust Deep Convolutional Autoencoders	385-403
<i>Waseem Hamdoon, Ahmet Zengin</i> Simulation of Cargo Unloading Problem: A Case Study on Estimating the Optimal Number of Trucks and Cranes	404-414
<i>Doğangün Kocaoğlu, Korhan Turgut, Mehmet Zeki Konyar</i> Sector-Based Stock Price Prediction with Machine Learning Models	415-426

<i>Emel Soylu</i> A Deep Transfer Learning-Based Comparative Study for Detection of Malaria Disease	427-447
<i>Fatma Akalin, Mehmet Fatih Orhan, Mustafa Büyükavcı</i> A Decision Support System For Detecting Stage In Hodgkin Lymphoma Patients Using Artificial Neural Network and Optimization Algorithms	448-461
<i>Oguzhan Katar, Ilhan Firat Kilincer</i> Automatic Classification of White Blood Cells Using Pre-Trained Deep Models	462-476

# Prediction of Unknown Terrorist Group Names Responsible for Attacks in Turkey

 Ibrahim A. Fadel<sup>1</sup>,  Cemil Öz<sup>2</sup>

<sup>1</sup>Corresponding Author; Dept. of Computer Engineering, Sakarya University; ibrahim.fadel@ogr.sakarya.edu.tr  
<sup>2</sup>Dept. of Computer Engineering, Sakarya University; coz@sakarya.edu.tr

Received 13 February 2021; Revised 9 June 2022; Accepted 23 August 2022; Published online 31 December 2022

## Abstract

In this paper, the dataset of real incidents that occurred in Turkey between 2013 and 2017 and are regarded as acts of terrorism without any doubt, according to Global Terrorism Database (GTD) is used to predict the group names responsible for unknown attacks. Principal Component Analysis (PCA) technique was used for feature selection. A novel voting method between five classification algorithms such as Random Forests, Logistic Regression, AdaBoost, Neural Network, and Support Vector Machine was used to predict the names. The results clearly demonstrate that the classification accuracy of all classifiers studied in this paper improved when PCA was used to select features as compared to selecting features without using PCA. The prediction of terrorist group names with PCA based feature reduction and the original features is carried out and the results are compared.

**Keywords:** prediction, classification, GTD dataset, PCA

## 1. Introduction

Since the September 11 terror attack, terrorism has become a global phenomenon and terrorist attacks are leading issues today and have become a focal point of concentration for different communities in the world. The term ‘terrorism’ is defined by the Central Intelligence Agency, U.S. state department and Department of defense as “premeditated, politically motivated violence perpetrated against noncombatant targets by subnational groups or clandestine agents, usually intended to influence an audience”[1].

This phenomenon is attracting the attention of various researches belonging to different organizations such as the National Union for the Study of Terrorism and Terrorism Responses (START). START is a division of the Center for Homeland Security of Excellence at the University of Maryland [2],[3] which monitors terrorist operations in the world and puts them in an open source database called Global Terrorism Database (GTD).

GTD is the most comprehensive base of operational information on terrorism in the world. The base contains information on terrorist events around the world from the year 1970 with annual updates. Based on this database, the Institute for Economics and Peace (IEP) publishes its annual Global Terrorism Index (GTI) on terrorism that assigns ranks to nations in the world according to the impact of terrorism.

According to the 2018 global terrorism index, The Middle East, to which Turkey belongs is reported to be the most affected region by terrorism. Turkey is ranked the 12th most affected country by terrorism in the world [4].

GTD dataset is used as source of the entire information related with the terrorist attacks examined in this work. From this dataset, terrorist attacks which occurred in Turkiye between 2013 and 2017 are referenced. The most active terrorist groups that are identified based on the dataset are Kurdistan Workers' Party (Partiya Karkerên Kurdistanê PKK), Islamic State of Iraq and the Levant (ISIL), Kurdistan Freedom Hawks (Teyrêbazên Azadiya Kurdistan TAK), Revolutionary People's Liberation Party/Front (Devrimci Halk Kurtuluş Partisi-Cephesi DHKP/C), Peace at Home Council (PHC), etc. However, there are a significant number of attacks which are not claimed by any of these known terrorist groups.

The rest of the paper is outlined as follows. Section 2 presents related works that have been done in the area. Our proposed methodology is presented in Section 3, Section 4 presents experiments, and the results obtained. Section 5 presents the results and discussion. The conclusion and future works are presented in Section 6.

## 2. Related Work

Prediction of terrorist groups after an attack is one of the most important steps for counter terrorism. As soon as we are able to find the involved group name, we will be able to make strategies to catch the culprits.

There is no international consensus on what counts as terrorism and what is not. However, Global Terrorism Index (GTI) defines terrorism as “the threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious, or social goal through fear, coercion, or intimidation.” Terrorism can be claimed by known terrorist groups, affiliated groups or individual terrorists. Terrorist attacks can have an enormous impact on wide sections of society [5].

The risks posed by a particular terror incident signify the magnitude of the act. In addition to major risks to people such as death, injuries and abduction, terrorism has a great effect on the economy too. Considering these and other minor disorders introduced due to the act of terrorists, responsible bodies such as law-enforcement agencies, police department and homeland security in general have to act accordingly. These acts which are generally known as counter terrorism may include community-based prevention and military operations. Counterterrorism requires understanding of the dynamic nature of terrorism, identification of patterns and determination of the magnitude of an attack, and consequently prioritizing the resources. However, the availability of large volumes of data related to terror makes manual analysis unimaginable. Traditional terrorist group name prediction techniques include email tracking, telephone signal information, and social network analytics [6]. These methods rely on manual analysis and are not efficient any more mainly because of the dynamic nature of terrorist groups and their actions. As important as it is, prediction requires more intelligent techniques which are reliable and can cope with the complexities associated with each terrorist acts. As is being effective in other prediction tasks, pattern recognition and machine learning techniques can be considered as a potential solution for terrorist group name prediction too. More importantly, prediction requires more intelligent techniques which are reliable and can cope with the complexities associated with each terrorist act. Communities from various fields have participated in one or more ways to provide tools that facilitate counter terrorism. Among these tools are crime category prediction [7], perpetrator prediction [6], [8], [9], [10], geographical and socioeconomic features [11],[12], future trend prediction and risk magnitude determination [11] tools. Machine learning techniques such as classification and clustering are the core for solutions provided by computer scientists and statisticians purposely for pattern discovery and therefore determine how far threatening a given terrorist group is.

There are a limited number of prior works relying on machine learning. However, these works report low accuracy and efficiency which can be attributed to feature redundancy and non-descriptiveness of datasets. Talreja et.al [6] proposed Factor Analysis of Mixed Data on the dataset to reduce the dimension of attributes and include only the twelve most prominent features to predict the perpetrators. Tolan and Soliman [9] studied five classification algorithms for terrorism prediction in Egypt. Their paper proposed mode-imputation and Litwise deletion approach to handle missing data and only six features were used based on manual, feature selection. Gohar et al [10] proposed a classification based approach for terrorist group prediction. In their work, it is stated that only seven attributes are selected for the classification task. However, there is no clear information on how these features are determined to be the most descriptive. Sachan and Roy [14] proposed a clustering based terrorist group prediction model which takes six attributes of an incident into account. Redundant features are removed and missing values are either deleted or edited based on other information sources. In their paper, there is no particular feature selection method, rather weight is assigned to each attribute representing its importance. The most

important features and their weights are determined through trial and error. Fatih et al. [15] proposed a crime prediction model that identifies and clusters incidents based on the similarity of attacks and attributes. Selection of the most important features involves the intervention of a domain expert. Python et al. [11] developed a model to predict terrorist attacks by training various machine learning models using data from GTD terrorist attacks committed between 2002 and 2016. They focused on geographic and socio-economic features to predict attacks on separate spatial time periods. Their results were impressive but inaccurate due to the wide geographical division because each region may contain different terrorist groups with different goals. In the study by Buffa et al. [12], the study target area was divided into hexagonal-grid cells of 25 square kilometers. They used five machine learning models and four spatial statistics to assess the validity of the results and improve inferences for the spatial processes between terrorist attacks. This analysis resulted in a Random Forest model that achieves an accuracy of 0.99 in predicting the presence or absence of terrorism, with a spatial accuracy of about 5 km. The results were validated by strong F1 and mean accuracy scores of 0.96 and 0.97, respectively. Inspired by the effectiveness of feature selection and dimension reduction, in this work, the redundant and non-descriptive features are identified and removed from the original dataset before training a given supervised learning algorithm. We proposed the application of PCA for feature selection and dimension reduction. The classification accuracies obtained from the resulting feature sets are better than state-of-the-art methodologies which rely on manual feature selection.

In GTD dataset, there are 1281 total incidents that occurred in Turkey between 2013 and 2017. In some cases, there is still an ambiguity to describe such incidents as terrorist attacks. In this work, we consider incidents where there is essentially no doubt as to whether they are acts of terrorism. Such incidents total to 890. From this total, 718 of them are identified by the entity that implemented them, which from now onwards are referred to as Known attacks. For example, an assailant opened fire on civilians celebrating the 2017 New Year outside Reina restaurant in Istanbul. At least 39 were killed and 69 injured including foreign tourists [16]. The assailant was arrested later and ISIL claimed responsibility of the attack.

Despite the terrorist groups' claim of their terrorist operations and their actions, there are still many terrorist attacks whose perpetrators have remained unknown. The remaining 172 instances (20%) of the 890 terrorist attacks in Turkey between 2013 and 2017 are not claimed by any group name, and as a result are called Unknown attacks. For example, the assailants opened fire on Ufuk Cafe in Istanbul in 02/01/2016, two people lost their lives and five others sustained injuries from the attack. No group claimed responsibility for the incident [17].

In this paper, we use classification algorithms to predict the names of the groups that might be responsible for such attacks. In order to predict the names of unknown attacks, we use the existing description of the known attacks. The GTD dataset describes each attack with 132 features, including the date and location of the incident, the weapons used, the nature of the target, and the number of casualties, etc. Some of the features in the dataset are redundant, consequently, they have no role in improving the prediction accuracy. At the same time, there are features which are completely irrelevant for the classification to be done. Such features need to be identified and removed before proceeding to the next stage. The process of determining and removing redundant, irrelevant features is termed as feature dimension reduction.

There are two principal algorithms for dimensionality reduction: Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The basic difference between these two is that LDA uses information of classes to find new features in order to maximize its separability while PCA uses the variance of each feature to do the same [18]. The idea behind PCA is simply to find a low-dimension set of axes that summarize data. [19] found that PCA can improve the predictive performance of machine learning algorithms in the classification of high dimensional data. In this paper, we use PCA to reduce the dimension of these features. We also evaluated the prediction accuracy to assess how representative the remaining features are.



The main contributions of this research are as follows:

1. Presents a tool that can assist law enforcement personnel to take advantage of the historical information of terrorist operations to identify potential terrorist organizations that may be the perpetrators of new attacks through the characteristics of their previous attacks. This will help the authorities to gain time in order to take the necessary measures to arrest the perpetrators quickly.
2. Predict the names of the terrorist groups that carried out a number of terrorist attacks in Turkey, which were classified as unknown.
3. Demonstrates the application of PCA to reduce data, limit dimensions and the utility of improving the performance of a variety of automated learning methods. In previous works, a certain number of attributes are selected and a focus is placed on the one-way learning method.
4. Describes a method of rating the classification of materials based on the difference between the total probability of the class assigned in all the algorithms compared to the total probability of other items, which means selection based on the collection of several different algorithms.

### 3. Proposed Technique

The proposed framework consists of several phases. These phases (shown in Figure 1) include: preprocessing to clean the dataset, splitting the dataset in two (known dataset: containing attacks with known responsible group name and the unknown dataset: containing attacks with unknown responsible group name). PCA is applied on a known dataset to select the most important features describing a given class. These features are used later by classification algorithms to build the prediction model. The resulting features from PCA from the known dataset is used for unknown dataset too. Then the predictor model was employed to predict the names.

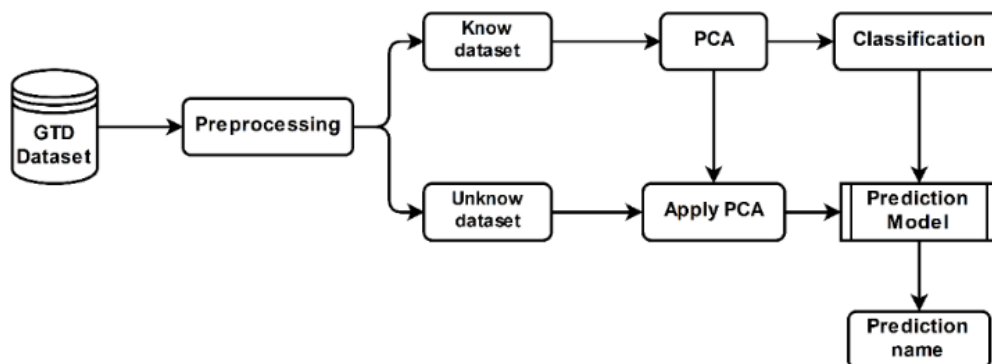


Figure 1 Proposed technique framework

#### 3.1 Dataset Pre-processing

Pre-processing is an essential step performed on the data set to make data more suitable for mining. In the examined dataset in this work, each terrorist incident was described by 134 attributes. Dataset preprocessing was carried out manually. Some attributes have been removed and some attribute values are grouped based on some specified conditions. The result of this procedure is a dataset without any missing values.

The criteria employed to remove the attributes is given as follows.

- The dataset contains a lot of missing values, so the attributes whose fields contain more than 50% missing values are deleted. 69 attributes were deleted accordingly.
- The attributes that contain additional relevant details about the attack such as: *summary* (a brief narrative summary of the incident), *addnotes* (more information that was not contained in any of the dataset fields), *location* (information specifying the location where the incident took place), *latitude*, *longitude*, ... etc.
- The attributes representing data collection methods about an attack such as the *Dbsource* attribute (identify the original data collection effort), *scite1*, *scite2*, and *scite3* (cites the various sources from which the incident information was compiled).
- Attributes which we found that the data fields are fixed for all fields. Because the dataset we selected includes only events that occurred in Turkey and were classified as terrorist acts. These attributes include: *country* (Country name), *region* (the regions which the country is located in) *crit1*, *crit2*, *crit3* (variables that show that the inclusion criteria are met), and *individual* (indicates person(s) who carried out the attack but do not belong to a known terrorist group or organization).
- Attributes containing subcategories of major classifications like *weapsubtype1\_txt* which shows the type of weapon used to carry out the attack.
- The features containing the number of casualties such as: *nkill* (number of killed), *nkillus* (The number of U.S. citizens killed), *nkillter* (number of perpetrators killed), *nwound* (number of wounded), *nwoundus* (The number of U.S. citizens wounded), and *nwoundte* (number of perpetrators wounded). These attributes were excluded, which is a result of the terrorist operation and has no direct impact on the name of the entity that carried out the operation.

The values of some attributes (types of instances) have been grouped as follows:

- *Day* and *month* attributes were integrated into one attribute called season and the values were distributed according to the four seasons. By closely analyzing the dataset, we note that the intensity of terrorist operations increases in the summer and the beginning of the fall (in the months of July, August and September) and decrease in winter and early spring (December until March).
- The attribute *provstate*, describe name of a place. The value of this field contains 43 city names and these cities are merged according to the region to which they belong. Turkey's provinces are distributed into 7 major regions.
- Attributes *targtype1\_txt* (captures the general type of target), *attacktype1\_txt* (captures the general method of attack type), and *gname* (name of the group that carried out the attack) were partially merged. Table 1 describes the merged values of each attribute.
- Attribute *natlty1\_txt* which includes nationality of the target that was attacked, it merged to 3 categories: Turkey, Foreign, and international.

### 3.2 Principal Component Analysis (PCA)

PCA is a technique used for data compression and feature extraction. Its main purpose is to analyze data to identify and find patterns in order to reduce the dimensions of the dataset into fewer dimensions which act as summaries of features with minimal loss of information [20].

Assume that  $X = x_1, x_2, \dots, x_n$  is a dataset consisting of  $n$  dimensional data vectors, our goal is to scale down this  $n$ -dimensional dataset to a  $k$ -dimensional subspace (where  $k < n$ ).

Table 1 Merged attribute values

Attribute	Category	#	New Category
targtype1_txt	Government (General)	59	Government
	Government (Diplomatic)	4	
	Military	59	Military and Police
	Police	275	
	Private Citizens & Property	203	Private Property
	Business	70	
	Educational Institution	46	Institutions
	Journalists & Media	17	
	NGO	2	
	Tourists	2	
	Religious Figures / Institutions	9	
	Utilities	18	
	Transportation	18	Transport
	Airports & Aircraft	2	
	Unknown	66	Other
	Other	4	
attacktype1_txt	Hostage Taking (Kidnapping)	60	Hostage
	Hostage Taking (Barricade Incident)	2	
	Hijacking	2	
	Unarmed Assault	2	Other
	Unknown	29	
gname	Turkish Communist Party/Marxist (TKP-ML)	2	Other
	Fetullah Terrorist Organization	1	
	Peoples' United Revolutionary Movement (HBDH)	1	
	The Independent Military Wing of the Syrian Revolution Abroad	1	
	Maoist Communist Party (MKP)	1	
	Free Syrian Army	1	
	People's Defense Unit (Turkey)	1	

The n-dimensional mean vector  $\mu$  is

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

The covariance matrix ( $Cov$ ) of the dataset is:

$$Cov = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \quad (2)$$

The eigenvalues and eigenvectors of the covariance matrix are calculated using

$$Cov_{vi} = \lambda_i v_i \quad (3)$$

Where  $\lambda$  = Eigenvalue,  $v$  =Eigenvector.

Let  $\Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_k]$ .  $\Lambda$  is a diagonal matrix of eigenvalues. The matrix  $V$  contains the eigenvectors  $V = [v_1, v_2, \dots, v_k]$  and is orthonormal  $V^T V = I_k$ . Where,  $I_n$  is the  $k \times k$  identity matrix.

Sort the eigenvectors by decreasing eigenvalues ( $\lambda_i \geq \lambda_{i+1}$ )

Let the matrix  $W = [v_1, v_2, \dots, v_k]$ , contain the first  $k$  eigenvectors

The low dimensional feature vector of a new input data is determined by

$$y = W^T \cdot x \quad (4)$$

When we applied the above equations to our dataset, from 51 principal components (PC), we found 31 components that are most contributing principal components. This contributes 95.9% of eigenvalues (Figure 2).

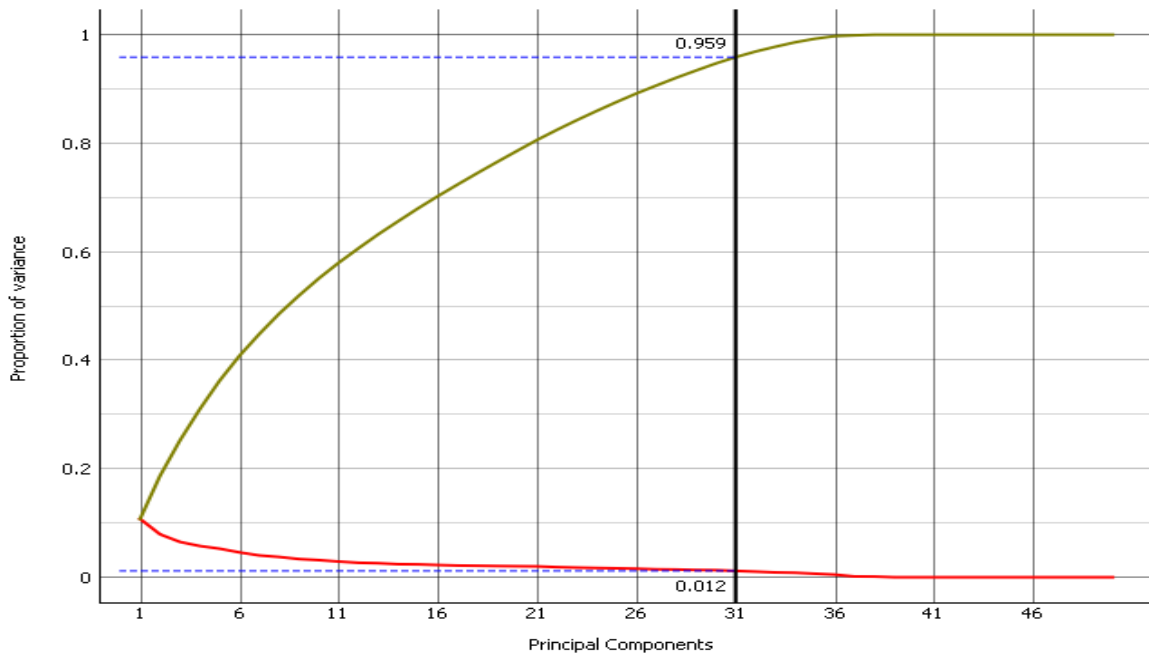


Figure 2 Principal components variance

Principle components also make distinction between classes much clearer. (Figure 3) shows the plot of the data points using only PC1 and PC7 principal components. It might be proper to use class tags to visualize the dataset as well. Looking at the graph, we can see that only two feature variables can be used to visualize the whole dataset properly as compared to having thirteen feature variables. Although the classes are not well differentiated but it helps to reduce the feature set without the loss of much information.

#### 4. Classification Algorithms

Classification and prediction are the prominent approaches for data mining in various fields. They are predictive models that predict the future trends based on some training datasets.

In the classification phase, a method of voting between 5 different machine learning algorithms was used. These algorithms are Random Forests, Logistic Regression, Adaptive Boosting (AdaBoost), Neural Network, and Support Vector Machine (SVM).

- **SVM** is a widely used machine learning algorithm. It can also be employed for both classification and regression purposes. The main idea of SVM is to construct maximum-margin hyperplane between any class data point within the training set, this can give a greater chance of new data being classified correctly [21].

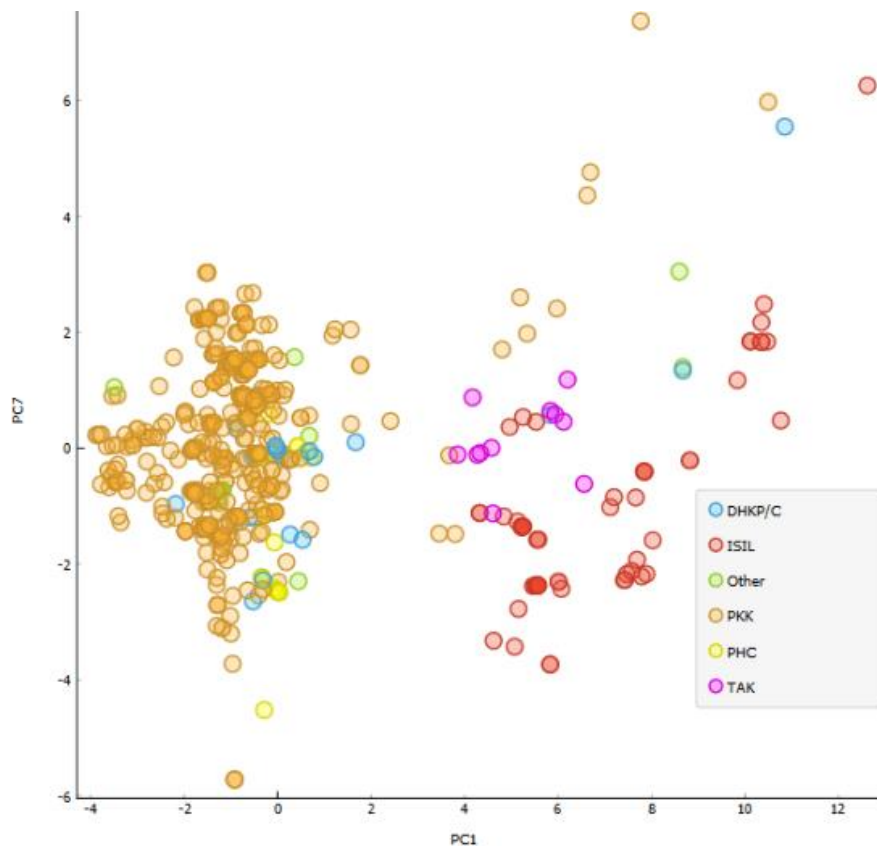


Figure 3 Clustering based on PC1-PC7

- **Random Forests** is an algorithm that is used for both classification and regression tasks by creating a forest and making it random. The created “forest”, is an ensemble of Decision Trees. The forest is trained using “bagging” method in most cases. Bagging is a combination of learning models to improve the overall result. Bagging can also be used for both classification and regression problems [22].
- **Logistic regression** is a probabilistic, statistical method for classifying data into discrete outcomes. It is named as ‘Logistic Regression’ because its underlying technique is quite the same as Linear Regression. But the biggest difference lies in what they are used for. Linear regression algorithms are used to predict/forecast values, but logistic regression is used for classification tasks [23].
- **AdaBoost** is an ensemble classifier that attempts to create a strong classifier from a number of weak classifiers. When used in conjunction with various types of learning algorithms, performance is improved. A weighted sum of the outputs of the 'weak learners' is used to represent the final output of the boosted classifier [24].
- **Neural Network** is a combination of units called neurons that are arranged in layers. These neurons convert an input vector into some output. Each neuron takes an input, applies some function (often nonlinear) on it and then passes its output to the next layer. In more general terms, neural networks are applied in a feed-forward fashion, i.e. Each layer forwards all its output to the next layer without feedback to the previous layer(s). A weighting mechanism is applied to the signals passing from one neuron to another. During the training phase, the weights are tuned so that the neural network adapts to a particular problem at hand [21].

All these algorithms were assembled in a voting algorithm to select the appropriate class. The voting is carried out according to the following formula

Assume we have  $k$  number of classes represented by  $C$ ,

$$c_j \in C, \quad j = \{1, \dots, k\} \quad (5)$$

and  $n$  classification algorithms represented by  $A$

$$a_i \in A, \quad i = \{1, \dots, n\} \quad (6)$$

the probability  $p$  of the class  $c_j$  in the Voting algorithm  $V$  is the average of the probabilities of these classes in the classification algorithms

$$p(V, c_j) = \frac{1}{n} \sum_{i=1}^n p(a_i, c_j) \quad (7)$$

The predicted class “ $Pred_c$ ” or the voted class “ $V_c$ ” in the voting algorithm is the class with the max probability.

$$V_c = \max \{p(V, c_1), p(V, c_2), \dots, p(V, c_k)\} \quad (8)$$

Let us consider 2 algorithms ( $a_1$  and  $a_2$ ) and 3 classes ( $c_1, c_2$ , and  $c_3$ ) as given in the Table 2.

Table 2 Voting algorithms

Classification Algorithm 1				Classification Algorithm 2				Voting Algorithm			
$p(a_1, c_1)$	$p(a_1, c_2)$	$p(a_1, c_3)$	$Pred_c$	$p(a_2, c_1)$	$p(a_2, c_2)$	$p(a_2, c_3)$	$Pred_c$	$p(V, c_1)$	$p(V, c_2)$	$p(V, c_3)$	$V_c$
0.53	0.07	0.4	$c_1$	0.45	0.03	0.62	$c_3$	0.44	0.05	0.51	$c_3$
0.05	0.15	0.8	$c_3$	0.01	0.10	0.89	$c_3$	0.03	0.125	0.845	$c_3$
0.58	0.32	0.1	$c_1$	0.18	0.77	0.05	$c_2$	0.38	0.545	0.075	$c_2$

#### 4.1 Performance measurement

Accuracy, Precision, Recall and F1-score metrics generally are used to evaluate the performance of different classification algorithms [25].

Since the dataset is multiple class, averaging the evaluation measures can give a view of the general results. There are two names to refer to averaged results: micro-averaged and macro-averaged results.

- In the Micro-average method, a sum of the individual true positives, false positives, and false negatives of the system for different sets is obtained to get the statistics about them.
- In Macro-average, the average of the precision and recall of the system on different sets is taken.

Since there is no balance between the number samples for each class in the dataset, (PKK represents 82.3% of the class while ISIL = 9.9, DHKP/C = 2.8, PHC=2.7 TAK = 1.5% and Other = 1.1), Micro-average is preferable if there is a class imbalance problem [26].

## 5. Results and Discussion

Each classifier was employed on known datasets both before and after feature dimension through PCA. The accuracy was estimated with 10-fold cross validation. The performance comparison of the classification learners on the two datasets is shown in Table 3. The results clearly show that PCA based feature dimension reduction led to an improved accuracy in all algorithms.

The comparative analysis based on results obtained using the proposed approach to that of other literature using GTD dataset is shown in Table 4.

Table 3 Accuracy results of selecting features using PCA ( $\oplus$ ) and without using PCA ( $\ominus$ )

Algorithm	Accuracy		F1 Micro-average	
	$\oplus$	$\ominus$	$\oplus$	$\ominus$
AdaBoost	92.5%	90.8%	96.1%	95.2%
Logistic regression	90.7%	88.2%	95.1%	93.7%
Neural Network	91.9%	90.4%	95.8%	95.0%
Random Forests	92.2%	90.5%	95.9%	95.0%
SVM	88.3%	86.9%	93.8%	93.0%
Voting	92.9%	91.2%	96.3%	95.4%

Table 4 Comparative results

		Current	Talreja et al.	Mohammed and Karabatak	Tolan and Soliman	Gohar et al
Dataset	Country	Turkey	India	Turkey	Egypt	world
	Period	2013-2017	1970-2015	2016	1970-2013	1970-2012
	Features	17	12	6	6	7
Classification algorithm	AdaBoost	92.5				
	Bayes Net			61.41		
	C4.5/J48		60	64.13	56.56	
	decision stump (DS)					84.97
	ID3				26.01	91.30
	KNN			51.1	73.03	83.43
	Logistic regression	90.7				
	NB			58.15	69.03	92.75
	Neural network	91.9				
	Random forest	92.2	58.5			
	SVM	88.3	73.2	59.78	75.42	
	Voting	92.9				93.40

In terms of predictions of names of terrorist groups in the unknown dataset. Table 5 shows the number of predicted names for each terrorist group using the algorithms in both cases. Note the difference in the number of operations per group depending on the algorithm and the case. However, all the algorithms and cases agreed to predict 0 times to the terrorist group PHC. This is logical because all their operations occurred between 15/July/2016 and 16/July/2016 when it announced its name and made a coup attempt to overthrow the government in Turkey.

Table 5 Predicted terrorist organizations' names and number of attacks using PCA ( $\oplus$ ) and without ( $\ominus$ )

Algorithm	PKK		ISIL		DHKP/C		TAK		PHC		Other	
	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\oplus$	$\ominus$
AdaBoost	144	132	22	25	3	6	2	5	0	0	1	4
Logistic Regression	122	139	24	19	16	10	7	3	0	0	3	1
Neural Network	127	123	25	29	11	10	6	7	0	0	3	3
Random Forest	146	144	20	21	1	3	2	4	0	0	3	0
SVM	110	103	34	33	13	18	8	9	0	0	7	9
Voting	147	142	18	20	3	3	3	5	0	0	1	2

## 6. Conclusion

This paper proposes a novel method for predicting responsible terrorist groups for unknown classified incidents. The proposed method consists of PCA technique for selecting features and a voting method between 5 best classification algorithms. The experiments are conducted using the GTD data set for the last 5 years terrorist incidents that occurred in Turkey. The proposed method using PCA obtained high results based on performance accuracy as compared to a method without using PCA. Our accuracy results show a significant improvement when compared to the results obtained by (Mohammed and Karabatak 2018) which do not exceed 64.13%. Moreover, an impressive result was obtained from the voting algorithm which is dependent on the sum of the probabilities resulting from all algorithms.

We suggest that future work can focus on the following:

- Improving the quality of terrorism-related data, by integrating GTD data with data from local authorities and the media.
- Linking the geographical factor of the terrorist operation with the areas of activity of the terrorist organization.
- Deeper studies of the relationship between the increase in terrorist operations at certain times and its relationship to political issues in the country, or to regional or international issues.

## References

- [1] C.C. Aggarwal, *Data Classification Algorithms and Applications*. London, England, CRC Press Taylor & Francis Group, 2015.
- [2] A. Babakura, M. N. Sulaiman and M. A. Yusut, "Improved method of classification algorithms for crime prediction". *Proc. - 2014 Int. Symposium on Biometrics and Security Tech., ISBAST 2014*, Kuala Lumpur, Malaysia, 26-27, August 2014.
- [3] BBC News, "Istanbul new year Reina nightclub attack leaves 39 dead," 2018. [Online]. Available: <https://www.bbc.com/news/world-europe-38481521> [Accessed: 09-Dec-2018].
- [4] E. S. Chris, *The Psychology of Terrorism: Theoretical understandings and perspectives*. Volume III, Lonon, England, Praeger Publishers, 2002.
- [5] Institute for Economics & Peace, *Global terrorism index 2018. Measuring the impact of terrorism*. Sydney, Australia, 2018.
- [6] J. Feng, H. Xu, S. Mannor and S. Yan, "Robust Logistic Regression and Classification," *Proc. - 27th Inter. Conf. on Neural Info. Processing Syst. (NIPS)*, Montréal, Canada, 08-13 December 2014.
- [7] F. Gohar, W. Haider and U. Qamar, "Terrorist Group Prediction Using Data Classification," *Proc. - Inter. Conf. on Artificial Intelligence and Pattern Recognition*, Kuala Lumpur, Malaysia, 17-19 November 2014.
- [8] GTD, "About the GTD," 2019 [Online]. Available: <https://www.start.umd.edu/gtd/about/> [Accessed: 09-Jan-2019].
- [9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. 2nd ed. New York, USA, Springer, 2008.
- [10] T. Howley, M. G. Madden, M. L. O'Connell, and A. G. Ryder, "The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data," *Knowledge-Based Syst.*, 19(5), 363–370, 2006.
- [11] A. Python, A. Bender, A. K. Nandi, P. A. Hancock, R. Arambepola, J. Brandsch, T. C. D. Lucas, "Predicting non-state terrorism worldwide," *Science Advances*, 7(21), 1-13, 2021.
- [12] C. Buffa, V. Sagan, G. Brunner and Z. Phillips. "Predicting Terrorism in Europe with Remote Sensing, Spatial Statistics, and Machine Learning," *ISPRS International Journal of Geo-Information*, 11(4), 1-12, 2022.



- [13] Hurriyet, "Kahvehaneye 58 hain kurşun - Son Dakika Haberler," 2018. [Online]. Available: <http://www.hurriyet.com.tr/gundem/kahvehaneye-58-hain-kursun-40048592> [Accessed: 09-Dec-2018].
- [14] I. T. Jolliffe, *Principal Component Analysis*. 2nd ed. New York, USA, Springer, 2002.
- [15] T. Kim, D. Park, D. woo, T. Jeong and S. Min, "Multi-class Classifier-Based Adaboost Algorithm," *Proc. - The Secd. Sino-foreign-interchange conf. on Intelligent Science and Intelligent Data Engineering*, Xi'an, China, 23 October 2011.
- [16] Z. C. Lipton, C. Elkan and B. Narayanaswamy, "Optimal Thresholding of Classifiers to Maximize F1 Measure," *Proc. - European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2014)*, Nancy, France, 15-19 September 2014.
- [17] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 228–233, 2001.
- [18] T. Feakin, "Terror, Security, and Money: Balancing the Risks, Benefits, and Costs of Homeland Security," *The RUSI Journal*, 157(4) 99, 2012.
- [19] F. Ozgul, Z. Erdem and C. Bowerman, "Prediction of past unsolved terrorist attacks," *Proc. - 2009 IEEE Inter. Conf. on Intelligence and Security Informatics (ISI 2009)*, Dallas/TX, USA, 26 June 2009.
- [20] D. M. W. Powers, *Evaluation: From Precision, Recall And F-Measure To Roc, Informedness, Markedness & Correlation*. Journal of Machine Learning Technologies, 2(1), 37–63. 2011.
- [21] A. Sachan and D. Roy, "TGPM: Terrorist Group Prediction Model for Counter Terrorism," *Inter. Journal of Comp. Appl.*, 44(10), 49–52. 2012.
- [22] M. Shermila, A. B. Bellarmine and N. Santiago, "Identity using Machine Learning Approach," *Proc. - 2nd Inter. Conf. on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 11-12, May 2018.
- [23] START, "Global terrorism database codebook: Inclusion criteria and variables 2018," 2018. [Online]. Available: <https://www.start.umd.edu/gtd/downloads/Codebook.pdf>. [Accessed: 02-Nov-2018].
- [24] D. Talreja, J. Nagaraj, N. J. Varsha and K. Mahesh, "Terrorism analytics: Learning to predict the perpetrator," *Proc. - 2017 Inter. Conf. on Advances in Computing, Communications and Informatics (ICACCI 2017)*, Mangalore India, 13, September 2017
- [25] Institute for Economics & Peace. *Global terrorism index 2019. Measuring the impact of terrorism*. Sydney, Australia, 2019.
- [26] G. M. Tolan and O. S. Soliman, "An Experimental Study of Classification Algorithms for Terrorism Prediction," *Inter. Journal of Knowledge Engineering-IACSIT*, 1(2), 107–112. 2015.

# Using of Hierarchical Loglinear Model in Multiway Frequency Tables and an Application on Suicide Cases

 Fatih Üçkardeş<sup>1</sup>

<sup>1</sup>University of Adiyaman, Faculty of Medicine, Department of Biostatistics and Medical Informatics, Adiyaman, Turkey; fatihuckardes@gmail.com

Received 25 May 2022; Revised 10 August 2022; Accepted 26 August 2022; Published online 31 December 2022

## Abstract

The aim of this study was to use Hierarchical Loglinear Model (HLLM) in the analysis of multiway frequency tables and to interpret the main and interaction effects of this model on suicide cases.

The data set used in this study was taken from the Turkish Republic State Statistical Institute (TUIK). A total of 6479 cases in 2016 and 2018 years were used in this analysis and the analyzes were made by considering gender, year and age variables.

As a result of HLLM analysis, Year, Gender and Age, which are the main effects in suicide cases, and the interactions of Year  $\times$  Gender and Gender  $\times$  Age were found significantly ( $P < 0.05$ ). There was a significant decrease in the suicide cases in 2018 compared to 2016 ( $P < 0.001$ ). In the sum of the years 2016 and 2018, among the age groups; 2: Suicide cases were observed in the 29-49 age group with a higher rate of 41.45%, while in the 1: 0-19 age group there were fewer suicide cases observed to 11.99%. When factor Gender is Male, factor Year changed from 50.61% to 49.39% at 2016 and 2018, respectively. However, when factor Gender is Female, factor Year changed from 55.71% to 44.29%. This differences in the amount of these changes caused significantly to the interaction of the Gender $\times$ Year.

The results has showed that, the main and interaction effects of multiway frequency tables can be interpreted by using HLLM analysis without another statistical method. Hence, it is thought that researchers may prefer HLLM models for the multiway frequency tables.

**Keywords:** contingency table, frequency table, loglinear model, suicide

## 1. Introduction

Categorical variables are widely used in the field of health. These variables are frequently analyzed for statistics with tests such as Pearson Chi-square, Likelihood Ratio or Fisher Exact test [1]. The use of Loglinear Model (LLM) method is very limited in the field of health. The main reason for this is that the method is not well known, its theoretical structure and the difficulty of interpreting the results.

LLM is a method that has been used to determine the relationships among variables of multiway frequency table obtained by cross-classifying sets of nominal, ordinal or discrete interval level variables. In this method, the variables can also be called as factor or categorigal variable. It allows to be evaluated as a contingency table of categorical factor [2]. The above mentioned tests are used for independence test in the statistical controls of contingency tables. However, when there are more than two categorical variables and interaction between categories out of the main effects, the above mentioned statistical methods are inadequate [3]. Interaction is the different effect of any level of a categorical variable or factor on various levels of other variable [4]. When investigating the presence of interaction in the contingency tables, the above-mentioned statistical methods are not used to explain this relationship [5]. Therefore, LLM, Configural Frequency Analysis (CFA) or Decision Tree (DT) methods are used for large data sets [6-9]. LLM is divided into three separate classes as general, logit and hierarchical. Hierarchical Loglinear Model (HLLM) is a method that checks whether the interaction effects are significant starting from the main effects in the analysis of data sets in multiway frequency table form of two or more discrete variables [5]. Therefore, it differs from the general and logit methods [10, 11].

Although Öğüş and Yazıcı [1], Şıklar et al. [3] and Yılmaz and Kesin [9] used LLM analysis in their studies, they interpreted through the Multidimensional scaling or Correspondence analysis (CA) methods because of the difficulties in interpreting the interaction effects.

Suicide is an important public health problem [12]. In a previous study, Topaloglu and Atay [13] reported that, according to a report published by the World Health Organization, the suicide cases are among the top ten causes of death in the world, and a person commits suicide every 40 seconds [14]. The total number of suicides in Turkey in 2016 and 2018 was 6479, with an average of 8.8 people committing suicide every day.

The aim of this study was to create a resource for researchers on the interpretation of interaction effects with HLLM analysis, by taking into account the number of suicides in Turkey in 2016 and 2018.

## 2. Materials and Methods

The suicide data used in this study were obtained from the open access database of the Turkish Republic State Statistical Institute (TUIK) [15]. Data of totally 6479 people in 2016 and 2018 were used. Three variables were considered within the scope of suicide data. Respectively, by Year (1: 2016 - 2: 2018), Gender (1: Male, 2: Female), Age (1: 0-19, 2:20-39, 3: 40-59 and 4: 60 and above) it is recorded.

Considering three categorical variables in this present study, saturated HLLM, which includes all main and interaction effects, is written as follows [6]:

$$\ln(m_{ijk}) = \theta + \lambda_i^A + \lambda_j^B + \lambda_{ij}^{AB} + \lambda_k^C + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} + \lambda_{ijk}^{ABC} \quad (1)$$

Here, the  $\lambda$ 's are called effects. The predicted cell count  $m_{ijk}$  based on the current hierarchical model. The prediction equation is the superscripts represent the variables and the subscripts represent the category numbers or levels of the variables. However, for the three categorical variables, in the saturated Loglinear model including all the main effects and interaction effects, the main effects of the A, B and C variables, respectively; AB, AC, BC and ABC terms show interaction effects [16].

### 2.1. Testing the fit of the model and choosing the model

Likelihood – Ratio ( $G^2$ ) and Pearson ( $\chi^2$ ) Chi-square statistics are used, respectively, to test the suitability of a saturated model considered in Equation (1) to the data.

$$\chi^2 = 2 \sum_{i,j,k} \frac{(f_{ijk} - \hat{m}_{ijk})^2}{\hat{m}_{ijk}} \quad (2)$$

and Likelihood – Ratio,

$$G^2 = 2 \sum_{i,j,k} f_{ijk} \ln \left( \frac{f_{ijk}}{\hat{m}_{ijk}} \right) \quad (3)$$

Here  $f_{ijk}$  is the observed number for a three-way table. These statistics are; It is used to test whether the agreement between the observed and expected frequencies calculated according to the model is significant [17].

NCSS package program version 7.0 was used for HLLM calculations [18]. In calculation; Delta value as module: 0.2, Model definition: Full model, and hierarchical model, Maximum number of repetitions: 25, maximum difference: 0.25, Alpha value: 0.05 was accepted as goodness of fit.

### 2.1.2. Parameter Estimation Section

The program uses the Maximum likelihood model parameter estimation algorithm suggested by Haberman [19] to determine the best saturated model. The Step Down method is used to determine the best saturated model, and the  $G^2$  test is used as the fit statistic of the model. In the step down method;

The algorithm starts the operation from the most complex model. It performs the process of finding the most suitable model by removing the terms from the model. The aim here is to determine the best model with the fewest parameters [20]. As a result, the NCSS Program proposes the best goodness of fit model among many prediction models.

The fact that the P-value of both  $G^2$  and  $\chi^2$  values is nonsignificant in the fit of the final model indicates a goodness fit of the saturated model. For this reason, it is expected that the best saturated model will come out with P-value nonsignificant.

### 3. Result

In Table 1, the significance of the terms that should be included in the HLLM model is analyzed. It is seen that the model with three or higher is significant. In the lower section, the significance results of whether single effects are significant or not are given. Since all three factors are significant, they will be included in the hierarchical model. Then, in the model selection section, the results of the best saturated model are given in Table 2.

Table 1 Multiple-Term Test Section

Multiple-Term Test Section					
K-Terms	DF	Like. Ratio Chi-Square	Prob Level	Pearson Chi-Square	Prob Level
1WAY & Higher	15	3078.48	<0.0001	3428.43	<0.0001
2WAY & Higher	10	162.19	<0.0001	173.55	<0.0001
<b>3WAY &amp; Higher</b>	<b>3</b>	<b>7.27</b>	<b>0.0637</b>	<b>7.25</b>	<b>0.0644</b>
K-Terms	DF	Like. Ratio	Prob		
1WAY Only	5	2916.28	<0.0001		
2WAY Only	7	154.92	<0.0001		
<b>3WAY Only</b>	<b>3</b>	<b>7.27</b>	<b>0.0637</b>		

Note: Simultaneous test that all interactions of order k are zero. These Chi-Squares are differences in the above table.  $P \leq 0.20$ : Significance reference value: 0.20.

Table 2 Statistical results for determination of the best saturated model

Step-Down Model-Search Section									
Step No	Best No	DF	Chi-Square	Prob Level	Deleted Term	DF	Chi-Square	Prob Level	Hierarchical Model
1	1	0	0.0	1.0000	None	0	0.0	<0.0001	ABC
2	1	3	7.3	0.0637	ABC	3	7.3	0.0637	BC,AC,AB
3	2	6	147.6	<0.0001	BC	3	140.3	0.0000	AC,AB
4	2	6	8.4	0.2135	AC	3	1.1	0.7825	BC,AB
5	2	4	19.7	0.0006	AB	1	12.5	0.0004	BC,AC
6	4	9	149.2	<0.0001	BC	3	140.8	<0.0001	AB,C
7	4	7	21.4	0.0033	AB	1	13.0	0.0003	BC,A
<b>Best model found: BC, AB</b>									
<b>4</b>	<b>4</b>	<b>6</b>	<b>8.4</b>	<b>0.2135</b>	<b>AC</b>	<b>3</b>	<b>1.1</b>	<b>0.7825</b>	<b>BC, AB</b>
<b>Model Section : Hierarchical Model: BC, AB</b>									

The main purpose of Table 2 was to exhibit the best model with the fewest terms. The NCSS Program defined the model in Step 4 as the best saturated model. The results of the goodness of fit of this saturated model are as follows.

The parameter estimation results of the saturated model were given in Table 3. As can be seen Table 3, the main effects and the interaction effects of the model were found to be significant ( $P < 0.05$ ).

Table 3 Statistical results of the main effects and the interaction effects of the saturated model

<b>Chi-Square Tests Section</b>								
DF	Like. Ratio Chi-Square	Prob Level	Pearson Chi-Square	Prob Level	Model			
6	8.35	0.2135	8.33	0.215	BC,AB			
<b>Parameter Estimation Section</b>								
Model	Number Cells	Count	Percent Count	Average Log(Count)	Effect (Lambda)	Effect Std. Error	Effect Z-Value	P value
Mean	16	6479	100	5.7632	5.7632	0.0155	371.78	<0.001
A: Year								
1	8	3366	51.94	5.8267	0.0635	0.0155	4.10	<0.001
2	8	3114	48.06	5.6997	-0.0635	0.0155	-4.10	<0.001
B: Gender								
1	8	4789	73.91	6.2395	0.4763	0.0155	30.73	<0.001
2	8	1691	26.09	5.2869	-0.4763	0.0155	-30.73	<0.001
C: Age								
1	4	777	11.99	5.2575	-0.5057	0.03	-16.86	<0.001
2	4	2686	41.45	6.3652	0.602	0.0221	27.21	<0.001
3	4	1912	29.51	5.962	0.1988	0.0253	7.87	<0.001
4	4	1105	17.05	5.4681	-0.2951	0.0293	-10.08	<0.001
B: Gender								
1	8	4789	73.91	6.2395	0.4763	0.0155	30.73	<0.001
2	8	1691	26.09	5.2869	-0.4763	0.0155	-30.73	<0.001
C: Year								
1	4	777	11.99	5.2575	-0.5057	0.03	-16.86	<0.001
2	4	2686	41.45	6.3652	0.602	0.0221	27.21	<0.001
3	4	1912	29.51	5.962	0.1988	0.0253	7.87	<0.001
4	4	1105	17.05	5.4681	-0.2951	0.0293	-10.08	<0.001
AB: Year, Gender								
1,1	4	2424	37.41	6.2518	-0.0512	0.0155	-3.30	<0.001
1,2	4	942	14.54	5.4016	0.0512	0.0155	3.30	<0.001
2,1	4	2365	36.5	6.2272	0.0512	0.0155	3.30	<0.001
2,2	4	749	11.56	5.1723	-0.0512	0.0155	-3.30	<0.001

BC: Gender, Age

1,1	2	437	6.75	5.3876	-0.3462	0.03	-11.55	<0.001
1,2	2	2008	31	6.9119	0.0704	0.0221	3.18	<0.001
1,3	2	1509	23.3	6.6262	0.188	0.0253	7.44	<0.001
1,4	2	833	12.86	6.0323	0.0879	0.0293	3.00	0.0014
2,1	2	339	5.24	5.1275	0.3462	0.03	11.55	<0.001
2,2	2	677	10.45	5.8186	-0.0704	0.0221	-3.18	<0.001
2,3	2	402	6.21	5.2977	-0.188	0.0253	-7.44	<0.001
2,4	2	271	4.19	4.9039	-0.0879	0.0293	-3.00	0.0014
1,2	2	942	14.54	5.4016	0.0512	0.0155	3.30	<0.001
2,1	2	2365	36.5	6.2272	0.0512	0.0155	3.30	<0.001
2,2	2	677	10.45	5.8186	-0.0704	0.0221	-3.18	<0.001
2,3	2	402	6.21	5.2977	-0.188	0.0253	-7.44	<0.001
2,4	2	271	4.19	4.9039	-0.0879	0.0293	-3.00	0.0014

According to Table 3, Year, Gender and Age main effects and Year  $\times$  Gender and Gender  $\times$  Age interaction effects were found significantly ( $P < 0.001$ ). There was a significant decrease in the cases of suicides in 2018 when compared to 2016 ( $P < 0.001$ ). Males (73.91%) attempted suicide at a higher rate than females (26.09%) in the total cases of suicides in 2016 and 2018 ( $P < 0.001$ ). Similarly, While the highest suicide rate was in the 29-49 age group (41.45%), the lowest was in the 0-19 age group (11.99%).

### 3.1. Interpretation of Interactions

The interactions of Year  $\times$  Gender and Gender  $\times$  Age were found to be significant in Table 3. We can construct the following two-way table of percentages from the Count column of Table 3.

Table 4 The table of Year  $\times$  Gender interaction percentages

Gender	Year		Total
	1:2016	2:2018	
<b>1: Male</b>	<b>50.61%</b>	<b>49.39%</b>	<b>% 100</b>
	$(2424/(2424+2365))*100$	$(2365/(2424+2365))*100$	
<b>2: Female</b>	<b>55.71%</b>	<b>44.29%</b>	<b>% 100</b>
	$(942/(942+749))*100$	$(749/(942+749))*100$	

As can be seen in Table 4, it was shown that when factor Gender is Male, factor Year changed from 50.61% to 49.39% at 2016 and 2018, respectively. However, when factor Gender is Female, factor Year changed from 55.71% to 44.29%. This difference in the amount of change is what causes Gender $\times$ Year to be significant (Figure 1). This type of table should be created for every significant term.

The suicide rate was found to be higher in men over the age of 20, whereas, the suicide rate in women was higher under the age of 20 (Table 5). This difference in the amount of change is what causes Gender  $\times$  Age to be significant (Figure 2).

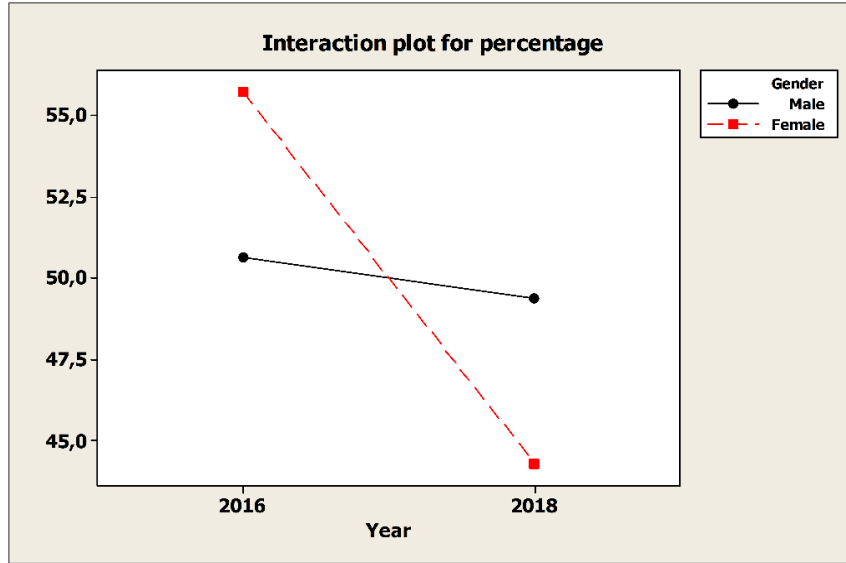


Figure 1 Percentage of Years × Gender interaction

Table 5 The table of Gender × Age interaction percentages

Age	Gender		Total
	1: Male	2: Female	
1:0-19	<b>56.32%</b> $(437/(437+339))*100$	<b>43.68%</b> $(339/(437+339))*100$	100%
2:20-39	<b>74.79%</b> $(2008/(2008+677))*100$	<b>25.21%</b> $(677/(2008+677))*100$	100%
3:40-59	<b>78.97%</b> $(1509/(1509+402))*100$	<b>21.03%</b> $(402/(1509+402))*100$	100%
4:60->	<b>75.45%</b> $(833/(833+271))*100$	<b>24.5%</b> $(271/(833+271))*100$	100%

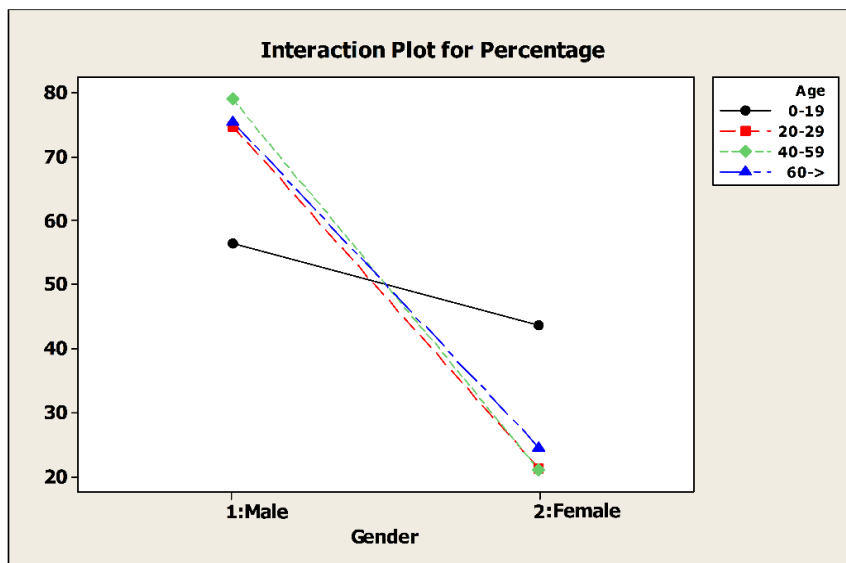


Figure 2 Percentage of Gender × Age interaction

#### 4. Discussion

It was determined that there was a significant decrease in the total number of cases from 2016 to 2018 in the number of suicides. The main reason for this may have been the Istanbul Declaration for Women, which came into force in 2017, and the women who are victims of domestic violence and violence against women, which came into force in 2017. One-click Women's Support Service (KADES) system may have reduced these deaths [21,22]. Similarly, the fact that women aged 0-19 have fewer suicide cases than men is consistent with both World Health Organization (WHO) and TUIK 2012 data [13]. However, it is highly probable that women in this age group have higher rates of suicide when compared to women in the older age group. This protection system developed by the state to protect women against domestic violence may be effective. It is thought that studies on this system may have reduced suicide rates, especially in women.

When suicide cases were evaluated in terms of gender, the suicide rates of men were found to be higher than women in all ages and years. Kaplan and Sadock [23] reported that men tend to commit suicide more than women. This result is consistent with this study.

Oğuzlar [5] tried to interpret the interactions, according to the estimation value as a result of the LLM analysis, and commented only as an increase or decrease. It is very difficult to make a biological interpretation of the estimation values of the LLM analysis. For this reason, many researchers have preferred the fit analysis.

Öncel and Erdugan [6] analyzed the main effects in the LLM analysis using the Chi-square independence test in a study on smoking addiction. Erdem [11] reported that if the number of categories or factors in the contingency table is more than two, the use of Chi-square independence tests becomes difficult and analysis may be impossible and so it has been argued that the LLM method should be used in such cases.

Many studies such as Yılmaz and Kesin [9], Yılmaz and Aktaş [24], Adıgüzel [25] and Kaşıkır [26] also used the LLM method. Since they could not interpret the interaction effects, they tried to interpret these interactions with Multidimensional scaling or Correspondence analysis (CA) methods. Although Yılmaz and Kesin [9] described the variation of the dimensions in the graphical interpretation of the CA analysis, they did not mention the interactions between the variables. In another studies, it has been compared the CA method with the LLM method and has been argued that these two methods are complementary to each other [27,28]. However, the CA method is a dimension reduction method for categorical variables, and allows variables to visually present the relationship between subcategories in a two or three dimensional space [9]. This method uses various distance or proximity measurements and normalization methods. According to these methods are obtained different graphical results, therefore, it should be kept in mind that performing CA analysis on variables whose interaction is significant as a result of LLM analysis may lead to different results [11].

Although three factors were considered in this study, tables using HLLM analysis may contain more factors. However, the number of factors being four or more may complicate the interpretation of the interaction. In such cases, matrix interaction graphs or alternatively decision tree methods can be used to interpret triple or higher order interactions [8].

Brzezinska [2] and Erdugan and Türkan [29] researchers suggested Akaike criteria (AIC), Bayes criteria (BIC) and Coefficient of Determination ( $R^2$ ) to determine the best model in LLM analysis. However, NCSS programme directly recommends the best model. Therefore, these criteria were not included in this study.

The menu of LLM is available in NCSS, SPSS and many other package programs. NCSS program proposals the best saturated LM model as a standard result compared to widely used SPSS package program.

Öncel and Erdugan [6] were solved manually the detailed analyzes of the contingency tables, the estimation of the model parameters and the statistical results of the parameter estimation section by the LLM. They emphasized these analyzes as a shortcoming of the SPSS package program. Öncel and Erdugan [6] reported that, as a result of the SPSS analysis, it was not clear that a meaningful result



would be obtained for which parameter and which level. Similarly, Şıklar et al. [3] firstly applied LLM analysis to the obtained data and it has been explained that it is necessary to use CA for the interpretation of interactions that are found to be significant.

In the NCSS program, the statistical results of the parameter estimation part of the best saturated HLLM are given in detail. There is no need to calculate manually or use any other method for these statistics. In this study, in addition to these statistics, a sample graphical interpretation has also been added to better understand the interpretation of interactions.

As a result, it has been shown that multiway frequency table using HLLM analysis can be analyzed without the need for any other statistical method and how interaction effects can be interpreted together with main effects. It is thought that researchers may prefer HLLM models for multiway frequency tables.

### Conflict of Interest

The author declare that there is not any conflict of interest regarding the publication of this manuscript.

### Ethics approval



Not applicable.

### References

- [1] E. Ögüş and C. A. Yazıcı, “Comparision of Log-Linear Analysis and Correspondence Analysis in Two-Way Contingency tables: A Medical Application”, *Balkan Med. J.*, vol. 28, pp.143-147, 2011.
- [2] J. Brzezińska, “Model Selection Methods In Log-Linear Analysis”, *Acta Uni.v Lodz Folia Oecon.*, vol. 285, pp.107-114, 2013.
- [3] E. Şıklar, V. Yılmaz and D. Çoşkun, “Eskişehir'deki Üniversitelerde Görevli Akademik Personelin İş Tatmini ve Duygusal Tükenmişliklerinin Log-Linear Modeller Ve Correspondence Analizi İle İncelenmesi”, *Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi*, vol. 26, no. 2, pp.113-134, 2011.
- [4] O. Düzgüneş, T. Kesici, O. Kavuncu and F. Gürbüz, “Araştırma ve Deneme Metotları”, *Ankara: Ankara Üniversitesi Ziraat Fakültesi Yayınları*, 1987.
- [5] A. Oğuzlar, “Hiyerarşik Logaritmik Doğrusal Modeller Arasından En Uygun Modelin Seçimi”, *Öneri*, vol. 6, no. 21, pp.235-245, 2004.
- [6] Y.S. Öncel and F. Erdugan, “Kontenjans tablolarının analizinde log-lineer modellerin kullanımı ve sigara bağımlılığı üzerine bir uygulama”, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, vol. 19, no. 2, pp.222-235, 2015.
- [7] N. Doğan and İ. Doğan, “Konfigürall Frekans Analizi ve İntihardaki 10 Yıllık Değişimin İncelenmesi”, *Euras. J. Fam. Med.*, vol. 6, no. 2, pp.77-81, 2017.
- [8] M. Koparal, N.U. Yılmaz, Ö.A. Küçük, A. Keskinrüzgar, F. Üçkardeş, “Classification Tree Method for Determining Factors Associated with Halitosis”, *BSJ Health Sci.*, vol. 4, no. 2, pp.91-97, 2021.
- [9] V. Yılmaz and F. K. Kesin, “Logaritmik Doğrusal Modeller ve Uyum Analizinin Birlikte Kullanımı: Lise Öğrencilerinin Sigara İçme Alışkanlıklarını Etkileyen Faktörlerin Belirlenmesi”, *Türkiye Klinikleri J. Biostat.*, vol. 10, no. 1, pp.65-86, 2018.
- [10] K. Özdamar, “*Paket Programları ile İstatistiksel Veri Analiz II*”, Eskişehir: Baskı Kaan Kitabevi, 2002.
- [11] A. Erdem, “*Uygunluk analizinde Logaritmik doğrusal modellerin kullanımı: Televizyon izleme eğilimleri üzerine bir uygulama*”, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, 2014.
- [12] M. Arslan, M. Duru and G. Kuvandık, “Hatay’da İntihar Girişiminde Bulunan Olguların Analizi”, *Adli Tıp Dergisi*, vol. 22, no. 3 pp.9-14, 2008.

- [13] E. Topaloğlu and A. Atay, "Kategorik Verilerin Analizinde Logaritmik Doğrusal Modellerin Kullanımı: İntihar Olasılığı Verileri Üzerine Bir Uygulama", *Optim. Ekon. Yönetim Bilim Derg.*, vol. 7, no. 2, pp.565-580, 2020.
- [14] S. Bayraktar, "Conceptual Issues Concerning Suicide in Children and Adolescents", *Mediterr. J. Soc.*, vol. 1, pp.139-159. 2015.
- [15] TÜİK, "Turkish Republic State Statistical Institute", 2021. [Online]. Available: <https://biruni.tuik.gov.tr/medas/?locale=tr>. [Accessed: 25-May-2022].
- [16] H. Toutenburg, "*Statistical Analysis of Designed Experiments*", Springer -Werlag New York Inc, 2002.
- [17] J. N. K. Rao, and A. J. Scott, "On Simple Adjustments to Chi-Square Tests with Sample Survey Data", *The Ann. Stat.*, vol. 15, no. 1, pp.385-397, 1987.
- [18] NCSS, "Number Cruncher Statistical System, Version 2007". NCSS, LLC. Kaysville, Utah, USA.
- [19] S. J. Haberman, "Log-linear models for frequency data: Sufficient statistics and Likelihood Equations", *The Annals of Statistics*, vol. 1, no. 4, pp.617-632, 1973.
- [20] J.K. Benedetti and B.M. Brown, "Strategies for the Selection of Log-Linear Models", *Biometrics*, vol. 34, pp.680-686, 1978.
- [21] Yargıtay, "Türkiye Cumhuriyeti Yargıtay Başkanlığı homepage", 2017. [Online]. Available: <https://www.yargitay.gov.tr/documents/IstanbulBildirgesiKitapcigi.pdf>. [Accessed: 25-May-2022].
- [22] KADES, "Kadın Destek Servisi", *Türkiye Cumhuriyeti İç İşleri Bakanlığı homepage*, 2017. [Online]. Available: <https://www.icisleri.gov.tr/kadin-destek-uygulamasi-kades>, [Accessed: 25-May-2022].
- [23] H.I. Kaplan and B. J. Sadock, "Klinik Psikiyatri", Klinik Psikiyatri. Abay E (Çev. Ed.). *Nobel Tıp Kitabevi*, İstanbul, 2004.
- [24] V. Yılmaz and C. Aktaş, "Üç Boyutlu Kontenjans Tablolarının Analizinde Log-Linear Modellerin Kullanımı ve Trafik Kazalarına Uygulanması", *OGÜ Sos Bil De.*, vol. 2, pp.169-182. 2001.
- [25] E. Adıgüzel, "Yeraltı Ocaklarındaki İş Kazalarının Aşamalı Logaritmik Doğrusal Modeller ve Uyum Analizi İle İncelenmesi", Yüksek Lisans tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 2008.
- [26] F. Kaşıkır, "Logaritmik Doğrusal Modeller ve Uygunluk Analizinin Birlikte Kullanımı: Lise Öğrencilerinin Sigara İçme Alışkanlıklarına Uygulanması", Yüksek Lisans tezi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 2012.
- [27] P.M. Van der Heijden and K.J. Worsley, "Correspondence analysis used complementary to loglinear analysis", *Psychometrika*, vol. 53, no. 2, pp.287-91, 1988.
- [28] Van der Heijden PGM, de Falguerolles A and A. J. de Leeuw, "Combined approach to contingency table analysis using correspondence analysis and log-linear analysis", *Journal of Appl. Statis.*, vol. 38, no. 2, pp.249-92, 1989.
- [29] F. Erdugan and H.A. Türkan, "Üç Yönlü Kontenjans Tablolarında Log-Linear Model ile İş Kazası Verilerinin İncelenmesi", *Karaelmas Fen ve Mühendislik Dergisi*, vol 7, no. 2, pp.462-468, 2017.

# A Review of Recent Developments on Secure Authentication using RF Fingerprints Techniques

 Hüseyin Parmaksız<sup>1</sup>,  Cihan Karakuzu<sup>2</sup>

<sup>1</sup>Corresponding Author; Bilecik Şeyh Edebali University, Department of IT; huseyin.parmaksiz@bilecik.edu.tr;

<sup>2</sup> Bilecik Şeyh Edebali University, Department of Computer Engineering; cihan.karakuzu@bilecik.edu.tr;

Received 7 March 2022; Revised 10 April 2022; Accepted 11 October 2022; Published online 31 December 2022

## Abstract

The Internet of Things (IoT) concept is widely used today. As IoT becomes more widely adopted, the number of devices communicating wirelessly (using various communication standards) grows. Due to resource constraints, customized security measures are not possible on IoT devices. As a result, security is becoming increasingly important in IoT. It is proposed in this study to use the physical layer features (PLF) of wireless signals as an effective method of increasing IoT security. According to the literature, radio frequency fingerprinting (RFF) techniques are used as an additional layer of security for wireless devices. To prevent spoofing or spoofing attacks, unique fingerprints appear to be used to identify wireless devices for security purposes (due to manufacturing defects in the devices' analog components). To overcome the difficulties in RFF, different parts of the transmitted signals (transient/preamble/steady-state) are used. This review provides an overview of the most recent RFF technique developments. It discusses various solution methods as well as the challenges that researchers face when developing effective RFFs. It takes a step towards the discovery of the wireless world in this context by drawing attention to the existence of software-defined radios (SDR) for signal capture. It also demonstrates how and what features can be extracted from captured RF signals from various wireless communication devices. All of these approaches' methodologies, classification algorithms, and feature classification are explained. In addition, this study discusses how deep learning, neural networks, and machine learning algorithms, in addition to traditional classifiers, can be used. Furthermore, the review gives researchers easy access to sample datasets in this field.

**Keywords:** IoT, security, deep learning, RFF, SDR

## Nomenclature Acronyms

AI/ML: artificial intelligence/machine learning	MLE: maximum likelihood estimates
AWGN: additive white gaussian noise	NLP: natural language processing
BRCD: Bayesian ramp change detector	OFDM: orthogonal frequency division multiplexing
BSCD: Bayesian step change detector	PCs: phase characteristics
CFO: carrier frequency offset	PCA: principal component analysis
CNN: convolutional neural network	PD: phase detector
CSIR: channel state information at the receiver	PLF: physical layer features
DCTF: differential constellation trace figure	PLS: partial least squares
DSP: digital signal processing	PSD: power spectral density
DWT: discrete wavelet transform	RF: radio frequency
ELM: extreme learning machine	RFF: radio frequency fingerprinting
EMD: empirical mode decomposition	RFID: radio frequency identification
FE: feature extraction	RSS: radio received signal strength
FFT: fast fourier transform	SDR: software defined radio
GSM: global system for mobile	SFO: sampling frequency offset
GLRTD: generalized likelihood ratio test detector	SNR: signal-to-noise ratio
HF: high frequency	SPoTS: starting point of transient signal
HHT: Hilbert-Huang transform	STSP: short training sequence preamble
IoT: internet of things	SVM: support vector machine
LTSP: long training sequence preamble	TS: transient signals
MCPD: mean change point detector	URH: universal radio hacker
MDA: multiple discrimination analysis	VFDTD: variance fractal dimension threshold detector

## 1. Introduction

Kevin Ashton coined the term IoT in a presentation (where the benefits of RFID technology to the company and its use were suggested) prepared for the Procter & Gamble Company in 1999. In general terms, it is possible to define the IoT as a system of devices that communicate with each other and form an intelligent network by connecting and sharing information, thanks to various communication protocols. Digital Agenda, published by the European Union, is an emerging technology and market that enables objects and applications to communicate between themselves, produce data and share this data. This structure is defined as “*an ecosystem of smart applications and services that make people's lives easier and raise their living standards*”. The European Technology Platform, on the other hand, defined it as “*a common network established between things/objects that can be physical and virtual, also have pre-defined functions and work in smart environments, and this network exchanges information with other networks and users*”. Today, one of the areas where technological developments are applied the fastest is objects. Both the vital convenience and benefits brought by technological innovations, and the rapid increase in the use of people by adapting to technology very quickly, communication with objects is the most current issue. With spread of smart devices, social structures have changed, and the phenomenon of “*Information Society*” has been fully formed. In the past, the information was based only on the information that people gave voluntarily, and the accuracy of the data received was often discussed. However, at this point, data is now collected with smart devices independently of the declaration of individuals, and the accuracy level increases. In this way, reliable knowledge will also increase with smart objects. Development of IoT concept and technology; changes the social structure by facilitating life, raising living standards, increasing productivity and contributing to economies. Like all good things, when it is not taken care of, its bad points are serious. The key point is information security. Information security problems related to smart objects, why the issue of security is really important and the precautions that can be taken are emphasized. In 2013, Russia's state channel Rossiya 24 claimed that the hacker irons produced in China and imported to the country contained a special wireless internet control chip, thus spying on the personal computers of the users by organizing a cyber-attack. Although this news may seem like exaggerated news or fake news at first, its accuracy has been determined in the examinations [1, 2].

It will be seen in real life from science fiction movies that the car we use is the target of attackers and causes accidents, smart alarm and lock systems are broken and cyber thefts occur, infiltrating wearable objects, detecting discomfort from body activities and the emergence of cyber murders. If we give an example from our house in a narrower area; if all objects in a house are managed from a single center; it is possible to seize that system with a cyber-attack, to start a fire by playing with the oven settings, to steal by turning off alarm system and opening the door, to copy all personal data on the computer or to violate the privacy of private life by watching the house from camera system. Information security violations that may occur in smart devices have a chance to be prevented by certain controls to be made by both the manufacturer and the user. Considering the most common security vulnerabilities in smart devices, it can be determined that “*Web Interface Configuration, Authentication/Authorization, Network Services, Encrypted Transport, Privacy, Mobile Applications, Cloud, Security Configurations, Software and Physical Security*” checkpoints should be made [2]. The security policy defines all the rules, regulations and procedures that must be followed to ensure system security and can be applied to many different areas. Some policies to prevent risks can be categorized as follows:

**Remote Access Policy:** It is the standardization of who can connect to the system, when and how, and what kind of devices can be connected to this system remotely.

**Information Privacy Policy:** It is the definition of which methods will be used to protect information depending on the level of sensitivity. Generally, more sensitive information has a higher level of security.

**Computer Security Policy:** Defines which computers users will use. This policy defines who will use certain computers and which programs will be used to protect a computer or whether a particular storage device will be used.

**Physical Security Policy:** Defines how physical assets are secured.

Password Policy: Determines how long a password should be changed, what type of passwords to use, and the criteria for defining password security levels.

The review's chronology continues as follows. First and foremost, what is an RF fingerprint, where it is used, the importance of using modern software defined radios instead of the antenna and oscilloscope used in the literature to capture the RF signal, which parts of the signal are used in determining the feature, effective algorithms in FE, what features are used in this area, and what classification algorithm is used and provides a general idea of how successful its methods have been. Furthermore, the extensive literature review and sample data sets are expected to shed light on the scientists who will work in this field.

## 2. Radio Frequency Fingerprinting and Data Acquisition

RFF is a technique that's used to identify the radio signals emitted by various devices. In monitoring radars, RFF is widely used in the military field. It's also used to authenticate wireless connections.

The process of extracting the radio signal's unique features involves firstly identifying the source code. The code then passes these features to a classifier. The data acquisition subsystem of a wireless device is used to acquire and digitize a radio signal. It is typically built into a device to minimize signal degradation due to noise [3].

In SDR, the software acts as the front end while the hardware provides the signal processing engine. The software that will allow this interaction is called GNU Radio. It is possible to create a "*flow graph*" which is a collection of interconnected signal processing blocks, by appropriately combining the blocks (to which algorithms and functions can be implemented) in GNU Radio. URH was created with theoretically oriented researchers in mind who want to focus on protocol logic rather than diving deep into HF and DSP. URH can perform spectrum analysis, signal recording, and protocol sniffing [4]. SigDigger, like GNU Radio and URH, is another software that can be used in the field of signal capture. SigDigger (digital signal analyzer written in Qt5 by BatchDrake for Unix systems) collaborates with three projects: Sigutils (DSP library that distributes the load using multi-core CPUs), Suscan (real-time signal analysis library), and SuWidgets.

Signal acquisition can be done either actively or passively. In the active mode, a radio signal is captured from a wireless device to be used for identification, and signal collection is used for sampling [5]. In passive mode, while the device communicate with other devices, radio signal get caught from it. As an example, mobile phones in GSM communication with passive reception base stations can be defined [6].

### 2.1 Software Defined Radios

SDRs is used in the literature for radio communication. Unlike hardware-based solutions, SDR is a software-defined radio technology based on radio and wireless communication protocols. Figure 1 provides a general perspective of the wireless world with SDR. Thanks to its reprogrammable feature, it meets the needs without the need for extra equipment. In this way, it strengthens the possibility of working on multi-functional and multi-band radio and wireless devices [7]. SDR is a major innovation development that develops a reconfigurable wireless communication system that replaces the traditional hardware communication devices implementation [8] SDR, it would be difficult and extra costly to install hardware from scratch or add new hardware to the existing system for minor design changes [9]. SDR allows the same hardware platform to be reused for many communications equipment with different protocols, reducing time to service and development cost to the end user [10]. The report in [7], it was expected SDR market is being worth more than \$29 billion for the year 2021. Global Industry Analysts, Inc. reports the following SDR market tendencies: (i) growing military interest in developing countries in communication/information and large-scale distribution systems; (ii) rising requisition for public safety and disaster preparedness applications; and (iii) the need for developing of virtual base stations. It is evaluated that SDRs with their physically small size and low power consumption are convenient to design and implement of systems of the future [11-13], vehicle-to-vehicle communication

systems [11], Global Navigation Satellite System sensors [12] and IoT applications [13], [14]. HackRF One is a low cost SDR. It is critical to know which frequency band range the communicating devices operate in when using the HackRF One (low cost) device for signal capture. The HackRF package includes the ANT500 (telescopic antenna). The frequency band range of many communication protocols can be used with HackRF, which operates in the 1MHz-6GHz range. GNU Radio can be used to create a programming interface [15].

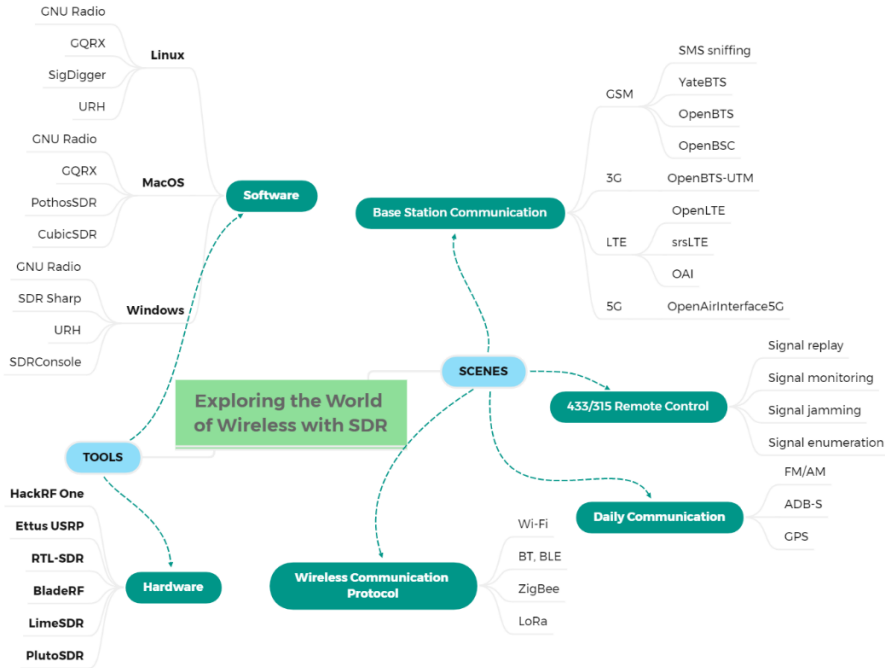


Figure 1 Exploring the Wireless World with SDR.

### 2.2 RF Fingerprints and Datasets

The RF spectrum given in Figure 2, which is part of the natural electromagnetic radiation spectrum, is between 3 kHz and 300 GHz frequency values. The spectrum used by wireless systems such as cell phones, radio and television broadcasts is in the critical frequency range. This spectrum covers frequencies in the [225 MHz to 3.7 GHz] range.

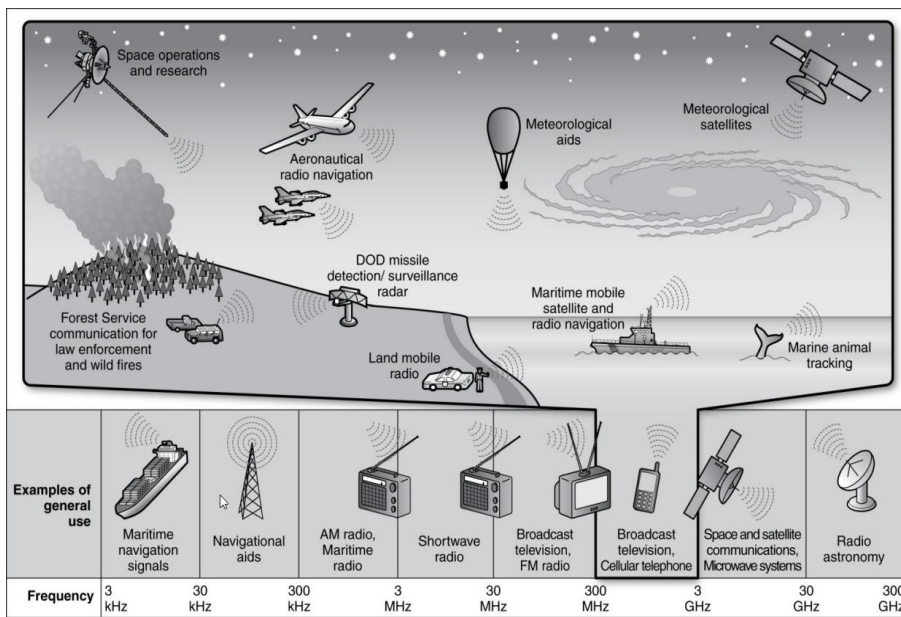


Figure 2 Dedicated Spectrum Uses and Federal Spectrum Uses with a Significant Value [16].

Sound perceivers identify the speaker by using unique variations and some aspects of the sound. RFF can mimic human speech in this regard. RFF uses the signal's time/frequency domain properties to automatically identify various radio and wireless devices. Almost all current and upcoming wireless communication standards employ OFDM [17].

What features of the signal are commonly extracted and what conclusions are drawn described below. In RF fingerprint capture, [18] uses SDR platform. Scrambling seed (from Descrambler), SFO (from Channel Estimator), CFO, and frame transient are the main features extracted (from OFDM Synchronizer). According to the article's conclusion, the results show that it is possible to identify Wi-Fi devices. And [19] conducted RFF on ZigBee devices using the SDR platform. In that study, DCTF, CFO, modulation shift, and I-Q shift properties were obtained. PSD coefficients are used in [20]. Because of the high performance of high-end receivers, it is emphasized when defining the RFF that identification accuracy is strictly related to the receiver. The identification accuracy of PSD coefficients and SNR is examined [21]. [22] used PSD as RFF for device identification. The identification performance degrades as the distance increases due to the multipath channel effect. The LTSP is subtracted from the time-domain signal received. The PSD is computed after the FFT. CFOs can also be calculated using a combination of different inputs.

In the literature, a number of learning datasets (protocol classifiers) for wireless communication have been published. Due to the acceleration in education, healthcare, e-commerce, computer vision and NLP in AI/ML and the lack of a common standard for organizing datasets, they are not yet integrated with a standard framework. Practitioners may be unable to access datasets because they are unaware of their existence [39]. Table 1 presents a summary excerpt from each explicitly available RF fingerprint datasets to educate those in this field.

Table 1 Summary Table of RFF Datasets

Made-up/ real-life	Freq. (GHz)	Waveform	Emmitter	Emmitter Count	Receiver	Dataset Ref.	Dataset Format
real-life	2.4	bluetooth	smartphones	86	TDS7404 Tektronix	[23]	.txt
real-life	2.4	out of standard	drone far controller	17	MSOS604A Keysight	[24]	.mat
real-life	1.09	ADS-B	aircraft	100	BladeRF	[25]	.mat
real-life	1.09	ADS-B	aircraft	>140	B210 (USRP)	[26]	.mat
made-up	2.45	Wi-Fi 2	X310	16	B210 (USRP)	[27]	SigMF
made-up	2.4065	out of standard	M100 Dji	7	X310 (USRP)	[27]	SigMF
made-up	2.685	Wi-Fi 2, LTE, 5G	X310	4	B210	[27]	SigMF
made-up	2.432	Wi-Fi 2/3	X310, N210	20	N210 (USRP)	[28]	SigMF

### 2.3 Classification of Features

The PLF are obtained by using the waveforms of the captured RF signals. It is categorized as position-dependent features and radio metrics that is position-independent features.

Position-independent Features: Due to defects in its analog components and manufacturing process, each transmitter has a separate RFF [29]. Device flaws are used to detect fingerprints used to identify devices. Some of these flaws include channel width, oxide thickness, and channel doping [30]. The primary goal of FE is to obtain an RFF profile that can be used to distinguish one transmitter from another. Previously, researchers [31] constructed an RFF using PSD and normalized PSD coefficients. Hall et al. [32] employs distinctive properties such as phase, amplitude, phase angle, and frequency. They use the DWT to extract these properties. The power amplifiers are the final component of the transmitter board. It is not easy for attackers to directly damage amplifiers with software. Power amplifier defects are also used in PL identification. Non-linear properties of power amplifiers can be

modeled using the Volterra series [33]. Passive Radiometric Device Identification System (PARADIS) was proposed by [34]. It makes use of frame size and phase errors, as well as I/Q origin offset and sync correlation. [35] uses transmitter phase shift and carrier frequency differences as fingerprints. It identifies devices with a second-order cyclic feature (SOCF). FE in radiometric techniques can be divided into transient and steady-state properties [34]. Transient-based methods [36] are adaptable but difficult to implement. It is based on time and frequency. I/Q instances are used as features in steady-state methods. Modulation-based methods have a better structure, but first the modulation scheme must be understood.

**Position-dependent Features:** The primary goals of RFF techniques (RFFTs) are to locate the device emitting the signal and the device from which signal originates [37]. RSS is an essential feature used in position-based RFFTs [38]. RSS is directly affected by the transmit power and channel attenuation of the transmitter. CSIR is another feature in classification and is extremely sensitive to motion. Furthermore, because location-related features are extremely sensitive to environmental changes, they cannot be used as individual fingerprints [30].

### 3. Feature Extracting

Transient features extracted from on/off transients or transmitted RF signal variations (envelope and phase shift of the transient signal) are used in device identification. The received signal is processed to extract stable features (such as SFO, CFO, and modulation features) [39]. A summary of studies with RFF in the literature is given in Table 2.

Table 2 An Overview of RFF Research Published in the Literature.

Based on	Year / Ref.	Parameter/ Method	Devices	Classification	Performance
modulation	2008 [34]	IQ offset, frequency error, phase and magnitude error, sync correlation.	802.11 NICs	SVM & k-NN	99.9% / SVM, 97% / k-NN
modulation	2009 [40]	spectral PCA features, modulation shape.	JCOP NXP 4.1 cards & e-passports	Mahalanobis distance	classification acc. 100%, identification acc. 97.5%
modulation	2017 [41]	IQ imbalance.	Matlab simulation	SVM	$\geq 90\%$ (SNR $\geq 15$ dB)
modulation	2019 [42]	IQ imbalance & DC offset	Phones, laptops & drones	CNN	98.6% (dataset:[27])
modulation	2020 [43]	time-domain RF signal	NI N210 & NI X310	CNN	Training and testing $\geq 87.41\%$
transient	2012 [44]	variance-based threshold.	Bluetooth transceivers	k-NN (obtaining energy envelope with STFT)	99.9%
transient	2009 [45]	variance-based threshold.	IEEE 802.15.4	Mahalanobis distance	$\geq 99.5\%$
transient	2014 [46]	phase based.	GSM phones	SVM	100%
wavelet	2009 [47]	Dual-tree complex wavelet transform	Wi-Fi 2 cards	Fisher-based MDA	$\geq 98\%$ (SNR $\geq 25$ dB)
wavelet	2019 [48]	Three-stage wavelet decomposition.	micro-UAV controllers	k-NN, SVM, DA, neural networks	k-NN 96.3%, SVM 96.84%
wavelet	2011 [49]	Wavelet packet decomposition, dynamic wavelet fingerprint.	Avery-Dennison AD 612 & Runway Gen 2	k-NN, SVM, LDC and QDC	99%



machine learning	2020 [50]	Time-domain RF signal	Four BS in the POWDER platform	CNN (augmented with triplet loss)	99.98% for 10 slices majority voting
machine learning	2020 [51]	RF signal spectrum (STFT method: RF signal to spectrum )	5 transmitters simulation	CNN	99.7%
deep-learning	2018 [52]	Bispectrum (Specific emitter identification (SEI))	E310, B210 & N210	CNN	>87%
deep-learning	2019 [53]	DCTF	CC2530 ZigBee modules	CNN	99.1% (SNR=30dB)
deep-learning	2020 [54]	Time-domain RF signal	Wi-Fi & ADS-B devices	CNN	Task 4F, 92.5% (per-transmission ADS-B accuracy)
deep neural networks	2020 [55]	Multiple data bursts	Dji M100 UAVs	CNN	>99%
transient & steady	2017 [56]	Empirical Mode Decomposition in SEI	Mobile phones & WLAN cards	SVM	transient >%93 (correct identification rate) SNR>0dB

### 3.1 Based on Modulation

The received frequency domain signal is used to FE. CNN (CFO) occurs when the carrier frequency is out of synchronization (when the signal down-conversion at the receiver (Rx) and the signal up-conversion at the transmitter (Tx) are inconsistent). The inter-carrier interference effect is caused by the CFO. OFDM performance is influenced by inter-carrier interference. SFO occurs when the sampling rate between the receiver and transmitter front ends is not synchronized. If the system is out of sync, the signal thus received may not be demodulated afterwards. The CFO is effective in synchronizing the system, it is calculated with the symbols LTSP and STSP. In the CFO's calculation, the literature uses the Moose algorithm [57] The CFO's  $\epsilon$  is represented by:

$$\mathbf{y}[\mathbf{n} + N_t] = \mathbf{y}[\mathbf{n}]e^{j\frac{2\pi N_t \epsilon}{N_t} F.T} \rightarrow \mathbf{Y}[\mathbf{n} + N_t] = \mathbf{Y}[\mathbf{n}]e^{2\pi\epsilon} \quad (1)$$

That is, the estimated CFO in the frequency domain:

$$\epsilon = \frac{1}{2\pi} \angle \left( \frac{\sum_{n=0}^{N_t-1} I_m\{y_1^*[n]y_2[n + N_t]\}}{\sum_{n=0}^{N_t-1} R_e\{y_1^*[n]y_2[n + N_t]\}} \right) \quad (2)$$

Even though the CFO is calculated and compensated at the receiver, it is calculated with LSTP for greater accuracy. The two are combined to calculate the OFDM system's CFO. It should be noted that the CFO is subject to change and thus requires constant supervision. When calculating SFO, the sliding window method is used to find the start of the data symbol [58]:

$$\delta = \arg \min \sum_{i=\delta}^{N_t-1+\delta} J_{SFO} \quad (3)$$

The cost function of estimated SFO:

$$J_{SFO} = |\mathbf{y}[\mathbf{n} + i] - \mathbf{y}[\mathbf{n} + N + i]| \quad (4)$$

A is the amplitude,  $\phi$  is the phase imbalance.  $y_I(t)$  In-phase (I),  $y_Q(t)$  quadrature (Q) paths outputs. If  $\hat{y}(t)$  is the ideal receive signal, the I-Q imbalance will have an effect on it:

$$\begin{aligned} \hat{y}(t) &= y_I(t) + y_Q(t) \\ &= R_e\{y(t)\} + jI_m\{Ae^{i\phi}y(t)\} \end{aligned} \quad (5)$$

I and Q are  $y_I(t) = \cos(\omega_0 t)$  and  $\widehat{y}_Q(t) = \sin(\omega_0 t)$ .  $\omega_0$  is the baseband signal. After RF signal is down-converted to baseband, the baseband signal that affects the I-Q imbalance [59] is:

$$\begin{aligned} \widehat{y}_I(t) &= \alpha \cos(\omega_0 t) + \widehat{\beta}_I \\ \widehat{y}_Q(t) &= \sin(\omega_0 t + \varphi) + \widehat{\beta}_Q \end{aligned} \quad (6)$$

Where  $\alpha = 1/A$  and  $\varphi$  are amplitude and phase errors caused by the I-Q imbalance defined above.  $\widehat{\beta}_I$  and  $\widehat{\beta}_Q$  are the DC biases of I and Q paths after down-converting. Removing these biases and substituting by  $\sin(\omega_0 t + \varphi) = \sin(\omega_0 t) \cos(\varphi) + \cos(\omega_0 t) \sin(\varphi)$ , the baseband signal has the following matrix form:

$$\begin{bmatrix} \widehat{y}_I(t) \\ \widehat{y}_Q(t) \end{bmatrix} = \begin{bmatrix} \alpha & \mathbf{0} \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} y_I(t) \\ y_Q(t) \end{bmatrix} \quad (7)$$

$\alpha$  and  $\varphi$  can be calculated as:

$$\begin{aligned} \langle y_I(t) \cdot y_I(t) \rangle &= \alpha^2 \langle \cos^2(\omega_0 t) \rangle = \frac{\alpha^2}{2} \\ \rightarrow \alpha &= \sqrt{2 \langle y_I(t) \cdot y_Q(t) \rangle} \end{aligned} \quad (8)$$

$$\begin{aligned} \langle y_I(t) \cdot y_Q(t) \rangle &= \frac{\alpha^2}{2} \sin(\varphi) \\ \rightarrow \varphi &= \sin^{-1} \left( (\alpha^2/2) \langle y_I(t) \cdot y_Q(t) \rangle \right) \end{aligned} \quad (9)$$

In the literature, CFO, SFO, amplitude shift and phase shift properties are commonly extracted from modulation-based signals. It makes use of the IQ components (in-phase and quadratic signal data) of signals collected at large scales from two different wireless standards (COTS: commercial ready and ADS-B: used for aircraft status updates). Modulations contain information providing unique signature about I-Q imbalance, phase noise, and carrier frequency shift while an information signal is being transmitted to an other device [54].

### 3.2 Based on transient

With non-stationary characteristics, it is not easy to separate TS and channel noise from each others. In this section, Bayesian Step Change Detector, Bayesian Ramp Change Detector, Variance Fractal Dimension Threshold Detector, Phase Detector, Average Point of Change Detector, Permutation Entropy, and Supremacy of Energy Criteria approaches are examined.

#### 3.2.1 Bayesian step change detector

Based on Higuchi's method in [65], the variance of the fractal dimension is calculated for successive parts of the signal. In this case, the fractal dimension variance between two consecutive sequences is proportional to ppDF (posteriori probability distribution function). The sample instant to which the maximum value calculated from the probability distribution function (pDF) belongs is found as the transient starting instant as in Fig 3. To do this, first, subsets of samples are rearranged:

$$X(m, k): X(m), X(m+k), \dots, X\left(m + \left\lceil \frac{N-m}{k} \right\rceil \times k\right) \quad (10)$$

$X(m, k)$  is the subset interval,  $m$  is the start time, and  $k$  is the interval time. Calculation the length of the curve  $L_m(k)$  is, its for each subset is:

$$L_m(k) = \left\{ \left( \sum_{i=1}^{\frac{N-m}{k}} |x(m+ik) - x(m+(i-1)k)| \right) \frac{N-1}{\left\lceil \frac{N-m}{k} \right\rceil k} \right\} / k \quad (11)$$

The mean value of  $k$  clusters is plotted, a log-log scale ( $L_m(k)$ ). After the curve fitting is done, the fractal dimension is calculated using the slope of the curve.

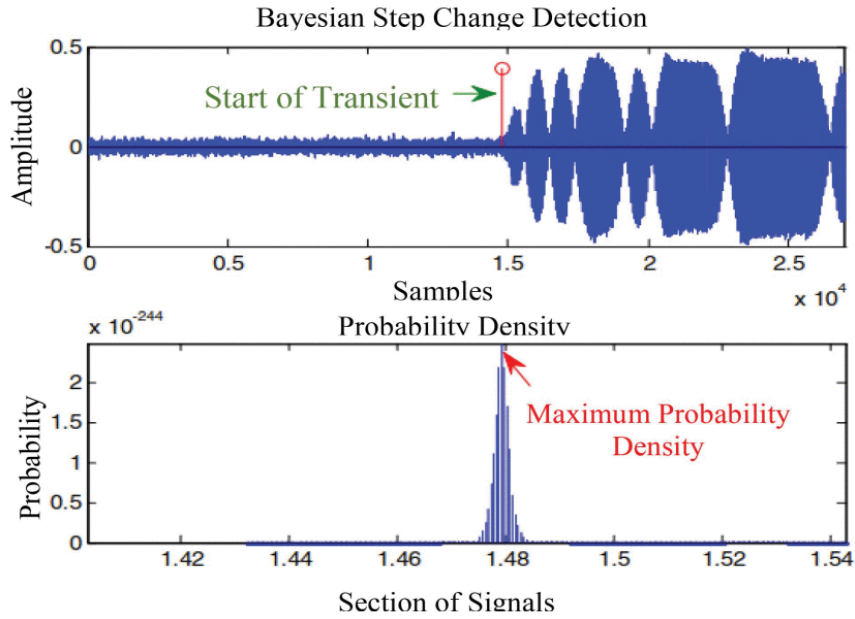


Figure 3 Bayesian step change detection on a sample signal [60]

The beginning of the transition ( $m$ ) is detected by tracking the ppDF in Equation 12. Here,  $N$  and  $d$  are the number of samples in a window and the fractal dimension respectively.

$$P(\{m\} | d) \propto \frac{1}{\sqrt{m(N-m)}} \left[ \sum_{i=1}^N d_i^2 - \frac{1}{m} (\sum_{i=1}^m d_i)^2 - \left( \frac{1}{N-m} (\sum_{i=m+1}^N d_i)^2 \right)^{\frac{N-2}{2}} \right] \quad (12)$$

### 3.2.2 Bayesian ramp change detector

Ureten and Serinken [61] proposed BRCD which is a modification of the BSCD. Transient due diligence is performed by estimating the point at which the signal's strength gradually increases. Prior to the transmission of actual data, typical transmission data includes channel noise. This signal's model is written in Equation 13 equation in matrix form.

$$d = Gb + e \quad (13)$$

$d$  is sample array in  $N \times 1$  dimension,  $G$  is a  $N \times M$  matrix of the basis functions estimated for each sample in the time series,  $b$  is an array in  $M \times 1$  dimension consist of linear coefficients, and  $e$  is  $N \times 1$  array of Gaussian noise examples. For change point determination, posteriori probability density is used [61]:

$$P(\{m\} | d, I) \propto \frac{[d^T d - d^T G(G^T G)^{-1} G^T d]^{\frac{N-m}{2}}}{\sqrt{\det(G^T G)}} \quad (14)$$

$I$  represents the pattern of the signal, the position of the starting point (SP) can be found in the matrix  $G$  in "as seen in Equation 15".

$$G^T = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 2 & 3 & \dots & N-m \end{bmatrix} \quad (15)$$

BRCD is more useful for Wi-Fi signals because it has 3 times lower standard deviation detection error than that of BSCD [62]. In fact, BRCD causes gradually an increase in power, such as Wi-Fi [60].

### 3.2.3 Variance fractal dimension threshold detector

VFDTD was proposed in [36]. It computes the fractal size of signal amplitude variance when detecting Wi-Fi transmitter transients. Furthermore, the VFDTD implementation is as follows [60]:

Calculation of fractal size of each segment of the signal given in Equation 16 where  $H$  denotes the Hurst index giving the correlation between  $\Delta X(t_i, \Delta t)$  and  $\Delta t$ .  $\Delta X(t_i, \Delta t)$  is amplitude difference between any two points of the signal, and  $\Delta t$  is sampling time that is  $\Delta t = |t_{i+1} - t_i|$ .

$$D(t) = 2 - H \tag{16}$$

Hurst index in the equation is calculated as in Equation 17 using least squares regression. In the equation,  $(x_i, y_i)$  pair is the pair of  $(\log(\Delta t_i), \log(\text{var}(\Delta X(t_i, \Delta t_i))))$ .

$$2H = \frac{N \sum_{i=1}^N x_i y_i - (\sum_{i=1}^N x_i)(\sum_{i=1}^N y_i)}{N(\sum_{i=1}^N x_i^2) - (\sum_{i=1}^N x_i)^2} \tag{17}$$

It is critical to select an appropriate time sequence and to ensure that there are enough  $(x_i, y_i)$  pairs. Next, we need to determine the SPoTS using the fractal size we have obtained, and then adjust the threshold  $\tau$  to be the average of the fractal size of the channel noise. If a set of values is less than the threshold value as given in Equation 18, then  $n$  is SPoTS.

$$D(n), D(n + 1), \dots, D(n + 450) \leq \tau \tag{18}$$

Figure 4 depicts the start of a wireless network card network core's temporal and fractal trajectory. The fractal dimension of the channel noise and crossover signal appears to differ significantly. These features determine the starting point's location, making it simple and quick. The threshold, on the other hand, is sensitive to noise and can only be determined through trial and error.

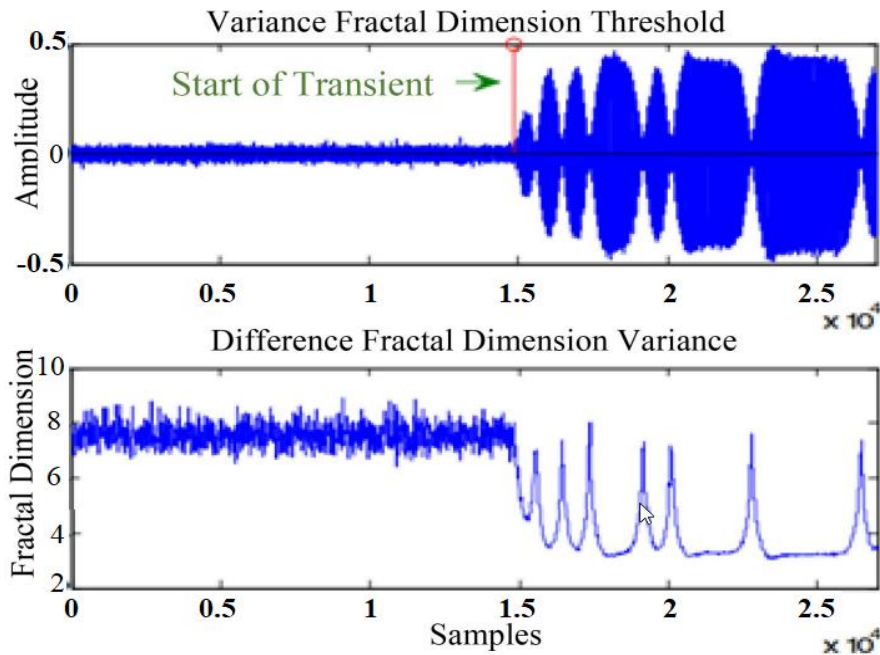


Figure 4 Variance fractal dimension threshold detection on a sample signal [60].

### 3.2.4 Phase detector

J. Hall proposed phase detection [32], which uses PCs. This method can be defined as follows: The Hilbert transform of a real signal can receive an analytical signal as in Equation 19 and 20.

$$X(t) = I(t) + jQ(t) \tag{19}$$

$$\theta(t) = \tan^{-1} \left[ \frac{Q(t)}{I(t)} \right] \tag{20}$$

$Q(t) = (s_q^a(n))$  and  $I(t) = (s_i^a(n))$ . However, the instantaneous phase of the signal in Equation 20, is unwrapped to remove the discontinuities caused by multiples of  $2\pi$  radians. Each element's AV (absolute value) in unwrapped vector is obtained as in Equation 21 in this method.

$$AV = \begin{cases} \theta(t) & |\theta(t) - \theta(t - 1)| \leq \pi \\ \theta(t) \pm 2\pi & \text{others} \end{cases} \quad (21)$$

TV (variance of phase) is calculated for each successive portion of AV to magnify the variation between the noise and transient portions of the signal as in Equation 22. To do this, size of a non-overlapping window (s) is used.

$$TV(i) = \text{var}(\overline{AV}(d + 1), \overline{AV}(d + 2), \dots, \overline{AV}(d + g)) \quad (22)$$

In the previous equation, i indeks takes values interval of  $[1, N/s]$ , g is  $i \times s$ , d is  $(i-1) \times s$ , and var represents the phase's variance. Finally, the difference in phase variance (PV) is calculated using Equation 23 in order to generate the fractal trajectory.

$$VT = |TV_i - TV_{i+1}|, i = 1, 2, \dots, \frac{N - w}{s} \quad (23)$$

It is obvious that the PV of a TS changes more slowly than the PV of noise. The onset of a TS and the detection of its fractal trajectory by PD are depicted in Figure 5. All in all, the onset of a transient state can be easily detected using this characteristic.

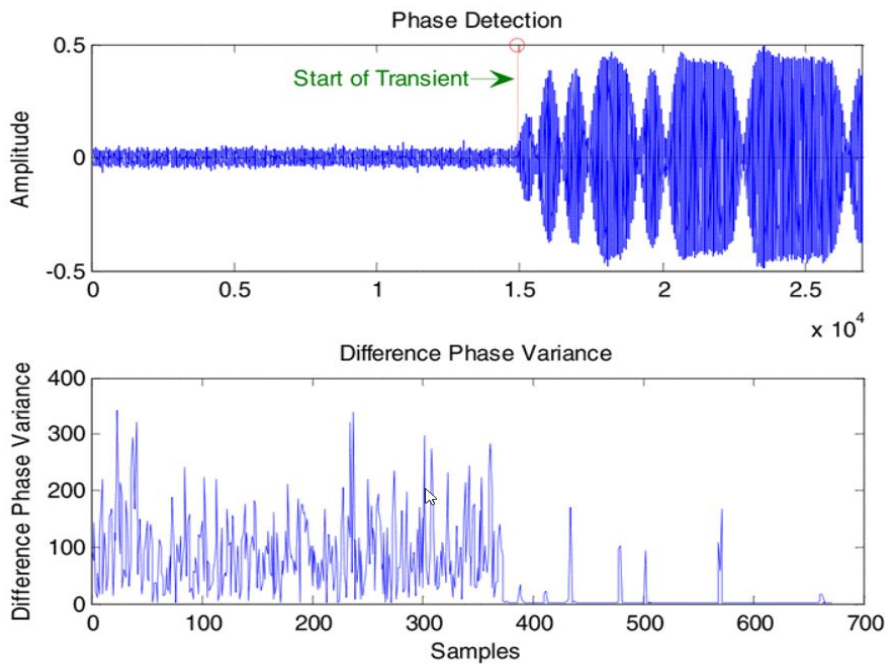


Figure 5 Phase detection on a sample signal [60].

Phase properties are used in PD sensing. Because noise has little effect on phase properties. It is quick and simple to detect the onset of the transient by changing the phase variance fractal trajectories; thus, its computational power is low and its robustness is high; however, there is a threshold problem [60].

### 3.2.5 Mean change point detector

In this method, difference between the statistics of the samples is taken as main principle. As can be seen from Figure 6, the sampling moment or index where the greatest difference is found is taken as the SPoTS [60].

The temporal vector is divided into two parts:  $x_1, x_2, \dots, x_{i-1}$ , and  $x_i, x_{i+1}, \dots, x_N$ . The average and statistics of each section are calculated as in following equations.

$$S_i = \sum_{n=1}^{i-1} (x_n - \bar{X}_{i1})^2 + \sum_{n=i}^N (x_n - \bar{X}_{i2})^2 \quad (24)$$

$\bar{X}$  is the mean of the combined partitions and the statistics ( $S$ ) of the real sample are expressed below:

$$S = \sum_{n=1}^N (x_n - \bar{X})^2 \quad (25)$$

The point having the largest amplitude of the  $S - S_i$  curve is the start point of the transient. The idea behind MCPD is to enlarge the difference, and then determine the where the maximum value occurs to be starting point of the transient [60].

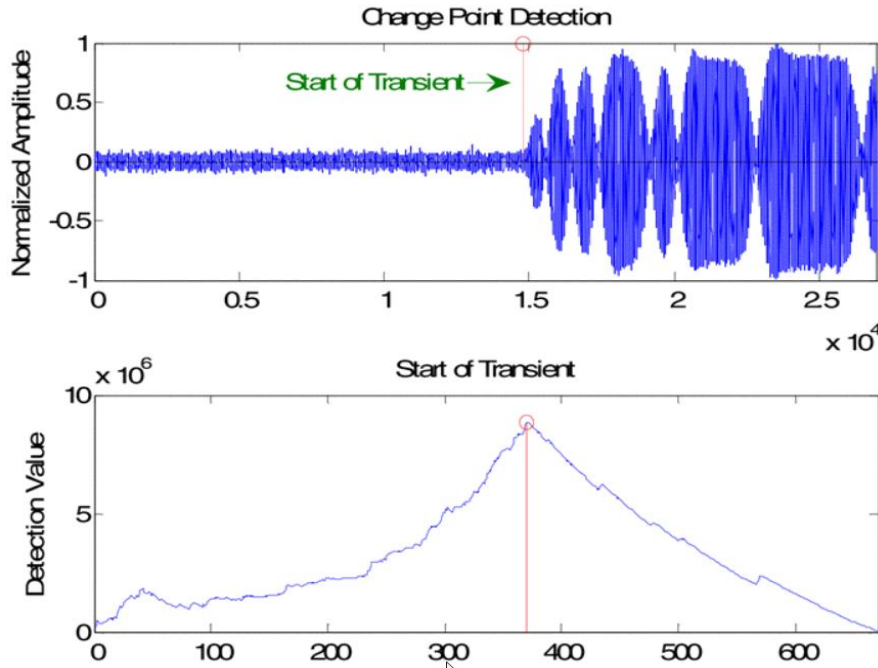


Figure 6 Mean change point detection on a sample signal [60].

### 3.2.6 Permutation entropy (PE) and generalized likelihood ratio test detector

Bandt-Pompe introduced PE, which can assess the irregularity and complexity of time series [63]. This method detects a TS using PE and GLRTDs. A GLRTD is used to determine the SP of the captured signal's PE [64]. The  $X_i, (i = 1, 2, \dots, N)$  time series are formed in an  $m$ -dimensional space as follows to calculate the PE:

$$X_i = [x(i), x(i + l), \dots, x(i + (m - 1)l)] \quad (26)$$

where  $l$  is the time lag,  $x(i)$  denotes the  $i$ -th point in  $m$ -dimensional space;  $1 \leq i \leq N - (m - 1)l$ . The actual  $X_i$  values in Equation 26 are then sorted in ascending as in Equation 27:

$$X_i = [x(i + (j_1 - 1)l) \leq x(i + (j_2 - 1)l) \leq \dots \leq x(i + (j_m - 1)l)] \quad (27)$$

When an equality occurs, sorting can be done according to their corresponding index of  $j$ . That is, if  $j_{n1} < j_{n2}$ , then the order is  $x(i + (j_{n1} - 1)l) \leq x(i + (j_{n2} - 1)l)$ , else the order is  $x(i + (j_{n2} - 1)l) \leq x(i + (j_{n1} - 1)l)$ . A permutation pattern  $\pi$  can be used to map the vector  $X_i$ :

$$\pi_i = [j_1, j_2, \dots, j_m] \quad (28)$$

$j$  in Equation 28 is the time index. One of the  $m!$  permutations of  $m$  different signs is  $\pi_i$ . The probability of finding  $\pi_i$  is easily calculated with  $p(\pi_i) = f(\pi_i) / (N - (m - 1)l)$ . In the equation  $f(\pi_i)$  is the number of occurrences of  $\pi$ . Finally, Shannon Entropy is used to calculate the PE [65]:

$$0 \leq H_p = - \sum_{j=1}^K p_j \ln p_j / \ln (m!) \leq 1 \quad (29)$$

In Equation 29,  $K$  is the number of different signs  $[\pi_1, \pi_2, \dots, \pi_{N-(m-1)l}]$ . SPoTS can be identified using PE data. Firstly, the PE trajectory of the transient can be computed using a rectangular window of length  $L_{wnd}$  that scrolls one sample at a time. A signal's PE is smaller than the noise series' PE. The main reason for this is the noise's irregularity. The following equation can be used to easily model the PE trajectory:

$$H_P(n) = \begin{cases} H_{pn}(n) & 1 \leq n \leq n_0 \\ H_{pt}(n) & n_0 \leq n \leq n_1 \\ H_{ps}(n) & n_1 + 1 \leq n \leq N \end{cases} \quad (30)$$

where  $n$  denotes the  $n$ th slide,  $H_p$  is the corresponding PE,  $N$  is the total of sliding,  $n_0$  is the first time there is a TS in the sliding window, and  $n_1$  is the last time there is a TS in the sliding window.  $H_{pn}$  is the probability of noise;  $H_{pt}$  is the probability of TS in the sliding window; and  $H_{ps}$  is the probability of stable signal. It is self-evident that  $H_{pn} > H_{pt} > H_{ps}$ .

PE begins to decrease when there is a transient in a sliding window, and PE changes slightly for a stable signal in the sliding window. The PE for slides with a TS is modelled as follows: [64]:

$$H_P(n) = \begin{cases} A_0 + w(n) & 1 \leq n \leq n_0 \\ A_0 + k \times (n - n_0) + w(n) & n_0 \leq n \leq N_0 \end{cases} \quad (31)$$

In Equation 31,  $w(n)$  denotes Gaussian noise with zero-mean and  $\sigma$  standard deviation;  $A_0$  is the mean of  $H_{pn}(n)$ ;  $k$  is the decreasing slope after  $n_0$ . When  $T_0$  is the mean PE,  $n_0$  is the first slide containing the TS;  $N_0$  denotes the changing point when  $n \leq N_0$ ,  $H_{pn}(n) > T_0$  and  $H_{pn}(N_0 + 1) \leq T_0$  and is computed as in Equation 32 and 33.

$$T_0 = \frac{\max(H_P) + \min(H_P)}{2} \quad (32)$$

The binary hypothesis test can be used to solve the transient detection problem:

$$H_0: A_0 + w(n) \\ H_1: \begin{cases} A_0 + w(n) & 1 \leq n \leq n_0 \\ A_0 + k \times (n - n_0) + w(n) & n_0 \leq n \leq N_0 \end{cases} \quad (33)$$

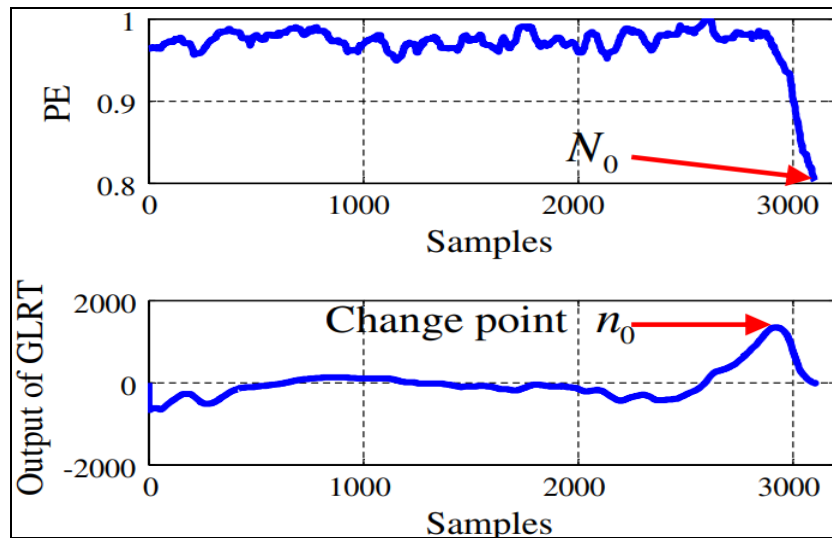


Figure 7 For a PE Trajectory (upper) Output of GLRT Dedector (lower) [64].

$H_P(n)$ 's GLRTD can be represented as: [66]:

$$L_G(x) = \frac{p(x; n_0, H_1)}{p(x; H_0)} = \frac{p(x; A_1 = \hat{A}_0, A_2 = \hat{A}_0 + \hat{k} \times (n - n_0), H_1)}{p(x; A_1 = \hat{A}_0)} \quad (34)$$

$p(x; n_0, H_1)$  and  $p(x; A_1)$  are computed as shown below; since  $A_0$  and  $k$  are unknown, instead of them their MLE can be used [67, 68].

$$p(x; A_1, A_2) = \frac{1}{(2\pi\sigma^2)^{N_0/2}} \exp \left[ -\frac{1}{2\sigma^2} \left( \sum_{n=1}^{n_0} (x(n) - A_1)^2 + \sum_{n=n_0+1}^{N_0} (x(n) - A_2)^2 \right) \right] \quad (35)$$

$$p(x; A_1) = \frac{1}{(2\pi\sigma^2)^{N_0/2}} \exp \left[ -\frac{1}{2\sigma^2} \left( \sum_{n=1}^{N_0} (x(n) - A_1)^2 \right) \right] \quad (36)$$

To determine  $A_0$  under the two hypotheses  $H_0$  and  $H_1$ , let the MLE of  $A_0$  under  $H_0$  and  $H_1$  be  $\hat{A}_{00}$  and  $\hat{A}_{01}$ , respectively.

$$\hat{A}_{00} = \hat{A}_0 = \frac{1}{N_0} \sum_{n=1}^{N_0} H_p(n) \quad (37)$$

$$\hat{A}_{01} = \hat{A}_0 = \frac{1}{n_0} \sum_{n=1}^{n_0} H_p(n) \quad (38)$$

The least squares fitting algorithm can estimate the MLE of slope  $k$  algorithm [65] and is provided below:

$$\hat{k} = \frac{(N_0 - n_0) \sum_{n=1}^{N_0-n_0} n H_p(n + n_0) - \sum_{n=1}^{N_0-n_0} n \sum_{n=1}^{N_0-n_0} H_p(n + n_0)}{(N_0 - n_0) \sum_{n=1}^{N_0-n_0} n^2 - (\sum_{n=1}^{N_0-n_0} n)^2} \quad (39)$$

The GLRTD is defined using the above equations as follows [64]:

$$\begin{aligned} & \text{Ln} \left( L_G(H_p(n)) \right) \\ &= \frac{1}{2\sigma^2} \left[ \sum_{n=1}^{N_0} (H_p(n) - \hat{A}_{00})^2 - \sum_{n=1}^{n_0} (H_p(n) - \hat{A}_{01})^2 \right. \\ & \quad \left. - \sum_{n=n_0+1}^{N_0} (H_p(n) - \hat{A}_{01} - \hat{k} \times (n - n_0))^2 \right] \end{aligned} \quad (40)$$

The GLRTD's maximum is the estimated SPoTS  $n_0$  [64]:

$$\hat{n}_0 = \arg \max_n \left[ \text{Ln} \left( L_G(H_p(n)) \right) \right] \quad (41)$$

As shown in Figure 7, when the PE trajectory falls down, the GLRTD output occurs in there, which can be identified as the change point  $n_0$ . It is reasonable to conclude that a very small number of signal samples in the sliding window cannot result in a noticeable decrease in the PE trajectory.

### 3.2.7 Superiority of energy criterion

This technique, which is widely used to predict the arrival time of signals in a variety of applications, is also a pioneer in detecting acoustic and electromagnetic partial discharges. The basic idea behind energy criterion (EC) is to characterize the arrival of a signal by a change in energy content. A sampled signal's ( $x$ ) energy ( $E_i$ ) is the sum of its amplitude values. [69], [70].

$$E_i = \sum_{k=0}^i x_k^2, i = 1, \dots, N \quad (42)$$

The length of the signal is represented by  $N$ . As follows, the signal is isolated from the noise component:

$$E'_i = E_i - i\delta = \sum_{k=0}^i (x_k^2 - i\delta) \quad (43)$$

$\delta$  in Equation 43, defined as in Equation 44.



$$\delta = \frac{E_N}{\vartheta \cdot N} \tag{44}$$

The  $\vartheta$  factor lessens the delay effect of  $\delta$ . As a result, the parameters that  $\delta$  influence are the total energy of the signal ( $E_N$ ) and the  $\vartheta$  factor. Two methods can be used to use the EC technique for transient SP detection: the EC (EC-a) method based on  $\mathbf{a}(\mathbf{n})$  features and the EC (EC- $\emptyset$ ) method based on  $AV(\mathbf{n})$  features.

When using the EC-a method to calculate ( $E'_i$ ), we first use the  $\mathbf{a}(\mathbf{n})$  features of the analytical signal found in Equation 45.

$$\mathbf{a}(\mathbf{n}) = \sqrt{(s_I^a(\mathbf{n}))^2 + (s_Q^a(\mathbf{n}))^2} \tag{45}$$

The energy curve's global minimum is then defined. The sample corresponding to the global minimum is used to determine the starting point of the transient. However, within a flat region, there may be several local minimums. In this case, the transition SP can be determined by selecting the region's first local minimum. It should be noted that the  $\delta$  factor chosen has a significant influence on the energy curve as seen in Equation 44. The value of the  $\delta$  factor under noise-free conditions is  $\delta = [1, 2, \dots, 100]$  [70]. When considering different SNR levels, the  $\delta$  factor value should be determined empirically. In this context, they discovered that when  $\delta = 30$  for the given data set, the detection accuracy increases significantly [71].

Figure 8 illustrated the energy curve computed using EC-a for  $\delta = 1, 2, 30$  and the discovered starting points.

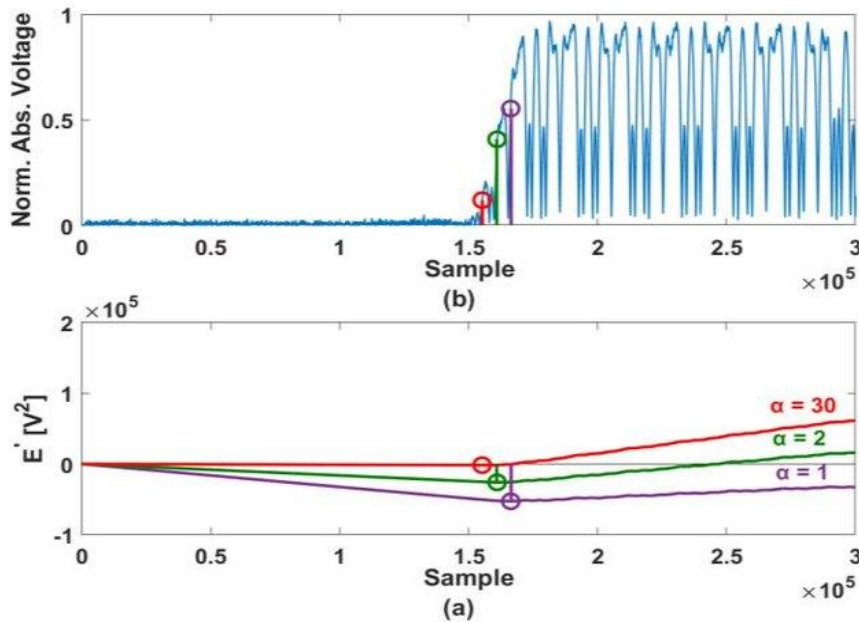


Figure 8 Illustration the energy curve obtained by EC-a method (lower) and the determined transient starting point (upper) [71].

The EC- a method is based on using  $AV(\mathbf{n})$  in Equation 46.

$$AV(\mathbf{n}) = \begin{cases} \emptyset(\mathbf{n}) & |\emptyset(\mathbf{n}) - \emptyset(\mathbf{n} - 1)| \leq \pi \\ \emptyset(\mathbf{n}) \pm 2\pi & \text{otherwise} \end{cases} \tag{46}$$

The basic logic is to generate another random signal with roughly equal variance by using the random change in the noise portion of the signal's unwrapped IP features. In the noise part of the signal, a monotonically increasing energy curve is expected to be obtained using this signal. The starting point is the global maximum point of the curve. As a result, the method begins by calculating the absolute differences between each mean window of the signal's unwrapped instantaneous PCs. After calculating  $E'_i$ , the maximum of the curve is shown in Figure 9(a). In the unwrapped instantaneous PCs

of the signal, the example corresponding to the window index providing the global maximum of the curve is defined in Figure 9(b). Figure 9(c) shows the determination of the starting point.

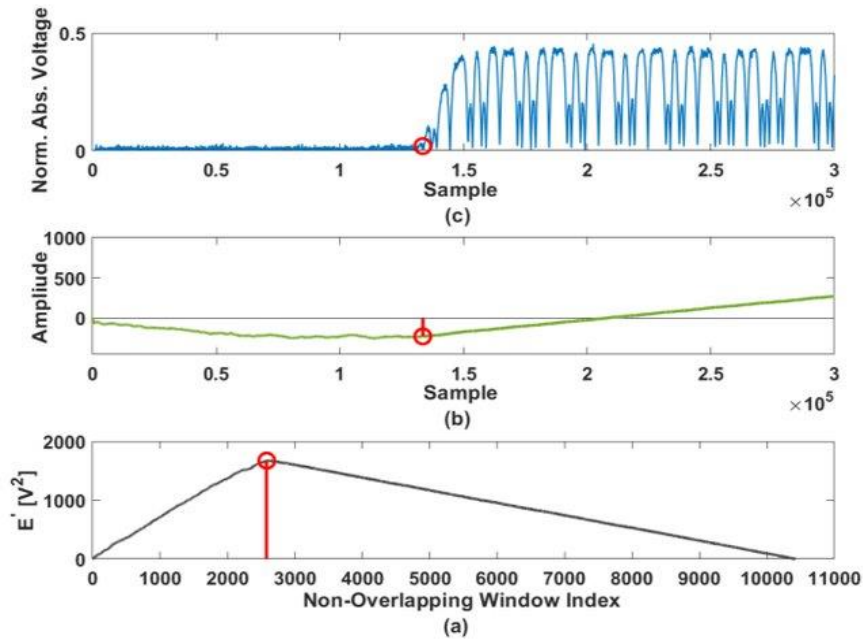


Figure 9 (a) Energy curve generated by the EC-  $\emptyset$  method, (b) instantaneous phase signal, (c) the determined of the starting point [71].

The transient-based signal characteristic recognition algorithms available in the literature are summarized in Table 3.

Table 3 Summary Table Transient Detection Algorithms.

Algorithms Ref.	Pros	Cons	Complexity	Success Rate	Signal/SNR
BSCD [72]	no threshold needed, high detection rate (hiDeR) (with suitable amplitude/without leading response).	weak detection (with small amplitude) for TS, need a long time, complicated calculation.	$O(n^3)$	% 80-85  Wi-Fi 1 transceiver	Radio/NA
BRCd [73]	no threshold needed, outperforming BSCD.	works well in signal models with linear power increases, complex calculation.	N/A	% 95  Wi-Fi 1 transceiver	Wi-Fi 1/NA
VFDTD [74]	hiDeR	threshold needed, highly sensitive to noise, need long time, complicated calculation.	$O(n^2)$	N/A	Radio/NA
PD [75]	fast and simple	less susceptible to noise, practically define a starting point, poor detection rate in low SNR.	$O(n)$	% 85-90  Wi-Fi 1 transceiver	Bluetooth/NA
MCPD [76]	no threshold needed, high detection, simple	takes long time to compute.	$O(n)$	% 90-92.5  8 different transmitters	Wi-Fi/6-30dB

PE & GLRT [64]	no threshold needed, hiDeR, detection of the start point is extremely accurate.	complicated calculation.	N/A	N/A	GSM/0-25dB
EC [71]	more effective different SNR levels	N/A	O(n)	N/A	Wi-Fi/-3-25db

### 3.3 Based on Steady-State

The unrivaled features extracted from the modulated signals are the focus of some steady-state studies. Gerdes and colleagues in their work [77], they proposed a based on steady-state RFF and preferred cards of the same manufacturer and model. The IEEE Ethernet 802.3 input part is used to identify the fingerprint profile, as well as the device emitting the signal. In the classification, a basic threshold and a matching filter application were used. [34] proposed a PARADIS. This system identifies the physical layer of a modulated signal based on five properties (frequency error, synchronous correlation, I/Q origin offset, magnitude and phase errors). The k-NN and SVM classifiers were used to create the RFF profile. To demonstrate the classifier's accuracy, 138 identical model Wi-Fi 1 signals are used. These signals were captured with a vector signal analyzer at distances in the interval of (3,15) meters from the antenna. With their proposed method, Shi and Jensen hoped to define Multiple Input Multiple Output devices. It has become a system comparable to PARADIS by utilizing the radiometric properties of these devices in modulation [78]. They used modulation-based approaches to classify RFID devices. They make use of spectral features from RFID transmitters as well as modulation features. Four different RFID transmitter classes and models are tested in the study (ISO 14443, HF 13.56 MHz) [40]. Frequency domain features were used in the study to identify RFF transmitters. The use of FFT allows for a great deal of flexibility in spectral feature selection. In laboratory testing, eight USRP transmitters are used. Optional feature selection the k-NN discriminator is used to generate the classification engine automatically. It achieves 97 and 66 percent accuracies at 30dB SNR and at 0dB SNR respectively. It also provides a less expensive alternative to the its counter approach requiring very high speed ADCs [79]. Suski and colleagues for their unique feature selection, they used the PSD coefficients in the Wi-Fi 2/3 signal input [80]. Table 4 details the IEEE 802.11 standards [81-89]. Integration employs the feature selection method, as opposed to other known feature selection methods (RELIEF-F, F Score, and Laplacian Score). The covariance feature is used as an RF fingerprint, and the K-Nearest Neighbor (KNN) classifier is used. The Spearman correlation coefficient is used to assess the method's stability [90]. It is a fact that the steady state component of the signal is not shared by all transmitters. On the other hand, transient part of the signal is always present. As a result, the study focuses on transient-based RFF. It is a significant difficulty to obtain the amplitude of the signal, in this context a higher sampling rate is needed to be able to detect the starting of the transient [79]. WLAN, RFID, and almost all other technologies use preamble as it simplifies receiver design at the start of transmission. Therefore, these approaches do not require a steady-state signal [31]. For deep learning RFF approaches, Yu et al. offer a general Denoising Auto Encoder based model. A partially stacking technique has also been developed for efficiently identifying ZigBee devices using both quasi-stable and steady-state RFFs. Under AWGN channels at lower SNRs (-10 dB to 5 dB), their suggested PSCDAE beats traditional CNN by 14 to 23.5 percent in terms of identification accuracy [91].

Table 4 Information of IEEE 802.11 Standards.

Release date	Standard	Common name	Freq. (GHz)	Modulation type	Bandwidth (MHz)	Data speed (bps)	Approx. range (meter)	Number of clients
1997	802.11	Wi-Fi 0	2.4	DSSS, FHSS	22	2 M	20-100	N/A
1999	802.11a	Wi-Fi 2	5	DSSS	20	54 M	35-120	N/A
1999	802.11b	Wi-Fi 1	2.4	CCK	22	11 M	35-140	N/A
2003	802.11g	Wi-Fi 3	2.4	OFDM	20	54 M	38-140	N/A
2009	802.11n	Wi-Fi 4	2.4 & 5	OFDM	20-40	600 M	70-250	<50
2013	802.11ac	Wi-Fi 5	5	OFDM	20-40-160	6.9 G	35-...	50-100

2019	802.11ax	Wi-Fi 6	2.4 & 5	OFDM, OFDMA	80-160	9.6 G	N/A	200-400
2020	802.11ax	Wi-Fi 6E	6	OFDMA	80-160	9.6 G	N/A	200-400
Expected in (2 <sup>nd</sup> half of 2022)	802.11be	Wi-Fi 7	2.4, 5 & 6	OFDMA	320	30 G	N/A	N/A

### 3.4 Other Methods

These approaches typically employ a proprietary wireless technology and/or extract additional signal and logical layer features [92], [47]. The PL is described by Danev et al. using the modulation pattern, spectral characteristics, and timing of device response signals. Timing and modulation are used to distinguish devices from various manufacturers, while spectral features are used to identify devices from the same manufacturer when fingerprints are used to identify devices [40]. Jana and Kasera identified access points in a wireless local area network using clock skew as a distinguishing feature [92]. In [93], the effectiveness of this technique for complex networks has been demonstrated. The results demonstrated that various access points could be distinguished with high accuracy. 802.11a OFDM signal devices are defined by a complex wavelet transform. MDA is used to categorize the features [47], [94]. To identify wireless devices, Suski et al. [95] generates an RF fingerprint. It makes use of the PSD of the Wi-Fi 2 preamble and spectral correlation is used for classification. When the SNR value of the captured packet frames is greater than six decibels, the average classification error rate is 20 percent in this method, which was tested on three devices. Recent research has focused on various RFID classes for PL identification [96], [97]. Periaswamy et al. [97], [98] used UHF-RFID tags to identify devices. According to the results of the study, the minimum power response feature can be used to identify devices with a 94.4 percent success rate. Recently, researchers looked into various signal characteristics and signal components [99], [6] in GSM devices. They identified and classified devices from four different manufacturers by using the intermediate and temporal parts of the GSM-GMSK burst signals. When the results of GSM signal identification are examined, it is discovered that the near temporal part is more effective in classification accuracy, while the mid-level part is less effective. Padilla et al. assess system performance using 20 Wi-Fi device datasets with 15 fingerprint samples per device. Both methods combine subspace transform-based feature reduction techniques with similarity-based analysis techniques such as PCA and PLS regression as identification methods. When only one device per manufacturer is used, accuracy is greater than 90%, and accuracy is around 70% when two devices per manufacturer are used [100]. To extract RFF features, the DCTF, a two-dimensional representation of the differential relationship of signal time series, is used. When defining devices, the developed DCTF-CNN is used [53]. Furthermore, HHT [101], EMD and Welch methods, which are employed in signal classification in several domains, will add to the literature if used to RFF [102].

## 4. Classification Methods

The classification methods used in the literature can be summarized as in Figure 10. As seen in the figure, methods divide into two main category as supervised and unsupervised. Unsupervised learning is not effective if there is prior tag information about devices. For Wi-Fi fingerprinting, infinite hidden Markov random field (ihMrf) based unsupervised clustering techniques are proposed using online classification algorithm and batch updates [103]. Transmitter features are used in [35], where Bayesian approach passively classifies equipments unsupervised.

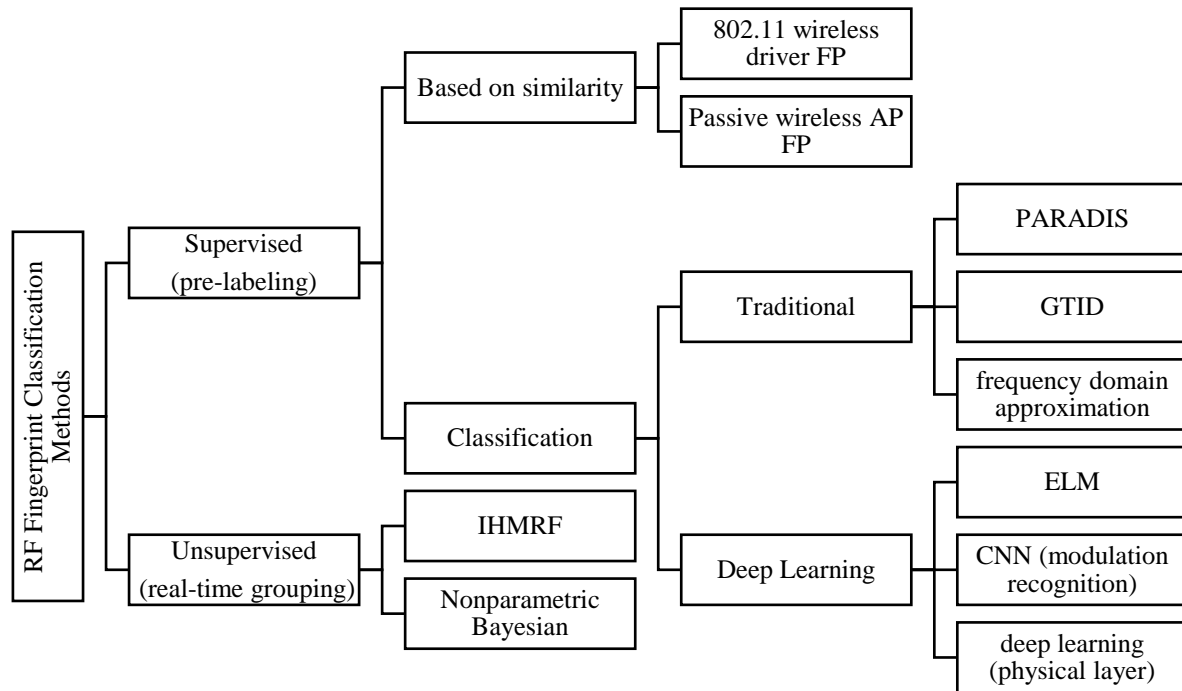


Figure 10 Perspective on RFF Classification.

In supervised learning, the network requires multiple labeled samples gathering prior to deployment to train for ML algorithm [104]. Below are studies using supervised learning-based methods in four different categories.

**Based on Likeness:** Comparing the observed signature of the transmitting device with records in a master database is necessary for similarity metrics. A passive fingerprint technique has been proposed in [105], to identify the Wi-Fi device driver running on an IEEE 802.11. Analysis of the collected traces and fingerprinting of device drivers is done using the Supervised Bayes approach. Using wavelet analysis [106] describes a passive black box-based technique that uses the time from TCP or UDP packet to determine type of access points. These techniques are based on priory knowledge about vendor-specific features.

**Classification-Based:** As can be seen in Figure 10, there are studies in the literature on classification-based supervised learning that makes use of RF features such as I/Q and phase imbalance, frequency error and RSS.

**Traditional:** In traditional classification, matching with pre-selected features is examined using the domain knowledge of the system. To do this, dominant features must be known beforehand. The method proposes a classification based on subtracting known input parts and calculating spectral ingredients. The log spectral energy property is given as input to the k-nearest neighbors (KNN) discriminant classifier [79]. PARADIS achieves 99% accuracy using SVM and KNN algorithms, fingerprinting 802.11 Wi-Fi devices, based on modulation specific errors in the frame [34]. A structure called GTID is proposed for physical device classification with artificial neural networks. This structure takes advantage of variations in clock skewness as well as hardware combinations of devices [107]. They investigated the problem of detecting and classifying micro-UAV control signals. The proposed detection method executes a Bayesian approach based on the Markov models of UAV and non-UAV classes, while the classification method relies on energy-time domain RF signal and uses features (skewness ( $\gamma$ ), variance ( $\sigma^2$ ), energy spectral entropy ( $H$ ), and kurtosis ( $\kappa$ )) extracted in this domain [48]. The mathematical calculations of these properties are given in the following equations.

$$\kappa = \frac{1}{L\sigma^4} \sum_{n=1}^L (\alpha(n) - \mu)^4 \quad (47)$$

$$\gamma = \frac{1}{L\sigma^3} \sum_{n=1}^L (\alpha(n) - \mu)^3 \quad (48)$$

$$\sigma^2 = \frac{1}{L} \sum_{n=1}^L (\alpha(n) - \mu)^2 \quad (49)$$

$$\mu = \frac{1}{L} \sum_{n=1}^L \alpha(n) \quad (50)$$

Choosing an appropriate feature set is an important huge duty when using many different features. When there are many devices, scalability problems may occur. This causes to increased computational complexity in training.

Deep Learning is also a popular approach in recent years using supervised learning. It is a network structure consisting of many layers, capable of solving complex problems by processing big data. It implements deep learning at the PL, focusing on modulation recognition using CNNs [108] [109]. However, it does not define a device as Riyaz et al. does, it only defines the modulation type used by the transmitter [104]. Three deep learning models (CNN, LSTM and MLP) were found successful in the literature, considering the characteristics of IQ samples, FFT results and spectrogram. In the literature, it appears that CNN is efficient in processing spatially relevant data such as images (spectrogram) in deep learning, whereas LSTM is efficient in temporally related time series (IQ samples) [110]. AlexNet, GoogleNet, VGG16, and ResNet are examples of popular CNN models. In the RF field, two new models are proposed as deep CNN architectures, inspired by Alex-Net and ResNet [54].

Extreme Learning Machines (ELM) was proposed for single layer neural networks by G. Huang in 2006. It presents a fast and not iterative numeric supervised learning. ELM provides good generalization performance at extremely fast learning speed [111, 112]. One of its most important features is that it does not require iterative calculations based on derivatives. It uses pseudo inverse computations for determining networks parameters [113]. This area is very untouched in RFF. A long time is required for the training of the above-mentioned structures used in deep learning. ELMs have the potential to reduce this time to extremely low values with an extremely fast operation. In this context, it is considered as an open field that is recommended to be used in the future.

## 5. Conclusion

Finally, this review focuses on the rapid development and widespread use of IoT and the security part. Considering the IoT's own hardware resources, the use of RFF to ensure security due to the error experienced during production at the physical layer draws attention. Therefore, RFF methods for Wi-Fi communication devices have been reviewed. Essentially, unique features from Wi-Fi communication devices are extracted and adapted to two-factor authentication systems for identification purposes. SDRs take the lead in signal capture and preprocessing to support different communication protocols. This review gives a summary of the most recent RFF detection and extraction techniques. FE methods used for different fingerprinting methods are detailed in this review.

## Acknowledgments

This study is supported by the project numbered 2021-01.BŞEÜ.01-01 within the scope of Bilecik Şeyh Edebali University Scientific Research Projects.

## References

- [1] (03.10.2022). *Hackersnewbulletin*. "Chinese Irons have hidden chips which serve malware in systems". Available: <http://www.hackersnewsbulletin.com/2013/11/russia-chinese-irons-hidden-chips-serve-malware-systems.html>
- [2] M. Z. Gündüz and R. Daş, "Internet of things (IoT): Evolution, components and applications fields," *Pamukkale University Journal of Engineering Sciences*, vol. 24, pp. 327-335, 2018.
- [3] D. Nouichi, M. Abdelsalam, Q. Nasir, and S. Abbas, "Iot devices security using rf fingerprinting," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, 2019, pp. 1-7.
- [4] J. Pohl and A. Noack, "Universal radio hacker: A suite for analyzing and attacking stateful wireless protocols," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, 2018.
- [5] B. Danev, D. Zanetti, and S. Capkun, "On physical-layer identification of wireless devices," *ACM Computing Surveys (CSUR)*, vol. 45, pp. 1-29, 2012.
- [6] M. D. Williams, M. A. Temple, and D. R. Reising, "Augmenting bit-level network security using physical layer RF-DNA fingerprinting," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, pp. 1-6.
- [7] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, state-of-the-art, and challenges," *Computer Communications*, vol. 128, pp. 106-125, 2018.
- [8] M. Gummineni and T. R. Polipalli, "Implementation of reconfigurable transceiver using GNU Radio and HackRF One," *Wireless Personal Communications*, pp. 1-17, 2020.
- [9] H. Miyashiro, M. Medrano, J. Huarcaya, and J. Lezama, "Software defined radio for hands-on communication theory," in *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2017, pp. 1-4.
- [10] M. Sruthi, M. Abirami, A. Manikoth, R. Gandhiraj, and K. Soman, "Low cost digital transceiver design for Software Defined Radio using RTL-SDR," in *2013 international mutli-conference on automation, computing, communication, control and compressed sensing (iMac4s)*, 2013, pp. 852-855.
- [11] W. Xiang, F. Sotiropoulos, and S. Liu, "xRadio: an novel software defined radio (SDR) platform and its exemplar application to vehicle-to-vehicle communications," in *International Conference on Ad-Hoc Networks and Wireless*, 2015, pp. 404-415.
- [12] J. Seo, Y.-H. Chen, D. S. De Lorenzo, S. Lo, P. Enge, D. Akos, *et al.*, "A real-time capable software-defined receiver using GPU for adaptive anti-jam GPS sensors," *Sensors*, vol. 11, pp. 8966-8991, 2011.
- [13] Y. Chen, S. Lu, H.-S. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge, "A low power software-defined-radio baseband processor for the Internet of Things," in *2016 IEEE international symposium on high performance computer architecture (HPCA)*, 2016, pp. 40-51.
- [14] Y. Park, S. Kuk, I. Kang, and H. Kim, "Overcoming IoT language barriers using smartphone SDRs," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 816-828, 2016.
- [15] J. N. Samuel, "Specific emitter identification for GSM cellular telephones," University of Pretoria, 2018.
- [16] B. Skorup, "Reclaiming federal spectrum: Proposals and recommendations," *Colum. Sci. & Tech. L. Rev.*, vol. 15, p. 90, 2013.
- [17] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio," in *Proceedings of the second workshop on Software radio implementation forum*, 2013, pp. 9-16.
- [18] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi devices using software defined radios," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016, pp. 3-14.
- [19] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid RF fingerprint extraction and device classification scheme," *IEEE Internet of Things Journal*, vol. 6, pp. 349-360, 2018.

- [20] S. U. Rehman, S. Alam, and I. T. Ardekani, "An overview of radio frequency fingerprinting for low-end devices," *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, vol. 6, pp. 1-21, 2014.
- [21] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Radio-frequency fingerprinting for mitigating primary user emulation attack in low-end cognitive radios," *IET Communications*, vol. 8, pp. 1274-1284, 2014.
- [22] T.-Y. Lin, C.-M. Lai, and C.-W. Chen, "Using SDR Platform to Extract the RF Fingerprint of the Wireless Devices for Device Identification," in *CS & IT Conference Proceedings*, 2020.
- [23] E. Uzundurukan, Y. Dalveren, and A. Kara, "A database for the radio frequency fingerprinting of Bluetooth devices," *Data*, vol. 5, p. 55, 2020.
- [24] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Drone remote controller RF signal dataset," *IEEE Dataport*, 2020.
- [25] Y. Liu, J. Wang, S. Niu, and H. Song, "ADS-B signals records for non-cryptographic identification and incremental learning," *IEEE, Piscataway, NJ, USA, Data Set*, 2021.
- [26] Y. Liu, J. Wang, H. Song, S. Niu, and Y. Thomas, "A 24-hour signal recording dataset with labels for cybersecurity and IoT," *IEEE, Piscataway, NJ, USA, Data Set*, 2020.
- [27] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A Comprehensive Survey on Radio Frequency (RF) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges," *arXiv preprint arXiv:2201.00680*, 2022.
- [28] A. Al-Shawabka, F. Restuccia, S. D'Oro, and T. Melodia, "Massive-Scale I/Q Datasets for WiFi Radio Fingerprinting," *Computer Networks*, vol. 182, p. 107566, 2020.
- [29] S. Dolatshahi, A. Polak, and D. L. Goeckel, "Identification of wireless users via power amplifier imperfections," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 1553-1557.
- [30] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 94-104, 2015.
- [31] P. Scanlon, I. O. Kennedy, and Y. Liu, "Feature extraction approaches to RF fingerprinting for device identification in femtocells," *Bell Labs Technical Journal*, vol. 15, pp. 141-151, 2010.
- [32] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase," *Wireless and Optical Communications*, pp. 13-18, 2003.
- [33] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, "Identifying wireless users via transmitter imperfections," *IEEE Journal on selected areas in communications*, vol. 29, pp. 1469-1479, 2011.
- [34] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008, pp. 116-127.
- [35] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device fingerprinting to enhance wireless security using nonparametric Bayesian method," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1404-1412.
- [36] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, pp. 222-233, 2020.
- [37] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 111-122.
- [38] R. Zekavat and R. M. Buehrer, *Handbook of position location: Theory, practice and advances* vol. 27: John Wiley & Sons, 2011.
- [39] Y. Ren, L. Peng, W. Bai, and J. Yu, "A practical study of channel influence on radio frequency fingerprint features," in *2018 IEEE International Conference on Electronics and Communication Engineering (ICECE)*, 2018, pp. 1-7.
- [40] B. Danev, T. S. Heydt-Benjamin, and S. Capkun, "Physical-layer Identification of RFID Devices," in *USENIX security symposium*, 2009, pp. 199-214.
- [41] Y. Huang, "Radio frequency fingerprint extraction of radio emitter based on I/Q imbalance," *Procedia computer science*, vol. 107, pp. 472-477, 2017.




- [42] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized radio classification through convolutional neural networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019, pp. 370-378.
- [43] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, *et al.*, "Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, 2020, pp. 646-655.
- [44] S. U. Rehman, K. Sowerby, and C. Coghill, "RF fingerprint extraction from the energy envelope of an instantaneous transient signal," in *2012 Australian Communications Theory Workshop (AusCTW)*, 2012, pp. 90-95.
- [45] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *2009 International Conference on Information Processing in Sensor Networks*, 2009, pp. 25-36.
- [46] Y. Yuan, Z. Huang, H. Wu, and X. Wang, "Specific emitter identification based on Hilbert-Huang transform-based time-frequency-energy distribution features," *IET communications*, vol. 8, pp. 2404-2412, 2014.
- [47] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Journal of Communications and Networks*, vol. 11, pp. 544-555, 2009.
- [48] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, 2019, pp. 1-13.
- [49] C. Bertocini, K. Rudd, B. Noursain, and M. Hinders, "Wavelet fingerprinting of radio-frequency identification (RFID) tags," *IEEE Transactions on Industrial Electronics*, vol. 59, pp. 4843-4850, 2011.
- [50] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. R. Chowdhury, "Trust in 5G open RANs through machine learning: RF fingerprinting on the POWDER PAWR platform," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 2020, pp. 1-6.
- [51] L. Zong, C. Xu, and H. Yuan, "A rf fingerprint recognition method based on deeply convolutional neural network," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020, pp. 1778-1781.
- [52] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE Communications Letters*, vol. 22, pp. 2591-2594, 2018.
- [53] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based RF fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 1091-1095, 2019.
- [54] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, *et al.*, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, pp. 50-57, 2020.
- [55] N. Soltani, G. Reus-Muns, B. Salehi, J. Dy, S. Ioannidis, and K. Chowdhury, "RF fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 15518-15531, 2020.
- [56] J.-H. Liang, Z.-T. Huang, and Z.-W. Li, "Method of empirical mode decomposition in specific emitter identification," *Wireless Personal Communications*, vol. 96, pp. 2447-2461, 2017.
- [57] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Transactions on communications*, vol. 42, pp. 2908-2914, 1994.
- [58] S. Kaur, C. Singh, and A. S. Sappal, "Effects and estimation techniques of symbol time offset and carrier frequency offset in OFDM system: Simulation and analysis," *International Journal of Electronics and Computer Science Engineering*, pp. 1188-1196, 2012.
- [59] S. Ellingson, "Correcting IQ imbalance in direct conversion receivers," *Argus Technical and Scientific Documents*, 2003.
- [60] L. Huang, M. Gao, C. Zhao, and X. Wu, "Detection of Wi-Fi transmitter transients using statistical method," in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, 2013, pp. 1-5.
- [61] O. Ureten and N. Serinken, "Bayesian detection of Wi-Fi transmitter RF fingerprints," *Electronics Letters*, vol. 41, pp. 373-374, 2005.

- [62] O. Ureten and N. Serinken, "Wireless security through RF fingerprinting," *Canadian Journal of Electrical and Computer Engineering*, vol. 32, pp. 27-33, 2007.
- [63] Y. Cao, W.-w. Tung, J. Gao, V. A. Protopopescu, and L. M. Hively, "Detecting dynamical changes in time series using the permutation entropy," *Physical review E*, vol. 70, p. 046217, 2004.
- [64] Y.-J. Yuan, X. Wang, Z.-T. Huang, and Z.-C. Sha, "Detection of radio transient signal based on permutation entropy and GLRT," *Wireless Personal Communications*, vol. 82, pp. 1047-1057, 2015.
- [65] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, p. 174102, 2002.
- [66] S. Kay, "Volume II: Detection Theory," *Fundamentals of Statistical Signal Processing; PTR Prentice Hall: Upper Saddle River, NJ, USA*, pp. 465-466, 1993.
- [67] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*: Prentice-Hall, Inc., 1993.
- [68] A. Candore, O. Kocabas, and F. Koushanfar, "Robust stable radiometric fingerprinting for wireless devices," in *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, 2009, pp. 43-49.
- [69] S. M. Markalous, S. Tenbohlen, and K. Feser, "Detection and location of partial discharges in power transformers using acoustic and electromagnetic signals," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 15, pp. 1576-1583, 2008.
- [70] C. Herold, T. Leibfried, S. Markalous, and I. Quint, "Algorithms for automated arrival time estimation of partial discharge signals in power cables," in *Proc. Int. Symp. High Volt. Eng.(ISH)*, 2007.
- [71] I. S. Mohamed, Y. Dalveren, and A. Kara, "Performance assessment of transient signal detection methods and superiority of energy criterion (EC) method," *IEEE Access*, vol. 8, pp. 115613-115620, 2020.
- [72] M. Barbeau, J. Hall, and E. Kranakis, "Detection of rogue devices in bluetooth networks using radio frequency fingerprinting," in *proceedings of the 3rd IASTED International Conference on Communications and Computer Networks, CCN*, 2006, pp. 4-6.
- [73] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, 2007, pp. 331-340.
- [74] C. Zhao, T. Y. Chi, L. Huang, Y. Yao, and S.-Y. Kuo, "Wireless local area network cards identification based on transient fingerprinting," *Wireless Communications and Mobile Computing*, vol. 13, pp. 711-718, 2013.
- [75] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Physica D: Nonlinear Phenomena*, vol. 31, pp. 277-283, 1988.
- [76] I. Mohamed, Y. Dalveren, F. O. Catak, and A. Kara, "On the Performance of Energy Criterion Method in Wi-Fi Transient Signal Detection," *Electronics*, vol. 11, p. 269, 2022.
- [77] R. M. Gerdes, T. E. Daniels, M. Mina, and S. Russell, "Device Identification via Analog Signal Fingerprinting: A Matched Filter Approach," in *NDSS*, 2006.
- [78] Y. Shi and M. A. Jensen, "Improved radiometric identification of wireless devices using MIMO transmission," *IEEE Transactions on Information Forensics and Security*, vol. 6, pp. 1346-1354, 2011.
- [79] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *2008 IEEE 68th Vehicular Technology Conference*, 2008, pp. 1-5.
- [80] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Radio frequency fingerprinting commercial communication devices to enhance electronic security," *International Journal of Electronic Security and Digital Forensics*, vol. 1, pp. 301-322, 2008.
- [81] E. J. Oughton, W. Lehr, K. Katsaros, I. Selinis, D. Bublely, and J. Kusuma, "Revisiting wireless internet connectivity: 5G vs Wi-Fi 6," *Telecommunications Policy*, vol. 45, p. 102127, 2021.

- [82] R. B. M. Abdelrahman, A. B. A. Mustafa, and A. A. Osman, "A Comparison between IEEE 802.11 a, b, g, n and ac Standards," *IOSR Journal of Computer Engineering (IOSR-JEC)*, vol. 17, pp. 26-29, 2015.
- [83] B. Mitchell, "Wireless Standards 802.11 a, 802.11 b/g/n, and 802.11 ac," *Verkköjulkaisu. Saatavissa: <http://compnetworking.about.com/cs/wireless80211/a/aa80211standard.htm> [viitattu 8.4. 2015]*, 2015.
- [84] R. Khanduri and S. Rattan, "Performance Comparison Analysis between IEEE 802.11 a/b/g/n Standards," *International Journal of Computer Applications*, vol. 78, pp. 13-20, 2013.
- [85] B. Bellalta, "IEEE 802.11 ax: High-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, pp. 38-46, 2016.
- [86] D.-J. Deng, K.-C. Chen, and R.-S. Cheng, "IEEE 802.11 ax: Next generation wireless local area networks," in *10Th international conference on heterogeneous networking for quality, reliability, security and robustness*, 2014, pp. 77-82.
- [87] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11 ax high efficiency WLANs," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 197-216, 2018.
- [88] D. López-Pérez, A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11 be extremely high throughput: The next generation of Wi-Fi technology beyond 802.11 ax," *IEEE Communications Magazine*, vol. 57, pp. 113-119, 2019.
- [89] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11 be, the future Wi-Fi 7," *IEEE access*, vol. 8, pp. 88664-88688, 2020.
- [90] Y. Li, Y. Lin, Z. Dou, and Y. Chen, "Research on RF Fingerprint Feature Selection Method," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1-5.
- [91] J. Yu, A. Hu, F. Zhou, Y. Xing, Y. Yu, G. Li, et al., "Radio frequency fingerprint identification based on denoising autoencoders," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 1-6.
- [92] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE transactions on Mobile Computing*, vol. 9, pp. 449-462, 2009.
- [93] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 93-108, 2005.
- [94] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet denoising to improve OFDM-based signal detection and classification," *Security and Communication Networks*, vol. 3, pp. 71-82, 2010.
- [95] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills, "Using spectral fingerprints to improve wireless network security," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, 2008, pp. 1-5.
- [96] D. Zanetti, B. Danev, and S. Capkun, "Physical-layer identification of UHF RFID tags," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, 2010, pp. 353-364.
- [97] S. C. G. Periaswamy, D. R. Thompson, and J. Di, "Fingerprinting RFID tags," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, pp. 938-943, 2010.
- [98] S. Chinnappa Gounder Periaswamy, D. R. Thompson, H. P. Romero, and J. Di, "Fingerprinting radio frequency identification tags using timing characteristics," in *Radio Frequency Identification System Security*, ed: IOS Press, 2010, pp. 73-81.
- [99] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improved wireless security for GMSK-based devices using RF fingerprinting," *International Journal of Electronic Security and Digital Forensics*, vol. 3, pp. 41-59, 2010.
- [100] J. Padilla, P. Padilla, J. Valenzuela-Valdés, J. Ramírez, and J. Górriz, "RF fingerprint measurements for the identification of devices in wireless communication networks based on feature reduction and subspace transformation," *Measurement*, vol. 58, pp. 468-475, 2014.
- [101] S. KARASU and Z. SARAÇ, "Güç kalitesi bozulmalarının hilbert-huang dönüşümü, genetik algoritma ve yapay zeka/makine öğrenmesi yöntemleri ile sınıflandırılması," *Politeknik Dergisi*, vol. 23, pp. 1219-1229, 2020.

- [102] M. Tosun and O. Çetin, "Ampirik Mod Ayırıştırması ve Welch Yöntemini Kullanarak Dört Sınıflı Motor Hayali EEG Sinyallerinin Derin Öğrenme ile Sınıflandırılması," *Avrupa Bilim ve Teknoloji Dergisi*, pp. 284-288, 2021.
- [103] F. Chen, Q. Yan, C. Shahriar, C. Lu, W. Lou, and T. C. Clancy, "On passive wireless device fingerprinting using infinite hidden markov random field," *submitted for publication*, 2017.
- [104] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, pp. 146-152, 2018.
- [105] J. A. Van Randwyk, J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, and D. Sicker, "Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting," Sandia National Lab.(SNL-CA), Livermore, CA (United States)2006.
- [106] K. Gao, C. Corbett, and R. Beyah, "A passive approach to wireless device fingerprinting," in *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, 2010, pp. 383-392.
- [107] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "GTID: A technique for physical device and device type fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, pp. 519-532, 2014.
- [108] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, 2016, pp. 213-226.
- [109] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," *arXiv preprint arXiv:1702.00832*, 2017.
- [110] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio Frequency Fingerprint Identification for LoRa Using Deep Learning," *IEEE Journal on Selected Areas in Communications*, 2021.
- [111] G. A. Kale and C. Karakuzu, "Multilayer extreme learning machines and their modeling performance on dynamical systems," *Applied Soft Computing*, vol. 122, p. 108861, 2022.
- [112] Z. Katılmış and C. Karakuzu, "ELM based two-handed dynamic turkish sign language (TSL) word recognition," *Expert Systems with Applications*, vol. 182, p. 115213, 2021.
- [113] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in EEG classification," *Mathematical Problems in Engineering*, vol. 2015, 2015.

# FA-AODV: Flooding Attacks Detection Based Ad Hoc On-Demand Distance Vector Routing Protocol for VANET

 Bugra Alp Tosunoglu<sup>1</sup>,  Cemal Kocak<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Faculty of Technology, University of Gazi, balp.tosunoglu@gazi.edu.tr, 06500, Ankara, Turkey

<sup>2</sup>Department of Computer Engineering, Faculty of Technology, University of Gazi, ccckocak@gazi.edu.tr, 06500, Ankara, Turkey

Received 15 September 2022; Revised 9 October 2022; Accepted 24 October 2022; Published online 31 December 2022

## Abstract

Vehicular Ad-Hoc Networks (VANET) is anticipated to be the most effective way of increasing performance and safety in transportation soon. VANETs are the sub-branch of Ad-Hoc Networks which provide safety and comfort features together with related services for the vehicle operators. RREQ flood attack mostly encountered in the literature for security of VANET. Due to the nature of the reactive protocols, the AODV routing protocol is quite open to attack types such as flood attack. Flood attacks occur in the network layer. The impact of flood attacks is not about victim nodes, it can be also affecting the whole network. A malicious attack that could be carried out in VANET could cause accidents that would cause a serious disaster. A malicious node could penetrate the IP addresses on a Flood Attack based User Datagram Protocol (UDP) to breakdown the data communication between two vehicles. The main purpose of this paper is to detect and prevent the flood attack, during the operation of the routing protocol and to decrease the end-to-end delay on the network.

**Keywords:** Vehicular AdHoc Networks, VANET Security, Flooding Attack, NS2

## 1. Introduction

Wireless networks made up of mobile nodes behaving arbitrarily and lacking infrastructure are known as mobile ad-hoc networks (MANET). There is no specified central control, such as a base station, in these kinds of networks. Another variant of MANET is the vehicular ad-hoc network (VANET), which is a more recent technology. Wireless communication between cars and between vehicles and the Roadside Unit (RSU) is made possible via VANET [1].

A group of engineers from IBM Cooperation and Delphi Delco Electronic Systems initially proposed the VANET. This team claims that the Inter-Vehicle Communication (IVC) and Roadside to Vehicles Communication (RVC) systems of vehicle communication are together referred to as VANET. VANET technology utilizes both cellular networks and Ad-Hoc Networks for maintain a connection. The infrastructure of the VANET, on the other hand, consists of a hybrid design that combines VLAN/Cellular, Ad-Hoc, and vehicular communications (V2V, V2I, and V2R). It is acknowledged that VANET is a part of the Intelligent Transport Systems (ITS). ITS facilitates communication amongst other vehicles by utilizing their safety and service applications [2-4]. VANET routing protocols, most notably as AODV and DSR, help create a route between the source node and the destination node. These routing protocols are divided into three main classes: proactive, reactive and hybrid.

For each route entry, the AODV protocol employs the destination sequence number; this sequence number provides a loop-free connection and the shortest way. This RREQ message is forwarded by the other mobile node after being propagated from the source. This message is subsequently transmitted to the intermediate node's neighboring node. This process is repeated until the packet reaches the destination or central node. The route entry in the routing table must be a legitimate entry, which implies that the item in the table must be less than a certain value. The hop count at the intermediate node is incremented by one as the RREQ packet moves across the network. If the node receives another RREQ message with the same ID, the packet is dropped. When an intermediate node or destination receives an

RREQ message and has a new valid route to the destination, it generates an RREP route reply message and sends it in response to the RREQ message [5].

The main purpose of the flood attacks is to consume the resources of the assets on the network. This type of attack is the category of large routing distortion that can lead to denial of service. Flood attacks occur in the network layer. The attacker continuously sends a RREQ message to the selected node. To respond to any incoming request, specific resources are allocated to the attacker by the target node. This behavior causes the destination node to run out of resources. IP address spoof-based flood attacks are a serious and still open security issue in wireless networks. IP address spoofing creates offensive fake routing packets using addresses that are assigned to others or never assigned. Several security solutions have been proposed to solve various problems related to flood attacks in the VANETs [6-8].

Our main goal is to provide secure and fast communication between the source node and the destination using the AODV protocol. We would like to give a suggestion that we limit to a solution for the detection and prevention of RREQ message type flood attacks. This proposed mechanism identifies not only the attack but also the source of the attack and isolates the attacker from the network. With proposed mechanism, end-to-end delays, and count of dropped packages will reduce, and the number of transmitted packets will increase.

The rest of this paper is organized as follows. In Section 2, related works on background of flood attack types on VANET and special subsection for flooding attack are mentioned. Section 3 describes the proposed FA-AODV. Section 4 presents the simulation parameters and performance metrics. The performance of FA-AODV is evaluated and compared with the default AODV as well as in Section 4. Section 5 draws the conclusions.

## 2. Related Work

In section definition of flood attack in AdHoc networks and previous studies on the type of floods that are subject to the study are discussed.

**Flood Attack:** Because reactive routing packets, such as AODV and DSR, set the route by using a route request, dependence on RREQ packets makes the reactive protocols vulnerable to flood attacks. RREQ flood attacks or data flood attacks depend on packets used on the network. The purpose of the malware node in RREQ flood attacks is to generate a flood of data by sending many RREQ packets of unknown targets on the network. If the target nodes are not present in the network, RREP packets will not be created, but RREQ packets will continue to be created by all nodes. The purpose of this type of attack is to consume bandwidth and network resources.

### 2.1. Flood Attacks In VANET

Using two well-known frameworks in uncertain reasoning, namely Bayesian Inference and Dempster-Shafer (D-S) evidence theory, innovative strategies for resisting RREQ flooding attacks in Wireless Ad Hoc Networks were proposed [9]. The current study describes the modeling of RREQ traffic and the development of an optimal method for detecting persistent RREQ flooding attacks using Bayesian Inference. Using D-S evidence theory, the method was further developed to identify high and low rate pulsed RREQ flooding attacks. The suggested solution effectively resisted any sort of flooding-based DDoS attack in Wireless Ad Hoc Network with decreased communication and memory cost, according to a detailed assessment utilizing mathematical modeling and simulation.

Al-Mehdhara et al [10], propose a secure VANET architecture that makes use of a Software-Defined Networking (SDN) controller and Neural Network Self-Organizing Maps (SOMs). A Multilayer Distributed SOM (MSOM) model based on two levels of clustering and classification is used to address the shortcomings of traditional SOMs and improve SOM efficiency. Experiment findings reveal that malicious traffic is detected at 99.67%, DDoS attacks are prevented, and system security is increased.

Another SDN study, have proposed the recognition of DDoS attacks to SD-VANET based on a combination of Hyperparameter optimization and feature selection. Initially, created a dataset

containing both the characteristics of normal network traffic and DDoS attack network traffic was obtained from SD-VANET topology. Minimum Redundancy Maximum Relevance (MRMR) feature selection algorithm was used to select the most distinctive features of the dataset. Bayesian optimization method boosted with hyperparameter optimization was applied using for classifiers in the learning phase. The best accuracy score attained using MRMR feature selection and Bayesian optimization-based decision tree classifier was 99.35% [11].

Zarei et al. [12], presented the LSFA-IoT strategy, which protects the AODV routing protocol as well as the IoT network against flooding. The authors divided the project into two major phases: the first involves identifying attackers using a physical layer intrusion and attack detection system, and the second involves detecting wrong events using Average Packet Transmission RREQ (APT-RREQ) messages. The simulation results demonstrated that the suggested technique outperformed the well-known IOT protocols REATO and IRAD.

The authors compared different ML approaches to detect malicious activity and the proposed Hybrid KSVM algorithm for DDoS attack. Their Hybrid KSVM algorithms gave better results with accuracy (92.46%) and precision (95.31%) compared to other ML algorithms [13].

The authors [14] worked on Information-centric networking (ICN)-based Named Data Networking (NDN), which they believe is the future of the internet in autonomous or semi-autonomous vehicles. NDN-based VANET suffers from several security attacks, one such attack is the Interest Flood Attack (IFA), which targets the core routing mechanism of NDN-based VANET. Their suggested approach can identify both normal and low-rate IFA. The results of their experiments reveal that their technique detects and mitigates both regular and low-rate IFA in the network.

Hasan et al. [15] proposed the FLOW-AODV algorithm for detecting the flood type attack when the IP address of the attacker was hidden. The smaller average delay in FLOW-AODV, as supported by the simulation results, means that it prevents redundant RREQ overflow from frequent flooding. In addition, results based on the appearance of multiple flood attackers' algorithm maintain almost 100% PDR with less than 200 ms. delay.

### 3. Material and Methods

#### 3.1. FA-AODV Algorithm and Simulation Styling

In this study, prevention of UDP flooding type attacks made on the AODV protocol is used in VANET for providing the continuation of the communication. For this, an algorithm named Flooding Attacks detection based Ad Hoc On-Demand Distance Vector Routing Protocol (FA-AODV) has been developed. As is known, AODV protocol works on demand. As the nodes are in mobility, the road information keeps changing continuously. For this reason, during the route discovery process, every node between the source and the destination nodes decides to either relay or drop the RREQ packets. In the scenario, attacks are realized using RREQ packets. During the attack number of packets given in the table, one is triggered by all the nodes simultaneously. Thus, the blockage of the network and prevention of the communication will be ensured.

Table 1 Number of UDP flooding attack packets triggered

<b>Number of UDP Flooding Attacks Packet</b>				
	<b>20 Node</b>	<b>15 Node</b>	<b>10 Node</b>	<b>5 Node</b>
Std. AODV	11339	10081	4449	3646
FA-AODV	9445	8135	5385	3221

Flood attacks are performed using UDP or ICMP packets. In these types of traffic types, in which the SYNC mechanism would not work, the attacker sends UDP packets to randomly or previously chosen ports. The attacked nodes investigate every packet that arrives to understand the services requested. This situation would decrease the performance in nodes that is; it increases the end-to-end delays. The attacked node would check the availability of the port assigned for the incoming request and by

concluding no ports are in listening state, it would be forced to send “Destination Unreachable” messages. These unlimited numbers of packets are processed. This situation would cause the data traffic it has with the neighboring nodes to be stopped. As a result of this attack, it would cause the victim node to be unreachable by the other nodes.

In figure 1, flow diagram for FA-AODV algorithm has been displayed. According to flow diagram and hypothetical scenario, all nodes on the network listen to packet communication of other neighbor nodes by working in “promiscuous mode”. On realizing next node exceeds previously defined packet drop threshold level, that node is identified as hostile. This related node will not be used for the next packet traffic.

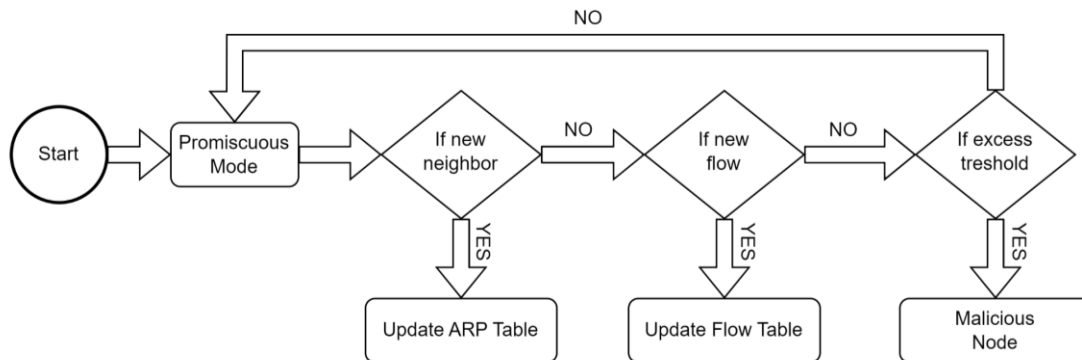


Figure 1 Flow Diagram for FA-AODV

A pseudo code developed for the FA-AODV algorithm is shown in pseudo code 1.

Code 1 FA-AODV Pseudo Code

1	Begin
2	If the sender / receiver listens to the data packet
3	Begin
4	If the expected packet
5	Begin
6	Deliver packet
7	Condition (node) = good
8	End
9	If the sender's packet timed out
10	Begin
11	If (forwarded packet)> threshold value
12	Begin
13	If Condition (node)! = Good
14	Begin
15	Generate Alarm to Source
16	Condition (Node) = Malicious
17	End
18	End
19	End
20	End
21	End

According to the pseudo code and the hypothetical scenario, algorithm to benefit of “promiscuous mode”. All nodes on the network listen to packet communication of other neighbors. When the next node exceeds threshold level, which is previously defined, that node is identified as malicious. Malicious node won't be used for the next traffic. Identifier nodes send a broadcast message to other nodes in



neighbors to a malicious node and drops the packet. It will mark the source node as an attacker, and it will drop the packets coming from the same source. If it is a standard packet, it will send an RREP packet and route initiation process will be started.

Thanks to the FA-AODV algorithm developed, with the dropping of RREQ packets sent in a flooding attack type with a purpose to determine a route, the malicious attacks have been prevented and network congestion has been removed at the same time. Thus, the regional blockage and the communication breakdown targeted by the attacker have been prevented. The ratio of the number of packets sent by the source node to the number of packets received by the destination node is described as data flow. With the FA-AODV algorithm suggested in present study, it is aimed that the data flow rate is increased. As the flow rate increases, recovery has been achieved in the average end-to-end delay.

### 3.2. Realized Simulation

NS-2 simulation involves many applications, protocols, network types, and network element and traffic model. NS-2 is an object-based simulator which Coded in C++. Object oriented extension of Tool command language (OTcl) will be used from Ns2 for interpreter to run user scripts during simulation. OTcl scripts help users to define detailed network topologies, to simulate featured protocol and applications, and to retrieve simulation results in a specific format. It has been developed by TCL Jhon Ousterhout and has emerged as a language suitable for fast development, able to supply graphic interface, compatible with many platforms and easy to use [17, 18].

In this study, network planning has been performed using the algorithms of the standard AODV and the FA-AODV developed according to the parameter values given in Table 2 on the NS-2 network simulator.

Table 2 Simulation parameters value

Parameters	Values
Channel	Wireless
Propagation	Two Ray Ground
Mac Protocol	802.11
Routing Queue	Queue Drop tail
Antenna	Omni Antenna
Energy Model	Battery
Simulation area (m)	1000*1000
Number of nodes	5, 10, 15, 20
Simulation stop time	6s
Mobility	20 - 25 m/s
Traffic type	CBR
Attacks type	UDP floods
Number of flood node	1, 2, 3, 4

Randomly chosen 5, 10, 15 and 20 nodes, distributed in the overall of the simulation domain given in Table 2 were generated. The speeds of the nodes are assigned with a speed of 0-25 m/s. The assigning of the speeds in this range is done randomly. For it to make sense, the running time is set to 2.5 m/s. The source and the destination packets were defined and the times at which the communications would start, and stop were specified in the TCL file. The nodes (vehicles) make communications as per the conditions stated in TCL configuration table. As of the 0,25th second of the simulation, the vehicles take off for different coordinates at 15-25 m/s speeds. To capture the destination node, UDP flooding attacks are initiated at the 0-1,5 seconds of the simulation using a CBR traffic over UDP traffic. At the 0-1 seconds of the simulation, the node2 (Vehicle numbered 0) is captured by attackers. Then, at the 1 second, flooding type attacks are started over Node2 as shown in Figure 2.

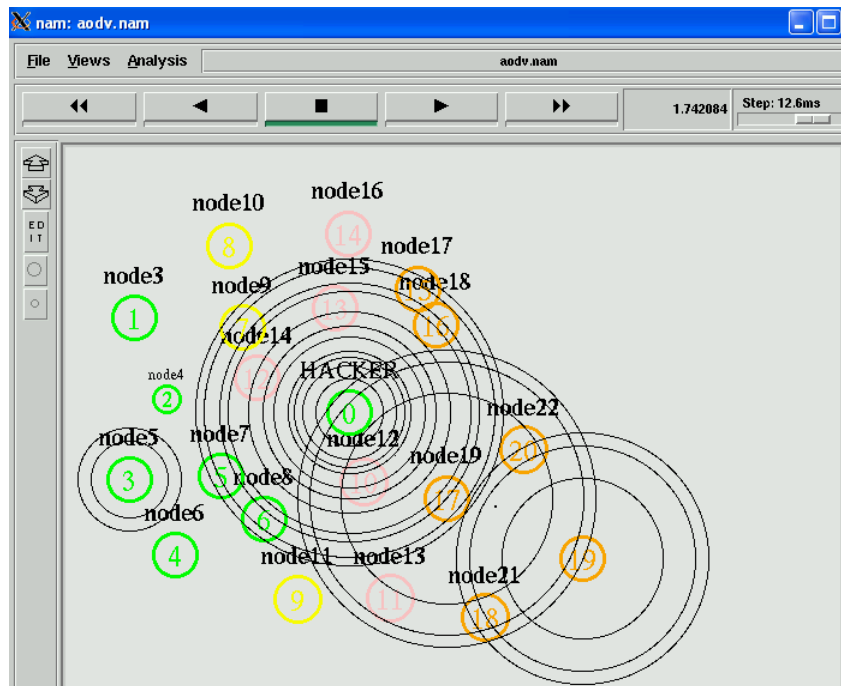


Figure 2 Malicious broadcast of Node2 to other nodes with a flood attack.

At the 1 – 2,5 seconds interval of the simulation, the dropped packets due to the algorithm that comes into play after the flood attacks performed are shown in Figure 3.

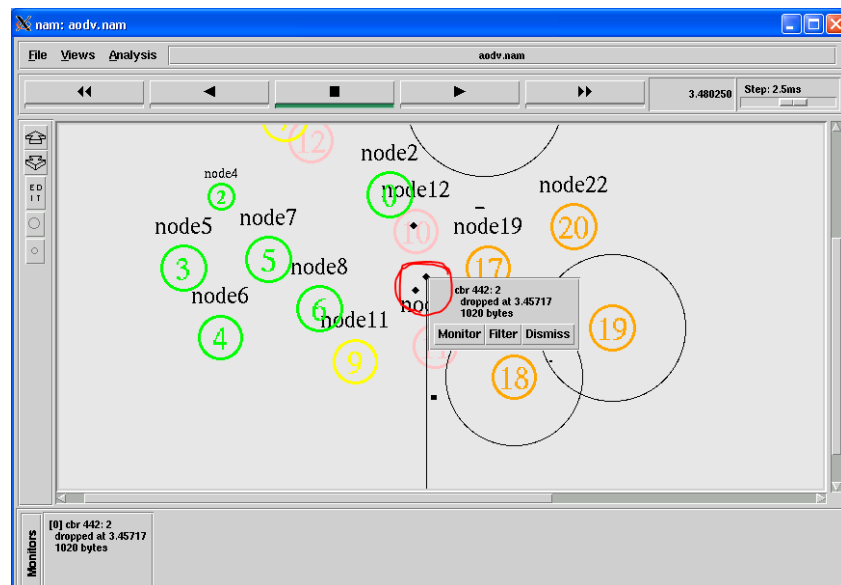


Figure 3 Dropped Packets

#### 4. FA-AODV Algorithm Performance Analysis

To evaluate the results of the suggested algorithm, simulation of two different models were run. The first of them is the implementation of UDP flooding attacks on the AODV protocol used in VANET. The second one is the simulation of the model in which the suggested FA-AODV algorithm is used. For both models, 5, 10, 15 and 20 nodes were used. There is one malicious node for five nodes (5-1, 10-2, 15-3, and 20-4). For performance criteria, the end-to-end average delay containing all the possible delays resulting from the queuing during the route discovery process, total of dropped packets and the number of packets that reached the destination in data transfer were compared.

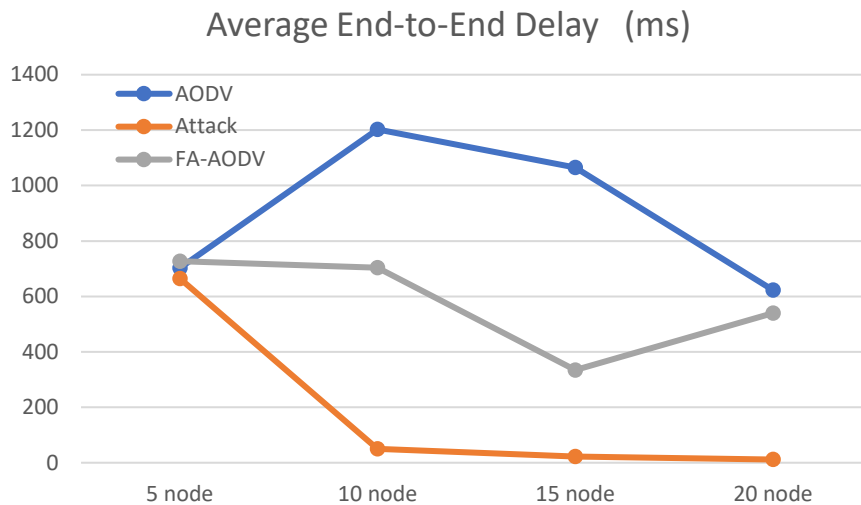


Figure 4 Average end-to-end delay

When the end-to-end average latency is examined, as seen in Figure 4, in the scenario which has 5 nodes, the delays between the scenarios are very close because of the distance between the vehicles. As the number of nodes under attack increases, average end-to-end reduces from the attacker node to the neighbor nodes because the packet transmission is too high. The proposed FA-AODV algorithm showed less latency compared to the standard AODV algorithm.

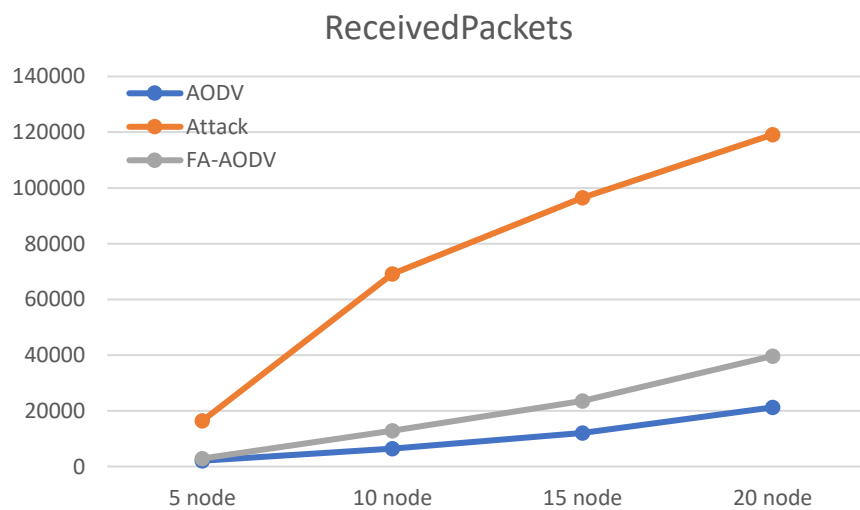


Figure 5 Received packets

The number of packets received by the attacked vehicles is given in figure 5. In the scenario where the standard AODV algorithm is used under attack, more packet reaches the target with the increase of attacker nodes. However, in the FA-AODV scenario, although the number of aggressors increased, results were very close to the non-attacked AODV scenario. In this way, fewer and safer packets were delivered to the target.

The studies in the literature were examined with the proposed method in terms of average end-to-end delay. The proposed method has provided less end-to-end delay times than [15,16]. The studies in the references [6,8] show a high similarity with the proposed method in terms of end-to-end delay times

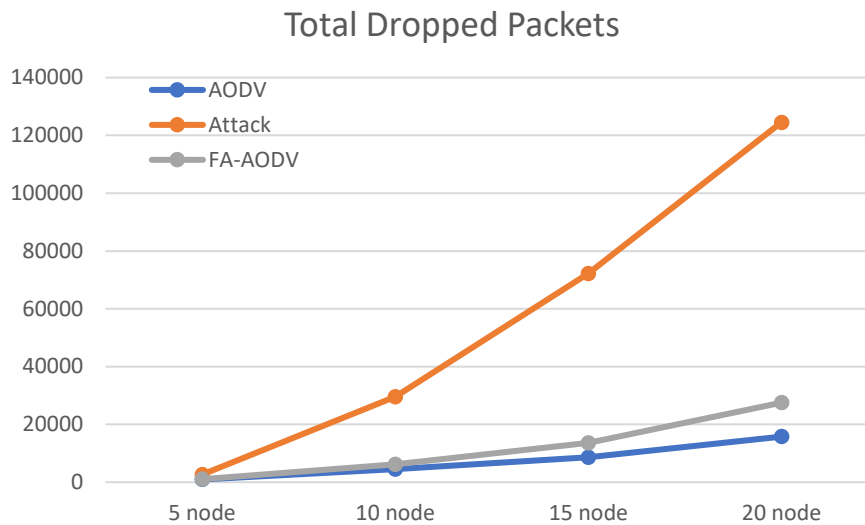


Figure 6 Dropped packets in total

In Figure 6, the comparison of the dropped packets uses the proposed FA-AODV algorithm instead of the attacked AODV protocol, and results obtained are close to the AODV protocol which works in the non-attack scenario due to isolating the attacker nodes from the network. The increase in the number of packets dropped indicates that the attacker blocked the communication and caused a network blockage. In the 5-nodes 1 attacker scenario, 2642 packets are dropped, while in the 20-node 4-attackers scenario, 124507 packets are dropped.

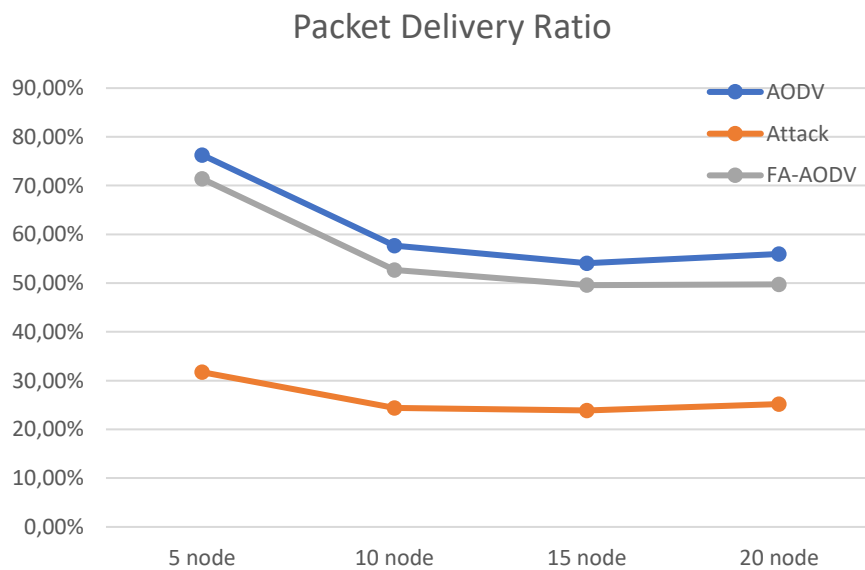


Figure 7 Packet Delivery Ratio

The Packet Delivery Ratio is another metric indicating the network quality. In the AODV protocol running under attack, packet transmission rates are reduced to 30%. As seen in Figure 7, the results of FA-AODV algorithm were close to the standard AODV algorithm results.

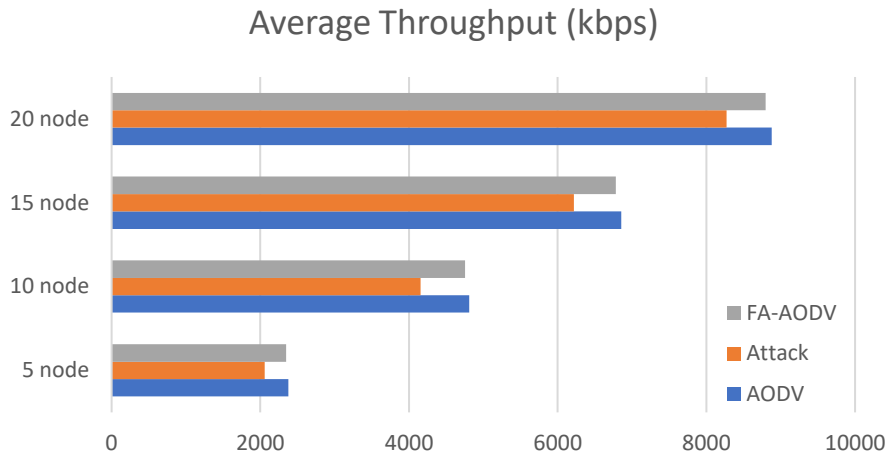


Figure 8 Average Throughput

The average throughput given in Figure 8 refers to the average data rate of successful data or message transmission over the connection.  $(\text{Received Size} / (\text{Stop Time} - \text{Start Time})) * (8/1000)$ . When the average throughputs were examined, it was observed that the FA-AODV algorithm results were close to the non-attack AODV algorithm results, due to packet transmission with under attack.

Table 3 Routing Load

Routing Load	AODV	Attack	FA-AODV
5 nodes	0,025	6,982	0,265
10 nodes	0,23	142,545	1,427
15 nodes	0,646	363,824	2,989
20 nodes	2,068	978,458	7,771

As can be seen in Table 3, the load of the network under attack increases exponentially as the number of nodes increases. This also seen in Figure 9 The FA-AODV algorithm has improved the network load when network is under attack.

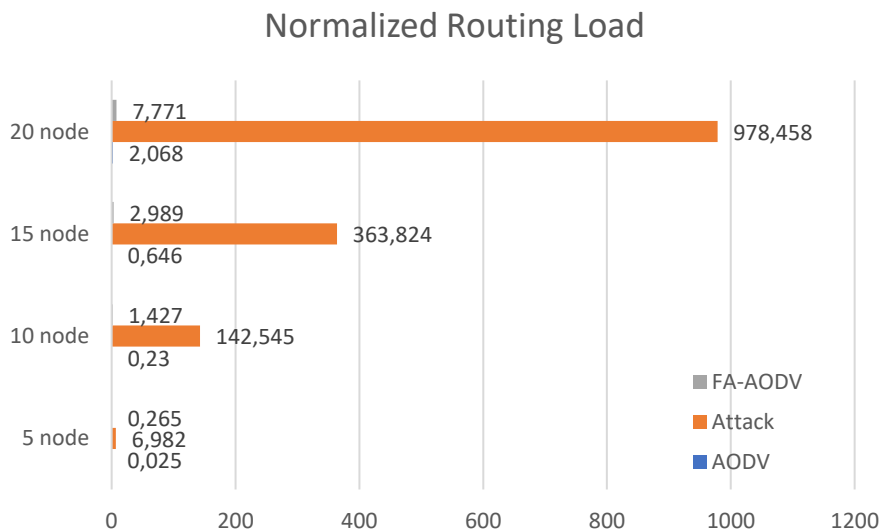


Figure 9 Routing Load

## 5. Conclusion

In this study, the FA-AODV algorithm that prevents UDP flooding type attacks made on the AODV protocol is used in VANET. The results are compared using two models, which are standard AODV and FA-AODV. For the performance analysis, average end-to-end delay, received packets and total dropped packets have been used. The FA-AODV algorithm checks the hop count and packet IDs in the RREQ packets and decides whether it is an attack or not. It has ensured the continuity of the communication by dropping the attacking packets. Therefore, the FA-AODV algorithm has not produced traffic load by preventing the network from getting congested. Despite the attacks, the performance of the moving nodes has been increased and continuation of a safe communication has been ensured. Therefore, FA-AODV algorithm prevents the UDP flooding type attacks and provides faster and safer communication. Safer communication has been established and the packets have been sent to the destination in the shortest time possible. Considerable improvements have been achieved in the average end-to-end delay and number of packets delivered. As it is considered that the attackers could conduct different types of attacks on the network, it is suggested that the algorithm be improved against the types of attacks apart from flooding. For the future work, we will plan to compare the proposed FA-AODV algorithm with other reactive routing protocols (DSR, TORA etc) used in VANET.

## References

- [1] S.-H. Kim and I.-Y. Lee, "A Secure and Efficient Vehicle-to-Vehicle Communication Scheme using Bloom Filter in VANETs," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 9–24, Mar. 2014, doi: 10.14257/ijasia.2014.8.2.02.
- [2] M. Y. Gadkari, "VANET: Routing Protocols, Security Issues and Simulation Tools," *IOSR Journal of Computer Engineering*, vol. 3, no. 3, pp. 28–38, 2012, doi: 10.9790/0661-0332838.
- [3] N. Arulkumar and E. G. D. P. Raj, "A simulation based study to implement Intelligent Transport Systems concepts in VANETs using AODV routing protocol in NS2," *2012 Fourth International Conference on Advanced Computing (ICoAC)*, vol. 1, no. 1, Dec. 2012, doi: 10.1109/icoac.2012.6416854.
- [4] J. Zhang, "Trust Management for VANETs," *International Journal of Distributed Systems and Technologies*, vol. 3, no. 1, pp. 48–62, Jan. 2012, doi: 10.4018/jdst.2012010104.
- [5] A. Kumar and M. Sinha, "Design and analysis of an improved AODV protocol for black hole and flooding attack in vehicular ad-hoc network (VANET)," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 4, pp. 453–463, May 2019, doi: 10.1080/09720529.2019.1637151.
- [6] M. J. Faghiniya, S. M. Hosseini, and M. Tahmasebi, "Security upgrade against RREQ flooding attack by using balance index on vehicular ad hoc network," *Wireless Networks*, vol. 23, no. 6, pp. 1863–1874, Apr. 2016, doi: 10.1007/s11276-016-1259-2.
- [7] VarshaGharu, M. Pawar, and J. Agarwal, "A literature survey on security issues of WSN and different types of attacks in network," *Indian Journal of Computer Science and Engineering*, vol. 8, no. 2, Apr. 2017.
- [8] M. Rmayti, Y. Begriche, R. Khatoun, L. Khoukhi, and D. Gaiti, "Flooding attacks detection in MANETs," *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, vol. 1, no. 1, Aug. 2015, doi: 10.1109/ssic.2015.7245675.
- [9] G. S. Ganpat Joshi, "A Novel Statistical Adhoc On-Demand Distance Vector Routing Protocol Technique Is Using for Preventing the Mobile Adhoc Network from Flooding Attack," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 1753–1765, Apr. 2021.
- [10] M. Al-Mehdhar and N. Ruan, "MSOM: Efficient Mechanism for Defense against DDoS Attacks in VANET," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, pp. 1–17, Apr. 2021, doi: 10.1155/2021/8891758.
- [11] J. Rabari and A. R. P. Kumar, "FIFA: Fighting against Interest Flooding Attack in NDN-based VANET," Jun. 2021. doi: 10.1109/iwcmc51323.2021.9498767.

- [12] M. Türkoğlu, H. Polat, C. Koçak, and O. Polat, "Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection," *Expert Systems with Applications*, vol. 203, no. 1, p. 117500, Oct. 2022, doi: 10.1016/j.eswa.2022.117500.
- [13] S. M. Zarei and R. Fotohi, "Defense against flooding attacks using probabilistic thresholds in the internet of things ecosystem," *Security and Privacy*, vol. 4, no. 3, Feb. 2021.
- [14] N. Kadam and K. R. Sekhar, "Machine Learning Approach of Hybrid KSVN Algorithm to Detect DDoS Attack in VANET," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021, doi: 10.14569/ijacsa.2021.0120782.
- [15] M. R. Hasan, Y. Zhao, Y. Luo, G. Wang, and R. M. Winter, "An Effective AODV-based Flooding Detection and Prevention for Smart Meter Network," *Procedia Computer Science*, vol. 129, no. 1, pp. 454–460, 2018, doi: 10.1016/j.procs.2018.03.024.
- [16] K. Saravanan and J. Vellingiri, "Defending MANET against flooding attack for medical application," *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, vol. 1, no. 1, Oct. 2017, doi: 10.1109/cesys.2017.8321328.
- [17] E. Altman and T. Jiménez, "NS Simulator for Beginners," *Synthesis Lectures on Communication Networks*, vol. 5, no. 1, pp. 1–184, Jan. 2012, doi: 10.2200/s00397ed1v01y201112cnt010.
- [18] "The Network Simulator - ns-2," *Isi.edu*, 2020. <https://www.isi.edu/nsnam/ns/> (accessed Sep. 14, 2022).

# The Effect of Numerical Mapping Techniques on Performance in Genomic Research

 Seda Nur Gülocak<sup>1</sup>,  Bihter Daş<sup>2</sup>

<sup>1</sup>Department of Software Engineering, Technology Faculty, Firat University; sedanurgulocak@gmail.com;

<sup>2</sup>Corresponding Author; Department of Software Engineering, Technology Faculty, Firat University;  
bihterdas@firat.edu.tr;

Received 19 October 2022; Accepted 1 November 2022; Published online 31 December 2022

## Abstract

In genomic signal processing applications, digitization of these signals is needed to process and analyze DNA signals. In the digitization process, the mapping technique to be chosen greatly affects the performance of the system for the genomic domain to be studied. The purpose of this review is to analyze how numerical mapping techniques used in digitizing DNA sequences affect performance in genomic studies. For this purpose, all digital coding techniques presented in the literature in the studies conducted in the last 10 years have been examined, and the numerical representations of these techniques are given in a sample DNA sequence. In addition, the frequency of use of these coding techniques in four popular genomic areas such as exon region identification, exon-intron classification, phylogenetic analysis, gene detection, and the min-max range of the performances obtained by using these techniques in that area are also given. This study is thought to be a guide for researchers who want to work in the field of bioinformatics.

**Keywords:** Numerical mapping techniques, Genomic analysis, DNA encoding schemes, Genomic signal processing, DNA sequence

## 1. Introduction

Deoxyribose nucleic acid (DNA) is the biological structure that is located inside the cells, which are the building blocks of the human body, and creates the genetic code in which all human characteristics are encoded. DNAs consist of sugar groups, phosphate groups and bases linked by ester bonds. These bases are Adenine, Thymine, Guanine, and Cytosine. In the DNA chain consisting of two long polymers, Adenine pairs with Thymine while Guanine pairs with Cytosine. The codes created by certain combinations of the building blocks called nucleic acids that makeup DNA are called genes. Genes; These are personal codes that determine all the characteristics of the body, such as eye color, height, hairstyle, or susceptibility to genetic diseases. A gene has exons and introns. The intron is the non-amino acid coding portion of a gene. Exons are the protein-coding parts of the gene. The triple arrangement of bases in DNA is referred to as codons and these codons code for the different amino acids that make up proteins. There are 64 possible codons in a DNA. Three of the codons (UAA, UAG, UGA) are termination or stop codons and do not code for any amino acid. Each of the remaining 61 codons codes for an amino acid, but since there are only 20 amino acids used in protein construction, there is more than one codon encoding the same amino acid [1]. Genetic information from DNA is transferred to RNA. This process is called transcription. The translation is the process of translating the code carried by the mRNA into proteins. Figure 1 shows the basic structure of a protein-coding gene related to transcription and translation in a eukaryotic organism.



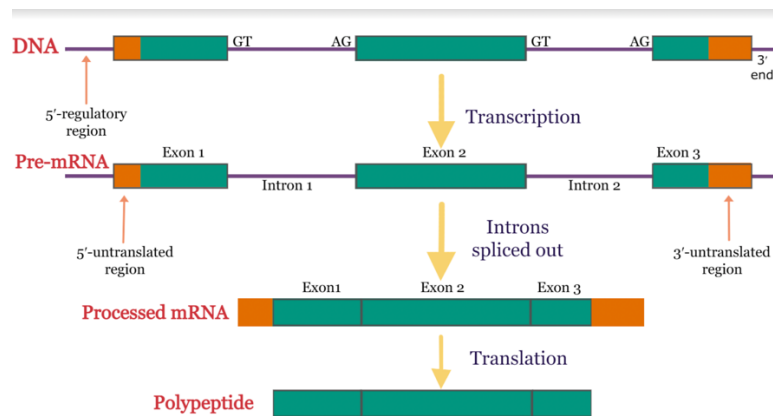


Figure 1 Gene and intergenic regions in a DNA

### 1.1. Literature Review

In this section, studies that examine the techniques used for digitization of DNA data in various genomic fields such as detection of exon regions from DNA sequences, exon-intron classification, phylogenetic analysis, interspecies similarity/difference, detection of disease, and gene detection are presented. Table 1 lists all studies over the past 15 years examining numerical mapping techniques used by genomic domains.

Table 1 All studies reviewed in the last 15 years

Reference Paper	Numerical mapping techniques	Genomic domain
Das et al.[3]	1. The Nucleotide Mapping Diplole moment mapping, Minimum entropy mapping, Trigonometric mapping, Variable mapping, Chaos game representation, Gray code mapping, Walsh code mapping, Pseudo-EIIP mapping 2. The Amino acid Mapping	Prediction of Exon regions
Wisesty et al. [4]	Dipole moment and alpha mapping, Binary representation, Genetic code context, EIIP, Complex prime numeric representation, Hyroathy index, P-adic mapping, Ionization-constant, The tetrahedron mapping Voss mapping, Z-curve, tetrahedron, Complex number representation, Integer and real representation, Trigonometric mapping, Paired numeric representation	Diagnosis of breast cancer
Kumar et al. [5]	Position Based Encoding, 2-bit Neural Network based encoding, Hamming Distance Based Encoding, Integer Number Encoding, Trigonometric Encoding, Autocorrelation Based Encoding and proposed Method walsh code	Detection of protein coding regions
Yu et al. [6]	1. Biochemical Properties Atomic number, Electron-Ion Interaction pseudopotential, Molecular mass representation, Thermodynamic properties 2. Primary-Structure Properties Dinucleotide representation, Ring structure, Inter nucleotide distance encoding, Triplet encoding, Frequency of occurrence mapping, Minimum entropy mapping 3. Cartesian-Coordinate Properties Integer and real number, Complex number, QPSK/PAM, DNA walk and paired numeric Method 4. Binary and Information Encoding Voss representation, Galois field, Error-Correction code, Ching representation	Genomic signal processing

5. Graphical Representation		
	CGR and CGR-walk, Tetrahedron, SOM based approach, Quaternion, H-curve and Z-curve	
Kumari et al. [7]	1.Fixed Mapping Voss, Tetrahedron, Complex, Integer, Real, Quaternion, QPSK 2.Cariable Mapping Complex representation of nucleotides by twiddle factor 3.Physico Chemical Property Based Mapping Atomic, Paired, DNA walk, Z-curve, EIIP, Pseudo-EIIP	Genomic signal processing
Das et al. [8]	Voss, Integer, Complex, Real, EIIP, Atomic Number, Paired Numeric, DNA Walk, Molecular Mass, Trigonometric, Entropy, Z- curve, Tetrahedron	Predicting of protein coding regions
Ahmad et al. [9]	Tetrahedron, 4-bit binary coding, Binary coding, Molecular mass, Z-curve, Pathogenity island coding, Entropic segmenttion coding, Paired nucleotide representation, Integer number, Autoregressive coding, Gradient source Localization, EIIP, Paired nucleotide atomic number, Complex number	Genomic signal processing
Jin et al. [10]	Methods based on graphical representation; 2-3-4-5-6 dimensional graphical methods, Chaos game representation, quaternion. Matrix mapping, Coding based on chemical properties, Codons, frequency values, Position statistics, Huffman coding, Euclidean distance coding	Detection of similarity between species
Mendizabal-Ruiz et al. [11]	Integer, Real, EIIP, Atomic number, Paired numeric, Voss, Tetrahedron, Z-curve, DNA walk	Identifcation of similarity of DNA sequences
Saini et al. [12]	Voss, Tetrahedron, Complex, EIIP, DNA walk, Integer number, Real number, Binary representation, 4-bit binary encoding, Paired nucleotide representation, Quaternion, Inter-nucleotide distance	Genomic signal processing
Mabrouk et al. [13]	Genetic code context, Frequency of nucleotide occurence, Atomic number, 2-bit binary, EIIP	Identification of protein coding regions
Das et al. [14]	Voss mapping, Integer mapping, Complex Mapping, Real Mapping, EIIP Mapping, Atomic Number Mapping, Paired Numeric Mapping, DNA Walk Mapping, The Modecular Mass Mapping	Identification of exon regions
Das et al. [15]	Integer Mapping, Reel Mapping, Atomic Mapping, Molecular Mass Mass Mapping, DNA Walk Mapping, Paired Numeric Mapping, Complex Digital Mapping, EIIP	Classification of exon and intron
Abo-Zahhad et al. [16]	Atomic number, Integer number, Real number, EIIP, Paired Numeric, DNA Walk	Prediction of donor ve acceptor in exon region
Abo-Zahhad et al. [17]	1. Fixed Mapping 2. Voss, Tetrahedron, Complex, Integer, Real, Quaternion 3. Physico Chemical Property Based Mapping EIIP, Paired numeric, DNA-walk, Z-curve	Classification of exon and intron
Kwan et al. [18]	Integer number, Single Galois Indicator, Paired nucleotide atomic umber, Atomic number, Molecular Mass, EIIP, Paired Numeric, Real Number, Complex Number, K-twin pair code, K-bipolar pair code, K-quaternion	Classification of exon and intron
Sharma et al. [19]	Voss, Tetrahedron, Z-curve, Complex, EIIP, Paired numeric, DNA walk, Frequency Nucleotide Occurence, Atomic number, Real number	Identifcation of exon region
Akalin et al. [20]	Real mapping, Moleculer Mass, EIIP, Shannon Entropy, Paired digital mapping technique	Classification of exon and intron
Akalin et al. [21]	Real mapping, Moleculer Mass, EIIP, Shannon Entropy, Paired digital mapping technique	Prediction of leukaemia
Akhtar et al. [22]	Voss, Tetrahedron, Z-curve, Complex, Queternion, EIIP, QPSK-PAM, Paired numeric	Prediction of exon regions

In this study, all numerical mapping techniques developed in the last 15 years in the literature and used to digitize DNA sequences were examined and the benefits and shortcomings of these numerical techniques in genomic study areas such as exon region detection, exon-intron classification, phylogenetic analysis. Also, disease-causing gene detection were emphasized. Digitization of DNA sequences is extremely important in order to achieve targeted high-performance accuracy in genomic studies such as detection of exon regions, exon-intron classification, disease-causing gene detection,

phylogenetic analysis. Therefore, in this study, all the digital mapping techniques of the last 15 years were introduced in detail and a review study presenting all the techniques was actualized.

## **1.2 Motivation**

Technological developments in biology and computers have advanced rapidly, and thus the emerging branch of bioinformatics has taken the lead among the most popular academic and industrial sectors today. Genome analysis is one of the most studied subjects in the field of bioinformatics, which is the synthesis of mathematics, statistics, computer science, molecular biology, and genetics. Although genomic studies seem to be aimed at basic scientific research, they will be indispensable for clinical informatics in the coming years. Our motivation for this review study is to analyze the effect of digital mapping techniques on the performance of the system in the most popular bioinformatics and genomic fields of study. In addition, while converting DNA analog signals into digital signals that can be understood by the computer in artificial intelligence applications, it is to guide researchers in choosing the correct digital coding technique that can best reflect the structure of DNA.

The remainder of this paper is organized as follows. In section 2, all numerical mapping techniques introduced in the literature by other authors in the last 5 years are searched and listed for this survey article. Section 3 highlights the frequency of use, performance, advantages, and drawbacks of numerical mapping techniques by genomic domains. In addition, mapping techniques that researchers can use according to genomic domains will be recommended along with their reasons. Finally, in Section 4 we conclude our survey with a brief summary.

## **2. DNA Numerical Mapping Techniques**

In this section, the coding techniques developed for the digitization of DNA sequences are comprehensively examined under five main headings. In the literature, coding techniques are also called different names as digital mapping techniques, numerical methods, and coding schemes. However, all the nomenclatures mean the same. 50 digital mapping techniques developed in the last 5 years are classified into five groups according to their general characteristics. These groups are cartesian coordinate coding techniques, biochemical and physicochemical coding techniques, binary and information coding, primary structure coding techniques, and graphically represented coding techniques. At the end of each of these five groups, there are collective digital signal plot graphs and tables of the digital coding techniques examined in that group. Graphs of digitized DNA signals using coding techniques include representations of the DNA sequence digitized by each coding technique applied to the DNA Fasta format dataset with reference number NR\_131216.1 from the NCBI database. Since the graphical value ranges of some mapping techniques are different, the numerical representation of these techniques is given in separate figures. In the general tables at the end of the examined groups, there is a brief explanation of the digitization technique in that group, the coding scheme, and the numerical version of this coding technique applied to a sample DNA sequence. Figure 2 shows the hierarchical scheme of all DNA mapping techniques.

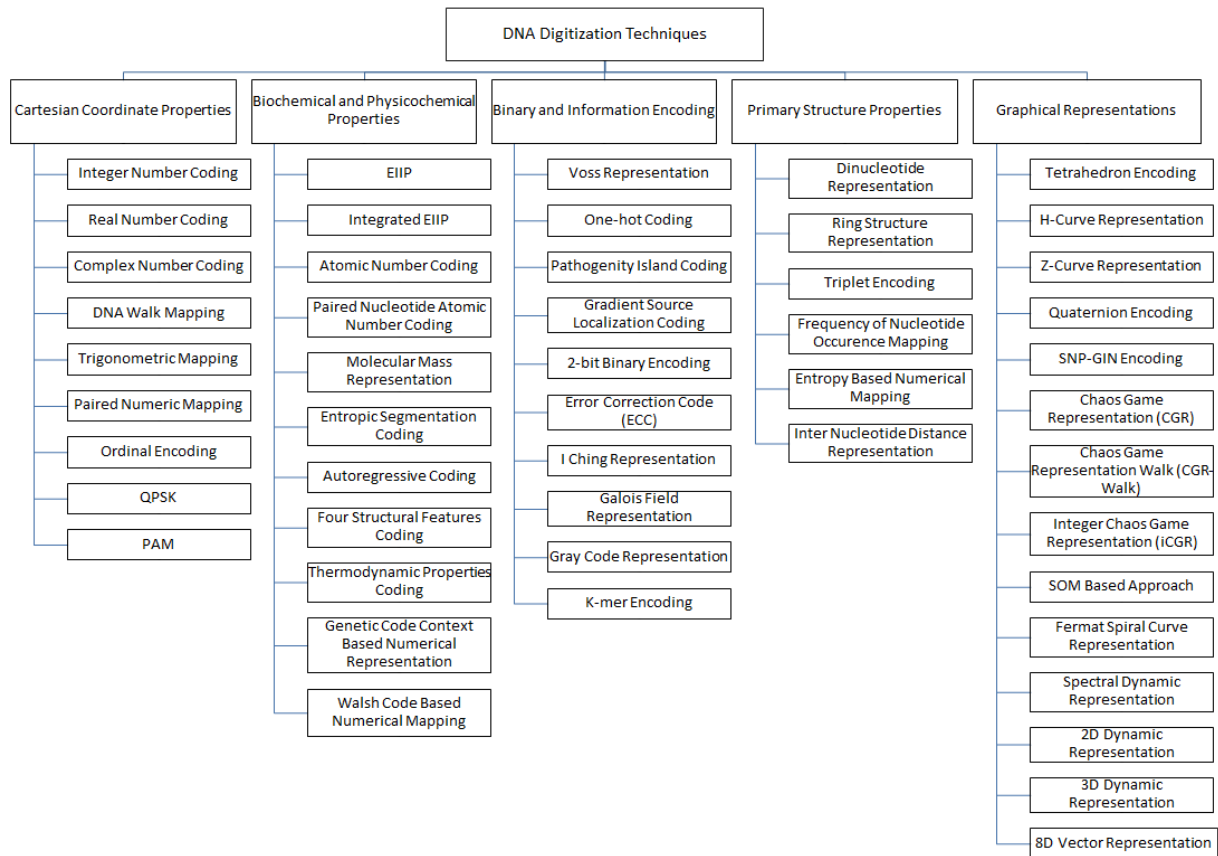


Figure 2 The hierarchical scheme of all DNA mapping techniques

### 2.1 Cartesian-Coordinate Properties Group

The first group of DNA numerical mapping techniques is cartesian coordinate properties (CCP) digitization techniques. Within this group, there are nine numerical coding techniques namely Integer Number Coding, Real Number Coding, Complex Number Coding, DNA Walk Mapping, Trigonometric Mapping, Paired Numeric Coding, Ordinal Encoding, QPSK (Quadrature Phase Shift Keying), PAM (Pulse Amplitude Modulation) are examined. Table 2 provides a brief summary of all the mapping techniques in the CCP group.

Table 2 The summary of all numerical coding techniques in cartesian-coordinate properties group

The name of technique	Coding Scheme	Numerical Representation	Definition
Integer Number Coding [6,15]	If Purin>Pirimidin T=0, C=1, A=2, G=3 T>A ve G>C ise A=0, C=1, T=2, G=3	$X=[AGCTACCGTG]$ $\hat{X}=[2, 3, 1, 0, 2, 1, 1, 3, 0, 3]$	Nucleotides are represented by integers.
Real Number Coding [6,15]	A=-1.5, T=1.5, C=0.5, G=-0.5	$X=[AGCTACCGTG]$ $\hat{X}=[-1.5, -0.5, 0.5, 1.5, -1.5, 0.5, 0.5, -0.5, 1.5, -0.5]$	Nucleotides are represented by real numbers.
Complex Number Coding [9]	A= -1, C= -j, G= j, T= 1	$X=[AGCTACCGTG]$ $\hat{X}(i) = [l, -l, j, j, l, -j, -j, -l, j, -l]$	Nucleotides are represented by complex numbers.
DNA Walk Coding [6]	Integer temsil; A= -1, C= 1 G=- 1, T= 1 Complex temsil;	$X=[AGCTACCGTG]$ $\hat{X}(i) = [-1, -2, -1, 0, -1, 0, 1, 0, 1, 0]$	Nucleotides are encoded by assigning integer or complex numbers and summing their

	A=1, C= -j G= -1, T= j		values along the DNA sequence.
Trigonometric Mapping Coding [3,22]	A= $\cos(\theta) + j \times \sin(\theta)$ C= $-\cos(\theta) - j \times \sin(\theta)$ G= $-\cos(\theta) + j \times \sin(\theta)$ T= $\cos(\theta) - j \times \sin(\theta)$	X=[AGCTACCGTG] $\hat{X}(i) = [0.5+0.8660i, -0.5+0.8660i, -0.5+0.8660i, 0.5+0.8660i, 0.5+0.8660i, -0.5+0.8660i, -0.5+0.8660i, 0.5+0.8660i, -0.5+0.8660i]$	Nucleotides are encoded by assigning trigonometric equations.
Paired Numeric Coding [15]	Purin(A&G)= 1 Pirimidin(C&T)= -1	X=[AGCTACCGTG] $\hat{X} = [1, 1, -1, -1, 1, -1, -1, 1, -1, 1]$	Nucleotides are encoded by assigning values according to their structural properties.
Ordinal Encoding [23]	A= 0.25, C= 0.50 G= 0.75, T= 1.00	X=[AGCTACCGTG] $\hat{X}(i) = [0.25, 0.75, 0.50, 1.00, 0.25, 0.50, 0.50, 0.75, 1.00, 0.50]$	Nucleotides are assigned sequential, linear values.
QPSK [24]	A= 1+j, G= -1+j C= -1-j, T= 1-j	X=[AGCTACCGTG] $\hat{X}(i) = [1+j, -1+j, -1-j, 1-j, 1+j, -1-j, -1-j, -1+j, 1-j, -1+j]$	2D QPSK constellation complex number values are assigned according to the complementary property of DNA.
PAM [6,25]	A= -1.5, G= -0.5 C= 0.5, T= 0.5	X=[AGCTACCGTG] $\hat{X}(i) = [-1.5, -0.5, 0.5, 0.5, -1.5, 0.5, 0.5, -0.5, 0.5, -0.5]$	Nucleotides are represented by 1D real numbers.

DNA sample datasets in Genbanks are available in Fasta format and are analog signals. Digitized signal representations of the first 100 bases of the sequence with reference number NR\_131216.1 retrieved from the NCBI database, with coding techniques in the "Cartesian-Coordinate Properties (CCP)" group are shown in Figure 3.

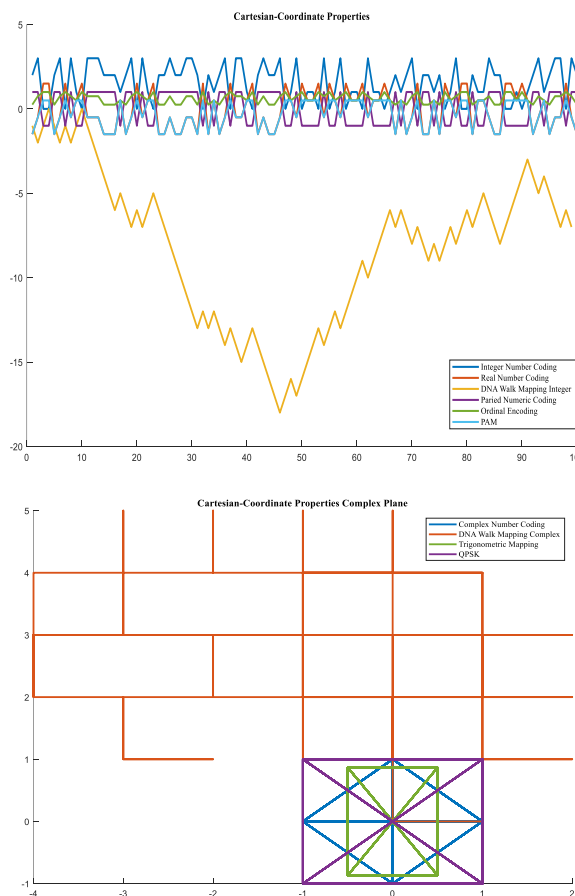


Figure 3 Numerical representations of CCP group techniques

## 2.2 Biochemical and Physicochemical Properties Group

The second group of DNA numerical mapping techniques is the biochemical and physicochemical (BPP) numerical techniques. Within this group, eleven coding schemes such as EIIP, Integrated EIIP, Atomic Number Coding, Paired Nucleotide Atomic Number Coding, Molecular Mass Representation, Entropic Segmentation Coding, Autoregressive Coding, Four Structural Features Coding, Thermodynamic Properties Coding, Genetic Code Context-Based Numerical coding, Walsh Code Based Numerical Mapping are examined. Table 3 provides a brief summary of all the mapping techniques in the BPP group.

Table 3 The summary of all numerical coding techniques in biochemical and physicochemical properties group

The name of technique	Coding Scheme	Numerical Representation	Definition
EIIP coding [3,6,9]	C=0.1340, T=0.1335, A=0.1260, G=0.0806	$X=[AGCTACCGTG]$ $\hat{X} = [0.1260, 0.0806, 0.1340, 0.1335, 0.1260, 0.1340, 0.1340, 0.0806, 0.1335, 0.0806]$	Energy values are assigned to the nucleotides
Integrated EIIP coding [26]	EIIP codes for 64 codons	$X=[AGCTACCGTG]$ $\hat{X} = [0.3406, 0.3481, 0.3935, 0.3935, 0.3940, 0.3486, 0.3935, 0.2947]$	EIIP energy values are assigned to DNA codons.
Atomic number coding [6]	C= 58, T= 66, A= 70, G= 78	$X=[AGCTACCGTG]$ $\hat{X} = [70, 78, 58, 66, 70, 58, 58, 78, 66, 78]$	Atomic numbers are assigned to nucleotides.
Paired nucleotide atomic coding [9]	A&G= 62, C&T= 42	$X=[AGCTACCGTG]$ $\hat{X}(i) = [62, 62, 42, 42, 62, 42, 42, 62, 42, 62]$	Atomic numbers are assigned to paired nucleotides.
Molecular mass coding [9,15]	C= 110, G= 150, A= 134, T= 125	$X=[AGCTACCGTG]$ $\hat{X}(i) = [134, 150, 110, 125, 134, 110, 110, 150, 125, 150]$	Molecular mass values are assigned to nucleotides.
Entropic segmentation coding [9,27,28]	12-Symbol alphabet A <sub>1</sub> , A <sub>2</sub> , A <sub>3</sub> , C <sub>1</sub> , C <sub>2</sub> , C <sub>3</sub> , G <sub>1</sub> , G <sub>2</sub> , G <sub>3</sub> , T <sub>1</sub> , T <sub>2</sub> , T <sub>3</sub> Calculates entropy by array	$X=[AGTTAGTGCT]$ $\hat{X}(i) = [A_1 G_2 S_3 T_3 S_1 T_1 A_2 G_3 T_1 G_2 C_3]$	DNA segments and stop codons are represented by the 18-symbol alphabet.
Autoregressive coding [9,29]	Propeller Twist ve DNA Bending Stiffness values for dinükleotides	$X=[AGCTACCGTG]$ Propeller Twist $\hat{X}(i) = [-14.00, -11.08, -14.00, -11.85, -13.10, -8.10, -10.03, -13.10, -9.45]$ Bending Stiffness $\hat{X}(i) = [60, 85, 60, 20, 60, 130, 85, 60, 60]$	According to the structural properties of DNA, propeller twist and DNA bending stiffness values and dinucleotides are coded.
Four Structural features coding [30]	DNA Bending Stiffness, Dublex Disrupt Energy, Dublex Free Energy, Propeller Twist values for dinükleotides	$X=[AGCTACCGTG]$ DNA bending stiffness $\tilde{x}_\alpha(n) = 60, 60, 60, 85, 60$ Dublex disrupt energy $\tilde{x}_\beta(n) = 16, 16, 13, 36, 19$ Dublex free energy $\tilde{x}_\gamma(n) = -15, -15, -15, -28, -17$ Propeller twist $\tilde{x}_\delta(n) = -1400, -1400, -1310, -1003, -94$	According to the four physical properties of DNA, the coding for the dinucleotide is performed according to the propeller twist value, DNA bending stiffness, duplex disrupts energy, and duplex free energy values.
Thermodynamic properties coding [6,31]	TC=5.6, GA=5.6, CA=5.8, TG=5.8, TA=6.0, AC=6.5, GT=6.5, CT=7.8, AG=7.8, AT=8.6, TT=9.1, AA=9.1,	$X=[AGCTACCGTG]$ $\hat{X}(i) = [7.8, 11.1, 7.8, 6.0, 6.5, 11.0, 11.9, 6.5, 5.8]$	Coding is performed by assigning enthalpy values of thermodynamic interactions of nucleotides.

	CC=11.0,GG=11.0, GC=11.1, CG=11.9		
Genetic code context (GCC) based numerical coding [13]	Assignment of GCC-based complex number representations to amino acids (Table 8)	$X=[AGCTACCGTG]$ Birinci çerçeve AGC TAC CGT İkinci çerçeve GCT ACC GTG $\hat{X}(i) = [0.05 + 88.7i, 0.6 + 88.3i, 1.88 + 193i, 0.06 + 125.1i, 0.60 + 181.2i, 1.32 + 141.4i]$	The DNA sequence is read with a reading frame as triplet codons. The sequence is digitized by assigning complex number values to amino acids.
Walsh Code Based coding [5]	A=W <sub>A</sub> =0000 T=W <sub>T</sub> =0011 G=W <sub>G</sub> =0101 C=W <sub>C</sub> =0110	$X=[AGCTACCGTG]$ $\hat{X}(i) = [00000101 \quad 011000110000$ $\quad 011001100101 \quad 00110101]$	The fourth-order Walsh codes are assigned to nucleotides

Digitized signal representations of the sample DNA sequence (NR\_131216.1) with EIIP and Integrated EIIP techniques are shown in Figure 4.

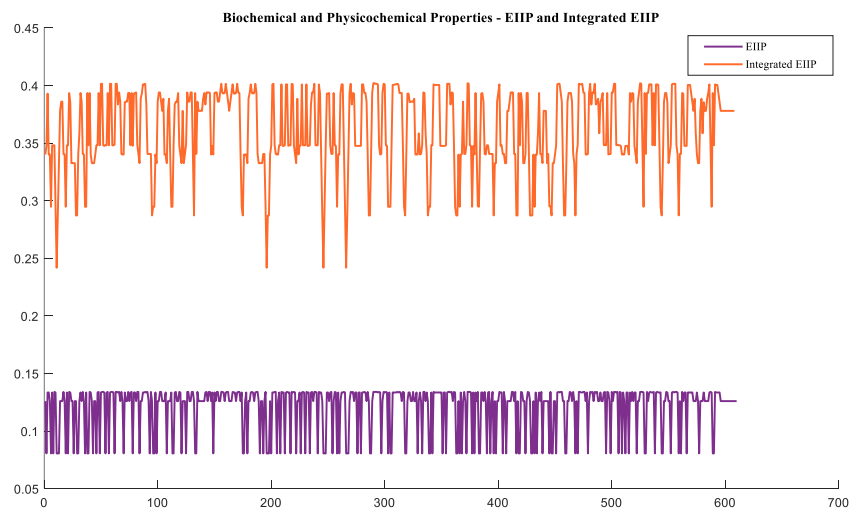


Figure 4 Numerical representations of EIIP and Integrated EIIP techniques

Digitized signal representations with the four structural features coding technique are shown in Figure 5.

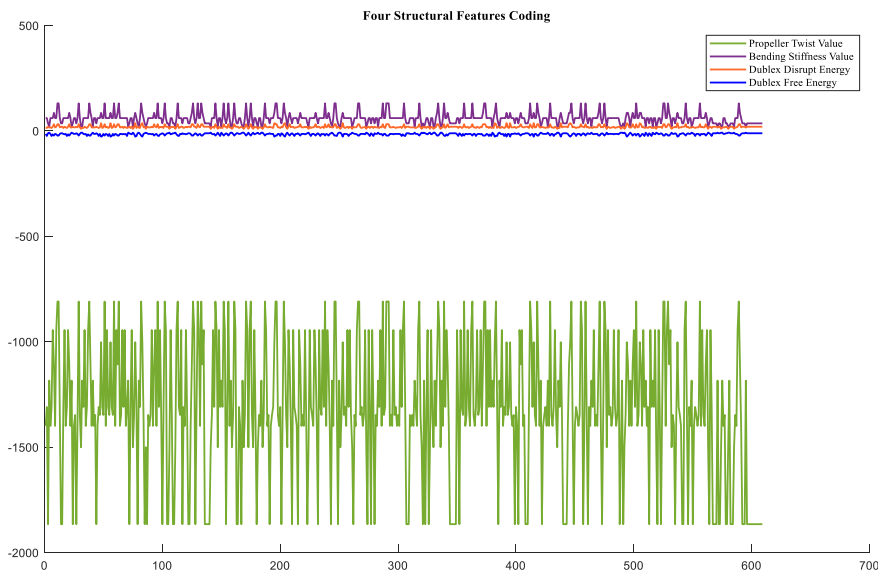


Figure 5 Numerical representations of the four structural features coding

Figure 6 gives the digitized signal plot of the reference sequence NR\_131216.1 using the GCC-based coding technique.

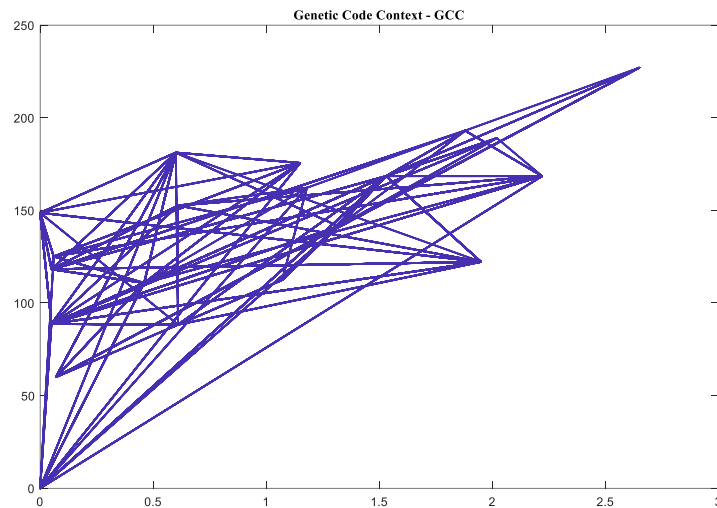


Figure 6 Numerical representations of the GCC coding Digitized signal representations of the first 100 bases of the NR\_131216.1 reference numbered sequences by coding techniques in the “Biochemical and Physicochemical Properties (BPP)” group are shown in Figure 7.

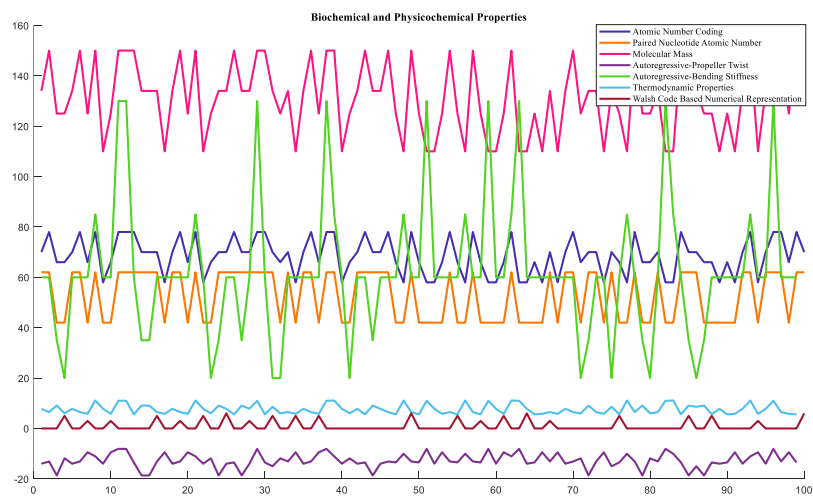


Figure 7 Numerical representations of BPP group techniques

### 2.3 Binary and Information Encoding Group

The third group of DNA numerical mapping techniques is cartesian coordinate (CCP) digitization techniques. Within this group, ten coding techniques as Voss Representation, One-Hot Coding, Pathogenicity Island Coding, Gradient Source Localization Coding, 2-bit Binary Encoding, Error Correction Code (ECC), I Ching Representation, Galois Field Representation, Gray Code Representation, K-mer Encoding are examined. Table 4 provides a brief summary of all the mapping techniques in the “Binary and Information Encoding” group.



Table 4 The summary of all numerical coding techniques in binary and information encoding group

The name of technique	Coding Scheme	Numerical Representation	Definition
Voss Coding [6]	S=[C, G, A, T], Cn=[1,0,0,0], Gn=[0, 1, 0, 0], An=[0, 0, 1, 0], Tn=[0, 0, 0, 1]	X=[AGCTACCGTG] A <sub>20</sub> = [1000100000] G <sub>20</sub> = [0100000101] C <sub>20</sub> = [0010011000] T <sub>20</sub> = [0001000010]	By creating four sequences, binary values are assigned according to the presence or absence of each base.
One-hot Coding [32]	A:1,0,0,0 T:0,1,0,0 C:0,0,1,0 G:0,0,0,1	X=[AGCTACCGTG] $\hat{X}(i) = [1000, 0001, 0010, 0100, 1000, 0010, 0010, 0001, 0100, 0001]$	Nucleotides are encoded with four-bit binary values.
Pathonetity Island Coding [9,33]	C&G= 1, A&T= 0	X=[AGCTACCGTG] $\hat{X}(i) = [0110011101]$	Binary values are assigned according to the presence and absence of pathogenicity islands.
Gradient Source Localization Coding [9]	A= 0, C=1, G=3, T= 2	X=[AGCTACCGTG] $\hat{X}(i) = [0, 3, 1, 2, 0, 1, 1, 3, 2, 3]$	Integer values are assigned to nucleotides based on gradient source localization
2-bit Binary Encoding [13]	A= 00, G= 10, T= 01, C= 11	X=[AGCTACCGTG] $\hat{X}(i) = [00, 10, 11, 01, 00, 11, 11, 10, 01, 10]$	Two-bit binary values are assigned to the nucleotides
Error Correction Coding [6,34]	x, y coordinates values according to group codons like Purine-Pyrimidine, Weak-Strong H bond, Amino-Keto	X=[AGCTACCGTG] Purine-Prymidine $\hat{X}(i) = [(1,1), (2,6), (3,5), (4,3), (5,6), (6,5), (7,3), (8,7)]$ Weak-Strong H bond $\hat{X}(i) = [(1,1), (2,2), (3,4), (4,7), (5,1), (6,3), (7,2), (8,5)]$ Amino-Keto $\hat{X}(i) = [(1,3), (2,7), (3,3), (4,6), (5,4), (6,5), (7,2), (8,0)]$	Nükleotidlere biyokimyasal özelliklerine göre x ve y koordinat değerleri atanır
IChing Coding [6,35]	Binary coding with 3 different I Ching tables according to amino acids	X=[AGCTACCGTG] $\hat{X}(i) = [110 100 001 010 100 001 010 101]$	Coding is performed with I Ching tables created according to the three biochemical properties of nucleic acids.
Galois Field Coding [36]	0=0 ⇔ 0 ⇔ A, x <sup>0</sup> =1 ⇔ 1 ⇔ C, x <sup>1</sup> =x ⇔ 2 ⇔ T, x <sup>2</sup> =x+1 ⇔ 3 ⇔ G	X=[AGCTACCGTG] $\hat{X}(i) = [0, 3, 1, 2, 0, 1, 1, 3, 2, 3]$	Nucleotides are assigned numerical values corresponding to their quadratic polynomial representation.
Gray Code Coding [36]	A= 00, T=01, C=10, G= 11	X=[AGCTACCGTG] $\hat{X}(i) = [0010011100 0101101110]$	Two-bit binary codes are assigned to nucleotides by ex-or operation.
K-mer Encoding [37,38]	1-mer coding A → [1,0,0,0] C → [0,1,0,0] G → [0,0,1,0] T → [0,0,0,1]	X=[AGCTACCGTG] $\hat{X}(i) = [1000, 0010, 0100, 0001, 1000, 0100, 0100, 0010, 0001, 0010]$	The DNA sequence is split into k-mer degments and coded with zeros and ones.

Figure 7(a) gives the digitized signal plot of the sample sequence using the Voss coding technique and Figure 7(b) IChing coding technique.

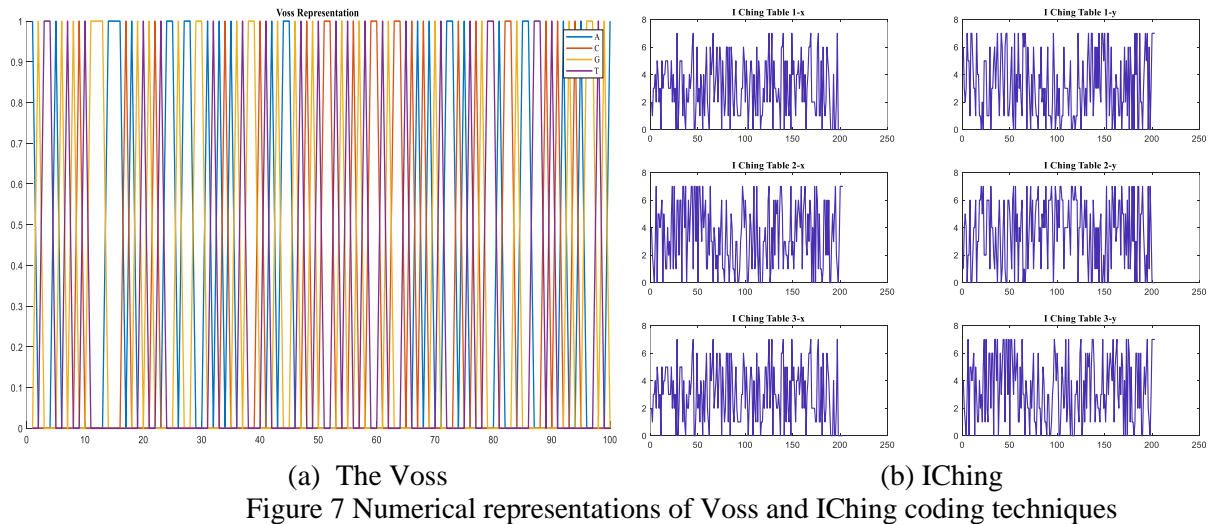
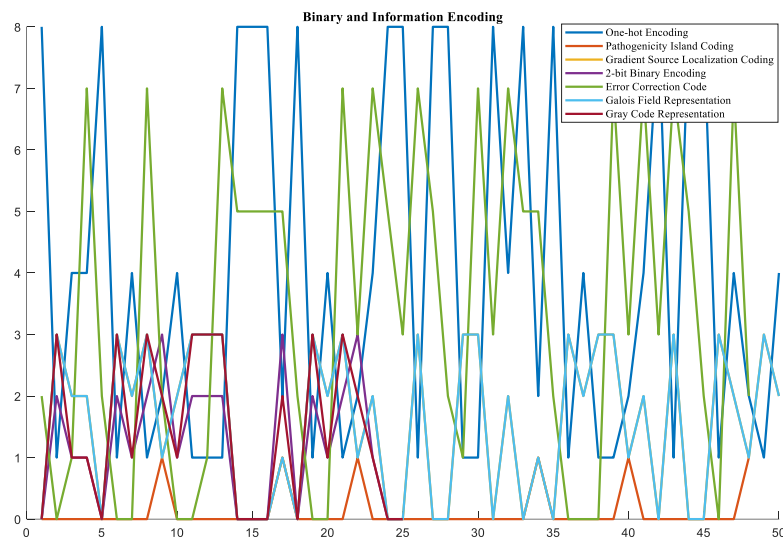


Figure 8 gives the digitized signal representations of the first 100 bases of the NR\_131216.1 reference numbered sequences with the coding techniques in the “Binary and Information Encoding” group.



### 2.4 Primary Structure Properties Group

The fourth group of DNA numerical mapping techniques are primary structure feature (PSP) digitization techniques. In this group, six coding techniques are examined, namely Dinucleotide Representation, Ring Structure Representation, Triplet Encoding, Frequency of Nucleotide Occurrence Mapping, Entropy Based Numerical Mapping, Inter Nucleotide Distance Representation. Table 5 provides a brief summary of all the mapping techniques in the "Primary Structure Properties" group.

Table 5 The summary of all numerical coding techniques in Primary Structure Properties group

The name of technique	Coding Scheme	Numerical Representation	Definition
Dinucleotide Representation [39,40]	16 dinucleotides are placed on the unit circle and coded according to their positions.	$X=[AGCTACCGTG]$ $\hat{X}(i) = [(\cos(\pi/2), \sin(\pi/2)),$ $(\cos(13\pi/4), \sin(13\pi/4)),$ $(\cos(15\pi/8), \sin(15\pi/8)),$ $(\cos(3\pi/4), \sin(3\pi/4)),$ $(\cos(5\pi/8), \sin(5\pi/8)),$	Dinucleotides are distributed evenly around a circle and coded with their coordinate values.

		$(\cos(\pi/8), \sin(\pi/8)), (\cos(2\pi), \sin(2\pi)), (\cos(11\pi/8), \sin(11\pi/8)), (\cos(\pi), \sin(\pi))]$	
Ring Structure Representation [6,41]	AG: (0, 1.5), CT: (0, -1.5), CA:(1,1), TG: (-1, -1), CG: (1, -1), TA: (-1, 1), GA: (1, 0),GT:(0.5, -1.25), GC: (-0.5, 1.25), TC: (-1, 0), AC: (-0.5, 1.25), AT: (0.5, 1.25), AA: (0, 1), TT: (0.5, 0), GG: (0,1), CC: (-0.5, 0).	$X=[AGCTACCGTG]$ $\hat{X}(i) = [(0, 1.5), (0,-1.5), (-0.5, 1.25), (1, -1), (-1, -1)]$	Dinucleotides are placed at the corners of the hexagon according to the six groups they are divided into according to their biochemical properties, and six coding schemes are obtained with six different combinations and coded with the corner coordinates of the hexagon.
Triplet Encoding [6,42]	64 codons are encoded by weights	$X=[AGCTACCGTG]$ $\hat{X}(i) = [15.6, 1.1, 11.3, 18.2, 16.4, 14.4, 2.1, 19.4]$	Nucleotide triplets and amino acid codons are quantified by weight.
Frequency of Occurrence Mapping [6,13]	C=0.27215, T=0.2056, A=0.24300, G=0.27909 or CG:0.01, GC: 0.043, CC: 0.047, GT:0 .049, GG: 0.050, AC: 0.054, TC: 0.057, GA: 0.061, TA:0.067, AG: 0.070, CT: 0.071, TG: 0.074, CA: 0.074, AT: 0.081, AA: 0.097, TT: 0 .097	$X=[AGCTACCGTG]$ $\hat{X}(i) = [0.070, 0.071, 0.054, 0.01, 0.074]$	Nucleotides or dinucleotides are coded with frequency values according to their frequency of occurrence.
Entropy Based Numerical Mapping [43]	Entropy values calculated according to the new formulas are assigned to 64 codons.	$X=[AGCTACCGTG]$ $\hat{X}(i) = [0.7222, 0.8331, 0.9086, 0.8331, 0.8118, 0.5363, 0.6259, 0.9818, 0.9998, 0.9954]$	The codons are coded by calculating the entropy values of the modified and fractional new equation of Shannon's entropy equation.
Inter Nucleotide Distance Representation [44]	Each base is encoded with the value of the base distance between the next itself and the same base.	$X=[AGCTACCGTG]$ $\hat{X}(i) = [4, 6, 3, 5, 5, 1, 3, 2, 1, 0]$	Each base in the DNA sequence is encoded with the base distance value between it and the same base that follows it.

Figure 9(a) gives the digitized signal plot of the reference sequence NR\_131216.1 using the Dinucleotide distance coding technique and Figure 9(b) The Ring Structure technique.

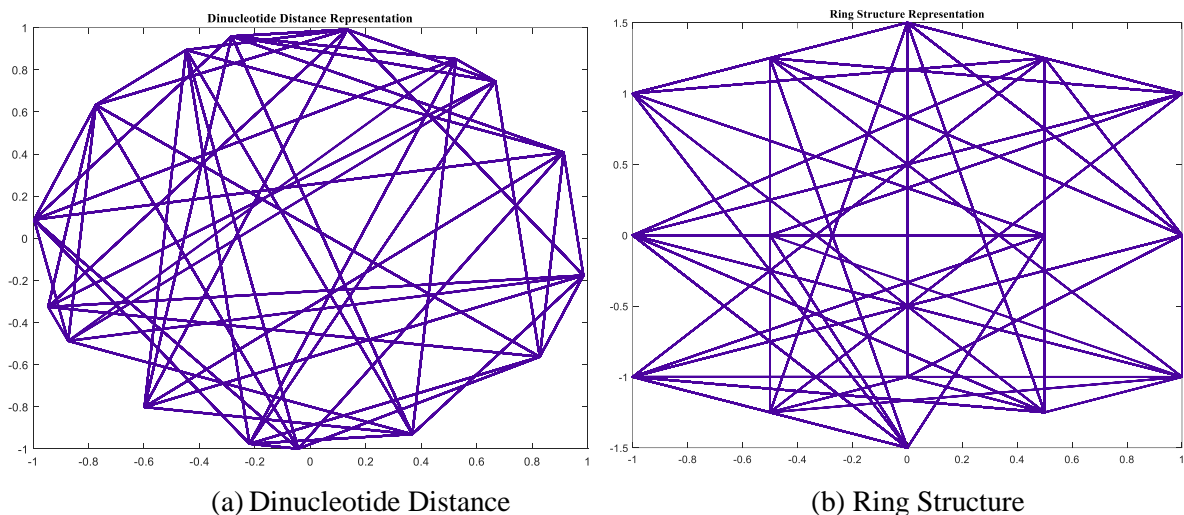


Figure 9 Numerical representations of Dinucleotide distance coding and the Ring Structure techniques

Figure 10 gives the digitized signal plot of the sample sequence using the Frequency of Nucleotide Occurrence coding technique.

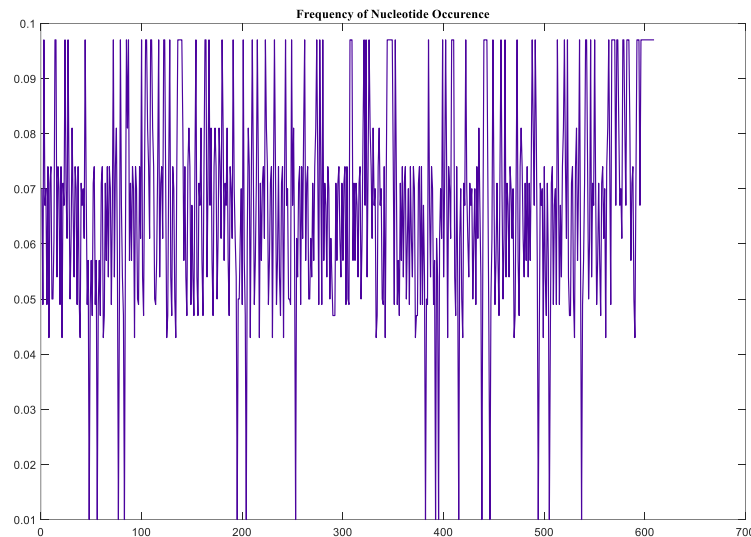


Figure 10 Numerical representation of Frequency of Nucleotide Occurrence Coding

Figure 11 gives the digitized signal representations of the first 100 bases of the NR\_131216.1 reference numbered sequences with the coding techniques in the “Primary Structure Properties” group.

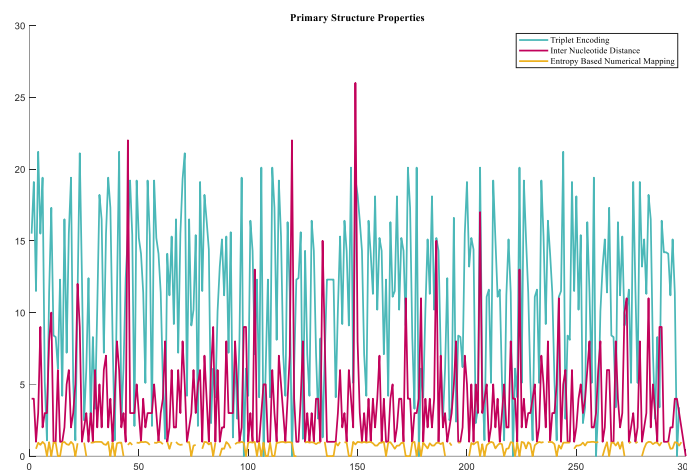


Figure 11 Numerical representation of Primary Structure Properties Group

## 2.5 Graphical Representation Group

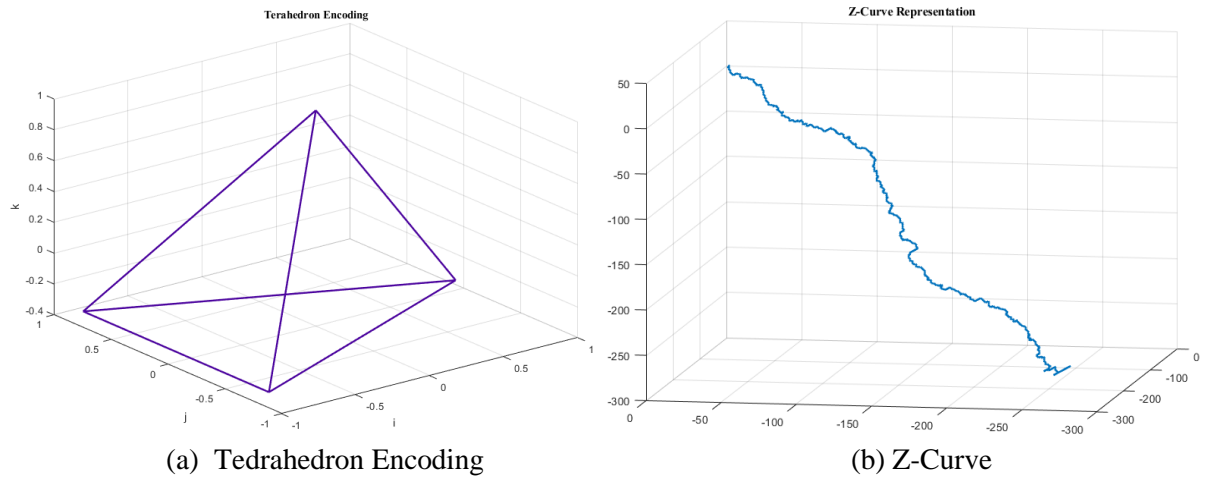
The fifth group of DNA numerical mapping techniques is Graphical Representation (GR) digitization techniques. Within this group, Tetrahedron Encoding, H-Curve Representation, Z-Curve Representation, Quaternion Encoding, SNP-GIN Encoding, Chaos Game Representation, (CGR), Chaos Game Representation Walk (CGR-Walk), Integer Chaos Game Representation (iCGR) Fourteen coding techniques are examined, namely, Som Based Approach, Fermat Spiral Curve Representation, Spectral Dynamic Representation, 2D Dynamic Representation, 3D Dynamic Representation, 8D Dynamic Representation. Table 6 provides a brief summary of all the mapping techniques in the "Graphical Representation" group.

Table 6 The summary of all numerical coding techniques in Graphical representation group

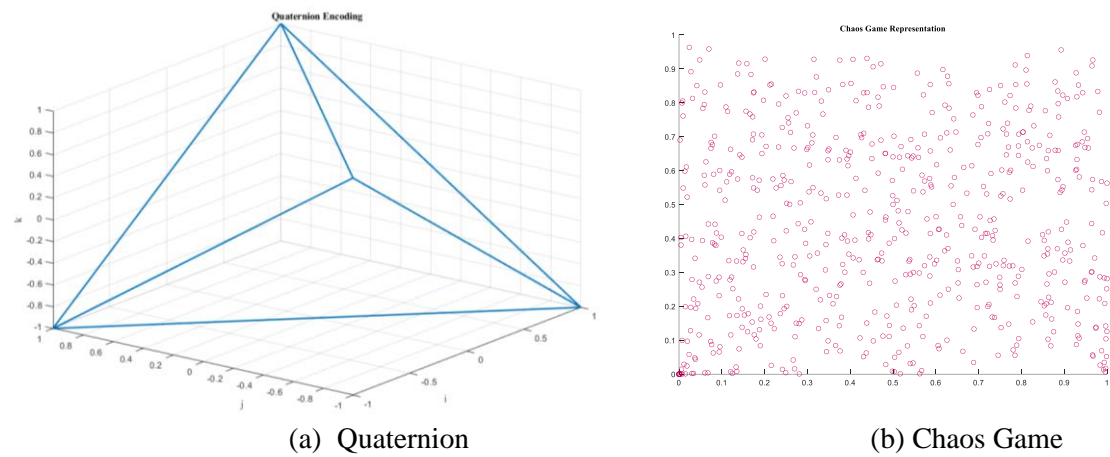
The name of technique	Coding Scheme	Numerical Representation	Definition
Tetrahedron Encoding [6,45]	$A = k,$ $G = -\frac{2\sqrt{2}}{3}i - \frac{\sqrt{6}}{3}j - \frac{1}{3}k,$ $C = -\frac{2\sqrt{2}}{3}i + \frac{\sqrt{6}}{3}j - \frac{1}{3}k,$ $T = \frac{2\sqrt{2}}{3}i - \frac{1}{3}k$	$X = [AGCTACCGTG]$ $\hat{X}_1 = \left[0, -\frac{\sqrt{2}}{3}, -\frac{\sqrt{2}}{3}, \frac{2\sqrt{2}}{3}, 0, -\frac{\sqrt{2}}{3}, -\frac{\sqrt{2}}{3}, -\frac{\sqrt{2}}{3}\right],$ $\frac{2\sqrt{2}}{3}, -\frac{\sqrt{2}}{3},$ $\hat{X}_2 = \left[0, -\frac{\sqrt{6}}{3}, \frac{\sqrt{6}}{3}, 0, 0, \frac{\sqrt{6}}{3}, \frac{\sqrt{6}}{3}, -\frac{\sqrt{6}}{3}, 0, \frac{\sqrt{6}}{3}\right],$ $\hat{X}_3 = \left[1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}\right]$	Nucleotides are placed at the four corners of the tetrahedron and coded by the numerical equations of the corners.
H-Curve Representation [6,46]	$A = \frac{1}{2}i - \frac{\sqrt{3}}{2}j,$ $T = \frac{1}{2}i + \frac{\sqrt{3}}{2}j,$ $C = \frac{\sqrt{3}}{2}i + \frac{1}{2}j,$ $G = \frac{\sqrt{3}}{2}i - \frac{1}{2}j$	$X = [AGCTACCGTG]$ $\hat{X}_1 = [-0.3660i, 0.3660i, 1.3660i, 1.3660i, -0.3660i, 1.3660i, 1.3660i, 0.3660i, 1.3660i, 0.3660i]$	Nucleotides are encoded with functions created by vectors $i$ , and $j$ on the $x$ , and $y$ axes.
Z-Curve Representation [6,47]	$\hat{X}_1(i) \begin{cases} X(i-1) + 1 & \text{if } X(i) = \\ X(i-1) + (-1) & \text{other} \end{cases}$ $\hat{X}_2(i) \begin{cases} X(i-1) + 1 & \text{if } X(i) = \\ X(i-1) + (-1) & \text{other} \end{cases}$ $\hat{X}_3(i) \begin{cases} X(i-1) + 1 & \text{if } X(i) = \\ X(i-1) + (-1) & \text{other} \end{cases}$	$X = [AGCTACCGTG]$ $\hat{X}_1 = [-1, 0, -1, 0, -1, -2, -3, -2, -1, 0]$ $\hat{X}_2 = [1, 0, 1, 0, 1, 2, 3, 2, 1, 0]$ $\hat{X}_3 = [1, 0, -1, 0, 1, 0, -1, -2, -1, -2]$	Nucleotides are encoded by sets of vectors towards the four faces of the tetrahedron.
Quaternion Encoding [6,18]	$A = i+j+k, C = -i+j-k, G = -i-j+k,$ $T = i-j-k$	$X = [AGCTACCGTG]$ $\hat{X}(i) = [i+j+k, -i-j+k, -i+j-k, i-j-k, i+j+k, -i+j-k, -i-j-k, -i-j+k, i-j-k, -i-j+k]$	Nucleotides are represented by 4D quaternion equations.
SNP GIN Encoding [49]	$1000 \rightarrow A A \text{ veya } A -$ $0100 \rightarrow C C \text{ veya } C -$ $0010 \rightarrow G G \text{ veya } G -$ $0001 \rightarrow T T \text{ veya } T -$ $1100 \rightarrow A C, 1010 \rightarrow A G$ $1001 \rightarrow A T, 0110 \rightarrow C G$ $0101 \rightarrow C T, 0011 \rightarrow G T$	$X = [AGCTACCGTG]$ $\hat{X}(i) \Rightarrow \text{Genotypes} \rightarrow A G C T A C C G T G$ Nucleotides $\rightarrow AGCTACCGTG$ Binary Encoding $\rightarrow 10100101110001100000$ Hexadecimal $\rightarrow A5C60$	GINs are created by assigning four pairs of SNPs to nucleotides.
Chaos Game Representation (CGR) [6,50]	$A: (0, 0), T: (1, 0), G: (1, 1),$ $C: (0, 1)$	$X = [AGCTACCGTG]$ $\hat{X}_1 = [(0,0), (0,0), (0.5, 0.5), (0.25, 0.75), (0.625, 0.375), (0.3125, 0.1875), (0.1563, 0.5938), (0.0781, 0.7969), (0.5391, 0.8984), (0.7695, 0.4492)]$	The DNA sequence is mapped according to the coordinate values within the unit square.
Chaos Game Representation Walk (CGR-Walk) [51,52]	$CGR_{RY}: A(0, 0), T(1, 0), C(0, 1), G(1, 1)$ $CGR_{MK}: A(0, 0), T(1, 0), G(0, 1), C(1, 1)$ $CGR_{WS}: A(0, 0), G(1, 0), C(0, 1), T(1, 1)$	$X = [AGCTACCGTG]$ Purine-Pyrimidine $\hat{X}_1 = [(0,0), (0,0), (0.5, 0.5), (0.25, 0.75), (0.625, 0.375), (0.3125, 0.1875), (0.1563, 0.5938), (0.0781, 0.7969), (0.5391, 0.8984), (0.7695, 0.4492)]$ Amino-Keto $\hat{X}_1 = [(0,0), (0,0), (0, 0.5), (0.5, 0.75), (0.75, 0.375), (0.375, 0.1875), 0.6875, 0.5938), 0.8438, 0.7969), (0.4219, 0.8984), 0.7109, 0.4492)]$ Weak-Strong $\hat{X}_1 = [(0,0), (0,0), (0.5, 0), (0.25, 0.5), (0.625, 0.75), (0.3125, 0.375), (0.1563, 0.6875), (0.0781, 0.8438), (0.5391, 0.4219), (0.7695, 0.7109)]$	The chaos game is performed in the form of a DNA walk, taking into account the thermodynamic properties of representative DNA.

Integer Chaos Game Representation (iCGR) [53]	A=(1,1), T=(-1,1), C=(-1,-1), G=(1,-1)	X=[AGCTACCGTG] $\hat{X}_1 = [(1,1), (3, -1), (-1,-5), (-9,3), (7,19), (-25,-13), (-89, -77), (39,-205), (-217, 51), (295, -461)]$	The Chaos game representation is performed with integers rather than floating-point numbers.
SOM Based Approach [6,54]	A: (0, 0, 0), T: (0.289, 0.5, 0.816), C: (0.866, 0.5, 0), G: (0, 1, 0)	X=[AGCTACCGTG] $\hat{X}_1 = [(0,0,0), (0,1,0), (0.866, 0.5, 0), (0.289, 0.5, 0.816), (0,0,0), (0.866, 0.5, 0), (0.866, 0.5, 0), (0,1,0), (0.289, 0.5, 0.816), (0,1,0)]$	Nucleotides are paired with all four corners. Coding is performed with the distance values between the AG and CT vertices.
Fermat Spiral Curve Representation [55]	Representation of the four sub-strings formed according to the positions A, T, C, and G in the Fermat spiral	X=[AGCTACCGTG] $\hat{X}_1$ (Aseq)= [1,0,0,0,5,0,0,0,0,0] $\hat{X}_1$ (Gseq)= [0,2,0,0,0,6,0,8,0,0] $\hat{X}_1$ (Cseq)= [0,0,0,0,0,0,0,9,0] $\hat{X}_1$ (Tseq)= [0,0,3,4,0,0,7,0,0,10]	Global and local location information of nucleotides in DNA is mapped on the Fermat spiral.
Spectral Dynamic Representation [56]	Representation of the effusions of each base by a series of lines	X=[AGCTACCGTG] $\hat{X}_1$ (A)= [1,0,0,0,1,0,0,0,0,0] $\hat{X}_1$ (G)= [0,1,0,0,0,1,0,1,0,0] $\hat{X}_1$ (C)= [0,0,0,0,0,0,0,0,1,0] $\hat{X}_1$ (T)= [0,0,1,1,0,0,1,0,0,1]	The distributions of DNA nucleotides are represented by four separate split line plots.
2D Dynamic Representation [57]	A=(-1,0), G=(1, 0), C=(0, 1), T=(0,-1)	X=[AGCTACCGTG] $\hat{X}_1 = [(-1,0), (0,0), (0,1), (0,0), (-1,0), (-1,1), (-1,2), (0,2), (0,1) (1,1)]$	DNA sequences are represented by point masses in the 2D Euclidean space.
3D Dynamic Representation [57,58]	A=(-1, 0, 1), G=(1, 0, 1), C=(0, 1, 1), T=(0, -1, 1)	X=[AGCTACCGTG] $\hat{X}_1 = [(-1,0,1), (0,0,2), (0,1,3), (0,0,4), (-1,0,5), (-1,1,6), (-1,2,7), (0,2,8), (0,1,9), (1,1,10)]$	DNA/RNA sequences are represented in the 3D plane.
8D Vector Representation [59,60]	A=(1, 0.2), T=(1, -0.2), C=(1, 0.3), G=(1, -0.3) $z_i = y_i / i \quad K = (m_z, v_z)$ $m_z = \frac{1}{n} \sum_{i=1}^n v_z =$ $\frac{1}{n} \sum_{i=1}^n (z_i - m_z)^2$	X=[AGCTACCGTG] $\hat{X}_1 = [(1, 0.2), (2, -0.2), (3, 0.2), (4,0), (5, 0.2), (6, 0.5), (7, 0.8), (8, 0.5), (9, 0.3), (10, 0)]$ Slope=1.5370e-04 Variance=0.0182+0.0200i	8D vectors are formed with mean, variance values from a zigzag plot of DNA/RNA sequences

Figure 12 (a) gives the digitized signal plot of the sample sequence using the tetrahedron encoding technique and (b) Z-Curve representation. Figure 13(a) gives the digitized signal plot of the sample sequence using the quaternion encoding Figure 13(b) Chaos Game representation. Figure 14 provides a digitized signal plot of the sample sequence according to the thermodynamic properties of purine-pyrimidine, amino-keto, and strong-weak H bonds using the Chaos game representation walk technique. Figure 15 gives the digitized signal plot of the sample sequence using the SOM-based coding technique.



(a) Tetrahedron Encoding (b) Z-Curve  
Figure 12 Numerical representations of Tetrahedron Encoding and Z-Curve techniques



(a) Quaternion (b) Chaos Game  
Figure 13 Numerical representations of Quaternion Encoding and Chaos Game techniques

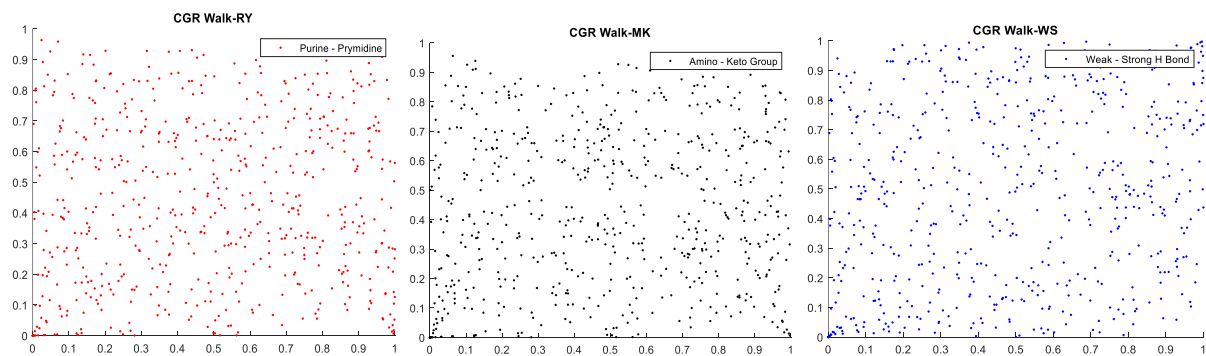


Figure 14 Numerical representation of Chaos Game Representation Walk (RY, MK, WS)

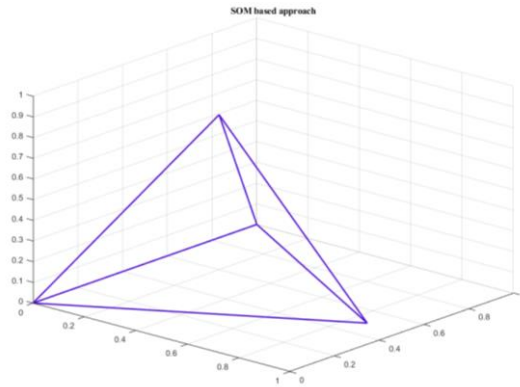


Figure 15 Numerical representation of SOM based coding

Figure 16 gives the digitized signal plot of the sample sequence using the Fermat spiral curve coding technique.

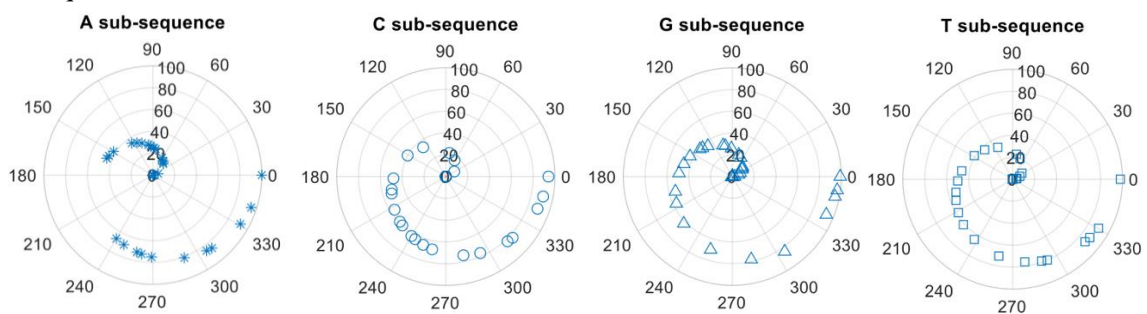


Figure 16 Numerical representation of Fermat spiral curve coding

Figure 17 gives a digitized signal plot of the sample sequence using the Spectral dynamic representation technique.

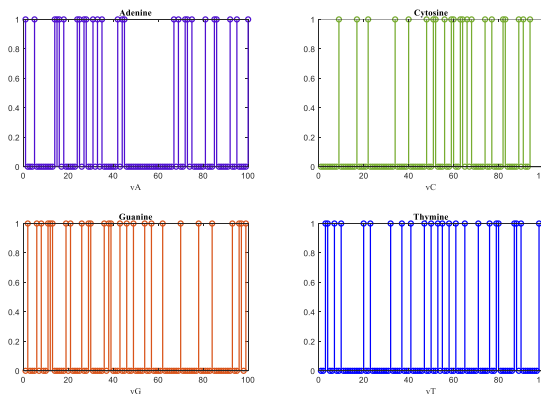


Figure 17 Numerical representation of Spectral dynamic representation

Figure 18(a) gives the digitized signal plot of the sample array using the 2D dynamic representation technique, Figure 18(b) 3D dynamic, Figure 18(c) 8D dynamic.



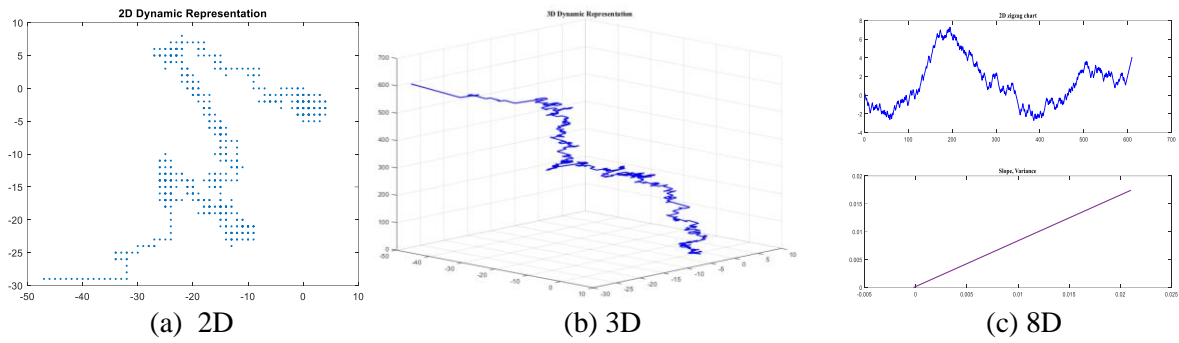


Figure 18 Numerical representations of 2D, 3D and 8D dynamic representations

### 3. Performance Comparison of Numerical Mapping Techniques in Genomic Fields

The aim of this review is to analyze how the digital mapping techniques (coding scheme -numerical methods), which are used for digitizing DNA sequences in bioinformatics studies and have gained popularity in recent years, affect performance in genomic fields. In this section, the frequency of use of DNA coding techniques for 4 popular genomic fields (identification of exon regions, exon-intron classification, phylogenetic analysis, gene detection) and the min-max range of the performances obtained using these techniques in that field was given. Figure 19 shows the most used numerical coding techniques for the identification of exon regions. Table 7 shows min-max performance intervals of the most used coding techniques for the identification of exon regions.

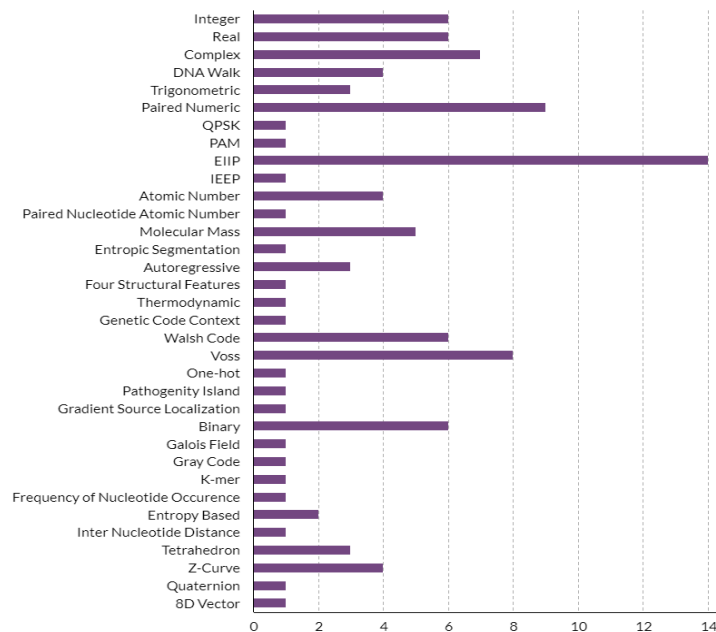


Figure 19 The most used coding techniques for the identification of exon regions

Table 7 Min-max performance intervals of the coding techniques in the identification of exon regions

Coding Technique	Min-Max Performance Interval (%)	Coding Technique	Min-Max Performance Interval	Coding Technique	Min-Max Performance Interval
Integer	36-81	EIIP	48-97	Voss	60-92
Real	50-81	IIEP	80-88	One-hot	70-90
Complex	64-75	Autoregressive	48-76	Binary	65-75
DNA-Walk	78-100	Four structural features	70-78	Pathogenity	60-70
Trigonometric	73-87	Thermodynamic	70-80	Gray code	65-80
Paired numeric	47-94	Genetic code context	66-79	K-mer	85-93
Frequency nucleotide occurence	81-100	Walsh code	83-91	8D vector	70-75
Z-curve	83-95	Galois field	65-82	Atomic number	39-86
Inter nucleotide distance	80-85	PAM	58-75	Entropic segmentation	72-86
Tetrahedron	71-79	Entropy-based	92-100	Gradient Source localization	65-80
Quaternion	70-75	Moleculer Mass	51-68	Paire dnucleotide atomic	80-86

As seen in Figure 19, the three most commonly used techniques for the identification of exon regions are EIIP, Paired numeric and Voss techniques, respectively. Looking at Table 7, it is seen that the performance values obtained in detecting exon regions with these three techniques are above 90%. However, although not used as often as these techniques in the literature, maximum 100% performance has been achieved in studies using Entropy-based, Frequency nucleotide occurrence, and DNA-Walk techniques. Therefore, this review is thought to increase the use of these techniques in the introduction of these techniques and in most genomic areas from now on. Figure 20 shows the most used numerical coding techniques for the classification of exon-intron.

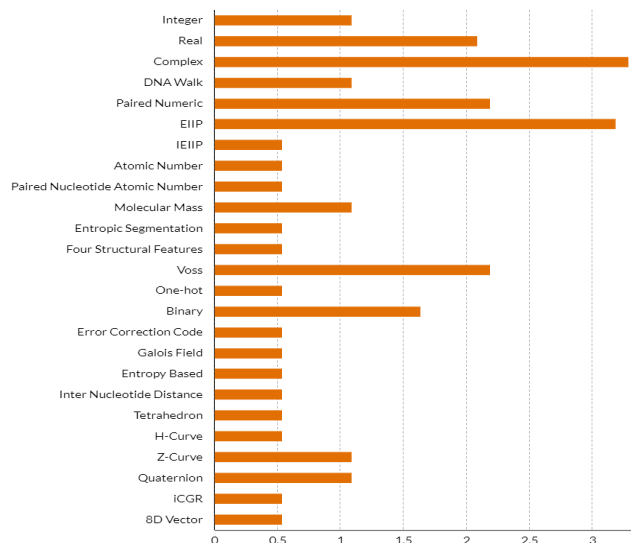


Figure 20 The most used coding techniques for the classification of exon-intron

Table 8 shows min-max performance intervals of the most used coding techniques for the classification of exon-intron.

Table 8 Min-max performance intervals of the coding techniques in the classification of exon-intron

Coding Technique	Min-Max Performance Interval (%)	Coding Technique	Min-Max Performance Interval	Coding Technique	Min-Max Performance Interval
Complex	60-80	DNA-Walk	67-95	Error correction code	60-70
Entropic segmentation	65-80	Integer	65-96	Entropy-based	92-96
8D vector	70-80	Real	38-61	Galois field	70-75
Z-curve	80-86	Binary	70-88	Paired nucleotide atomic	60-75
Voss	75-88	Paired-numeric	75-95	Molecular mass	59-65
Tetrahedron	60-75	Quaternion	66-95	IEIIP	80-92
EIIP	85-95	Inter nucleotide atomic	75-85	Atomic number	58-76
Four structure featus	75-82	One-hot	64-80	H-curve	60-76

As seen in Figure 20, the three most commonly used techniques for the identification of exon regions are Complex, EIIP and Voss techniques, respectively. Looking at Table 8, while the performances of Complex and Voss techniques were above 80%, the performance increased up to 95% in studies conducted with the EIIP technique. Apart from these, Entropy-based, Integer, and EIIP techniques were used in studies with the highest performance in exon-intron classification. Figure 21 shows the most used numerical coding techniques for the phylogenetic analysis.

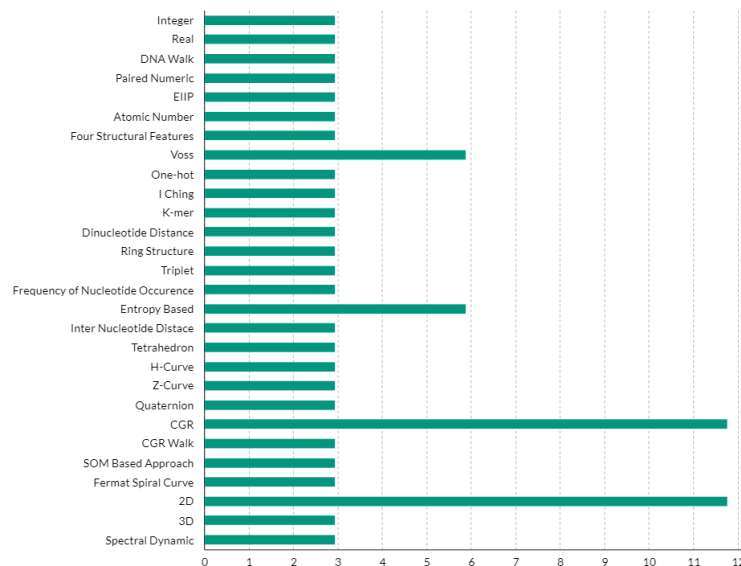


Figure 21 The most used coding techniques for the phylogenetic analysis

Table 9 shows min-max performance intervals of the most used coding techniques for the phylogenetic analysis

Table 9 Min-max performance intervals of the coding techniques in the phylogenetic analysis.

Coding Technique	Min-Max Performance Interval (%)	Coding Technique	Min-Max Performance Interval	Coding Technique	Min-Max Performance Interval
CGR	80-90	Voss	42-75	Quaternion	65-72
2D	78-90	Tetrahedron	36-70	Entropy	82-90
Fermat spiral	70-80	Z-curve	70-100	Four structural features	60-68
Integer	75-84	H-curve	70-80	CGR Walk	50-60
Real	80-100	DNA Walk	75-88	Sprectral Dynamic	55-65
EIIP	86-98	Dinucleotide	66-85	One-hot	75-80
Atomic number	80-98	Inter nucleotide distance	75-80	Inter nucleotide distance	55-72
Paired numeric	80-100	3D	80-90	Triplet encoding	75-83

As seen in Figure 21, the two most commonly used techniques for the phylogenetic analysis are CGR ve 2D techniques. Voss and Entropy-based techniques are the second most frequently used techniques after these. Looking at Table 9, Real and Z-Curve techniques were used in the highest performing studies for phylogenetic analysis. After these, EIIP and atomic number techniques were used in the studies with the highest performance. Figure 22 shows the most used numerical coding techniques for the detection of gene.

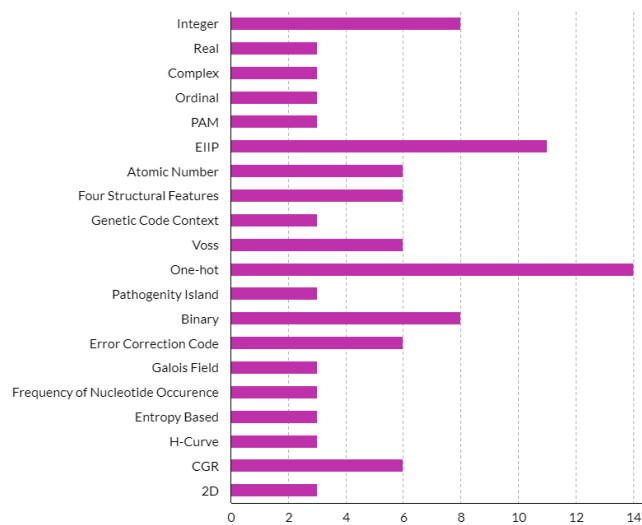


Figure 22 The most used coding techniques for the detection of gene

Table 10 shows min-max performance intervals of the most used coding techniques for the detection of gene.

Table 10 Min-max performance intervals of the coding techniques in the detection of gene.

Coding Technique	Min-Max Performance Interval (%)	Coding Technique	Min-Max Performance Interval	Coding Technique	Min-Max Performance Interval
CGR	70-83	Complex	65-75	One-hot	78-96
Four structural	70-75	Integer	63-99	Voss	80-98
2D	70-75	Real	80-97	Pathogenity Island	65-75
Error correction code	72-78	Binary	91-100	EIIP	61-100
H-curve	70-80	Galois Field	70-75	Atomic number	65-97
Entropy	80-100	Genetic code context	62-79	Frequency of nucleotide	72-100

As seen in Figure 23, the three most commonly used techniques for gene detection are One-hot, EIIP, and Integer techniques, respectively. Looking at Table 10, success performances of 96%, 100%, and 99%, respectively, were obtained in gene screening studies using these techniques. Apart from these, 100% maximum success performance has been achieved in studies using the frequency of nucleotide technique and the Entropy-based technique, although it is not used very often.

#### 4. Conclusion

This study is an attempt to review the DNA numerical mapping techniques used in the analysis of DNA sequences and to present the advantages and disadvantages of each technique to researchers. Each coding technique is exemplified in a DNA sequence, showing how that DNA sequence is digitized. Then, the frequency of use of these coding techniques in the 4 most popular study areas in the last 10 years and the max-min range of the performances obtained using these coding techniques were analyzed. This review will guide researchers in developing new coding techniques, and will facilitate previous researchers to improve their work. It will also guide researchers in discovering new techniques using innovative ideas.

#### References

- [1] R. H. Thomas. "Molecular Evolution and Phylogenetics. Masatoshi Nei and Sudhir Kumar. Oxford University Press, Oxford. 2000. pp. 333. Price £65.00, hardback. ISBN 0 19 513584 9.," *Heredity*, vol. 86, no. 3, pp. 385–385, 2001, doi: 10.1046/j.1365-2540.2001.0923a.x.
- [2] M. Akhtar, J. Epps, and E. Ambikairajah. "Signal Processing in Sequence Analysis: Advances in Eukaryotic Gene Prediction", *IEEE J. Sel. Top. Signal Process.*, vol. 2, no 3, pp 310-321, Jun. 2008, doi: 10.1109/JSTSP.2008.923854.
- [3] L. Das, J. K. Das, S. Mohapatra and S. Nanda. "DNA numerical encoding schemes for exon prediction: a recent history", *Nucleosides, Nucleotides & Nucleic Acids*, vol.40, no 10, pp. 985-1017, Oct. 2021, doi: 10.1080/15257770.2021.1966797.
- [4] U. N. Wisesty, T. R. Mengko and A. Purwarianti. "Gene mutation detection for breast cancer disease: A review", *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 830, no 3, pp. 032051, Apr. 2020, doi: 10.1088/1757-899X/830/3/032051.
- [5] M. Raman Kumar and N. K. Vaegae. "A new numerical approach for DNA representation using modified Gabor wavelet transform for the identification of protein coding regions", *Biocybernetics and Biomedical Engineering*, vol. 40, no 2, pp. 836-848, Apr. 2020, doi: 10.1016/j.bbe.2020.03.007.

- [6] N. Yu, Z. Li, and Z. Yu. "Survey on encoding schemes for genomic data representation and feature learning—from signal processing to machine learning", *Big Data Mining and Analytics*, vol. 1, no 3, pp. 191-210, Sep. 2018, doi: 10.26599/BDMA.2018.9020018.
- [7] P. K. Kumari, "A Survey on Numerical Representation of DNA Sequences", *Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146*, Apr. 2018.
- [8] L. Das, J. K. Das, S. Nanda and S. Mohapatra. "DNA Coding Sequence Prediction: A Review", içinde *2018 International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC)*, Oct. 2018, vol. 1, pp. 1-6. doi: 10.1109/AESPC44649.2018.9033278.
- [9] M. Ahmad, L. T. Jung and A.-A. Bhuiyan. "From DNA to protein: Why genetic code context of nucleotides for DNA signal processing? A review", *Biomedical Signal Processing and Control*, vol. 34, pp. 44-63, Apr. 2017, doi: 10.1016/j.bspc.2017.01.004.
- [10] X. Jin et al. "Similarity/dissimilarity calculation methods of DNA sequences: A survey", *Journal of Molecular Graphics and Modelling*, vol. 76, pp. 342-355, Sep. 2017, doi: 10.1016/j.jmglm.2017.07.019.
- [11] G. Mendizabal-Ruiz, I. Román-Godínez, S. Torres-Ramos, R. A. Salido-Ruiz and J. A. Morales, "On DNA numerical representations for genomic similarity computation", *PLOS ONE*, vol. 12, no 3, p. e0173288, Mar. 2017, doi: 10.1371/journal.pone.0173288.
- [12] S. Saini and L. Dewan. "Comparison of Numerical Representations of Genomic Sequences: Choosing the Best Mapping for Wavelet Analysis", *Int. J. Appl. Comput. Math*, vol. 3, no 4, pp. 2943-2958, Dec. 2017, doi: 10.1007/s40819-016-0277-1.
- [13] Mabrouk, M.S. Advanced Genomic Signal Processing Methods in DNA Mapping Schemes for Gene Prediction Using Digital Filters --Gene prediction, Digital filters, 3- Base periodicity, Exon, Intron, Bioinformatics, Genomic signal processing", *American Journal of Signal Processing*, p. 13, 2017.
- [14] L. Das, J. K. Das and S. Nanda, "Identification of exon location applying kaiser window and DFT techniques", içinde *2017 2nd International Conference for Convergence in Technology (I2CT)*, Apr. 2017, pp. 211-216. doi: 10.1109/I2CT.2017.8226123.
- [15] B. Das and I. Turkoglu. "Classification of DNA sequences using numerical mapping techniques and Fourier transformation, Journal of the Faculty of Engineering and Architecture of Gazi University, 2016, doi: 10.17341/gazimmfd.278447.
- [16] M. Abo-Zahhad, S. M. Ahmed and S. A. Abd-Elrahman. "Integrated Model of DNA Sequence Numerical Representation and Artificial Neural Network for Human Donor and Acceptor Sites Prediction", *IJITCS*, vol. 6, no 8, pp. 51-57, July. 2014, doi: 10.5815/ijitcs.2014.08.07.
- [17] M. Abo-Zahhad, S. M. Ahmed and S. A. Abd-Elrahman. "Genomic Analysis and Classification of Exon and Intron Sequences Using DNA Numerical Mapping Techniques", *IJITCS*, vol. 4, no 8, pp. 22-36, July. 2012, doi: 10.5815/ijitcs.2012.08.03.
- [18] H. K. Kwan, B. Y. M. Kwan and J. Y. Y. Kwan, "Novel methodologies for spectral classification of exon and intron sequences", *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no 1, p. 50, Feb 2012, doi: 10.1186/1687-6180-2012-50.
- [19] S. D. Sharma, K. Shakya and S. N. Sharma, "Evaluation of DNA mapping schemes for exon detection", in *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, Mar. 2011, pp. 71-74. doi: 10.1109/ICCCET.2011.5762441.
- [20] F. Akalin and N. Yumusak. "Classification of exon and intron regions obtained using digital signal processing techniques on the DNA genome sequencing with EfficientNetB7 architecture", *GUMMFD*, 37:3 (2022) 1355-1371.
- [21] F. Akalin and N. Yumusak. "Classification of ALL and CML malignancies being among the main types of leukaemia with graph neural networks and fuzzy logic algorithm," *GUMMFD*, Mar. 2022, doi: 10.17341/gazimmfd.1022624.
- [22] L. Das, S. Nanda and J. K. Das. "An integrated approach for identification of exon locations using recursive Gauss Newton tuned adaptive Kaiser window", *Genomics*, vol. 111, no 3, pp. 284-296, May. 2019, doi: 10.1016/j.ygeno.2018.10.008.
- [23] A. C. H. Choong and N. K. Lee. "Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method", içinde *2017 International Conference*


- on *Computer and Drone Applications (IconDA)*, Nov. 2017, pp. 60-65. doi: 10.1109/ICONDA.2017.8270400.
- [24] N. Chakravarthy, A. Spanias, L. D. Iasemidis and K. Tsakalis. "Autoregressive Modeling and Feature Analysis of DNA Sequences", *EURASIP J. Adv. Signal Process.*, vol. 2004, no 1, pp. 952689, Jan. 2004, doi: 10.1155/S111086570430925X.
- [25] R. M. Kumar and N. K. Vaegae. "Walsh code based numerical mapping method for the identification of protein coding regions in eukaryotes", *Biomedical Signal Processing and Control*, vol. 58, no. 101859, Ap. 2020, doi: 10.1016/j.bspc.2020.101859.
- [26] B. Das, S. Toraman, and I. Turkoğlu. "A novel genome analysis method with the entropy-based numerical technique using pretrained convolutional neural networks," *Turk J Elec Eng & Comp Sci*, vol. 28, no. 4, pp. 1932–1948, Jul. 2020, doi: 10.3906/elk-1909-119.
- [27] P. Bernaola-Galván, I. Grosse, P. Carpena, J. L. Oliver, R. Román-Roldán and H. E. Stanley. "Finding Borders between Coding and Noncoding DNA Regions by an Entropic Segmentation Method", *Phys. Rev. Lett.*, vol. 85, no 6, pp. 1342-1345, Aug. 2000, doi: 10.1103/PhysRevLett.85.1342.
- [28] D. Nicorici and J. Astola. "Segmentation of DNA into Coding and Noncoding Regions Based on Recursive Entropic Segmentation and Stop-Codon Statistics", *EURASIP J. Adv. Signal Process.*, vol. 2004, no 1, pp. 832471, Dec. 2004, doi: 10.1155/S1110865704309212.
- [29] N. Y. Song and H. Yan. "Autoregressive modeling of DNA features for short exon recognition", in *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2010, pp. 450-455. doi: 10.1109/BIBM.2010.5706608.
- [30] Q. Zheng, T. Chen, W. Zhou, L. Xie and H. Su. "Gene prediction by the noise-assisted MEMD and wavelet transform for identifying the protein coding regions", *Biocybernetics and Biomedical Engineering*, Vol. 41, no 1, 2021, doi: 10.1016/j.bbe.2020.12.005.
- [31] R. Harrison, Y. Li and I. Măndoiu, Ed. *Bioinformatics Research and Applications: 11th International Symposium, ISBRA 2015 Norfolk, USA, June 7-10, 2015 Proceedings*, c. 9096. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-19048-8.
- [32] Z. Abbas, H. Tayara and K. T. Chong. "4mCPred-CNN—Prediction of DNA N4-Methylcytosine in the Mouse Genome Using a Convolutional Neural Network", *Genes*, vol. 12, no 2, Feb. 2021, doi: 10.3390/genes12020296.
- [33] P. Liò and M. Vannucci. "Finding pathogenicity islands and gene transfer events in genome data", *Bioinformatics*, vol. 16, no 10, pp. 932-940, Oct. 2000, doi: 10.1093/bioinformatics/16.10.932.
- [34] L. Zhang, F. Tian, S. Wang and X. Liu. "A novel coding method for gene mutation correction during protein translation process", *Journal of Theoretical Biology*, vol. 296, pp. 33-40, Mar. 2012, doi: 10.1016/j.jtbi.2011.11.031.
- [35] F. Castro-Chavez. "Defragged Binary I Ching Genetic Code Chromosomes Compared to Nirenberg's and Transformed into Rotating 2D Circles and Squares and into a 3D 100% Symmetrical Tetrahedron Coupled to a Functional One to Discern Start from Non-Start Methionines through a Stella Octangula", *J Proteome Sci Comput Biol*, vol. 2012, no 1, pp. 3, 2012, doi: 10.7243/2050-2273-1-3.
- [36] M. Raman Kumar and V. Naveen Kumar. "A Numerical Representation Method for a DNA Sequence Using Gray Code Method", içinde *Soft Computing for Problem Solving*, Singapore, 2020, pp. 645-654. doi: 10.1007/978-981-15-0184-5\_55.
- [37] L. Deng, H. Wu, X. Liu and H. Liu. "DeepD2V: A Novel Deep Learning-Based Framework for Predicting Transcription Factor Binding Sites from Combined DNA Sequence", *International Journal of Molecular Sciences*, vol. 22, no 11, Jan. 2021, doi: 10.3390/ijms22115521.
- [38] Q. Zhang, Z. Shen and D.-S. Huang, "Modeling in-vivo protein-DNA binding by combining multiple-instance learning with a hybrid deep neural network", *Sci Rep*, vol. 9, no 1, p. 8484, June. 2019, doi: 10.1038/s41598-019-44966-x.
- [39] M. Randić, D. Butina and J. Zupan, "Novel 2-D graphical representation of proteins," *Chemical Physics Letters*, vol. 419, no. 4, pp. 528–532, Feb. 2006, doi: 10.1016/j.cplett.2005.11.091.
- [40] Z. Liu, B. Liao, W. Zhu and G. Huang, "A 2D graphical representation of DNA sequence based on dual nucleotides and its application", *International Journal of Quantum Chemistry*, vol. 109, no 5, pp. 948-958, 2009, doi: 10.1002/qua.21919.

- [41] A. T. M. Bari, M. Reaz, A. T. Islam, H.-J. Choi, and B.-S. Jeong. "Effective Encoding for DNA Sequence Visualization Based on Nucleotide's Ring Structure", *Evolutionary bioinformatics online*, vol. 9, pp. 251-61, July. 2013, doi: 10.4137/EBO.S12160.
- [42] S. Zou, L. Wang and J. Wang. "A 2D graphical representation of the sequences of DNA based on triplets and its application", *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2014, no 1, pp. 1, Jan 2014, doi: 10.1186/1687-4153-2014-1.
- [43] B. Das and I. Turkoglu. "A novel numerical mapping method based on entropy for digitizing DNA sequences", *Neural Comput & Applic*, vol. 29, 8: 207-215, Apr. 2018, doi: 10.1007/s00521-017-2871-5.
- [44] A. Sankar, A. Nair and M. Thiru. "Visualization of genomic data using inter-nucleotide distance signals", Jan. 2005.
- [45] Das, B. "A deep learning model for identification of diabetes type 2 based on nucleotide signals". *Neural Comput & Applic* (2022). <https://doi.org/10.1007/s00521-022-07121-8>
- [46] Das, B. "An implementation of a hybrid method based on machine learning to identify biomarkers in the Covid-19 diagnosis using DNA sequences", *Chemometrics and Intelligent Laboratory Systems* (2022),v. 230, 104680, [tps://doi.org/10.1016/j.chemolab.2022.104680](https://doi.org/10.1016/j.chemolab.2022.104680)
- [47] C.-T. Zhang and J. Wang. "Recognition of protein coding genes in the yeast genome at better than 95% accuracy based on the Z curve", *Nucleic Acids Research*, vol. 28, no 14, pp. 2804-2814, Tem. 2000, doi: 10.1093/nar/28.14.2804.
- [48] C. Yu, M. Deng, L. Zheng, R. L. He, J. Yang and S. S.-T. Yau, "DFA7, a New Method to Distinguish between Intron-Containing and Intronless Genes", *PLOS ONE*, vol. 9, no 7, pp. e101363, Tem. 2014, doi: 10.1371/journal.pone.0101363.
- [49] R. R. Garafutdinov, A. R. Sakhabutdinova, P. A. Slominsky, F. G. Aminev and A. V. Chemeris, "A new digital approach to SNP encoding for DNA identification", *Forensic Science International*, vol. 317, no. 110520, Dec. 2020, doi: 10.1016/j.forsciint.2020.110520.
- [50] T. Hoang, C. Yin and S. S.-T. Yau, "Numerical encoding of DNA sequences by chaos game representation with application in similarity comparison", *Genomics*, vol. 108, no 3, pp. 134-142, Oct 2016, doi: 10.1016/j.ygeno.2016.08.002.
- [51] W. Deng and Y. Luan, "Analysis of Similarity/Dissimilarity of DNA Sequences Based on Chaos Game Representation", *Abstract and Applied Analysis*, vol. 2013, p. e926519, Mar. 2013, doi: 10.1155/2013/926519.
- [52] Z.-G. Yu and V. Anh. "Time series model based on global structure of complete genome", *Chaos, Solitons & Fractals*, vol. 12, no 10, pp. 1827-1834, Aug. 2001, doi: 10.1016/S0960-0779(00)00147-8.
- [53] C. Yin, "Encoding and Decoding DNA Sequences by Integer Chaos Game Representation", *Journal of Computational Biology*, vol. 26, no 2, pp. 143-151, Feb. 2019, doi: 10.1089/cmb.2018.0173.
- [54] A. P. Boyle et al., "Comparative analysis of regulatory information and circuits across distant species", *Nature*, vol. 512, no 7515, Aug. 2014, doi: 10.1038/nature13668.
- [55] Z. Mo et al., "One novel representation of DNA sequence based on the global and local position information", *Sci Rep*, vol. 8, no 1, p. 7592, May. 2018, doi: 10.1038/s41598-018-26005-3.
- [56] D. Bielińska-Wąż and P. Wąż, "Spectral-dynamic representation of DNA sequences", *Journal of Biomedical Informatics*, vol. 72, pp. 1-7, Aug. 2017, doi: 10.1016/j.jbi.2017.06.001.
- [57] A. Czerniecka, D. Bielińska-Wąż, P. Wąż, and T. Clark, "20D-dynamic representation of protein sequences", *Genomics*, vol. 107, no 1, pp. 16-23, Jan. 2016, doi: 10.1016/j.ygeno.2015.12.003.
- [58] D. Zhang. "A New Numerical Method for DNA Sequence Analysis Based on 8-Dimensional Vector Representation", *Journal of Applied Mathematics and Physics*, vol. 7, no 12, Dec. 2019, doi: 10.4236/jamp.2019.712204.
- [59] F. Ben Nasr, and A. E. Oueslati, "CNN for human exons and introns classification", içinde *2021 18th International Multi-Conference on Systems, Signals Devices (SSD)*, Mar. 2021, pp. 249-254. doi: 10.1109/SSD52085.2021.9429303.
- [60] A. Rokas, "Phylogenetic Analysis of Protein Sequence Data Using the Randomized Axelerated Maximum Likelihood (RAXML) Program", *Current Protocols in Molecular Biology*, vol. 96, no 1, pp. 19.11.1-19.11.14, 2011, doi: 10.1002/0471142727.mb1911s96.





## Process Mining in Manufacturing: A Literature Review

 Yüksel Yurtay

Sakarya University, Department of Computer Engineering; Sakarya, Turkey; yyurtay@sakarya.edu.tr;

Received 22 June 2022; Revised 30 September 2022; Accepted 17 November 2022; Published online 31.12.2022

### Abstract

Process mining in manufacturing is a newly expanding field of research in the application of data mining and machine learning techniques and the focus of business processes. Although it is an exciting subject of the recent past and business processes, sufficient research has not been done. Decision support systems such as enterprise resource planning, customer relationship management, and management information systems store the most valuable resource data of process details and event logs. In the advanced information systems of tomorrow, the process management, analysis, and modelling functions of modern enterprises will take their place as a necessity. As a requirement, the fundamental purpose of process mining in production is to refine data from event logs, automatically create process models, compare models with event logs, and improve and make development continuous. Our study points to the definitions of studies published in the context of production with different titles and keywords. At the same time, it contributes to the access of researchers scanning in the main title of production and brings it to their attention. It is based on the literature review and primary stages of business process mining publications in the last decade with a production focus. An overview is discussed as a roadmap for future research with meaningful results.

**Keywords:** production process mining, business process mining, business process management, production, literature review

### 1. Introduction

Businesses consume a lot of resources to execute, analyze and manage business process models. Companies that want to reduce resource consumption tend to apply techniques and approaches that can structure a more efficient operation. Owning new solutions for business processes and operations with process mining instruments is an exciting business agenda. The primary purpose of process mining is to obtain process-centred information by analyzing event logs through business information systems. In the context of the process, it makes visible the actual situation, information and the secret, curable and problematic recipes of the operation of the business. With the process view being achieved, it is seen that business processes form the centre of the organization [3]. At this point, the meaningful association of the data owned by the enterprises includes the most appropriate functioning of the organizations in the context of the process. Suppose more than one organization contributes to the operation of the process. In that case, it requires consolidation of many business processes because organizations work interactively under the pressure of interdependence and intend to create shared value. Actions such as business process management, modelling and analysis are topics of increasing interest for organizational managers oriented towards a specific goal [4].

Today's organizations use process-aware information systems (PAIS) in automated modern business environments [14]. PAIS runs on structures where complex business processes are described as a "naturally distributed system" [2]. Information systems containing many open process models, such as supply chain management systems (SCM), financial systems (FS), and customer relationship management systems (CRM), can be given as examples. The processing and analysis of data obtained through business processes in information systems are manually performed [6,9,11]. With this workload in information systems, flexibility and definition have not received enough attention, as well as the developed open process concepts [1]. As the consistency or comparison of business process models emerged as a need, the first signs of process mining began to be seen. In the recent past, process mining was recognized as one of the significant innovations in business process management in 2012[7]. Then, key concepts, manifestos, roadmap and challenges were defined as a research area [5]. The process

mining research area, which uses data mining techniques to discover automatically and dynamically consistent process models and detect anomalies and their causes, has been announced [1]. Today, process mining is considered a broad discipline that combines the meaningful association, analysis, improvement, execution, and monitoring of the data that makes up the process under the umbrella of data science with information from information technology and management sciences [5]. As a new discipline, process mining focuses on the flexibility, dynamism, comparability and possible anomaly detection of dynamic models [34]. In particular, process mining studies continue to find a place for improvement, development, monitoring and detection of problems in production organizations. In addition, it is known to provide many contributions, such as determining optimal processes, improving product quality and production processes, and easy and fast performance analysis [35].

Its contributions stand out under five headings;

- It paves the way for model discovery,
- Detection of deviations in the process,
- Implementation of conformity check based on alignment,
- Ease of diagnosis in performance improvement,
- Increase process visibility and diagnosis with findings.

Process mining studies and research, which attract the necessary attention in production organizations, continue to attract attention in the scientific field. In our scans, it is seen that the titles given to the publications do not include the phrase "production" or "process mining in production". This situation makes it difficult for researchers and decision-makers in organizations to access research and applications carried out in the context of production. The article draws attention to the distribution in the main headings to facilitate access to the publications. Our motivation is to make the gains visible by drawing attention to the titles in the classification and entry of emerging publications. In addition, the answer is found under which main headings scientific publications on process mining are compiled in the context of production. It also helps to consolidate publications in the context of production by emphasizing the importance of including the title "production". Because facilitating access to publications to be made in the context of production, approaches and similarities used in publications will accelerate and guide various production processes waiting for solutions. At the same time, it will help process mining mature for sectors/areas that have not yet been researched and applied as a new discipline.

This article provides a general framework for process mining in the production context by focusing on the fundamental concepts of the process mining research field. Process mining types, tools, techniques and event logs are the prominent topics of the article. This research study has three main contributions;

- 1-The topics and content are compiled in the context of production process mining.
- 2-It presents the processing of event logs and the grouping of tools in production process mining.
- 3- The interest or analysis of the organizations engaged in production within the process mining framework is shared.

Process mining application points and literature reviews in production are shared. The second part of the study explains the basic concepts and current studies of process mining. The third section describes the form of the literature study, while the fourth section presents the literature research results and discussion. The fifth section includes the results evaluation and future productions.

## **2. Business process mining: an overview**

Organizations' workflows have changed significantly with the innovations in computers and communication. Changing business processes brought complexity and increased the need for information systems. Visualization is critical to improving the understandability of business processes. Key actions of business process mining predict performance, provide insights, discuss responsibilities and analyze compliance. At the same time, it stands out as a new research area defined and promoted through process mining. Therefore, it most closely defines business process mining manifests of process

mining [10]. The primary function of process mining is to view the actual process and discover and improve it with the help of the analysis of event logs obtained from existing information systems [3]. In particular, a business process is a cross-section of any case or process instance. The activity is the executing transaction part of the business process segment. An event is a well-defined step in a process in a particular situation. The event is the first assumption that refers to the activity or task in the process example. The second assumption is that the events in question are sequential [13]. Event logs contain information about incidents and activities and store information about employees, devices, timestamps, and contributors in the data environment. Business process mining aims to bridge the gap between business process management (BPM), and workflow management (WFM) approaches on the one hand and data mining (DM), Machine Learning (ML) and business intelligence (BI) on the other. Process mining focuses on end-to-end process models, while DM, ML and BI concentrate on data. The results can detect/diagnose inefficiencies, performance, bottlenecks, risks and deviations. All disciplines affecting these points will, directly and indirectly, benefit from the process mining results. Institutions that improve and explore their processes will increase competitiveness while obtaining performance and productivity contributions. The following sections briefly share the stages, types, perspectives, tools, techniques, and main problems of process mining in achieving competitive advantage.

Stages of process mining;

- Extracting process models from event logs or performing automatic process discovery,
- Comparison of model and event logs, detection of unusual executions and compliance control,
- With the help of new models, techniques and technologies, steps can be taken for improvement and forward development.

In Figure 1, the context of process mining is shared with its significant concepts. Data such as transactions (messages, work orders, etc.) and event logs of business processes in daily life are stored. Data is filtered, cleaned and extracted. The software performs the desired process mining (discovery, compliance and development) steps.

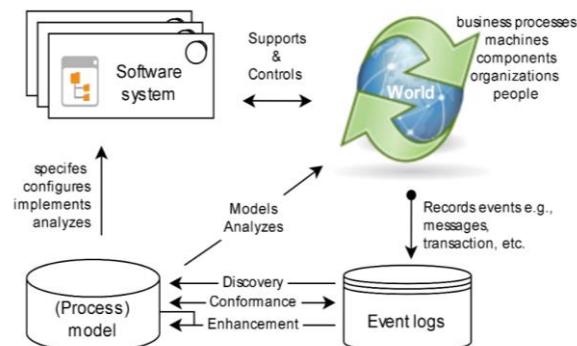


Figure 1 Process mining context [10]

Knowledge systems are the working plane of mining and automation Technologies. Process mining activities include data extraction, filtering and cleaning from information systems. Because, considering that data can be shared with various sources, it is necessary to correct existing data anomalies. Therefore, a set of manifests of event data is an implementation directive. In Figure 2, the principles to be considered in practice are below.

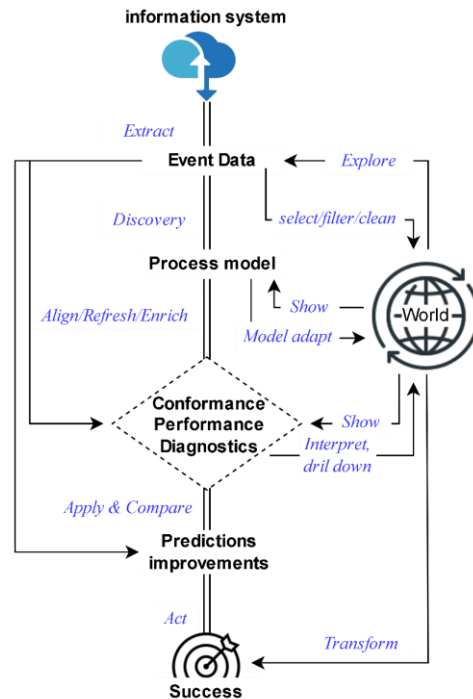


Figure 2 The high-level view [14]

Figure 2 provides a high-level view of the information system to the success achieved. The process mining flow is defined as the upper level, from acquiring event data by inference from information systems to success [14]. Flow can be rearranged for unusual process flows. The guiding principles [15] to be considered in the figure are listed below.

1. The quality of the process mining result is related mainly to the data and its quality; it should be acted on in this context.
2. Extracting meaningful event data with concrete questions is a requirement.
3. Basic control flow (concurrency, selection, etc.) structures should be supported.
4. It should be related to events and model elements.
5. Models derived from event data provide insights into reality.
6. The continuity of process mining should be ensured.

## 2.1. Process mining types

It analyzes three types of event logs under process mining, discovery, compliance, and development [10].

- Discovery requires creating an event log and generating a model without prior knowledge. Typically, the model explored is a process model, similar to a Petri net, BPMN (Business Process Model and Notation), or UML (Unified Modeling Language) activity diagram. In addition, the discovered model can also describe other perspectives, such as social networking [10].

- Compliance is an audit technique that uses event logs and the model as input. Explain the similarities and differences between the output, model and event logs. The fit check can be used to show the conformity or nonconformity of reality to the model, as in recorded event logs [10]. Compliance control is used mainly to detect, explain and analyze deviations in the process [16]. Two different types of metrics are used. It is the accuracy observed in the behaviour of the process model and the degree of openness it represents [17].

- Build is a model development technique that uses an event log and model as input. The resulting output defines the enhanced and extended model. It uses to process information describing event log records

to develop and extend the existing process model [10]. The process model can be changed, corrected and demonstrated at this stage.

## 2.2. Process mining perspectives

Process analysis of institutions is carried out using process execution data. Process data also stores information about its execution. The process perspective reveals the details of how a process is executed. Thus, the process perspective is based on the data and activities of an event log [10]. From the process perspective, the activity sequence is the study's focus. The work also referred to as the control-flow perspective, is to characterize all possible paths with Petri net or other process notations [10]. Control flow stakeholders, people, roles, and the relational status of departments are defined from the organizational perspective. The primary purpose is to observe the event logs and look for answers to many questions such as processing time estimates, bottlenecks, service levels, and resource usage [10].

## 2.3. Process mining software tools and techniques

Process mining software analyzes hidden options, anomalies, and business processes using event logs. Process mining software tools primarily include process discovery, design, analysis and recovery functions. Provides feedback and recommendations for process design, improvement and development. Process analytics examines event log data to detect potential problems. Process recovery or revitalization takes the process one step further for dynamic improvement.

Examples of process mining software include ProM (Open Source, TU/e), ARIS PPM (Software AG), Icris (Icris), Process Mining, BusinessOptix, Disco (Fluxicon), and LANA Process Mining (Lana Labs).

Preferred software should provide the necessary features to implement priority use cases and processes. It should be possible to process commonly used event logs in selecting software in which the user and place of use stand out. Otherwise, it will increase the user's workload in obtaining insights.

Algorithms used in the general framework can be classified into three main categories.

1. Deterministic algorithms: Since the algorithm's output, which accepts all variable data as input, is constant, it produces repeatable models [3].
2. Heuristic algorithms are solution approaches based on insights when the algorithm cannot conclude [3].
3. Genetic algorithms: Better solution models are sought by combining features and adding random variations in operations carried out with an arbitrary starting point. Although the approach is powerful and complex, it can yield many models [18].

## 2.4. Process mining problems

In addition to many techniques applied in process mining, it also contains difficulties [19],[20]. In practice, the most crucial challenge of process mining is the preparation and processing of data. There are difficulties with event data such as non-process-oriented, incomplete, scattered, noisy, and mismatched timestamps. Concept shift concerning processes is another challenge [21]. Concept shift means that the process may change during the analysis. Data may vary due to changing business conditions. Choosing the appropriate data range based on changing business conditions can simplify the solution. Therefore, event data also changes due to changes in process models or information systems. For example, the content mismatch between old and new data will create a concept shift. Four different conceptual shifts are introduced as sudden, gradual, repetitive and incremental [22]. The different levels of detail of event logs is another challenge that should not be ignored [44].

Since process mining is an emerging discipline, various problems and difficulties arise during its application in business processes [10]. As process mining challenges and possible solutions is a topic of interest, the process mining manifest [10] may be needed for further reading.

Data is one of the most valuable assets of the organization's business processes. The data waiting to be evaluated added value in proportion to its impact on the entire functioning of the organization by refining potentially hidden patterns. For example, it allows them to correctly understand the processes operating during mining operations, check compliance and improve them. For organizations, the primary purpose of process mining is to obtain action-oriented information by refining event logs obtained from existing information systems. On the other hand, with the increasing amount of data, there is a need for techniques and methods to process the growing data. Case studies exist to refine the event logs of organizations built on distributed systems [24],[25]. It should be remembered that mining process models through event logs of different structured information systems can be challenging.

### 3. Research design and methodology

The research aims to make the current situation visible regarding process mining in production. It is to compile information about recent studies of software tools, types and perspectives with a focus on process mining within the scope of production. The Scopus and Google Scholar databases, including data sources, books, scientific journals and conference papers, were searched for the purpose. In Figure 3, the stages of the literature review process are given with the evaluation parameters.

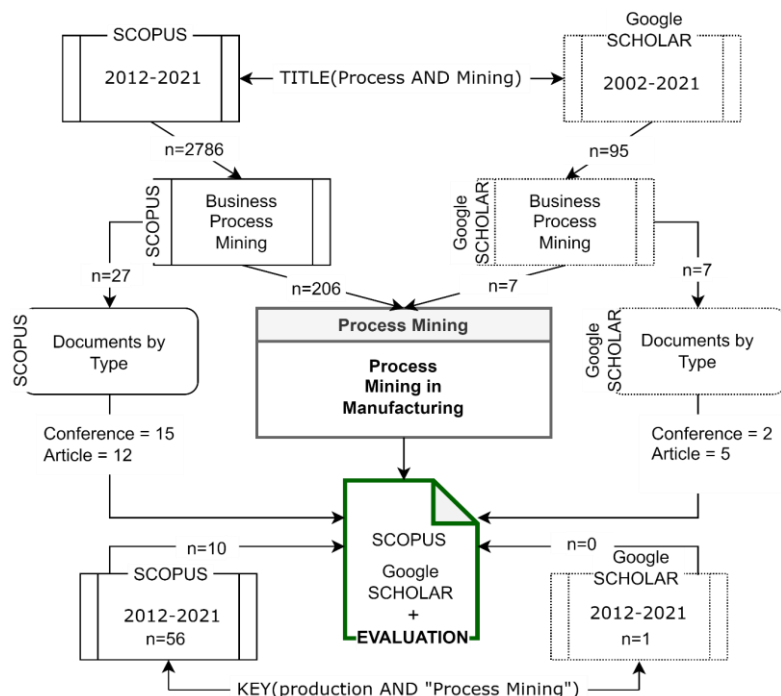


Figure 3 Literature review process

It is seen that the process mining studies carried out in the focus of production have started to attract new attention. So much so that the studies are compiled under business process mining in scientific publication environments. Under the title of business process mining, it is categorically collected in the content of production, management and accounting. While the categories do not give clear boundaries for new work, they define a facilitating framework for research. With the increase in interest and studies, the classes will become apparent as frames defined under a separate title. The research reports the trend and development of production-oriented studies from two different source collection points.

It resulted in 2786 articles initially obtained from the SCOPUS resource. Research criteria start with process mining and narrow down to business process mining. They are reduced from 206 business process mining articles to 27 production content articles. When the production and process mining titles on the same source are searched together in keywords, 56 papers are compiled. Out of the 56 article

intersections, ten articles are obtained as a process mining study in production. Among the articles scanned in our research, no article with the title "production process mining" was found.

Secondly, the number of articles obtained from the Google SCHOLAR resource is 95. The number of essays compiled in the scanning process narrowed by business process mining is 7. Scanned papers are gathered over the title. An article is obtained by searching the keywords "production" and "process mining". It is noteworthy that the related articles, titles, keywords and content are increasing daily. There was no study on a topic with the range of process mining in production and the expression of production process mining. In the literature review, the abstracts of the related articles were analyzed and scanned as in Table 1 and written according to their titles. In Table 1, examples of the articles that are the research subject are given.

Table 1: Sample publications summarizing the work done

Year	Title	Authors	Purpose / Type of source	Keywords:	Conceptual / Theoretical framework	Major themes
2022	Process Mining for Dynamic Modeling of Smart Manufacturing Systems: Data Requirements [26].	J.Friederich etc.	Designing consistent data structures / Research	*Model generation *Discrete event simulation *Process mining *Machine behavior *Reliability models	Developing new models and identifying data requirements by using process mining algorithms	Dynamic modeling for smart manufacturing systems in process mining
2020	Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach [27].	S.Saraeian etc.	Anomaly detection in production / Research	*Event-based anomaly detection *Additive manufacturing *Business process management system *Process mining technique *Game theory modeling *Distributed production system	Intrusion detection through the integration of process mining and game theory techniques.	Process improvement/development in production
2020	An extended model for remaining time prediction in manufacturing systems using process mining [28].	A. Choueiri etc.	Estimating cycle times in production / Research	*Process mining *Remaining time prediction *Multiple linear regression	Remaining time prediction	Process improvement in production
2017	ERP Post Implementation Review with Process Mining: A Case of Procurement Process [29].	M.Er etc.	Workflow process editing and development / Research	*Post Implementation review *Process Mining *Procurement *SAP Materials Management	Purchasing process definition/editing and development	Improvement in the purchasing process
2015	Process Mining Techniques in Conformance Testing of Inventory Processes: An Industrial Application [30].	Z.Paszkiwicz	inventory processes analysis / Research	*Process mining *Business process intelligence *Inventory management *Quality management *Warehouse management system.	Compliance control in inventory processes	Improvement in inventory management processes

Although the studies in Table 1 are process mining studies that focus on production systems, they are mostly keyed as methods or analyses used as categories.

#### 4. Research results and discussion

It is seen that the articles studied in terms of the use of process mining are primarily and mostly control-flow, and secondly, case/time and organizational perspective. It is determined that there are studies for analyzing the expectation process that is prioritized in the field of exciting process mining [3]. Especially in process analysis, decision-makers who want to improve flow and control are representatives of different production sectors. The case/time perspective is studied for the needs of the process, such as analysis of processing, waiting, output times, discovering bottlenecks, anomalies and case frequencies. When evaluated from a higher level, studies have been carried out in the organization's title to support



strategic decisions and obtain an organizational perspective. While many papers apply more than one perspective, no significant model combines perspectives [3].

The article types reviewed in our searches are aligned in order of discovery, development, and relevance. It is seen that researchers care more about process improvement and solutions to problems. Emerging and prominent improvement options are flow and temporal requirements. Mining studies for compliance checking are less common in scanned articles and other sources [3]. Motivation is a need to identify compliance and deviations and make recommendations. They compare the model that emerged/discovered in the work of decision-makers with the reference model, with software tools such as "ProM" and "Disco", respectively. The digitization and evaluation of the appropriate metrics take place through the reports produced by the software tools. Considering the application areas, are listed as "Healthcare", "IT", "Finance", "Manufacturing", and "Education"[3]. The ranking is predictable, considering the rapidly changing digital world's expectations put pressure on industries. Especially in the health sector, it is known that the development, improvement or personalization of the variety of services given to the patient [31], [32], [33], [34], [35], [36], [37],[38], [39], [40] has come to the fore in the recent past. The contribution of the IT industry to other fields is an undeniable fact. In this respect, the software process, positive/negative user activities and service management process continue to attract attention [41],[42],[43]. While the finance sector has attracted interest in every period, analysis [23],[45] has been focused on deposit, customer and loan service points in the recent past. At the same time, studies on security [46] continue to attract attention.

Even when the "Scopus" publication source of process mining is scanned for the last ten years, it is seen that the increasing interest continues (Chart 1). When the graph is examined, it is seen that institutions continue to apply for process mining instruments in a short time, and positive feedback is received. With the active use of software tools developed for solutions that can meet the needs of decision-makers, the application areas have differentiated. The development of the tools and the positive results in the application areas have highlighted the interest in process mining. Research results on "Scopus" and "Google Scholar" contain similar results. Below, the research results are shared via the "Scopus" publication source.

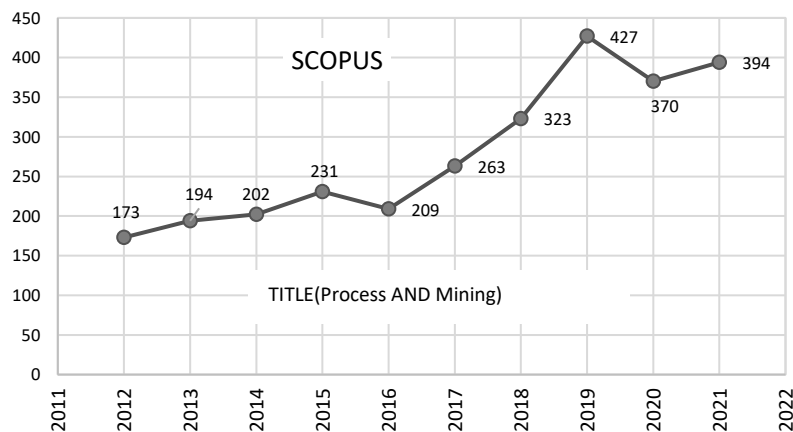


Chart 1 Number of publications with "process mining" in the title

Graph 2 of the application areas of interest in the same source and date range is shared. When the summaries of the studies appearing in Graph 2 are examined, it is seen that they are defined under different titles instead of the production title. Remarkably, "Computer Science" is continuing the development of algorithms and software tools in this new field of process mining. So much so that the modelling and development of mining techniques in all industries with processes led to publications in the areas of "Engineering", "Mathematics", and "Decision Sciences". Process mining in production is gathered under the title "Business, Management and Accounting".

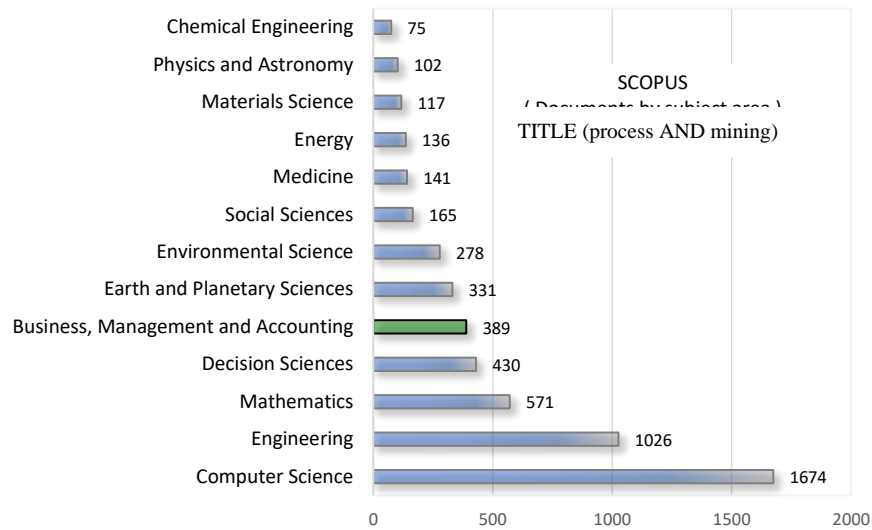


Chart 2 Documents by subject area (in the title)

Process mining solutions for production are shared under the "Business, Management and Accounting" title and the "Business" sub-title. In the "Business" heading, there are solutions for the production process and all solutions for business processes. From this point of view, the variety and number of methods in the production title will pave the way for examining process mining in production under a separate heading.

The number of studies obtained with the keywords "KEY (manufacture OR production) AND KEY (business AND process AND mining)" in the scanned articles is 56. The distribution of article subject areas is given in graph 3.

As can be seen in Graph 1, it is seen that the number of studies on "process mining" in the scanned article titles continues to increase rapidly. When the article contents are examined in Graph 2, scientific publications related to mining research are in the first three places. This output is geared towards developing tools and scientific approaches in the maturing field of "process mining". It determines the sectoral titles and boundaries of the studies following the first three lines. Remarkably, the studies defined by "Decision Sciences" are at the centre of attention of all sectors. So much so that the most effective support used in planning and management issues is sought under "decision sciences". Starting from the fourth row in Graph 2, the number of publications related to sectoral fields is shared. "Business, Management and Accounting" is one of the common sectoral areas. This title covers processes such as business, production, management and cost. As the variety and number of publications in the scope increase, new titles will emerge.

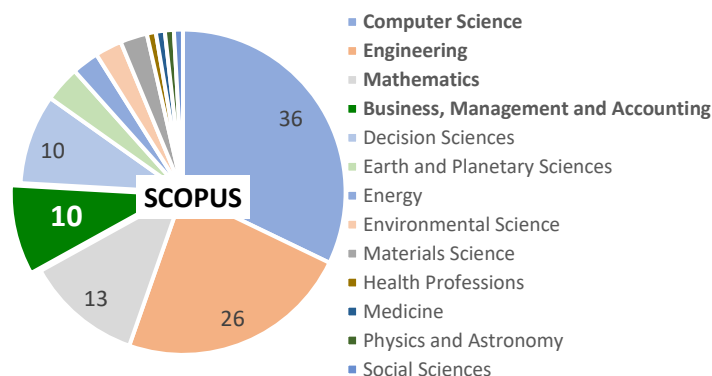


Chart 3 Documents by subject area (in keywords)

"Business, Management and Accounting" corresponds to 17.8% of the articles. Likewise, research on the content of production processes in 10 articles is collected under "Business". Although studies in the context of production are categorically under this title, the number of titles will increase with the increase in acquisition and use. Chart 4 shows the number of publications by keywords that support this situation. It is seen that the number of publications increases according to the keywords that support this situation. When queried with the term "KEY ( manufacture OR production ) AND KEY ( business AND process AND mining )", the number of accessible publications in the last four years is shared in graph.4.

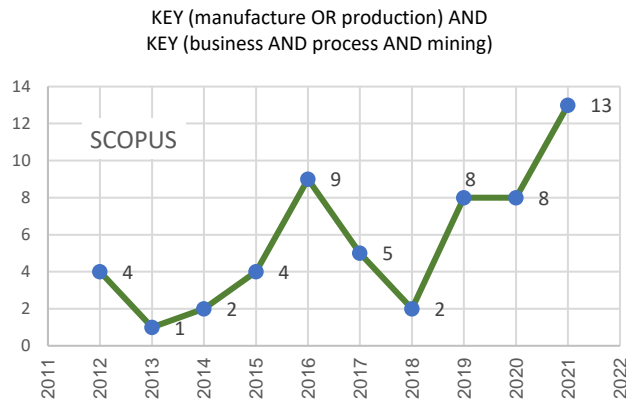
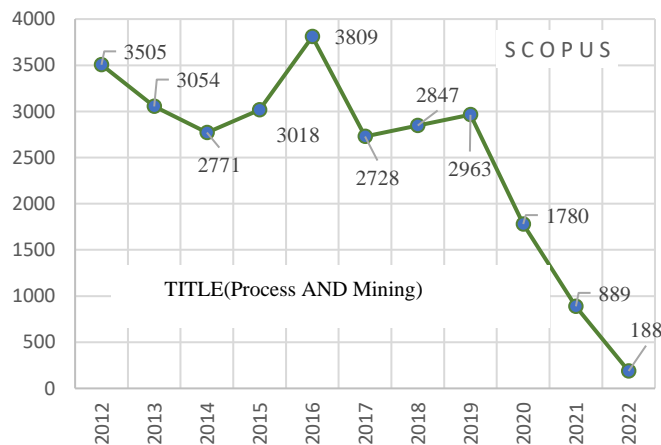


Chart 4 Number of publications by keywords

Graph 5 shows the citation numbers of process mining publications in the context of production for the last ten years. Process mining tends to decline after the 2019 pandemic as it corresponds to the application in the field.



Char 5: The number of citations of publications with the title "Process Mining" between 2012-2022.

Looking at the last four years, research studies defined under "Business" in the context of production continue to increase. Sharing the data of production processes by decision-makers and announcing their applications will attract attention to production processes.

Therefore, after the readings, the possible benefits of use can be listed as follows, focusing on application in production processes today.

1. Measurement and optimization of machine process performance
2. Improve and better understand operations
3. Identifying automation opportunities

4. Reducing operating costs
5. Contribution to production planning
6. Support for resource management

Considering the listed benefits, awareness and interest in the production processes of organizations will increase. The interest turned into practice will open the door to businesses' competitive advantage in global markets.

Knowing what you have in your system is a necessary and sufficient condition for change, transformation and competitive advantage. All resources are brought together with the help of intuitive dashboards and investigative interfaces, optimizing the collected data by analyzing the differences in your systems. Ensuring the synchronization of resources with customer expectations and needs with daily business operations is a challenging task. At this point, traceable process metrics, synchronization, and catching and correcting instant performance deviations are the result of process intelligence. Also, process mining provides a quick solution to understanding, sharing and improving the current situation. On the other hand, the correlation and meaningful association of the data supports the decision points. Decisions based on data and accurate process information empower organizations against risks. Therefore, a process-oriented shift enables organizations in all sectors to compare and position their current situation as they should in the future. Research conducted in this framework and solutions in the industries will increase the acceleration of the transformation process. In this context, analysis and solutions in the sectors will contribute to the transformation process. Working in a new discipline and using new research methods has lost the sensitivity of the title of "production". The term "production", especially in the title and keywords, will facilitate access to publications. At the same time, it was evaluated that the publications being easily accessible or found would contribute positively to the search and transformation process of the sectors.

## **5. Conclusion**

Adequate information on performance, optimization, improvement in production processes, detection of downtime/faults and a better understanding of the process are among the indicators for the purpose. How to analyze actionable operational data to achieve the goal. How to provide end-to-end visibility into your system that hosts your crucial process metrics and key performance indicators (KPIs). How to analyze baseline models automatically. How to get the real reasons and compare the variables in the poorly performing processes. These and many more questions are among the questions that the manufacturing sectors urgently need to find answers to. One of the easy ways to reach answers and solutions is to access scientific publications. Our study draws attention to the definitions in titles and keywords in accessing publications. The article draws a general picture in which research and studies carried out within the scope of production are compiled.

In the article, it is given under which headings the publications are distributed in the production field to increase the gains in mining works and production processes. While this distribution includes new methods and subheadings, the term "production" is not included in the top heading. Our study shows the place, status and result of production processes in process mining research areas quantitatively. At the same time, emphasis is placed on using the term "production" in developing mining studies in the context of access. The importance and conceptual framework of process mining are shared in the first stage. The second stage discusses the scope and development of business process mining. To answer our question, it is seen that the publications are classified under identical/similar titles in two different scientific databases. Both database sources show that the publications are classified under "Business, Management and Accounting" instead of "Production". The results obtained share limited examples of scientific publications. Apart from production, there is a need for research in the known titles of different sectors. Research needs more databases and methods for a general description. The results obtained in our study were scanned, evaluated and limited based on abstract, title and keywords. While the publications specific to the scope of process mining in production are compiled under the title of "business", they also include different business process publications. The results of changing the scan

sequence in Charts 3 and 4 show that process mining efforts are increasing in business and production. The works that emerge with a more accurate title definition or keywords will be more visible in the main title of the paper. Making the definitions correct in the classification of the studies, which is the article's main subject, will open the way for easier access to the researchers. As a result, it is evaluated that the remarkable process mining studies will continue to grow in production as the quality and access to mining techniques and scientific approaches increase.

In future studies, it is noteworthy that mining studies and research of the production processes of enterprises with big data infrastructures, whose benefits we clearly show, are a niche area.

Three critical benefits, in particular, attract the attention of relevant employees.

1. Critical workflows and operational processes: User error is minimized by improving continuity and business process performance. Continuity in improvement is ensured by process mining.
2. Adaptive technology: Warning and action are taken before potential bottlenecks occur. The automation potential can be determined, e.g. batch processing.
3. Scalable: It can manage millions of events, tasks, multiple processes and workflows. A digital twin can be created at an appropriate scale.

The automation era in production has brought industries to the brink of extensive data analysis. For this reason, the gaps between organizations' physical and digital structures make it difficult for developing institutions to mine. However, developments continue rapidly in creating dynamic manufacturing businesses capable of objective decision-making based on interconnected facts. Researches and solutions on process mining in production provide practical benefits in the digital twin of organizations, intelligent process management and mining studies.

It will take place among the decision support tools quickly if comprehensive and high-quality process mining research is carried out in production. Interagency mining research is another interesting topic. In addition, process mining over the Internet and refining process and event data in real-time are other research subjects.

## References

- [1] S. Suman ve I. Pogarcic, "Development of ERP and Other Large Business Systems in the Context of New Trends and Technologies," *DAAAM Proceedings*, 1. bs, vol. 1, pp. 0319-0327, B. Katalinic, Ed. DAAAM International Vienna, 2016, doi: 10.2507/27th.daaam.proceedings. 047.
- [2] W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining: a research agenda," *Computers in Industry*, vol. 53, no 3, pp. 231-244, 2004, doi: 10.1016/j.compind.2003.10.001.
- [3] D. Dakic, S. Sladojevic, T. Lolic, and D. Stefanovic, "Process Mining Possibilities and Challenges: A Case Study," *2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, Eyl. pp. 000161-000166, 2019. doi: 10.1109/SISY47553.2019.9111591.
- [4] S. Smirnov, H. A. Reijers, M. Weske, and T. Nugteren, "Business process model abstraction: a definition, catalog, and survey," *Distrib Parallel Databases*, vol. 30, no 1, pp. 63-99, Feb. 2012, doi: 10.1007/s10619-011-7088-5.
- [5] F. W. Breyfogle, "Implementing six sigma: smarter solutions using statistical methods," 2nd ed., ISBN: 978-0471265726, Hoboken, NJ: Wiley, 2003.
- [6] M. Özcan ve S. Peker, "Designing a Data Warehouse for Earthquake Risk Assessment of Buildings: A Case Study for Healthcare Facilities," *Sakarya University Journal of Computer and Information Sciences*, vol. 4, no 1, pp. 156-165, 2021, doi: 10.35377/saucis.04.01.872729.
- [7] B. Kaya, "Analysis of the Association Between Vitamin D Deficiency and Other Diagnoses of Patients by Data Mining Techniques," *Sakarya University Journal of Computer and Information Sciences*, pp. 50-58, 2020, doi: 10.35377/saucis.03.01.677676.
- [8] W. M. P. van der Aalst, M. La Rosa, and F. M. Santoro, "Business Process Management: Don't Forget to Improve the Process," *Bus Inf Syst Eng*, c. 58, sy 1, ss. 1-6, Feb. 2016, doi: 10.1007/s12599-015-0409-x.


- [9] W. M. P. van der Aalst, "Business Process Management: A Comprehensive Survey," *ISRN Software Engineering*, vol. 2013, pp. 1-37, 2013, doi: 10.1155/2013/507984
- [10] F. Daniel, K. Barkaoui, and S. Dustdar, Ed., "Business process management workshops: BPM 2011 International Workshops," *Clermont-Ferrand, France*, vol 99, pp. 169-194, August 29, 2011, *Revised selected papers. Part I*. Berlin; New York: Springer, 2012.
- [11] W. M. Van der Aalst, M. La Rosa, and F. M. Santoro, "Business process management," *Business & Information Systems Engineering*, vol. 58, pp. 1-6, 2016.
- [12] D. Duplakova, M. Teliskova, J. Duplák, J. Torok, M. Hatala, J. Steranka, and S. Radchenko, "Determination of Optimal Production Process Using Scheduling and Simulation Software," *Int. j. simul. model.*, vol.17, no 4, pp. 609-622, 2018, doi: 10.2507/IJSIMM17(4)447..
- [13] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters, "Workflow mining: A survey of issues and approaches," *Data & Knowledge Engineering*, vol. 47, no 2, pp. 237-267, Kas. 2003, doi: 10.1016/S0169-023X(03)00066-1.
- [14] ProcessMining, "ProcessMining homepage", 2022. [Online]. Available: <http://www.processmining.org/process-discovery.html>. [Accessed: 04-June-2022].
- [15] W. Van der Aalst, "Process Mining Manifesto," *Conference: Proc. of Business Process Management Workshops*, Berlin, Heidelberg, 2012.
- [16] W. M. P. van der Aalst, "Process Mining: Discovery, Conformance and Enhancement of Business Processes," Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-19345-3.
- [17] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no 1, pp. 64-95, 2008, doi: 10.1016/j.is.2007.07.001.
- [18] M. Er, N. Arsad, H. M. Astuti, R. P. Kusumawardani, and R. A. Utami, "Analysis of production planning in a global manufacturing company with process mining," *JEIM*, vol. 31, no 2, pp. 317-337, 2018, doi: 10.1108/JEIM-01-2017-0003.
- [19] F. Daniel, K. Barkaoui, and S. Dustdar, "Business process management workshops," *BPM 2011 International Workshops, Clermont-Ferrand, Revised selected papers. Part I*. Berlin; New York: Springer, pp. 169-194, France, August 29, 2011.
- [20] W.M. Van Der Aalst, "Process Mining: Data science in action," *2nded., Springer, Berlin, Heidelberg*, pp.3-23, 2016.
- [21] R. P. J. C. BOSE, W. M. P. Van DerAlst, I. Z. Liobaite, and P. Echenizkiy, "Dealing with concept drift in process mining," *IEEE Trans. on Neur. Net. and Lear. Syst.*, 2013.
- [22] V. Mittal and I. Kashyap, "Online Methods of Learning in Occurrence of Concept Drift," *IJCA*, vol. 117, no 13, pp. 18-22, 2015, doi: 10.5120/20614-3280.
- [23] Märuşter and N. R. T. P. van Beest, "Redesigning business processes: a methodology based on simulation and process mining techniques," *Knowl Inf Syst*, vol. 21, no 3, pp. 267-297, 2009, doi: 10.1007/s10115-009-0224-0.
- [24] P. Zellner, M. Laumann, and W. Appelfeller, "Towards Managing Business Process Variants within Organizations - An Action Research Study," *2015 48th Hawaii International Conference on System Sciences*, HI, USA, pp. 4130-4139, 2015. doi: 10.1109/HICSS.2015.495.
- [25] Leemans and W. M. P. van der Aalst, "Process mining in software systems: Discovering real-life business transactions and process models from distributed systems," *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Ottawa, ON, Canada, 2015, pp. 44-53. doi: 10.1109/MODELS.2015.7338234.
- [26] Friederich, J., Lugaresi, G., Lazarova-Molnar, S., & Matta, A. "Process mining for dynamic modeling of smart manufacturing systems: Data requirements." *Procedia CIRP*, 107, 546-551. 2022. doi.org/10.1016/j.procir.2022.05.023
- [27] Saraeian, S., & Shirazi, B. "Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach." *Computers & Industrial Engineering*, 146, 106584. 2020. doi.org/10.1016/j.cie.2020.106584
- [28] Choueiri, A. C., Sato, D. M. V., Scalabrin, E. E., & Santos, E. A. P. "An extended model for remaining time prediction in manufacturing systems using process mining." *Journal of Manufacturing Systems*, 56, 188-201. 2020. doi.org/10.1016/j.jmsy.2020.06.003

- [29] Mahendrawathi, E. R., Zayin, S. O., & Pamungkas, F. J. Erp post implementation review with process mining: A case of procurement process. *Procedia Computer Science*, 124, 216-223, 2017. doi.org/10.1016/j.procs.2017.12.149
- [30] Paszkiewicz, Z. "Process mining techniques in conformance testing of inventory processes: An industrial application." *Springer Berlin Heidelberg*, c.160, ss.302-313, 2013. doi.org/10.1007/978-3-642-41687-3\_28
- [31] A. Rozinat, I. S. M. de Jong, C. W. Gunther, and W. M. P. van der Aalst, "Process Mining Applied to the Test Process of Wafer Scanners in ASML," *IEEE Trans. Syst., Man, Cybern. C*, vol. 39, no 4, pp. 474-479, 2009, doi: 10.1109/TSMCC.2009.2014169.
- [32] A. P. Kurniati and I. Atastina, "Implementing process mining to improve COBIT 5 assessment program or managing operations," *Journal of Theoretical and Applied Information Technology*, vol. 72, no. 2, pp. 191–198, 2015.
- [33] C. Huang, H. Cai, Y. Li, J. Du, F. Bu, and L. Jiang, "A Process Mining Based Service Composition Approach for Mobile Information Systems," *Mobile Information Systems*, vol. 2017, pp. 1-13, 2017, doi: 10.1155/2017/3254908.
- [34] J. Samalikova, R. J. Kusters, J. J. M. Trienekens, and A. J. M. M. Weijters, "Process mining support for Capability Maturity Model Integration-based software process assessment, in principle and in practice: PROCESS MINING SUPPORT FOR CMMI-BASED SOFTWARE PROCESS ASSESSMENT," *J. Softw. Evol. and Proc.*, vol. 26, no 7, pp. 714-728, 2014, doi: 10.1002/smr.1645.
- [35] J. De Weerd, A. Schupp, A. Vanderloock, and B. Baesens, "Process Mining for the multi-faceted analysis of business processes—A case study in a financial services organization," *Computers in Industry*, vol. 64, no 1, pp. 57-67, 2013, doi: 10.1016/j.compind.2012.09.010.
- [36] E. Kim, S. Kim, M. Song, S. Kim, D. Yoo, H. Hwang, and S. Yoo, "Discovery of Outpatient Care Process of a Tertiary University Hospital Using Process Mining," *Healthc Inform Res*, vol. 19, no 1, pp. 42, 2013, doi: 10.4258/hir.2013.19.1.42.
- [37] W. M. P. van der Aalst, S. Guo, and P. Gorissen, "Comparative Process Mining in Education: An Approach Based on Process Cubes," *Data-Driven Process Discovery and Analysis*, P. Ceravolo, R. Accorsi, and P. Cudre-Mauroux, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 203, pp. 110-134, 2015. doi: 10.1007/978-3-662-46436-6\_6.
- [38] D. Antonelli and G. Bruno, "Application of Process Mining and Semantic Structuring Towards a Lean Healthcare Network," *Risks and Resilience of Collaborative Networks*, L. M. Camarinha-Matos, F. Bénaben, ve W. Picard, Ed. Cham: Springer International Publishing, vol. 463, pp. 497-508, 2015, doi: 10.1007/978-3-319-24141-8\_46.
- [39] V. Aisa, A. P. Kurniati, and A. W. Yanuar Firdaus, "Evaluation of the online assessment test using process mining (Case Study: Intensive English Center)," *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, Nusa Dua, Bali, Indonesia, 2015, pp. 472-477. doi: 10.1109/ICoICT.2015.7231470.
- [40] B. Vázquez-Barreiros, D. Chapela, M. Mucientes, M. Lama and D. Barea, "Process mining in IT service management: A case study," *In CEUR Workshop Proceedings*, vol. 1592, pp. 16–30, 2016.
- [41] B. T. Greyling and W. Jooste, "The application of business process mining to improving a physical asset management process: A case study", *SAJIE*, vol. 28, no 2, 2017, doi: 10.7166/28-2-1691.
- [42] R. Pérez-Castillo, B. Weber, I. G.-R. de Guzmán, and M. Piattini, "Process mining through dynamic analysis for modernising legacy systems," *IET Softw.*, vol. 5, no 3, pp. 304, 2011, doi: 10.1049/iet-sen.2010.0103.
- [43] G. Sedrakyan, J. De Weerd, and M. Snoeck, "Process-mining enabled feedback: 'Tell me what I did wrong' vs. 'tell me how to do it right,'" *Computers in Human Behavior*, vol. 57, pp. 352-376, 2016, doi: 10.1016/j.chb.2015.12.040.
- [44] R. S. Mans, W. M. P. van der Aalst, and R. J. B. Vanwersch, "Process Mining in Healthcare: Evaluating and Exploiting Operational Healthcare Processes," *Cham: Springer International Publishing*, pp. 1-91. 2015. doi: 10.1007/978-3-319-16071-9.

- [45] M. Bozkaya, J. Gabriels, and J. M. van der Werf, "Process Diagnostics: A Method Based on Process Mining" *2009 International Conference on Information, Process, and Knowledge Management*, Cancun, Şub. 2009, pp. 22-27. doi: 10.1109/eKNOW.2009.29.
- [46] M. Sahlabadi, R. C. Muniyandi and Z. Sukur. "Detecting abnormal behavior in social network websites by using a process mining technique," *Journal of Computer Science*, vol.10, no 3, pp.393–402, ISSN: 1549-36362014, doi: 10.3844/jcssp.2014.393.402.



# Software Development for the Use of Generalized Parabolic Blending in Data Prediction Processes

 Hakan Üstünel<sup>1</sup>

<sup>1</sup> Corresponding Author; Department of Software Engineering, Faculty of Engineering, Kırklareli University, Kırklareli, Türkiye; hakanustunel@hotmail.com; hakanustunel@klu.edu.tr

Received 27 May 2022; Revised 7 June 2022; Accepted 18 November 2022; Published online 31 December 2022

## Abstract

Parabolic blending (PB) is one of the important topics in applied mathematics and computer graphics. The use of generalized parabolic blending (GPB) for different scenarios adds flexibility to the polynomial. Overhauser (OVR) elements is a special case in GPB ( $r=0.5$ ,  $s=0.5$ ). GPB can also be used in estimation. In this study, data obtained from thickness distribution of a 3mm thick high impact polystyrene product after thermoforming using a mold was used for data estimation. For this purpose, software has been developed. The software development steps and formula usages are explained. Using the developed software, polynomials for GPB and default PB (OVR) were created. The data set was compared with the  $y$  values produced by the polynomials for certain  $x$  values. At the end of the research, it was determined that the results obtained from the GPB were 0.1728 percent more accurate than the data obtained from the PB for the default values.

**Keywords:** Generalized parabolic blending, computer graphics, visualization, software development, thickness distribution

## 1. Introduction

Curves have been an important subject of mathematics and applied sciences for hundreds of years. It has been possible for curves to be applied in many different fields and to be defined in different ways in this old subject. In its general definition, it is possible to define a curve with at least a quadratic function.

A curve can also be created by defining a function for the curve using existing vertices (control points, vectors). The basic approach in this matter can be shown by defining the smallest degree polynomial passing through all control points, as in the Lagrangian interpolation polynomial or the Newton divided difference polynomial. Parabolic curves also find a place in the areas of practical solutions of theoretical calculations such as multi-segment trajectory tracking of the redundant space robot for smooth motion [1], free vibration analysis of a variable stiffness composite laminate square plate with circular cutout [2] and describing the viscoelastic behavior of a shape memory polymer blend [3].

Curves can also be defined using the parabolic blending (PB) approach by following a different path than curve interpolation. PB, which is preferred especially in places where smooth transition should occur, is an important subject of computer graphics. Overhauser [4] created the primary definitions of PB. In later studies, PB was also referred to as Overhauser elements, Overhauser's parabolic blending interpolation, or Overhauser's curve [5-7]. Brewer & Anderson [8] presented the applications of PB in computer graphics. Schneider [9] has worked on adding tension to the formula. PB can be applied in boundary conditions [10-14] as Boundary Element Method (BEM). PB has also been an important field of study in the estimation process [18].

Having four control vertices/points/vectors in PB is sufficient to define the curve. The main difference between PB and the interpolation polynomial generation process is that the weights of the control vertices on the curve are calculated in different ways. When the interpolation polynomial is constructed for the curve passing through a multi-element series of vertices, the polynomial will have a high degree. This will cause cumbersome operations, especially for programs with multiple repetitions.

Hadavinia et al. [17] named their work as general parabolic blending (GPB) elements by formulating the variation of the parameters used to give the general form of the PB and corresponding to the intermediate vertices. Thus, Overhauser (OVR) elements are shown as a special case of GPB. They claimed that the results obtained with the GPB elements would be more accurate than the OVR, and they formulated the parameter variation for the intermediate vertices in its general form. They supported their claims with the results they obtained from the sample functions they had used in their studies.

In this study, the notation that Hadavinia et al. [17] call GPB is explained with examples for different parameters. Afterwards, multi-repetitive software was developed to be used in the estimation process to be run on a real dataset. The results obtained are visualized to facilitate comparison.

## 2. Parabolic Blending for Changing Parameters

The process of creating a cubic between two parabolas is called PB [17,18] (also called OVR elements). Three vertices are selected from four control vertices ( $P_1$ - $P_4$ ) to form two parabolas (Figure 1).  $P_1$ - $P_3$  defines the  $P(r)$  parabola, while  $P_2$ - $P_4$  defines the  $Q(s)$  parabola [4,18].

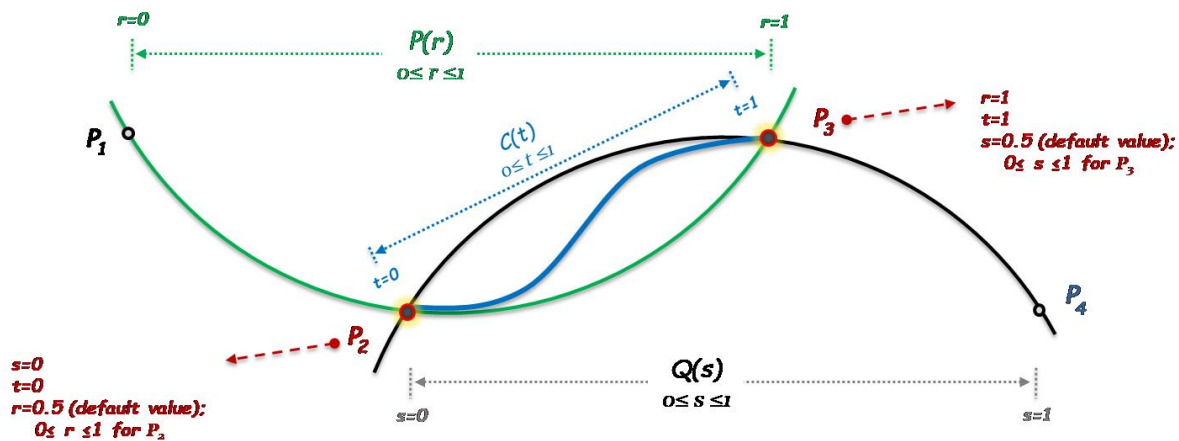


Figure 1 Parabolas, parameters and control vertices that define parabolic blending

With the effect of the weights of these two parabolas,  $C(t)$  is given its general form. The effect of  $P(r)$  and  $Q(s)$  on  $C(t)$  is linear (line equation) (Formula 1).

$$C(t) = (1 - t)P(r) + t * Q(s) \tag{1}$$

In Formula 1, the weight of the parabolas on  $C(t)$  changes depending on the value of  $t$  ( $0 \leq t \leq 1$ ). For the value of  $t$  0.5, the weights of the parabolas on  $C(t)$  are equal. If the  $t$  value is less than 0.5, the effect of  $P(r)$  on the curve is greater than that of  $Q(s)$ . If the  $t$  value is greater than 0.5, the effect of  $P(r)$  on the curve is less than that of  $Q(s)$ . Depending on the  $t$  value, any vertices on  $C(t)$  ( $x,y$ ) can also be reached depending on the control vertices. Equations of parabolas  $P(r)$  and  $Q(s)$  must be formed.

If the general  $P(r)$  equation is created depending on the  $r$  values, Formula 2 is obtained ( $0 \leq r \leq 1$ ).  $[B]$  is the coefficient matrix to be calculated based on the control vertices where the  $r$  parameter is equalized.

$$P(r) = [r^2 \quad r \quad 1][B] \tag{2}$$

At this stage, the  $r$  value is used to give  $P(r)$  its general form. If the  $r$  value is 0, the  $P_1$  is obtained, and if the  $r$  value is 1, the  $P_3$  is obtained. Any  $r$  value between these two vertices is equated with the  $P_2$  to form the parabola equation. The value of  $r$  for  $P_2$  will give  $P(r)$  its general form. The default  $r$  value for  $P_2$  is 0.5 in parabolic blending.

$$\begin{aligned} r = 0 & \Rightarrow P(0) = P_1; & P_1 &= [0^2 \quad 0 \quad 1][B] \\ r = 0.5 & \Rightarrow P(0.5) = P_2; & P_2 &= [0.5^2 \quad 0.5 \quad 1][B] \\ r = 1 & \Rightarrow P(1) = P_3; & P_3 &= [1^2 \quad 1 \quad 1][B] \end{aligned}$$

Formula 3 occurs when the accepted  $r$  values for control vertices are combined to the side. If the  $r$  value was 0.3 for  $P_2$ , the row with index 1 of the matrix would be [0.32 0.3 1].

$$\begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0.25 & 0.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} [B] \quad (3)$$

If the matrix that creates the  $r$  values for the selected control vertices is called  $[Mp]$  and then  $[B]$  is left alone, Formula 4 is obtained.

$$\begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} = [Mp][B]; \quad [B] = [Mp]^{-1} \begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} \quad (4)$$

If Formula 4 is substituted in Formula 2, Formula 5 is obtained.

$$P(r) = [r^2 \quad r \quad 1][Mp]^{-1} \begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} \quad (5)$$

As can be seen, depending on the  $r$  value at the  $P_2$ ,  $[Mp]$  and therefore  $[Mp]^{-1}$  will change, so the  $P(r)$  equation will also change (Formula 5). This change will also affect the general form of  $C(t)$  (Formula 1).

While creating the  $P(r)$  equation, it has been accepted that the  $P_2$  will be reached if the  $r$  value is 0.5 (default state for PB). Until this stage, the equation will change, since  $[Mp]$  values will change for different values of the  $r$  value. For example, if  $P_2$  corresponds to  $r=0.2$  or  $r=0.7$  value instead of 0.5 value of  $r$ ;

$$[Mp] = \begin{bmatrix} 0 & 0 & 1 \\ 0.04 & 0.2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad [Mp] = \begin{bmatrix} 0 & 0 & 1 \\ 0.49 & 0.7 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Formula 6 is obtained when the operations performed in  $P(r)$  are repeated for the vertices  $P_{2-4}$  to calculate the  $Q(s)$  equation. While creating the  $Q(s)$  equation, it has been accepted that  $P_3$  will be reached if the  $s$  value is 0.5 (default state for PB). Similar to  $P(r)$ , it matters for the general form of  $Q(s)$  at which  $P_3$  is reached, versus which value of  $s$  ( $0 \leq s \leq 1$ ).

$$Q(s) = [s^2 \quad s \quad 1][Mq]^{-1} \begin{bmatrix} P2 \\ P3 \\ P4 \end{bmatrix} \quad (6)$$

If Formula 5 and Formula 6 are substituted in Formula 1, Formula 7 is obtained.

$$C(t) = (1 - t) * [r^2 \quad r \quad 1][Mp]^{-1} \begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} + t * [s^2 \quad s \quad 1][Mq]^{-1} \begin{bmatrix} P2 \\ P3 \\ P4 \end{bmatrix} \quad (7)$$

After Formula 7 is created, the value of  $r$  and  $s$  must be calculated for any value of  $t$  (for vertices where  $P(r)$  and  $Q(s)$  intersect with  $C(t)$ ;  $P_2, P_3$ ). The variation of  $r$  with respect to  $t$  is their line equation (Formula 8).

$$r = k1 * t + k2 \quad (8)$$

In Formula 5, the  $k_1$  and  $k_2$  values must be calculated. If  $P_2$  is reachable for the default value (0.5) of  $r$ , Formula 9 is obtained.

$$\begin{aligned}
 P2: r = 0.5; \quad t = 0 \Rightarrow r &= k_1 * t + k_2 & 0.5 &= k_1 * 0 + k_2 \Rightarrow k_2 = 0.5 \\
 P3: r = 1; \quad t = 1 \Rightarrow r &= k_1 * t + k_2 & 1 &= k_1 * 1 + 0.5 \Rightarrow k_1 = 0.5 \\
 & & r(t) &= 0.5 * t + 0.5
 \end{aligned} \tag{9}$$

If  $P_2$  is reachable for the 0.3 of  $r$ , Formula 10 is obtained.

$$\begin{aligned}
 P2: r = 0.3; \quad t = 0 \Rightarrow r &= k_1 * t + k_2 & 0.3 &= k_1 * 0 + k_2 \Rightarrow k_2 = 0.3 \\
 P3: r = 1; \quad t = 1 \Rightarrow r &= k_1 * t + k_2 & 1 &= k_1 * 1 + 0.3 \Rightarrow k_1 = 0.7 \\
 & & r(t) &= 0.7 * t + 0.3
 \end{aligned} \tag{10}$$

For different values of  $r$ , Formula 8 will change. If a pattern is extracted using Formula 8, Formula 9 and Formula 10, the equation  $k_1=1-r$  and  $k_2=r$  can be formed. In this way, Formula 11 is obtained when Formula 5 is generalized.

$$r(t) = (1 - r) * t + r \tag{11}$$

With  $r(t)$  in Formula 11, an instant  $r$  value depending on the  $t$  value is obtained. It is the position of  $P_2$  that gives its general form to  $P(r)$ , with the  $r$  parameter on the right side of the equation. In order not to confuse the terms in the next stages,  $r_{instant(t)}$  will be used instead of  $r(t)$  and after the equality is solved, the value obtained is called  $r_{instant}$  and the formula is rearranged (Formula 12).

$$r_{Instant}(t) = (1 - r) * t + r \tag{12}$$

As with the  $t$ -dependent change in the  $r$  value, the  $t$ -dependent change in the  $s$  value is also a line equation (Formula 13).

$$s = k_3 * t + k_4 \tag{13}$$

In Formula 13, the  $k_3$  and  $k_4$  values must be calculated. If  $P_2$  is reachable for the default value (0.5) of  $s$ , Formula 14 is obtained.

$$\begin{aligned}
 P2: s = 0; \quad t = 0 \Rightarrow s &= k_3 * t + k_4; & 0 &= k_3 * 0 + k_4 \Rightarrow k_4 = 0 \\
 P3: s = 0.5; \quad t = 1 \Rightarrow s &= k_3 * t + k_4 & 0.5 &= k_3 * 1 + 0 \Rightarrow k_3 = 0.5 \\
 & & s(t) &= 0.5 * t
 \end{aligned} \tag{14}$$

If  $P_2$  is reachable for the 0.1 of  $s$ , Formula 15 is obtained.

$$\begin{aligned}
 P2: s = 0; \quad t = 0 \Rightarrow s &= k_3 * t + k_4; & 0 &= k_3 * 0 + k_4 \Rightarrow k_4 = 0 \\
 P3: s = 0.1; \quad t = 1 \Rightarrow s &= k_3 * t + k_4 & 0.1 &= k_3 * 1 + 0 \Rightarrow k_3 = 0.1 \\
 & & s(t) &= 0.1 * t
 \end{aligned} \tag{15}$$

For different values of  $s$  the Formula 13 will change. If a pattern is extracted using Formula 13, Formula 14 and Formula 15, the equation  $k_3=r$  and  $k_4=0$  can be formed. In this way, Formula 16 is obtained when Formula 13 is generalized.

$$s(t) = s * t \tag{16}$$

With  $s(t)$  in Formula 16, an instant  $s$  value depending on the  $t$  value is obtained. It is the position of  $P_2$  that gives its general form to  $Q(s)$  with the  $s$  parameter on the right side of the equation. In order not to confuse the terms in the next stages,  $s_{instant(t)}$  will be used instead of  $s(t)$  and after the equality is solved, the value obtained is called  $s_{instant}$  and the formula is rearranged (Formula 17).

$$s_{Instant}(t) = s * t \tag{17}$$

As a result, it is first necessary to determine which  $r$  and  $s$  values will be used to reach the vertices  $P_2$  and  $P_3$ , respectively. Since these values will change the values of the matrices in the formula for  $P(r)$

and  $Q(s)$  (Formula 5, Formula 6), the weights of the parabolas on  $C(t)$  will also change. In the next step, the  $r$  and  $s$  values for a given  $t$  value will be calculated using generalized formulas (Formula 12, Formula 17), and should be written into the general formula (Formula 7).

### 3. Methodology

In this section, the dataset and the pattern used in the research and the purpose of the research are mentioned. The dataset of this study was taken from a previous study, in which the author of this study was one of the researchers [18]. This dataset was created as a result of measuring the thickness distribution of a 3mm thick high impact polystyrene product after thermoforming using a mold (Column 2 in Table 2). In a study by Ekşi and Üstünel [16], PB was used to estimate the thickness distribution of the thermoformed plate. With the coded program, correct estimation was performed with a 4.3866 percent error value. In the same study, while determining the PB steps,  $P_2$  and  $P_3$  vertices were accessed for the 0.5 value of the  $r$  and  $s$  parameters; that is, the default values were used. In this research, while determining the PB steps, instead of a fixed value such as 0.5 for the  $r$  and  $s$  parameters, access to the  $P_2$  and  $P_3$  vertices was provided for the values in a varying range depending on a certain precision value. This process can be expressed as optimizing the PB process depending on the  $r$  and  $s$  values that change independently of the  $t$  value. In order to compare the results obtained in the optimization of the PB with the results of the previous research [16], the same dataset and pattern were used in this study.

The aim of this research is to ensure that more accurate results can be produced in the estimation processes by optimizing PB. For this purpose, software has been developed, and its steps are explained. Afterwards, the results of the error calculations performed using the software were examined. In  $C(t)$ , the value of  $y$  is calculated based on the calculated  $x$  value for any value of  $t$ . Then, the error value is calculated by comparing this data with the dataset ([16], Table 3 Column 2-4) regarding the thickness distribution of the thermoformed material. In this study, after this step, the case where the  $r$  and  $s$  parameters are 0.5 will be called the default PB (instead of OVR elements). In addition, the situation in which  $C(t)$  also changes with each change of the  $r$  and  $s$  parameters will be called GPB, depending on the nomenclature of Hadavinia et al. [17].

The flow chart of the developed program is shown in Figure 2. Under this heading, flowchart sections are explained. In Section 1, variable definitions and value assignments, especially precision, are carried out. The program has been implemented as a Visual Studio 2019 C# form application. The results obtained from the calculations were transferred to an spreadsheetfile by the program. The output file (spreadsheet file) is processed in MatLAB® in the next step, and the results are finally visualized.

The developed software mainly consists of the following parts. Assigning the initial values (section 1), selecting the precision values to be used in the calculations (section 2), performing the assignments for the  $P(r)$  values (Section 3), performing the assignments for the  $Q(s)$  values (Section 4), for the  $C(t)$  values performing assignments and generating error values and saving them to the spreadsheet file (Section 5).

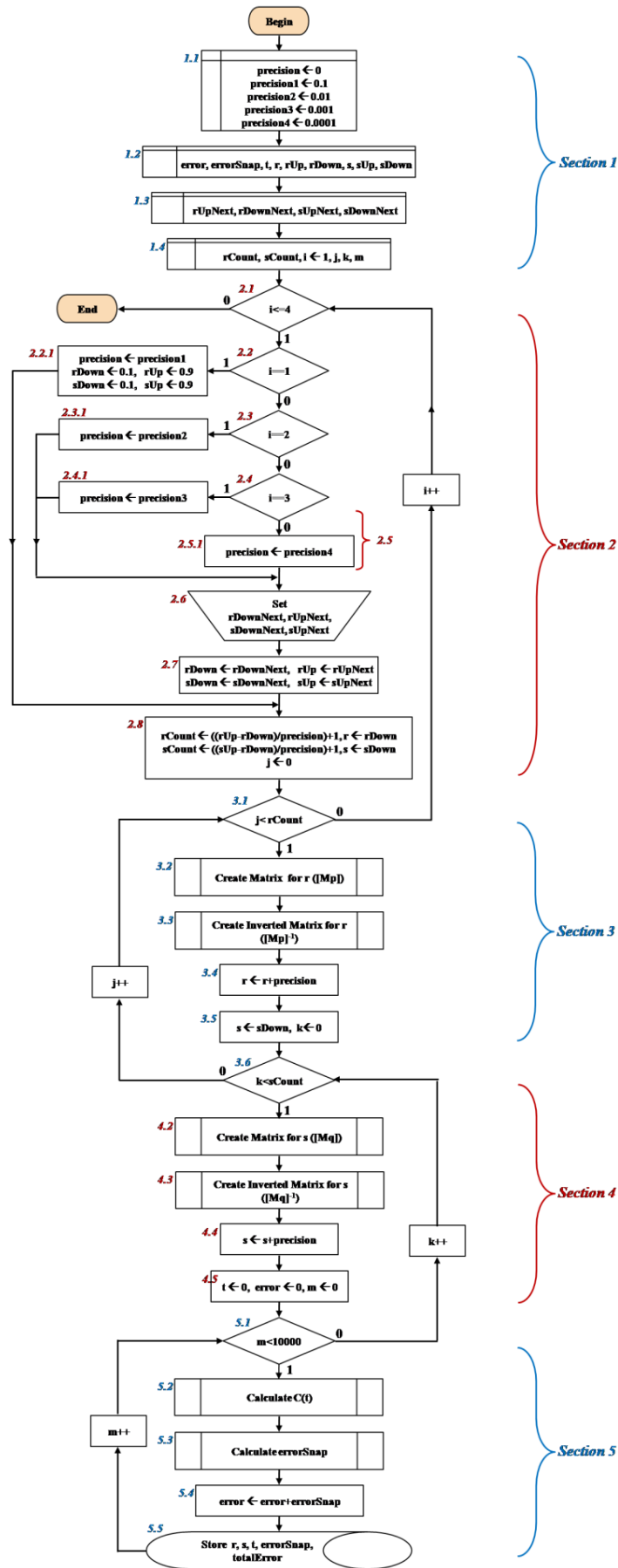


Figure 2 General flow diagram of the software

Assignment operations required for four different precision (i) values are performed in Section 2. For example, if the value of i is equal to 1 (Section 2.2.1), the precision value will be 0.1. A range for r and s will be defined, and C(t) will be calculated on this range. The closed interval [0.2, 0.9] is defined for r and s. A value of 0.1 (precision1) is determined as r/sDown and a value of 0.9 as r/sUp. For r and s values, the number of iterations to be made in the outer two-dimensional loop must be specified (Section 2.8).

After the first cycle of the loop, the results were filtered and examined. In case the i value is 2, 3 and 4 (Section 2.2 – Section 2.4), the necessary value assignment (r/sDown,r/sUp) operations were performed manually (Section 2.6 and Section 2.7). If i is equal to 3 (Section 2.4), precision value 0.001 (Section 2.4.1) will be selected.

For example, if the interval values for r and s were chosen as [0.45, 0.55], [0.32, 0.51], respectively, the calculated values for precision=0.01 (i=2) would be as shown below.

$$rCount \leftarrow \left( \frac{rUp - rDown}{precision} \right) + 1$$

$$rCount \leftarrow \left( \frac{0.55 - 0.45}{0.01} \right) + 1$$

$$rCount \leftarrow 11$$

$$sCount \leftarrow \left( \frac{sUp - sDown}{precision} \right) + 1$$

$$sCount \leftarrow \left( \frac{0.51 - 0.32}{0.01} \right) + 1$$

$$sCount \leftarrow 20$$

Thus, calculations will be made for rCount\*sCount (11\*20) with 220 different C(t) (Formula 7). P(r) will be in the row of the two-dimensional loop, where C(t) will be calculated for different r and s values (Section 3). P(r) will be in the row of the two-dimensional loop where C(t) will be calculated for different r and s values. For a certain value of r, the matrix [Mp] (Section 2.2) and the inverse matrix [Mp]<sup>-1</sup> (Section 2.3) must be calculated. Then, the precision value, which has been assigned a value based on the value of i, is assigned to r. At each step of the loop, the s value is determined as the lower limit (sDown) of the range.

The operations for r in Section 3 are repeated for s in Section 4. At the end of Section 4, the error and t values are also reset.

After generating C(t) for a given r and s values, rinstant and sinstant values are calculated based on the t value. These three parameters are written into their places in Formula 7. This process is repeated twice for the x and y values of C(t). The x values of the control vertices are used in the calculation for x, and the y values of the control vertices are used for y. Thus, x and y values are produced for any value of t.

For a given r-s combination, a C(t) is generated. In this study, C(t) is calculated by choosing the sensitivity value 10<sup>-5</sup> for the sequential increase of the t value. For any value of t, when x parameters of control vertices are used in Formula 7, x value for C(t) is produced. For the same t value, when the y values of the control vertices are used in Formula 7, the y value for c(t) is produced. Thus, (x,y) values are produced for a certain t value. When this x value produced by the program matches the x value in the dataset, the relative difference between the y value in the dataset and the y value produced by the program creates the error value. The resulting error value is calculated by averaging these error values (16 vertices in the dataset).

In the reference study, this error value was calculated as 4.386 for the default r-s values used for PB [16]. In this study, the aim is to obtain a lower error rate by changing C(t) and by changing the r-s values of PB.

#### 4. Experimental Study and Results

A program was written using C# programming language to generate different C(t) equations using different r\*s combinations. The results produced by the program were compared with the reference dataset (Column 2 in Table 2). The program exports these comparison results to the spreadsheet files. As a result of filtering the data in the exported file separately for each different precision value, the interval value for r and s in the next step (precision) is determined. An example of the results produced by the program for different r and s parameters is shown in Figure 3. The blue colored curve (“Real”) is the reference dataset used in the study.

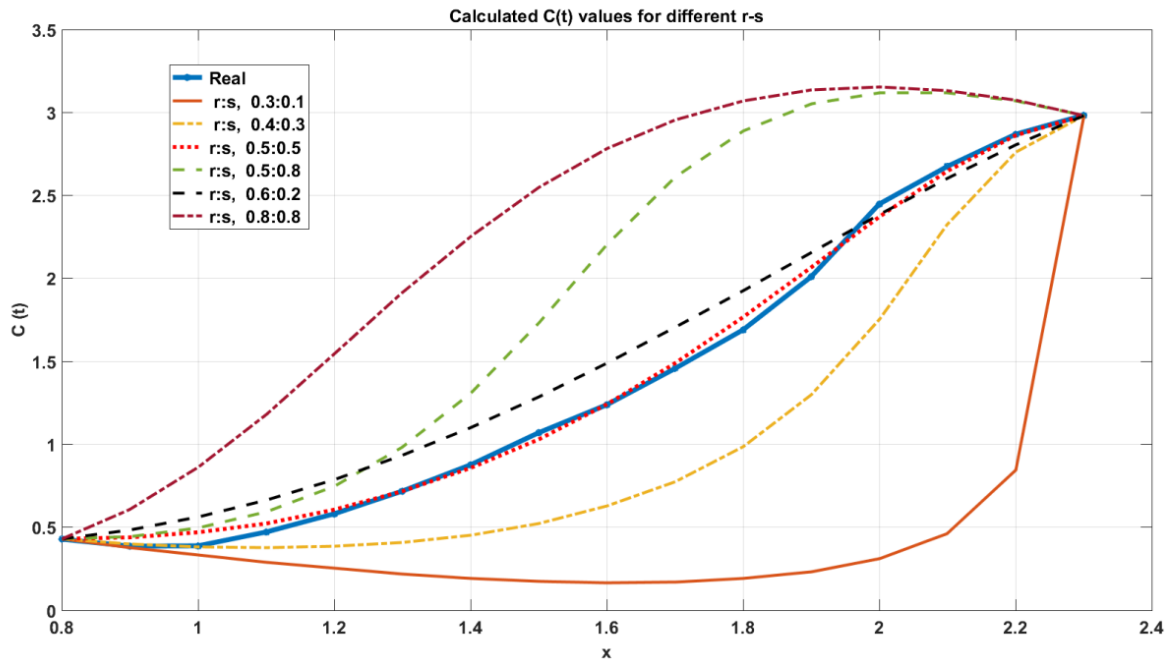


Figure 3 Calculated c(t) values for different r-s → P<sub>2</sub>-P<sub>3</sub>; precision ← 0.1

The r and s interval values for precision=0.1 were determined as [0.2, 0.9] and [0.1, 0.8], respectively. Therefore, the r value is calculated as 8 and the s value as 8 (Figure 2, Section 2.2.1).

$$rCount \leftarrow \left( \frac{rUp - rDown}{precision} \right) + 1$$

$$rCount \leftarrow \left( \frac{0.9 - 0.2}{0.1} \right) + 1$$

$$rCount \leftarrow 8$$

$$sCount \leftarrow \left( \frac{sUp - sDown}{precision} \right) + 1$$

$$sCount \leftarrow \left( \frac{0.8 - 0.1}{0.1} \right) + 1$$

$$sCount \leftarrow 8$$

In this way, 8\*8 (r\*s) C(t) formulas are created (Formula 7). 64 error values are calculated by comparing the values calculated using these formulas with the measured thickness distribution of a 3mm thick high impact polystyrene product after thermoforming using a mold values ("Real" in Figure 3). The error values obtained for some r and s values from these 64 different error values are shown in Table 1 (Branch A).



Table 1 Examples of error values calculated based on r and s values for different precision

Branch	R	S	Average Error
A precision 0.1 i=1	0.3	0.8	44.5248
	0.4	0.7	16.3331
	0.5	0.5	<b>4.3861</b>
	0.5	0.8	36.0096
	0.7	0.5	35.2096
	0.9	0.5	46.6825
B precision 0.01 i=2	0.49	0.53	<b>4.2374</b>
	0.49	0.51	4.3546
	0.49	0.54	4.239
	0.48	0.59	4.6175
	0.41	0.61	17.7456
C precision 0.001 i=3	0.487	0.539	4.2201
	0.487	0.541	<b>4.2148</b>
	0.487	0.543	4.22
	0.491	0.531	4.2416
	0.502	0.501	4.557
	0.502	0.506	4.5979
D precision 0.0001 i=4	0.487	0.5405	4.2139
	0.487	0.5407	<b>4.2133</b>
	0.4876	0.5387	4.2168
	0.4881	0.546	4.248
	0.4937	0.5403	4.3049
	0.4862	0.5259	4.4408

The data related to these error values obtained were transferred to MatLAB® and graphed (Figure 4). In this graph, the vertices where the error value is small (r,s) are seen as darker blue.

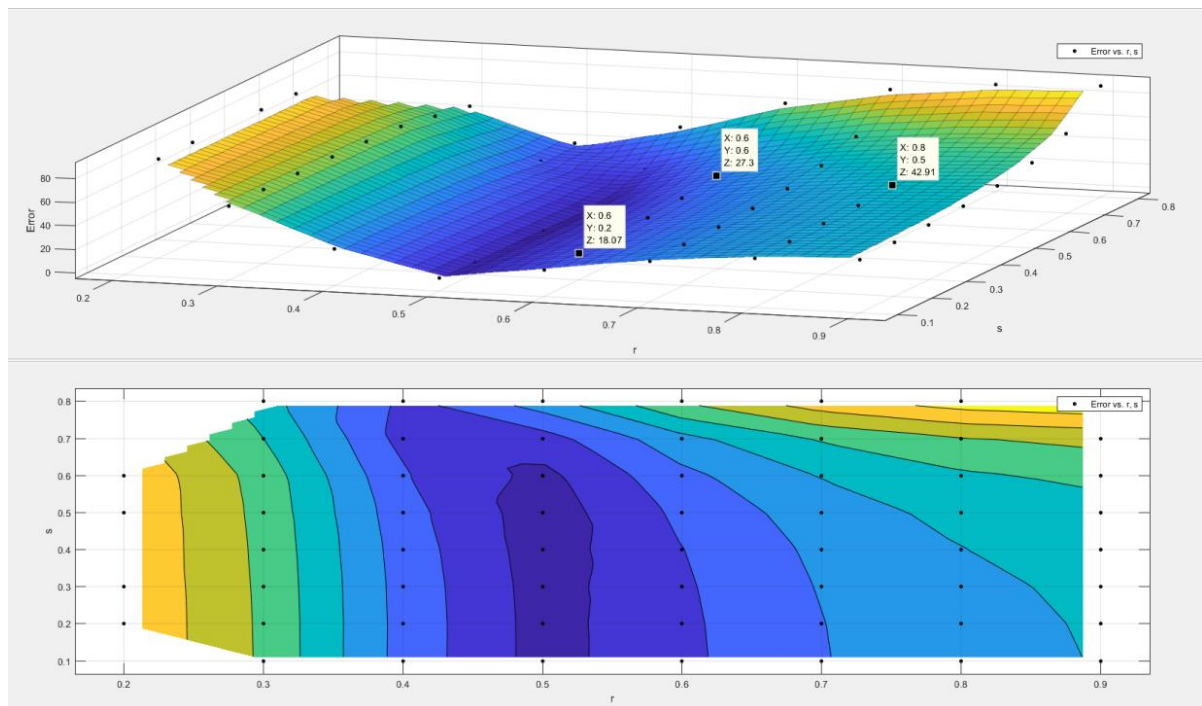


Figure 4 Visualization of the results achieved, precision  $\leftarrow 0.1$

The r and s interval values for precision=0.01 were determined as [0.40, 0.60] and [0.10, 0.70], respectively. Therefore, the r value is calculated as 21 and the s value as 61 (Figure 2, Section 2.3.1 - 2.8).

$$rCount \leftarrow \left( \frac{rUp - rDown}{precision} \right) + 1$$

$$rCount \leftarrow \left( \frac{0.60 - 0.40}{0.01} \right) + 1$$

$$rCount \leftarrow 21$$

$$sCount \leftarrow \left( \frac{sUp - sDown}{precision} \right) + 1$$

$$sCount \leftarrow \left( \frac{0.70 - 0.10}{0.01} \right) + 1$$

$$sCount \leftarrow 61$$

In this way, 21\*61 (r\*s) C(t) formulas are created, and 1,281 error values are calculated by comparing the values calculated using these formulas with the dataset. Some error values for r and s from these 1,281 different error values are shown in Table 1 (Branch B). In addition, the results obtained are shown in Figure 5 as a graph.

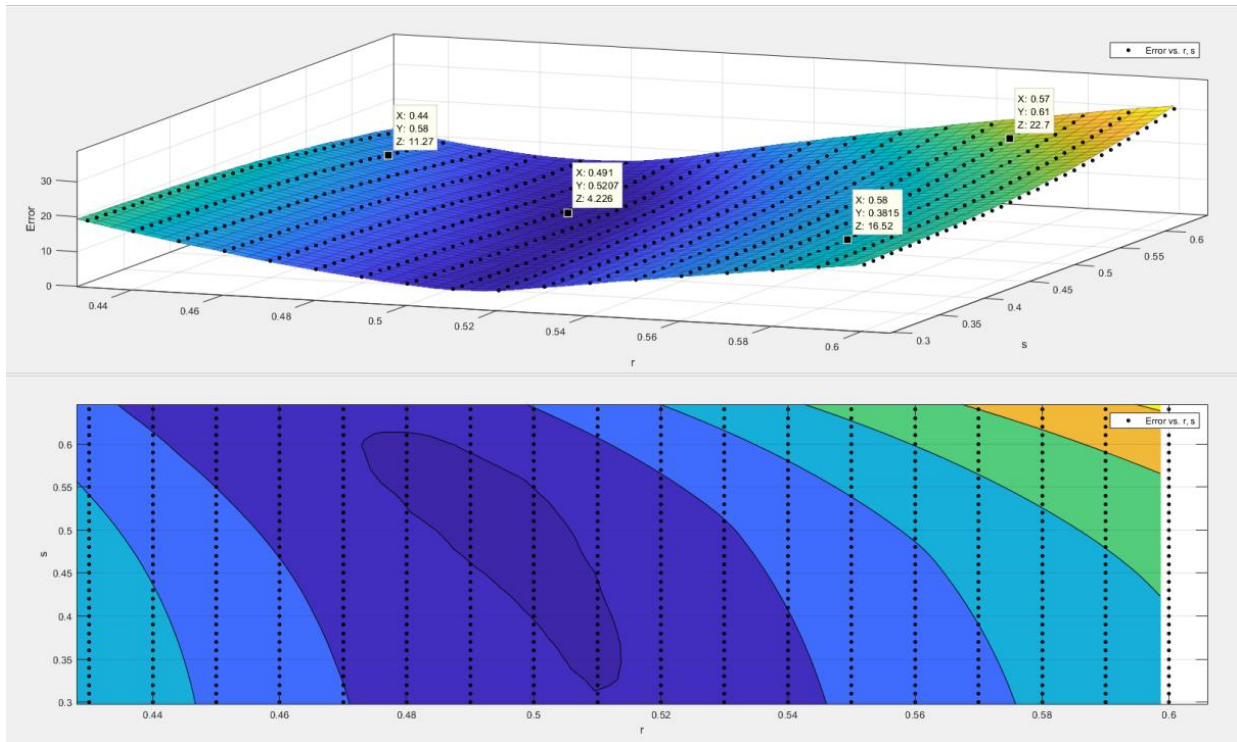


Figure 5 Visualization of the results achieved, precision=0.01

The r and s interval values for precision=0.001 were determined as [0.480, 0.510] and [0.480, 0.570], respectively. Therefore, the r value is calculated as 31 and the s value as 91 (Figure 2, Section 2.4.1 - 2.8).

$$rCount \leftarrow \left( \frac{rUp - rDown}{precision} \right) + 1$$

$$rCount \leftarrow \left( \frac{0.510 - 0.480}{0.001} \right) + 1$$

$$rCount \leftarrow 31$$

$$sCount \leftarrow \left( \frac{sUp - sDown}{precision} \right) + 1$$

$$sCount \leftarrow \left( \frac{0.570 - 0.480}{0.001} \right) + 1$$

$$sCount \leftarrow 91$$

In this way, 31\*91 (r\*s) C(t) formulas are created, and 2,821 error values are calculated by comparing the values calculated using these formulas with the dataset. Some error values for r and s from these 2,821 different error values are shown in Table 1 (Branch C). In addition, the results obtained are shown in Figure 6 as a graph.

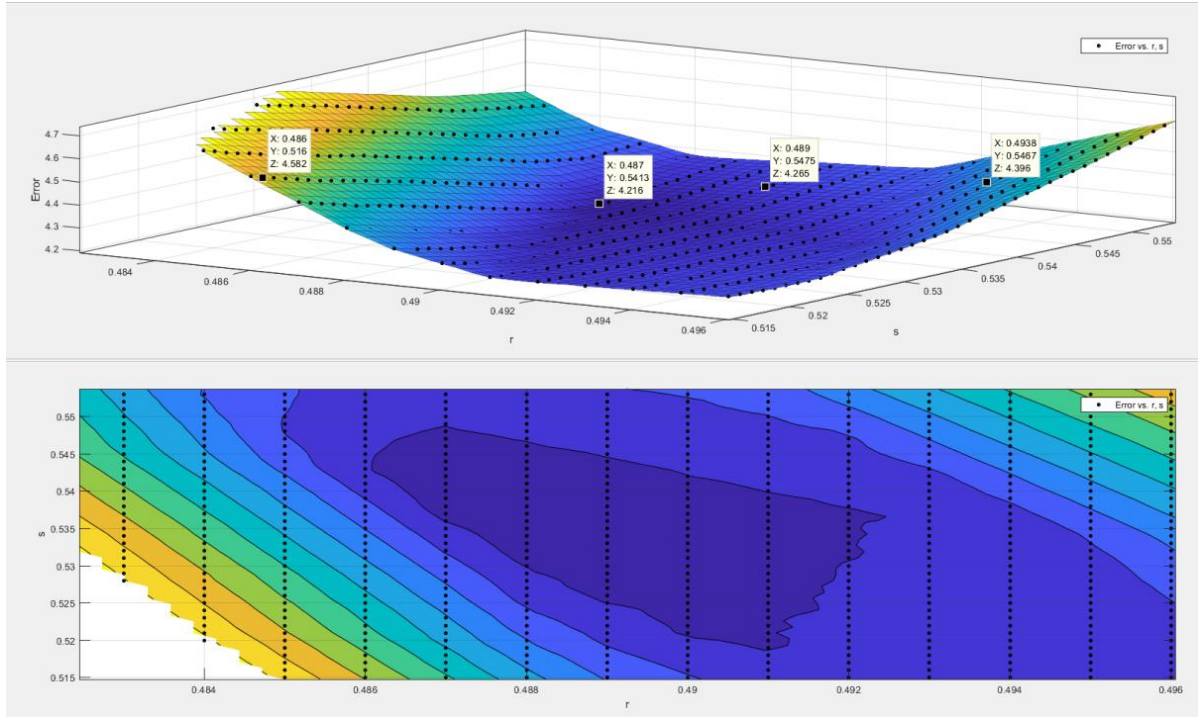


Figure 6 Visualization of the results achieved, precision ← 0.001

The r and s interval values for precision=0.0001 were determined as [0.4860, 0.4940] and [0.5175, 0.5475], respectively. Therefore, the r value is calculated as 81 and the s value as 301 (Figure 2, section 2.5.1 – 2.8).

$$rCount \leftarrow \left( \frac{rUp - rDown}{precision} \right) + 1$$

$$rCount \leftarrow \left( \frac{0.4940 - 0.4860}{0.0001} \right) + 1$$

$$rCount \leftarrow 81$$

$$sCount \leftarrow \left( \frac{sUp - sDown}{precision} \right) + 1$$

$$sCount \leftarrow \left( \frac{0.5475 - 0.5175}{0.0001} \right) + 1$$

$$sCount \leftarrow 301$$

In this way, 81\*301 (r\*s) C(t) formulas are created, and 24,381 error values are calculated by comparing the values calculated using these formulas with the dataset. Some error values for r and s from these 24,381 different error values are shown in Table 1 (Branch D). In addition, the results obtained are shown in Figure 7 as a graph.

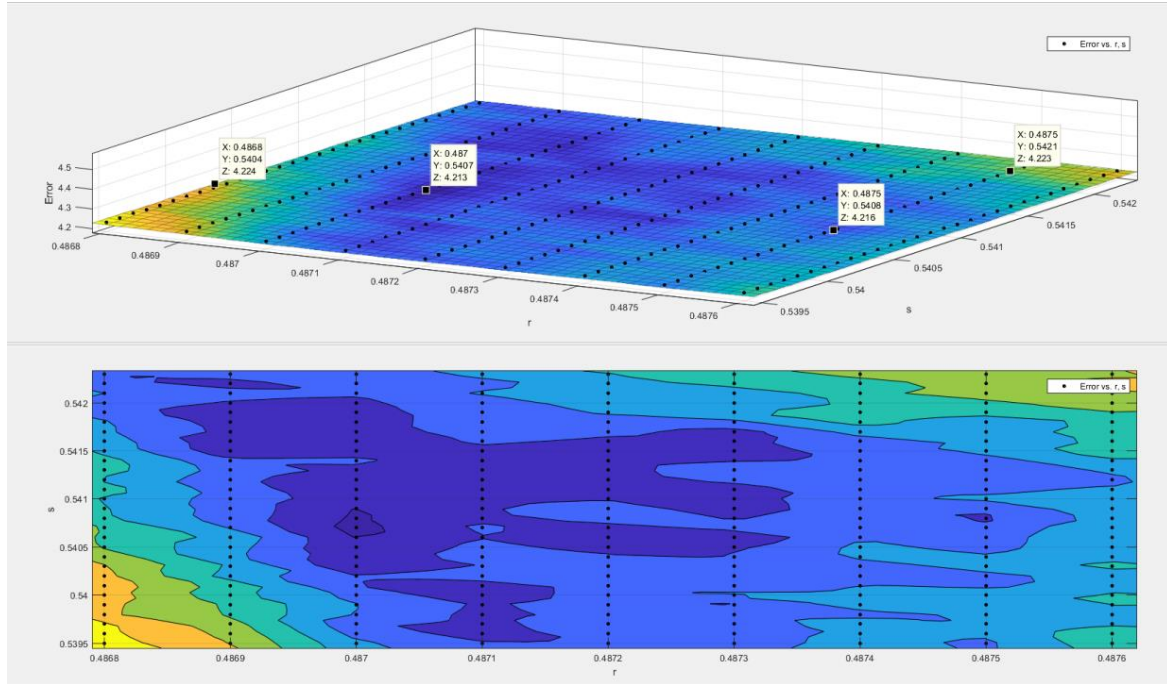


Figure 7 Visualization of the results achieved, precision  $\leftarrow 0.0001$

The error value decreased as the interval value for r and s values was reduced in a controlled manner, taking into account the precision value. The error value for r=0.487, s=0.5407 was calculated as 4.2133 percent (Table 1, Branch D). This value was calculated as 4.3861 percent for r=0.5, s=0.5 (Table 1, Branch A) for PB's default state. This value also confirms the value obtained in the reference study. By optimizing the PB, the error value was reduced by 0.1728 percent.

The total absolute percentage relative error value for generalized C(t) was calculated as 67.4132 ( $\sum_{n=8}^{23} |relativepercentageerror|$ ), and the average absolute percentage relative error value was calculated as 4.2133 ( $(\sum_{n=8}^{23} |relativepercentageerror|)/16$ ) (Table 2). This value means that the program produced a more accurate result in this study than in the previous study [16].

Table 2 PB curve values (default PB and GPB) for dataset

X axis values	Measured thickness value (mm)	Default PB(r=0.5; s=0.5)		GPB (r=0.4870; s=0.5470)	
		Y axis value produced by the program, C(t)	Relative error (%)	Y axis value produced by the program, C(t)	Relative error (%)
0.8	0.4283	0.4283	0	0.4283	0
0.9	0.3866	0.4381	13.3213	0.4328	11.9503
1	0.3883	0.4694	20.8859	0.4582	18.0015
1.1	0.4716	0.5244	11.1959	0.5069	7.4852
1.2	0.5816	0.6055	4.1094	0.5817	0.0172
1.3	0.7183	0.7153	0.4177	0.6856	4.5524
1.4	0.8766	0.8563	2.3158	0.8221	6.2172
1.5	1.07	1.0312	3.6262	0.9948	7.028
1.6	1.2383	1.2417	0.2746	1.2068	2.5438
1.7	1.4566	1.4881	2.1626	1.4597	0.2128

Table 2 PB curve values (default PB and GPB) for dataset (cont.)

<i>X axis values</i>	<i>Measured thickness value (mm)</i>	<i>Default PB(r=0.5; s=0.5)</i>		<i>GPB (r=0.4870; s=0.5470)</i>	
		<i>Y axis value produced by the program, C(t)</i>	<i>Relative error (%)</i>	<i>Y axis value produced by the program, C(t)</i>	<i>Relative error (%)</i>
1.8	1.69	1.7667	4.5385	1.7509	3.6036
1.9	2.01	2.0676	2.8657	2.0691	2.9403
2	2.4483	2.3711	3.1532	2.3896	2.3976
2.1	2.675	2.6471	1.043	2.6738	0.0449
2.2	2.8683	2.8606	0.2685	2.8803	0.4184
2.3	2.9833	2.9833	0	2.9833	0

The mean percent error with the GPB was calculated as 4.2133. This value was 4.3861 for the default PB (Table 1 Branch A and [16]). By optimizing the PB, an improvement of 0.1728 percent was achieved in the error value.

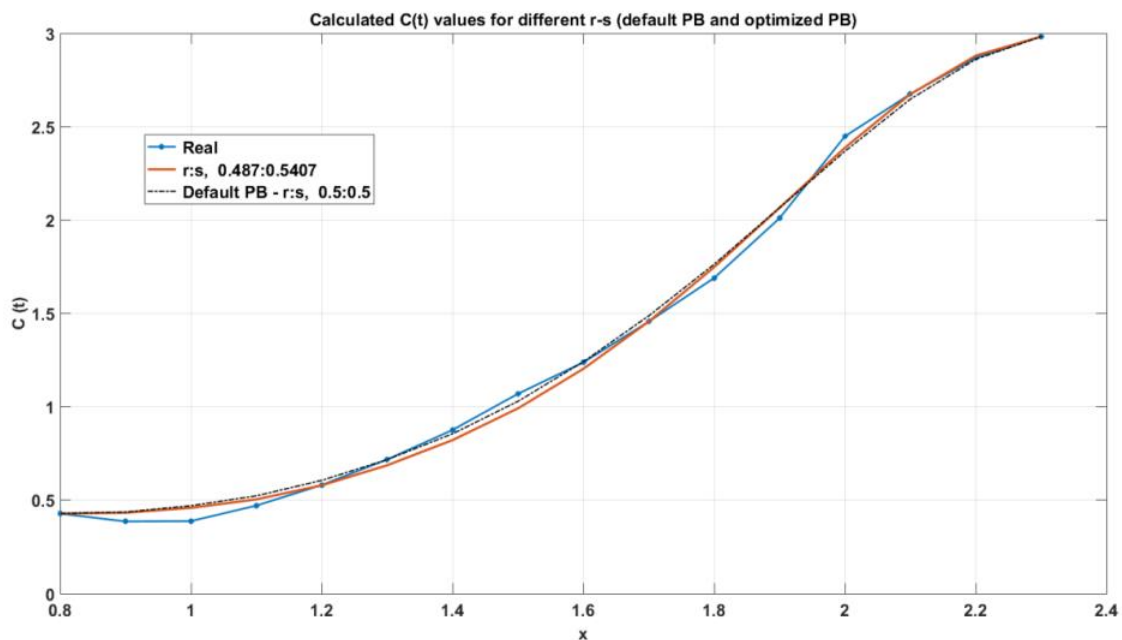


Figure 8 Default PB and GPB versus dataset (real measurement results)

## 5. Conclusion

The comparison of the calculated  $C(t)$  for the default  $r:s$  values and the calculated  $C(t)$  for the generalized  $r:s$  values with the dataset is shown in Figure 8. When the figure is examined, it is seen that  $C(t)$  calculated with the GPB outside the range of [1.2, 1.6] approaches the dataset with less error than the GPB.

The dataset used in this research was created as a result of measuring the thickness distribution of a 3 mm thick high-impact polystyrene product after thermoforming using a mold. For this dataset, the most suitable  $r$  and  $s$  values that can be used with a precision of  $10^{-4}$  were calculated as  $r=0.487$  and  $s=0.5407$ . These  $r$  and  $s$  values may vary for another dataset. With the algorithm and software developed in this study, the most appropriate  $r$  and  $s$  values can be determined for a new dataset. In every study where PB can be applied, the optimization steps revealed in this study can be applied.

OVR is a special case of GPB for the values for which it will take the  $r$  and  $s$  parameters. For this reason, it is not possible to say unconditionally that GPB produces more accurate results in all cases or for all types of datasets. Hadavinia et al. [17], working on sample functions, stated that GPB can produce more accurate results than OVR. In this study, the results obtained by working on a real dataset were evaluated. The results obtained confirm the results and claims of Hadavinia et al. In addition, the results obtained through this study have been visualized in an easy-to-follow manner, and a software development process algorithm has been presented for use in further studies. In addition, it will be less costly for the software development process to fit curves in a flexible structure using four vertices instead of creating a high-order interpolation polynomial using all vertices.

## References



- [1] Shrivastava, A., & Dalla, V. K. (2022). Multi-segment trajectory tracking of the redundant space robot for smooth motion planning based on interpolation of linear polynomials with parabolic blend. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 09544062221088723, doi.org/10.1177/09544062221088723
- [2] Hachemi, M., Hamza-Cherif, S. M., & Houmat, A. (2017). Free vibration analysis of variable stiffness composite laminate plate with circular cutout. *Australian Journal of Mechanical Engineering*, doi.org/10.1080/14484846.2017.1385694
- [3] Ben Abdallah, A., Kallel, A., Hassine, T., Gamaoun, F., & Tcharkhtchi, A. (2022). Modeling of viscoelastic behavior of a shape memory polymer blend. *Journal of Applied Polymer Science*, 139(13), 51859, doi.org/10.1177/09544062221088723
- [4] Overhauser, A. W. (2005). Analytic definition of curves and surfaces by parabolic blending. arXiv preprint cs/0503054.
- [5] Liu, Y., & Rizzo, F. J. (1991). Application of Overhauser C 1 Continuous Boundary Elements to "Hypersingular" BIE for 3-D Acoustic Wave Problems. In *Boundary elements XIII* (pp. 957-966). Springer, Dordrecht, doi.org/10.1007/978-94-011-3696-9\_75
- [6] Walters, H. G., & Gipson, G. S. (1994). Evaluation of overhauser splines as boundary elements in linear elastostatics. *Engineering analysis with boundary elements*, 14(2), 171-177, doi.org/10.1016/0955-7997(94)90093-0
- [7] Durodola, J. F., & Fenner, R. T. (1996). Overhauser triangular elements for three-dimensional potential problems using boundary element methods. *International journal for numerical methods in engineering*, 39(24), 4183-4198, doi.org/10.1002/(SICI)1097-0207(19961230)39:24<3C4183::AID-NME38%3E3.0.CO;2-9
- [8] Brewer, J. A., & Anderson, D. C. (1977). Visual interaction with overhauser curves and surfaces. *ACM SIGGRAPH Computer Graphics*, 11(2), 132-137, doi.org/10.1145/965141.563883
- [9] Schneider, W. (1986). A simple technique for adding tension to parabolic blending interpolation. *Computers & Mathematics with Applications*, 12(11), 1155-1160, doi.org/10.1016/0898-1221(86)90019-2
- [10] Qian, X., Yuan, H., Zhou, M., & Zhang, B. (2014). A general 3D contact smoothing method based on radial point interpolation. *Journal of Computational and Applied Mathematics*, 257, 1-13.
- [11] El-Abbasi, N., Meguid, S. A., & Czekanski, A. (2001). On the modelling of smooth contact surfaces using cubic splines. *International Journal for Numerical Methods in Engineering*, 50(4), 953-967, doi.org/10.1002/1097-0207(20010210)50:4<3C953::AID-NME64%3E3.0.CO;2-P
- [12] Chung, K. H., Kim, J. W., Ryu, K. W., Lee, K. T., & Lee, D. J. (2006). Sound generation and radiation from rotor tip-vortex pairing phenomenon. *AIAA journal*, 44(6), 1181-1187, doi.org/10.2514/1.22548
- [13] De Almeida Barros, P. L., & de Mesquita Neto, E. (2000). Singular-ended spline interpolation for two-dimensional boundary element analysis. *International Journal for Numerical Methods in Engineering*, 47(5), 951-967.
- [14] Kunz, T., & Stilman, M. (2012). Time-optimal trajectory generation for path following with bounded acceleration and velocity. *Robotics: Science and Systems VIII*, 1-8.

- [15] Burgoyne, C. J., & Crisfield, M. A. (1990). Numerical integration strategy for plates and shells. *International journal for numerical methods in engineering*, 29(1), 105-121, doi.org/10.1002/nme.1620290108
- [16] Ekşi, O., & Üstünel, H. (2020). Application of parabolic blending for the estimation of thickness distribution in thermoformed products. *Journal of Elastomers & Plastics*, 0095244320959801, doi.org/10.1177%2F0095244320959801
- [17] Hadavinia, H., Travis, R. P., & Fenner, R. T. (2000). C1-continuous generalised parabolic blending elements in the Boundary Element Method. *Mathematical and Computer Modelling*, 31(8-9), 17-34, doi.org/10.1016/S0895-7177(00)00057-1
- [18] Rogers DF and Adams JA. *Mathematical elements for computer graphics*. 2nd ed. New York: McGraw-Hill, 1989.

### Notation List

P, Q	: Parabolas
C	: Cubic function (curve)
P <sub>n</sub>	: n <sup>th</sup> point (It can take integer values between 1 and 4)
r,s,t	: Parameters of P, Q and C respectively ( <i>it can take float values between 0 and 1, including limit values</i> )
[B]	: Coefficient matrix for P
k <sub>n</sub>	: Coefficients of the line equation used in the calculation of t-dependent change in the r and s values ( <i>1 and 2 for r, 3 and 4 for s</i> )
r <sub>instant</sub> (t)	: The calculated r value for any value of the t parameter
s <sub>instant</sub> (t)	: The calculated s value for any value of the t parameter
rCount	: The number of steps in the loop for the precision value set for r ( <i>in the software</i> )
sCount	: The number of steps in the loop for the precision value set for s ( <i>in the software</i> )
rUp, rDown)	: Upper and lower limit value for r value used when calculating rCount ( <i>in the software</i> )
sUp, sDown)	: Upper and lower limit value for s value used when calculating sCount ( <i>in the software</i> )

# Using Multi-Label Classification Methods to Analyze Complaints Against Cargo Services During the COVID-19 Outbreak: Comparing Survey-Based and Word-Based Labeling\*

 Tolga Kuyucuk<sup>1</sup>,  Levent Çağrı<sup>2</sup>

<sup>1</sup>Department of Information Systems Engineering, Sakarya University, tolga.kuyucuk1@ogr.sakarya.edu.tr

<sup>2</sup>Corresponding Author; Department of Information Systems Engineering, Sakarya University, lcalli@sakarya.edu.tr

Received 26 May 2022; Revised 18 November 2022; Accepted 21 November 2022; Published online 31 December 2022

## Abstract

This study investigates how cargo companies managed their last-mile activities during the Covid-19 outbreak and suggest a solution to the adverse outcomes. The data used in the study included complaints made about cargo companies from sikayetvar.com between February 2020 and September 2021 and was collected using Python language and the Scrapy module web scraping methods. Multi-label classification algorithms were used to categorize complaints based on assessments of training data obtained according to the topics. Results showed that parcel delivery-related themes were the most often complained about, and a considerable portion were delay issues.

**Keywords:** Covid-19, web scraping, cargo companies, customer complaints, multi-label classification, text mining

## 1. Introduction

A series of pneumonia cases of unknown origin emerged in Wuhan, China, in the late days of December 2019, triggering an investigation that the source of this infection is thought to be linked to the Wuhan South China Seafood Market. In January, the Chinese government reported this situation to the World Health Organization (WHO), and the pathogen causing this epidemic was defined as a new type of coronavirus on January 7, 2020, and named Covid-19 [1]. The virus expanded to other nations and had a huge global impact when the number of cases increased rapidly till January 23, 2020 [2]. The first Covid-19 case in Türkiye was seen on March 11. WHO declared that this newly emerged virus was a pandemic on the same date [3]. Due to the coronavirus, nearly 527 million cases have been reported globally as of May 24, 2022, and as a result of this outbreak, it permanently influenced all aspects of consumers' needs and daily habits with a new lifestyle [4]–[7].

During the pandemic, e-commerce has increased globally, especially in countries where e-commerce was previously less developed have been more remarkable. The growth of the digital economy will undoubtedly continue to be influenced by the changing shopping and payment behaviors of customers brought about by the pandemic conditions [8]. In 2020, Türkiye's e-commerce volume increased 66% compared to 2019 and reached 226.2 billion TL from 136 billion TL [9]. In 2021, Türkiye's e-commerce volume climbed by 69% from the previous year and reached 381.5 billion TL, while the orders increased by 46%, from 2 billion 297 million to 3 billion 347 million [10]. The asymmetrical growth that resulted inevitably affected the delivery step, the most crucial part of e-commerce. According to Güven [11], customers' main complaints during the Covid-19 outbreak were discovered to be customer service / live assistance and the delivery process, considering complaints received on e-commerce sites.

Additionally, Parlakkılıç et al. [12] found a significant negative relationship between trust and cargo tracking during the Covid-19 period. A similar problem has also been mentioned from the industry side; for example, Etid's [13] report states that the high instantaneous increase in demand during the pandemic led to problems in the logistic organization, warehouse, and order preparation operations resulting in

---

\* This study was produced from the master's thesis prepared by Tolga KUYUCUK with the supervisor Levent ÇALLI at Sakarya University, Institute of Natural Sciences.



customer dissatisfaction. In this regard, it is strategically vital for cargo companies to identify the reasons that trigger consumer complaints during the delivery phase of e-commerce to take the necessary precautions.

In this sense, it is thought that this research will make an academic and practical contribution to the literature from a different perspective. First of all, this research fills the gap in the literature with sample size and puts forward an alternative method considering studies carried out by a relatively small number of complaints in the local literature. Hence, in this study, much more complaints were classified by machine learning approaches, and a method will be proposed for the literature. In addition, in extraordinary situations such as Covid-19, practical contributions to cargo firms will be provided according to research findings.

## 2. Literature Review

When a consumer has an issue with a product or service they purchase, they have three options under the complaint behavior. The first of these options is to cease buying the relevant brand and promote negative word of mouth among the social circle. The second is the direct contact of the consumers with the company, and the last way is to initiate the legal process [14]. Today, with social media applications, it has become easier for the consumer to use these different methods of complaint behavior simultaneously, and it has provided convenience for consumers in terms of solutions for complaints [15][16]. For example, *sikayetvar.com* [17] has emerged as an online platform that acts as a bridge between customers and brands by allowing companies to resolve complaints and increase customer satisfaction. Users who visit the system can easily see the brands that received the most complaints or those with the most resolved complaints. In this sense, the platform, which creates pressure on brands, has gained much popularity in Türkiye and has resolved approximately 2 million complaints. This platform is also a resource used in academic studies. For instance, Çallı and Çallı [16], who conducted research into the complaints mentioned by airline passengers during the Covid-19 pandemic on the *sikayetvar.com* platform, highlighted the service quality issues and proposed potential solutions for both low-cost carriers and full-service airlines. In another study that analyzed complaints regarding private hospitals, topics and sub-topics were revealed using data from *sikayetvar.com* [18]. In a qualitative study, Güler [19] found that half of the complaints against banks during the pandemic period were related to credit cards via *sikayetvar.com*.

A limited number of studies in the academic literature focus on the topic of customer complaints made against cargo services at *sikayetvar.com*. All of these studies use a qualitative method, which means they can only deal with a small number of complaints. In this regard, this study is expected to fill a gap in the relevant literature in terms of the methodology applied and the number of complaints obtained. In the study conducted by Burucuoğlu and Yazar [20], which was prepared by considering 300 complaints, the main complaint topics for cargo services operating in Türkiye before the pandemic are; business processes, product-related problems, courier-related issues while receiving the parcel, delivery/distribution-related issues, pricing, communication, and personnel. According to Gürce and Tosun's [21] findings, the most common complaint themes for the cargo services were providing the promised service, timely delivery, fulfilling good service, willingness to assist, and sincere problem-solving. The study considered 300 complaints made to various online shopping sites and found that one of the most frequently mentioned complaint topics in the Covid-19 period is mainly related to cargo services and was revealed as unfair shipping charges, sending to the wrong address, not receiving the orders, and not delivering the product on time [11]. In another study considering 690 complaints against online shopping sites during the Covid-19 outbreak [22], it is stated that similar complaints about the delivery process that Güven [11] mentioned are the second most frequently mentioned issues.

Considering local research about complaints other than *sikayetvar.com* against cargo services, Deniz and Gödekmerdan [23] found that delay is the most common problem while determining the factors that cause dissatisfaction in cargo transportation in Türkiye with the survey method. According to Akkan [24], the first two most common service issues are delivery times, such as the delivery of the parcel late or delivery later than promised, and communication difficulties, like not answering calls or leaving a note even though the customer is at home for Turkish customers. Duran et al. [25] evaluated the opinions

of consumers about cargo services within the framework of five factors. The first is logistics values, including parcel delivery time and a widespread transportation network. Reliability, which includes the concepts of timely delivery of the product and easy communication with the customer, is another concept evaluated within the model's scope. Delivery speed is evaluated under the time factor, which includes the delivery at the promised time, the return process, and the provision of information to the customer. The factor that includes price-quality consistency, compensation for faulty situations, and promotional elements is the economic expense. Finally, the concept of personnel and service consists of dimensions that include fulfilling expectations, assurance, being kind to the customer, product follow-up, and solving complaints in a short time. A study examining 300 complaints about the three largest cargo companies operating in Türkiye stated that the customers mostly expressed problems related to not being found at the address, issues with the delivery, not delivering to the address, and the attitudes and behaviors of the service personnel [20].

The practical solution to customer complaints is a critical factor in customer satisfaction and loyalty. For example, in their study, Cho et al. [26], considering the complaints of e-commerce customers, determined that customer service, product, price, delivery problems, misleading information, security & trust issues, tracking and tracing, and promotion are general complaints topics. They state that online customers should be provided with the best service, customer demands and complaints should be responded to more quickly than offline customers, and strategies should be developed in line with the product category, such as giving more detailed information with different multimedia tools for cosmetic products.

If businesses manage complaints effectively, they can develop their products/services to meet the expectations of their customers. A dissatisfied customer decides whether to leave or stay based on the complaint's solution. If the business handles this process successfully, it will offer an excellent opportunity for the firm, considering that acquiring new customers is five or six times more expensive than retaining existing customers [27].

## 2.1 Machine Learning for Complaint Classification

Many online customers experiencing issues with delivery service companies due to the pandemic have been looking for answers by posting their complaints on online complaint sites. Examining each complaint in a sector with a high volume of complaints can be costly for firms' budgets since it requires more human resources and intelligence. Instead, reviewing only a few complaints with machine learning algorithms allows future complaints to be categorized more quickly and at a lower cost.

The concept of machine learning is an area of Artificial Intelligence that attracts excellent attention in the digital world and is a crucial component of digitization solutions. Depending on the types and categories of training data, methods such as supervised, unsupervised, semi-supervised, and reinforcement learning are used [28]. Basically, machine learning (ML) can be defined as a continually changing computing program that, in some ways, mimics human intelligence by learning from its surroundings [29]. Regression, classification, clustering, dimensionality reduction, ensemble methods, neural nets, deep learning, transfer learning, reinforcement learning, natural language processing, and word embeddings are commonly used machine learning approaches applied to any data scenario [30].

Text mining, one of the fundamental techniques in data mining, is described as discovering knowledge by the computers automatically extracting information from various unstructured or structured textual sources [31], [32]. Natural Language Processing (NLP) is a branch of artificial intelligence in which computers efficiently analyze and understand human language with machine learning. While sentiment and grammatical structure can be extracted with NLP from language, frequency, correlation, and word patterns may be revealed with text mining as statistical indicators with a multidiscipline approach [33]–[35].

A more complex situation is encountered in complaint cases in text mining practices. While the analysis process assumes that each item belongs to a single class in classification problems, this is not possible in complaints due to their nature. For example, a consumer complaining about undelivered cargo may also say he could not reach customer service in the same complaint message. Multi-label classification

is known as a solution for this type of challenge, which may categorize each complaint such that it may be allocated to more than one topic using machine learning techniques [36].

In this study, Python language, which has become one of the leading technologies in creating models for the industry and developing new methods for researchers, together with machine learning libraries, was used to perform multi-label classification using the Scikit-multilearn [37].

### 3. Methodology

#### 3.1 Text Mining Process

The data for the study were collected using Python scripts and the Scrapy module from the sikayetvar.com website, an online complaint management platform, between February 2020 and September 2021. The database consists of 16332 customers who received service from cargo companies in Türkiye and expressed their dissatisfaction on the sikayetvar.com website. Complaints against cargo companies in Türkiye during the Covid-19 outbreak were classified using machine learning and multi-label classification algorithms.

The qualitative approach was used to discover the most common complaints of cargo customers during the pandemic period by reviewing a random sample of the complaints in the database considering the literature findings. The frequent complaints were formed under six topics as follows; *delayed or not delivered parcel, the note was written "customer was not at home" was left at the door or parcel not brought to the door, customer service has not answered the call, returning processes, parcel not received or delivered, and hygiene rules related issues.*

The dataset's first 3000 rows were used as training data and were refined using natural language processing (NLP) techniques. Labels were assigned to the training data using two approaches. The first method used a written python script to label complaints, while the second method included the survey method to label complaints based on participant responses. The text mining process used in the study is presented in Figure 1.

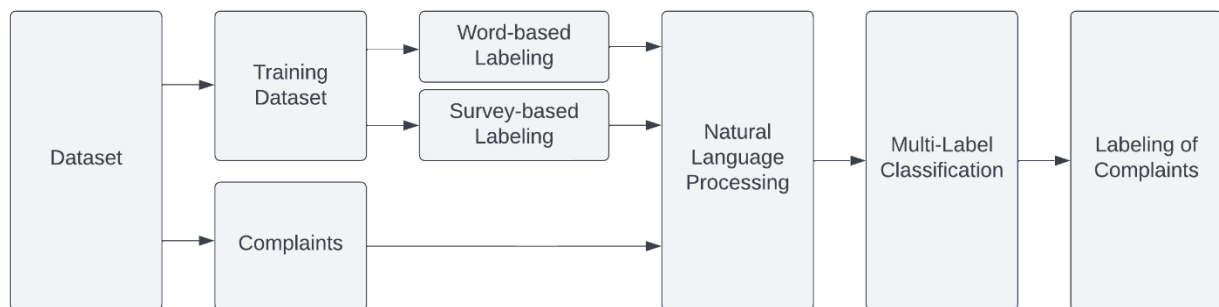


Figure 1 Text Mining Process

Logistic Regression, OneVsRest Classifier, and vectorization methods were used for multi-label classification.

##### 3.1.1 Multi-Label Classification

This section explains the results of word-based and survey-based labeling methods for selecting the appropriate topics for each complaint in the training dataset.

### 3.1.1.1 Word-Based Labeling

After tokenizing each complaint in the training dataset, a script written in Python programming language checked whether the determined words related to the relevant topic for labeling the complaint to the appropriate topics. This process is illustrated in figure 2.

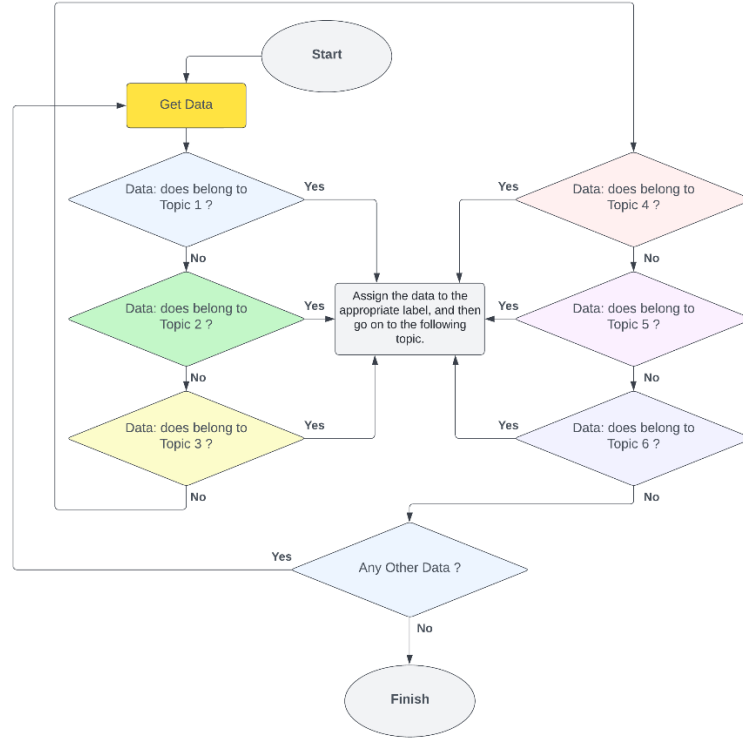


Figure 2 Word-based Labeling Flowchart

### 3.1.2.2 Survey-Based Labeling

Survey-based labeling aims to get more accurate labeling with human intelligence by having different participants assign labels to each complaint in the training dataset. Three thousand complaints from the training dataset were primarily uploaded to the Google Sheets platform in Turkish and English for Turkish and foreign participants to access the survey. The numbers generated between 3000 and 1 were assigned to the complaints as temporary random IDs, and a dynamic structure was obtained with the page renewed every 4 minutes.

1	Yurtici Kargo Evde Olmama Rağmen Kargom Gelmedi.-> 71117690850 numaralı gönderi ile tarafıma kargo gönderildi. Kendim şu an Covid 19 tanısı ile yaklaşık 1 haftadır evde karantına olmama rağmen güya arkadaşlar gelmiş kapıyı çalmış telefonla aramışlar ama ulaşamamışlar. Evde karantinadayım ve telefonum kesinlikle çalmadı. Bu yüzden şu an kargom şubede ve kargo olarak gelende tedavim için gerekli ilaç. Şubeyi aradığımda da bugün getirmenin mümkün olmayacağını söylüyor. (ŞİKAYET NO: 960)	FORM	Yurtici Cargo Shipping Despite not being at home I did not come -> number 71117690850 with the shipment sent to the cargo. Despite myself to be that Covidien has approximately 1 week with 19 diagnostic quarantine at home they allegedly stole a phone call friends come across the door but could not reach. I'm definitely quarantine at home and the phone did not ring. So now required for drug treatment in branch offices and cargo from the cargo. When I call the Branch says they will not be able to bring today. (Complaint NO: 960)	1
2	PTT Kargo Dağıtım Çıkamayan Kargo-> 9 Temmuz 2020 tarihinde PTT iş yerine gelen KP0273****95 takip numaralı kargom hala dağıtım çıkmadı! Dakikalarca hatta bekleme rağmen müşteri hizmetlerine ulaşamıyorum! Pandemi nedeniyle bir çok alışverişimizi internette yaptığımız bu dönemde PTT şubeye gidip paketimi teslim almak zorunda kalacağım! (ŞİKAYET NO: 1640)	✎	PTT Cargo Distribution could not get out of Shipping -> Post Office came to the job on July 8, 2020 KP0273**** 95 cargo tracking number'm still waiting for distribution! Although I can not get to wait for minutes or even customer service! I'm going to get in this period went to deliver my package from the post office branch we did a lot of shopping our internet because the pandemic! (Complaint NO: 1640)	2
3	Surat Kargo Teslim Süresi-> 20.06.2020 tarihinde Ayvalık şubesine paketim gelmiş. 1 hafta olacak hala teslim etmiyorlar şubede kurye dağıtım gözüktüyor 3 gündür nereden nereye dağıtıyorlar anlamıyorum. Şubeyi sitede belirtilmiş numaradan arıyoruz açan yok. O zaman neden varlar kapatsınlar dükkani. Ama gördüğüm kadariyle Surat Kargo'dan herkes şikayetçiymiş. Bahane olarak korona sürecini gösteriyor ama bir tek Surat Kargo'yu vurmuş anlaşılın diğer firmalar ürünleri getiriyor. Kargoyla iş çok yapıyorum ama ilk defa bir kargoyla bu kadar uğraştım (ŞİKAYET NO: 2264)		Surat Cargo Shipping Delivery Time -> Date 6/20/2020 package came in Ayvalık branch. 1 week still looks courier delivery will be delivered in 3 days they do not understand the branches from where they distribute. No number listed on the site are looking to open a branch. Then why would they have shut shop. But as I see Surat Cargo He was complaining from everyone. Show choir as an excuse, but brings the process to other companies products apparently hit a one-Surat Cargo. I'm doing a lot of work, but first a courier to courier I fought so hard (NO COMPLAINTS: 2264)	3

Figure 3 Google Form Survey

Complaints with IDs between 1 and 10 were shown to the participants to show different complaints to the different participants at different times. The participant who accesses the web page is directed to the survey page created on Google Form by clicking the "Form" button, as seen in Figure 3. The participant is asked to determine which of the ten different complaints belong to the pre-determined categories on the survey page.

### 3.1.2 Text Vectorization Algorithms

Vectorization is a crucial stage in NLP for machine interpretation of data by transforming textual materials into meaningful numerical representations [38]. This study used Tf-Idf Vectorizer, Hashing Vectorizer, and Count Vectorizer methods.

One of the most widely used text vectorization algorithms in today's information retrieval systems is the Term frequency-Inverse Document Frequency (TF-IDF) [39]. TF-IDF method weights word counts by measuring how often they appear in documents. The equation is as follows.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) 100 \quad (1)$$

The description of terms used in equation 1 is as follows [40];

$w_{ij}$  : The weight for Term  $i$  in document  $j$ .

$N$  : The number of documents in the collection.

$tf_{ij}$  : The term frequency of Term  $i$  in document  $j$ .

$df_i$  : The document frequency of Term  $i$  in the collection.

The second method used for text vectorization is hashing. Tokens are stored as strings in the hashing-vectorizer, and the hashing trick is used to encode features as numerical indexes [41]. The hashing trick creates a unique association between the input and the hash value and replaces the authenticity of a large quantity of information with a much smaller hash value [42]. The Count Vectorizer is used as the third method for text vectorizing in this study as a simple technique based on the count of word occurrences in the document [41].

### 3.1.3 Classification

One-vs-rest (OvR) method was used for the multi-label classification with Python coding, as seen in figure 4. OvR applies binary classification methods for multi-class classification by splitting the multi-class dataset into multiple binary classification models. Then, each binary classification problem is used to train a binary classifier, and the most confident model is used to make predictions [43].

```

1 import numpy as np # linear algebra
2 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
3
4
5 from sklearn.feature_extraction.text import HashingVectorizer
6 from sklearn.feature_extraction.text import TfidfVectorizer
7 from sklearn.feature_extraction.text import CountVectorizer
8 from sklearn.linear_model import LogisticRegression
9 from sklearn.multiclass import OneVsRestClassifier
10 from sklearn.pipeline import Pipeline
11
12
13 train = pd.read_excel('input/train.xls')
14 sikayetsayisi=len(train)
15 yontem=HashingVectorizer
16
17 print(train.info())
18 print(train.describe())
19 print(train.icerik.head())
20
21 #creating x and y
22 x=train.loc[:, 'icerik']
23
24 z=[0,1,2,3,4,5]
25 z[0]=train.loc[:, 'Gecikti veya Dağıtım Çıkmadı']
26 z[1]=train.loc[:, 'Evrde yok notu düğüldü veya Kapıya Getirilmedi']
27 z[2]=train.loc[:, 'Telefonlara Cevap Verilmedi']
28 z[3]=train.loc[:, 'İade Süreci']
29 z[4]=train.loc[:, 'Teslim Alınmadı veya Teslim Edilmedi']
30 z[5]=train.loc[:, 'Hiçyen Kurallarına Uyulmadı']
31
32 y = train.drop(['id', 'icerik'], axis=1)
33
34 tks = '[A-Za-z0-9]+(?:\\s+)'
35
36
37 pl = Pipeline([
38     ('vec', yontem(token_pattern = tks)),
39     ('clf', OneVsRestClassifier(LogisticRegression()))
40 ])
41
42 # Fit to the training data
43 pl.fit(x,y)

```

Figure 4 Example of Python Coding

Logistic regression was used as the prediction method. The first five rows of the database are shown in Table 1 as an example of the coding procedure. The mean score of each category in the training dataset was used to determine categories for each complaint, and scores above the mean score were coded as 1, as seen in Table 2.

Table 1 Category Prediction Scores of Complaints

ID	Delayed or Not Delivered Parcel	The Note Was Written "Customer Was Not at Home" Was Left at The Door or Parcel Not Brought to The Door	Customer Service Has Not Answered the Call	Returning Processes	Parcel Not Received or Delivered	Hygiene Rules Related Issues
1	0,2612	0,4362	0,44955	0,1009	0,0873	0,0633
2	0,3179	0,2796	0,4277	0,2112	0,2756	0,094
3	0,6167	0,3059	0,26176	0,1664	0,1481	0,5855
4	0,3715	0,1817	0,64245	0,0859	0,1601	0,0408
5	0,6186	0,2144	0,16975	0,0903	0,0893	0,0556

Table 2 Binary Coding of Each Complaint

ID	Delayed or Not Delivered Parcel	The Note Was Written "Customer Was Not at Home" Was Left at The Door or Parcel Not Brought to The Door	Customer Service Has Not Answered the Call	Returning Processes	Parcel Not Received or Delivered	Hygiene Rules Related Issues
1	0	1	1	0	0	0
2	0	0	1	1	1	1
3	1	0	0	0	0	1
4	0	0	1	0	0	0
5	1	0	0	0	0	0

Analysis processes were carried out with each text vectorization algorithm to discover the general complaint topics. The topics that lead to the most significant number of customer complaints were identified, and the rates of complaints topics according to the cargo firms were calculated.

## 4. Results

### 4.1 Density Map

All complaints containing the words corona, covid, and pandemic listed on the sikayetvar.com website were acquired within the specified period with Python code using the Scrapy module, data including company name, number of reads, and created date. Complaints regarding cargo companies were isolated, and a density map was generated based on March 2020 to September 2020 data, as seen in Figure 5.

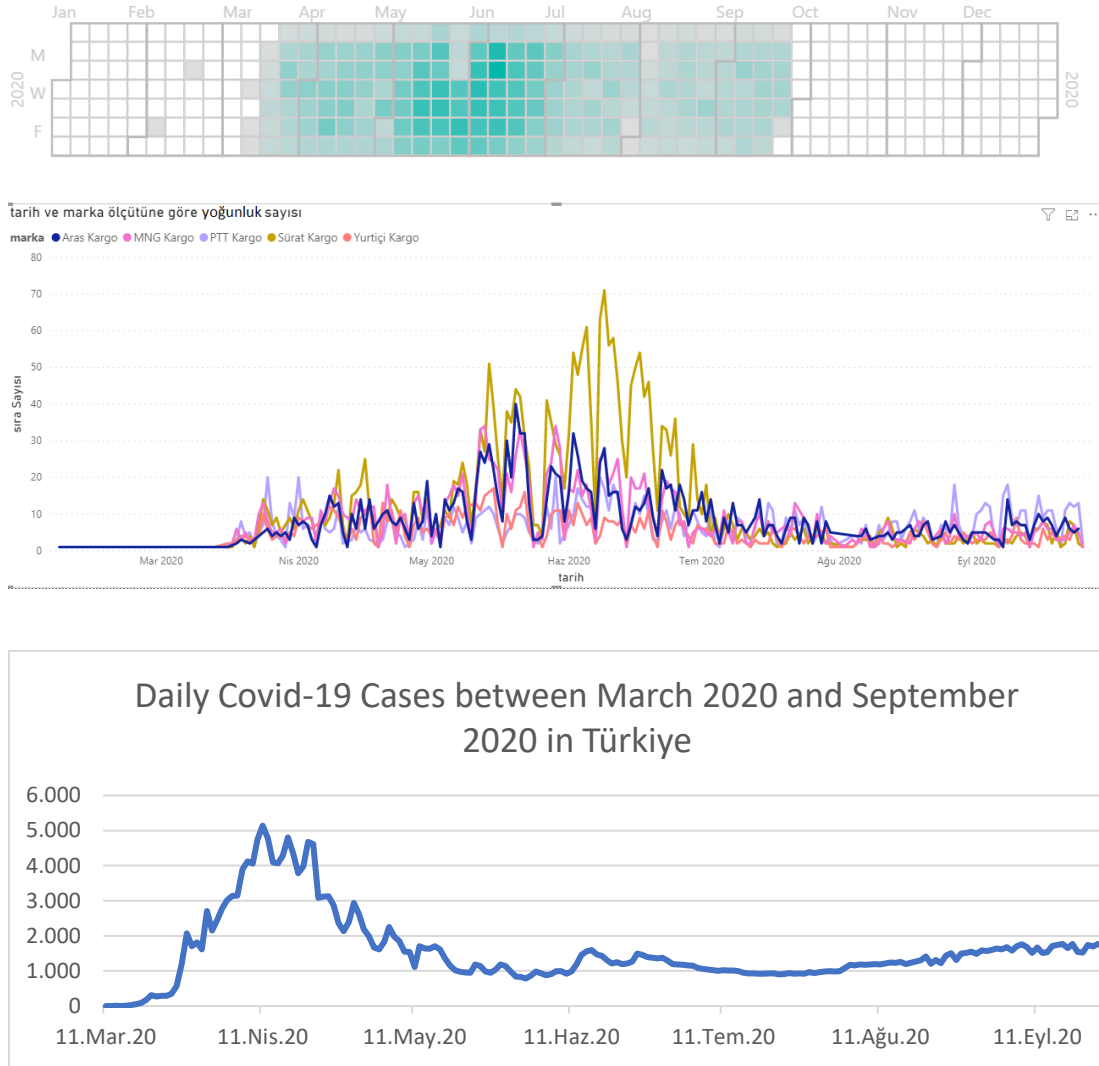


Figure 5 Density Map

Accordingly, it is observed that the most intense complaints are made between the end of May 2020 and the beginning of July 2020, which shows a pattern with the Covid-19 cases in Türkiye that are most intense between April 2020 to May 2020 [44]. Following the peak of Covid-19 cases in April and May 2020, it was observed that complaints increased between May and July 2020.

#### 4.2 Prediction Results

A computer program was created to generate the training data by determining the words and word groups that are assumed to be related to each category, as illustrated in Figure 6. The first 3000 complaints are used for creating the training dataset in this method.

```

280 if 'dağıtım' in icerik:
281     dagitimdegisken=dagitimdegisken+1
282 if 'çıkmadı' in icerik:
283     dagitimdegisken=dagitimdegisken+1
284 if 'çıkış şube' in icerik:
285     dagitimdegisken=dagitimdegisken+1
286 if 'gelmedi' in icerik:
287     dagitimdegisken=dagitimdegisken+1
288 if 'beklet' in icerik:
289     dagitimdegisken=dagitimdegisken+1
290 if 'süredir' in icerik:
291     dagitimdegisken=dagitimdegisken+1
292 if 'gecik' in icerik:
293     dagitimdegisken=dagitimdegisken+1
294 if 'bekli' in icerik:
295     dagitimdegisken=dagitimdegisken+1
296 if 'hala çıkış' in icerik:
297     dagitimdegisken=dagitimdegisken+1
298 if 'geç' in icerik:
299     dagitimdegisken=dagitimdegisken+1
300 if (dagitimdegisken*100)/len(splitWords)>oran:
301     print("Gecikti veya Dağıtım Çıkmadı")
302     print(dagitimdegisken)
344 if 'telefonu' in icerik:
345     telefondegisken=telefondegisken+1
346 if 'telefona' in icerik:
347     telefondegisken=telefondegisken+1
348 if 'telefonlar' in icerik:
349     telefondegisken=telefondegisken+1
350 if 'açmadı' in icerik:
351     telefondegisken=telefondegisken+1
352 if 'cevap' in icerik:
353     telefondegisken=telefondegisken+1
354 if 'açmıyor' in icerik:
355     telefondegisken=telefondegisken+1
356 if 'ulaşlamıyor' in icerik:
357     telefondegisken=telefondegisken+1
358 if 'ulaşama' in icerik:
359     telefondegisken=telefondegisken+1
360 if 'açan yok' in icerik:
361     telefondegisken=telefondegisken+1
362 if (telefondegisken*100)/len(splitWords)>oran:
363     print("Telefonlara Cevap Verilmedi")
364     print(telefondegisken)
    
```

Figure 6 Example of Word-Based Labeling Code

Table 3 shows the confusion matrix based on 53.982 predictions using word-based labeling. The method's accuracy rate was calculated to be 92%.

Table 3 Confusion Matrix Based on Word-Based Labeling

	% 92		
	Prediction: No	Prediction: Yes	Total
Actual: No	33528	1406	34934
Actual: Yes	2874	16174	19048
Total	36402	17580	

Two hundred forty-nine participants filled out the online survey, which was used to generate training data using human intelligence. A total of 1887 questionnaires were found to be suitable for the research after the filtering process. Table 4 shows the confusion matrix prepared according to 33.966 predictions, according to the model trained with survey-based labeling. The accuracy rate of this method was calculated as 87%.

Table 4 Confusion Matrix Based on Survey-Based Labeling

	% 86		
	Prediction: No	Prediction: Yes	Total
Actual: No	21631	805	22436
Actual: Yes	3620	7910	11530
Total	25251	8715	

According to the results, the distribution of the complaint topics is shown in table 5. While considering the means values, It is seen that the most common complaint topics are as follows respectively: delayed or not delivered parcel, customer service has not answered the call, parcel not received or delivered, returning processes, and hygiene rules-related issues.

Table 5 Distribution of Predicted Complaint Topics by Methods

Method	Text Vectorization	The Note Was Written					
		Delayed or Not Delivered Parcel	"Customer Was Not at Home" Was Left at The Door or Parcel Not Brought to The Door	Customer Service Has Not Answered the Call	Returning Processes	Parcel Not Received or Delivered	Hygiene Rules Related Issues
Word-Based	TF-IDF Vectorizer	8156	6176	6902	4189	5696	3104
	Count Vectorizer	8614	6336	7610	3431	5100	1078
	Hashing Vectorizer	8075	5770	7001	3894	5375	4017
Survey-Based	TF-IDF Vectorizer	7019	5991	6514	4751	7706	5196
	Count Vectorizer	5941	6265	5418	2542	6568	2958
	Hashing Vectorizer	7198	5961	6590	4617	7473	5605
	Mean	7501	6083	6673	3904	6320	3660



Another goal of this research is to identify the most common complaints about each cargo operator during the pandemic. Table 6 shows the five cargo firms' complaint topics based on the prediction results.

Table 6 Distribution of Predicted Complaint Topics According to Cargo Firms

	Delayed or Not Delivered Parcel	The Note Was Written "Customer Was Not at Home" Was Left at The Door or Parcel Not Brought to The Door	Customer Service Has Not Answered the Call	Returning Processes	Parcel Not Received or Delivered	Hygiene Rules Related Issues
<b>Firm A</b>	1549	1074	1337	961	1494	984
<b>Firm B</b>	1725	1413	1550	1127	1853	1183
<b>Firm C</b>	730	672	715	499	855	612
<b>Firm D</b>	1660	1383	1548	1102	1823	1218
<b>Firm E</b>	1354	1447	1364	1059	1678	1195

Complaints involving non-delivery and delays are more common, while those about return processes and hygiene rules are relatively less frequent, as seen in Table 6.

## 5. Conclusion

As a result, our study shows that customers primarily complained about the delay or lack of delivery of their cargo. The least common complaint was that cargo staff did not follow the hygiene rules during the pandemic. According to the research results, the number of complaints about each topic and the total number of complaints reveals the best and worst parts of the leading cargo companies in Türkiye. While some companies have difficulties in specific areas, some have had a relatively successful period. In this context, adopting the decision support method used in the study to companies is critical for reviewing real-time complaints and discovering missing or unsatisfactory situations. The findings of this study reveal how the leading cargo companies in Türkiye manage the pandemic within the scope of the sample data. If cargo companies want to achieve customer satisfaction, receiving fewer complaints is the best method to be accomplished. They should focus on removing their shortcomings to be more successful and observe their competitors' activities by reviewing complaints.

When the research findings are compared with the relevant literature, Burucuoglu and Yazar's [20] findings show high similarities in consideration of the topic called delivery/distribution, which includes; the issues of "you were not at the address" note, the return of the cargo without the knowledge of the customer, delivery to the wrong address, no delivery to the address, no delivery, late delivery, and delivery to the wrong person is considered as one of the most intense complaints of the customers that revealed. Güven [11], Kocabaş [22], and Tosun & Gürce's [21] studies partially show similarities with our findings, such as delayed or not delivered parcels, parcels not received or delivered, and returns processes. The issues related to hygiene rules revealed the complaint topic different from the relevant literature considering cargo services in our study.

Within the scope of the research, it was seen that the word-based labeling method estimated a total of 100,524 complaints, while the survey-based method estimated a total of 104,313 complaints. When we examined it for accuracy, the word-based label assignment method was found to be 92%, whereas the survey-based method was 87%. Although the survey-based labeling method predicts many more complaints, the prediction accuracy rate is decreased. The accuracy rates of each method are shown in Table 7.

Table 7 Accuracy Rates of Methods

Word-Based					Survey-Based				
TfidfVectorizer					TfidfVectorizer				
Prediction					Prediction				
No Yes Total					No Yes Total				
Actual	No	11059	519	11578	Actual	No	6950	252	7202
	Yes	1075	5341	6416		Yes	1467	2653	4120
Total 12134 5860 17994					Total 8417 2905 11322				
Accuracy 91%					Accuracy 84%				
HashingVectorizer					HashingVectorizer				
Prediction					Prediction				
No Yes Total					No Yes Total				
Actual	No	10356	879	11235	Actual	No	6339	553	6892
	Yes	1778	4981	6759		Yes	2078	2352	4430
Total 12134 5860 17994					Total 8417 2905 11322				
Accuracy 85%					Accuracy 76%				
CountVectorizer					CountVectorizer				
Prediction					Prediction				
No Yes Total					No Yes Total				
Actual	No	12113	8	12121	Actual	No	8342	0	8342
	Yes	21	5852	5873		Yes	75	2905	2980
Total 12134 5860 17994					Total 8417 2905 11322				
Accuracy 99%					Accuracy 99%				

The CountVectorizer vectoring method has the highest accuracy rate of 99% in both labeling approaches. This result is believed to be related to the method's operating basis. The CountVectorizer vectoring approach makes vectorization by counting the words in the document. Since the training and test data were comprised of the same samples throughout the testing of the training data, it is likely that the approach discovered identical word counts on the same samples and, as a result, made highly accurate predictions. It should be noted that this result may be misleading.

The period with the highest number of complaints followed a period in which the number of Covid-19 cases increased in Türkiye dramatically for the first time. During this period, the Turkish Ministry of Health requested that citizens stay home. The main reason for the increase in complaints is the growing number of customers who prefer to order via the internet rather than traditional methods and that most cargo companies were caught unprepared for this demand. As a result, cargo companies should need to anticipate future demand and create a variety of operational approaches in the case of similar scenarios in the future.

In general, it is seen that most complaints are experienced during the delivery of the parcels, which is the number of delayed cases is relatively high. In order to reduce the number of complaints about this theme, cargo businesses must increase the number of branches in proportion to the increase in business volume and employees.

Another finding based on the research findings is that branch staff, and customer service professionals experience a lack of control in the face of increased business volume, just like field personnel. Customers have complained about the carelessness of customer service and the lack of consideration for their requests (for example, delivery in the branch, although customers were requesting delivery at the door). Undoubtedly, this research field should be handled with an interdisciplinary approach, as in this study, and the underlying causes of the complaints should be dealt with in more detail. For example, from the perspective of the management or human resources discipline, it can be said that the perception of burnout in the employees who were recruited with insufficient training due to the workload may be an important reason for these complaints. In this scenario, the cargo companies devoting emphasis to the in-company training and considering the aspects that motivate their personnel may significantly minimize the number of complaints. According to recent studies [11], [12] done during the pandemic, the number of orders on e-commerce sites has steadily grown, and logistics problems have become more challenging. In this sense, our study findings are similar to the literature studies.

## 6. Limitation and Future Studies

Several limitations must be considered in evaluating the findings of this study. The first limitation is that the data used in this study was obtained only from a single web page. Using data from various platforms may reveal different outcomes. The second limitation of the study is that the complainants' demographic characteristics are unknown, and it cannot be determined whether they have written more than one complaint about the same problems or whether the complaints written are genuine. The third limitation is about methods. In the survey-based labeling method, the accuracy of the participants' answers could not be checked.

Furthermore, the study solely used the logistic regression method to make the prediction. In this context, using different methods will be beneficial for the relevant literature, as it will provide the chance to make a comparison in future studies. Future studies in this field may improve predicted consistency by increasing the amount of learning data and employing various methods for estimating word weights or implementing different classification algorithms.

## References

- [1] V. Surveillances, “The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (COVID-19) in China,” *Zhonghua Liu Xing Bing Xue Za Zhi*, vol. 41, no. 2, pp. 145–151, 2020, doi: 10.3760/cma.j.issn.0254-6450.2020.02.003.
- [2] Z. Y. Zu et al., “Coronavirus Disease 2019 (COVID-19): A Perspective from China,” *Radiology*, vol. 296, no. 2, pp. E15–E25, 2020, doi: 10.1148/radiol.2020200490.
- [3] F. Budak and Ş. Korkmaz, “Covid-19 Pandemi Sürecine Yönelik Genel Bir Değerlendirme: Türkiye Örneği,” *Sos. Araştırmalar ve Yönetim Derg.*, no. 1, pp. 62–79, May 2020, doi: 10.35375/sayod.738657.
- [4] GoogleNews, “Koronavirüs (COVID-19),” 2022. <https://news.google.com/covid19/map?hl=tr&mid=%2Fm%2F02j71&gl=TR&ceid=TR%3Atr>.
- [5] M. Bulut, “Analysis of The Covid-19 Impact on Electricity Consumption and Production,” *Sak. Univ. J. Comput. Inf. Sci.*, vol. 3, no. 3, 2020, doi: 10.35377/saucis.03.03.817595.
- [6] B. Kaya and A. Günay, “Twitter Sentiment Analysis Based on Daily Covid-19 Table in Turkey,” *Sak. Univ. J. Comput. Inf. Sci.*, vol. 4, no. 3, 2021, doi: 10.35377/saucis...932620.
- [7] L. Çallı and B. A. Çallı, “Covid-19 Aşı Tereddütüne Sahip Hekimlerin Gizli Dirichlet Ayrımı (GDA) Algoritmasıyla Twitter Paylaşımlarının Konu Modellemesi,” in *8th International Management Information Systems Conference*, 2021, no. October, pp. 91–103.
- [8] V. Alfonso, C. Boar, J. Frost, L. Gambacorta, and J. Liu, “E-Commerce in the Pandemic and Beyond,” *BIS Bull.*, vol. 220, no. 44, 2021, [Online]. Available: <https://ideas.repec.org/p/bis/bisblt/36.html>.
- [9] E-TicaretBilgiPlatformu, “2020 Yılı İstatistikleri (Ocak - Aralık),” 2021. <https://www.eticaret.gov.tr/istatistikler>.
- [10] ETBİS, “Elektronik Ticaret Bilgi Sistemi (Etbis) 2021 Yılı Verileri,” 2022. [https://www.eticaret.gov.tr/dnnqthgzvawtdxraybsaacxtymawm/content/FileManager/Dosyalar/2021 Yılı E-Ticaret Bülteni.pdf](https://www.eticaret.gov.tr/dnnqthgzvawtdxraybsaacxtymawm/content/FileManager/Dosyalar/2021%20Yılı%20E-Ticaret%20Bülteni.pdf).
- [11] H. Güven, “Covid-19 Sürecinde E-Ticaret Sitelerine Yöneltilen Müşteri Şikâyetlerinin İncelenmesi,” *J. Turkish Stud.*, vol. Volume 15, no. Volume 15 Issue 4, pp. 511–530, 2020, doi: 10.7827/turkishstudies.44354.
- [12] A. Parlaklıç, M. Üzmez, and S. Mertoğlu, “How Does Covid-19 Pandemic Effect Online Shopping in E-Commerce?,” *J. Bus. Digit. Age*, vol. 3, no. 2, pp. 117–122, 2020, doi: 10.46238/jobda.823955.

- [13] Etid, “EY Parthenon & ETİD COVID 19 Yönetici ve KOBİ Anketleri,” 2020. [Online]. Available: [https://assets.ey.com/content/dam/ey-sites/ey-com/tr\\_tr/pdf/2020/07/ey-turkiye-parthenon--etid--covid-19-anketleri.pdf](https://assets.ey.com/content/dam/ey-sites/ey-com/tr_tr/pdf/2020/07/ey-turkiye-parthenon--etid--covid-19-anketleri.pdf).
- [14] L. Jean Harrison-Walker, “E-complaining: A content analysis of an Internet complaint forum,” 2001.
- [15] D. Istanbuluoglu, S. Leek, and I. T. Szmigin, “Beyond exit and voice: developing an integrated taxonomy of consumer complaining behaviour,” *Eur. J. Mark.*, vol. 51, no. 5–6, pp. 1109–1128, 2017, doi: 10.1108/EJM-04-2016-0204.
- [16] L. Çallı and F. Çallı, “Understanding Airline Passengers during Covid-19 Outbreak to Improve Service Quality: Topic Modeling Approach to Complaints with Latent Dirichlet Allocation Algorithm,” *Transp. Res. Rec. J. Transp. Res. Board*, p. 036119812211120, 2022, doi: 10.1177/03611981221112096.
- [17] sikayetvar.com, “About Us,” 2021. <https://www.sikayetvar.com/hakkimizda>.
- [18] D. Gündüz Hoşgör and H. Hoşgör, “Sağlık Hizmeti Tüketicileri Perspektifinden Özel Hastane Şikâyetlerinin İncelenmesi (Sikayetvar.Com Örneği),” *Hacettepe Sağlık İdaresi Derg.*, vol. 22, no. 4, pp. 823–842, 2019, [Online]. Available: <https://orcid.org/0000-0002-1377-4617>.
- [19] H. N. Güler, “Koronavirüsü (COVID-19) Günlerinde Bankalara İletilen Müşteri İtiraz ve Şikâyetlerinin İncelenmesi,” *Avrasya Sos. ve Ekon. Araştırmaları Derg.*, vol. 7, no. 4, pp. 85–99, 2020.
- [20] M. Burucuoğlu and E. E. Yazar, “Üçüncü Parti Platformda Kargo Firmalarına Yapılan Müşteri Şikâyetlerinin İçerik Analizi,” *Ekon. ve Sos. Araştırmalar Derg.*, vol. 16, no. 1, pp. 99–114, 2020.
- [21] M. Y. Gürce and P. Tosun, “Kargo Hizmetlerine İlişkin Müşteri Şikâyetleri: Bir İçerik Analizi,” *J. Bus. Res. - Turk*, vol. 3, no. 9, pp. 177–196, 2017, doi: 10.20491/isarder.2017.294.
- [22] İ. Kocabaş, “Covid- 19 Döneminde E - Şikâyet Yönetimi Perspektifinden Müşterilerin Çevrimiçi Alışverişte Karşılaştıkları Sorunlar,” *Selçuk İletişim Derg.*, vol. 15, no. 1, pp. 323–359, 2022.
- [23] A. Deniz and L. Gödekmerdan, “Müşterilerin Kargo Firmalarının Sunduğu Hizmetlere Yönelik Tutum ve Düşünceleri Üzerine Bir Araştırma,” *Atatürk Üniversitesi Sos. Bilim. Enstitüsü Derg.*, vol. 15, no. 2, pp. 379–396, 2011.
- [24] S. Kapıkıran, F. Öztürk, and E. Akkan, “Kargo Hizmetlerine Yönelik Hizmet Hatası Seviyesi , Hizmet Telafisi ve Tatminin Müşteri Sadakati Üzerindeki Etkisi Belirlemeye Yönelik Pilot,” pp. 0–2, 2021.
- [25] A. H. Özyayın, S. Çelikkaya, and G. Duran, “Kargo Hizmetlerinin Tüketici Davranışlarına Etkisi Üzerine Bir Çalışma: Süleyman Demirel Üniversitesi Örneği,” *Enderun Derg.*, vol. 3, no. 2, pp. 86–97, 2019.
- [26] Y. Cho, I. IM, R. Hiltz, and J. Fjermestad, “An analysis of online customer complaints: Implications for Web complaint management,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2002-Janua, no. c, pp. 2308–2317, 2002, doi: 10.1109/HICSS.2002.994162.
- [27] M. N. Alabay, “Müşteri Şikâyetleri Yönetimi,” *Uluslararası Yönetim İktisat ve İşletme Derg.*, vol. 8, no. 16, 2012.
- [28] S. Ray, “A Quick Review of Machine Learning Algorithms,” in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects, COMITCon 2019*, 2019, pp. 35–39, doi: 10.1109/COMITCon.2019.8862451.
- [29] I. El Naqa, R. Li, and M. J. Murphy, *Machine Learning in Radiation Oncology*. Cham: Springer International Publishing, 2015.

- [30] J. Castañón, “10 Machine Learning Methods that Every Data Scientist Should Know,” 2019. <https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9>.
- [31] T. Kwartler, “What is Text Mining?,” in *Text Mining in Practice with R*, 2017, pp. 1–15.
- [32] L. Çalli, F. Çalli, and B. Alma Çalli, “Yönetim Bilişim Sistemleri Disiplininde Hazırlanan Lisansüstü Tezlerin Gizli Dirichlet Ayırımı Algoritmasıyla Konu Modellemesi,” *MANAS Sos. Araştırmalar Derg.*, pp. 2355–2372, 2021, doi: 10.33206/mjss.894809.
- [33] M. Roukalova, “Text Mining vs. Natural Language Processing,” *Scion Analytics*, 2021. <https://scionanalytics.com/text-mining-vs-natural-language-processing>.
- [34] A.-H. Tan, “Text Mining: The state of the art and the challenges,” *Proc. PAKDD 1999 Work. Knowl. Discovery from Adv. Databases*, vol. 8, pp. 65–70, 1999, doi: 10.1.1.38.7672.
- [35] S. Kazan and H. Karakoca, “Product Category Classification with Machine Learning,” *Sak. Univ. J. Comput. Inf. Sci.*, vol. 2, no. 1, pp. 18–27, 2019, doi: 10.35377/saucis.02.01.523139.
- [36] A. C. P. L. F. de Carvalho and A. A. Freitas, “A Tutorial on Multi-label Classification Techniques,” 2009, pp. 177–195.
- [37] P. Szymanski and T. Kajdanowicz, “Scikit-multilearn: A python library for multi-label classification,” *J. Mach. Learn. Res.*, vol. 20, no. 6, pp. 1–22, 2019.
- [38] A. K. Singh and M. Shashi, “Vectorization of text documents for identifying unifiable news articles,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 305–310, 2019, doi: 10.14569/ijacsa.2019.0100742.
- [39] A. Aizawa, “An information-theoretic perspective of tf-idf measures,” *Inf. Process. Manag.*, vol. 39, no. 1, pp. 45–65, Jan. 2003, doi: 10.1016/S0306-4573(02)00021-3.
- [40] W. Zhang, T. Yoshida, and X. Tang, “A comparative study of TF\*IDF, LSI and multi-words for text classification,” *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, 2011, doi: 10.1016/j.eswa.2010.08.066.
- [41] S. Kaur, P. Kumar, and P. Kumaraguru, “Automating fake news detection system using multi-level voting model,” *Soft Comput.*, vol. 24, no. 12, pp. 9049–9069, 2020, doi: 10.1007/s00500-019-04436-y.
- [42] J. Deepakumara, H. M. Heys, and R. Venkatesan, “FPGA implementation of MD5 hash algorithm,” *Can. Conf. Electr. Comput. Eng.*, vol. 2, pp. 919–924, 2001, doi: 10.1109/ccece.2001.933564.
- [43] J. Brownlee, “One-vs-Rest and One-vs-One for Multi-Class Classification,” *Machine Learning Mastery*, 2020. <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/>.
- [44] TurkishMinistryofHealth, “COVID-19 Bilgilendirme Platformu,” 2022. <https://covid19.saglik.gov.tr/>.

# Pseudo-Supervised Defect Detection Using Robust Deep Convolutional Autoencoders

 Mahmut Nedim Alpdemir

TÜBİTAK, Informatics and Information Security Research Center (BİLGEM); nedim.alpdemir@tubitak.gov.tr;

Received 29 October 2022; Accepted 22 November 2022; Published online 31 December 2022

## Abstract

Robust Autoencoders separate the input image into a *Signal(L)* and a *Noise(S)* part which, intuitively speaking, roughly corresponds to a more stable background scene (L) and an undesired anomaly (or defect) (S). This property of the method provides a convenient theoretical basis for divorcing intermittent anomalies that happen to clutter a relatively consistent background image. In this paper, we illustrate the use of Robust Deep Convolutional Autoencoders (RDCAE) for defect detection, via a pseudo-supervised training process. Our method introduces synthetic simulated defects (or structured noise) to the training process, that alleviates the scarcity of true (real-life) anomalous samples. As such, we offer a pseudo-supervised training process to devise a well-defined mechanism for deciding that the *defect-normal discrimination* capability of the autoencoders has reached to an acceptable point at training time. The experiment results illustrate that pseudo supervised Robust Deep Convolutional Autoencoders are very effective in identifying surface defects in an efficient way, compared to state of the art anomaly detection methods.

**Keywords:** robust autoencoders, anomaly detection, defect detection, machine learning, convolutional neural networks

## 1. Introduction

Detecting data samples with deviating features compared to a set of examples deemed as "normals", constitutes a major research area. Research communities with varying precedence and focus employ different names for this problem, such as outlier detection, novelty detection, anomaly detection, defect detection, noise detection, deviation detection or exception mining. Despite the variation in naming, the fundamental problem is to define a region in the feature space that represents "the normal" for a data set, and subsequently identify all cases that lie outside the boundaries of that region. The application areas of the problem include fraud detection, structural defect detection, intrusion detection, time-series monitoring, loan application processing, medical condition monitoring, motion segmentation, detecting novelty in text etc. [1]. The solution space to the problem has been explored by different communities resulting in a partitioning along several axes, reflecting approaches, methodologies and tools adopted by those communities. Many surveys on the subject probe into different methods, techniques and approaches employed to solve the problem [1, 2, 3]. As more recent developments in *Deep Learning* [4] offer promising results in extracting relevant features in an automated way, in particular for computer vision applications [5], more recent surveys such as [6] and [7] provide a deep learning centric account of the subject.

Anomaly detection problem is relatively difficult to solve in general. Therefore, most of the techniques in the literature tend to solve a specific instance of the general problem based on the type of application, the type of input data and model, the availability of labels for the training data, and the type of anomalies. The problem of defect detection in flat surfaces, where scarcity of defect samples is a common issue, is a good example of a research domain that stands to benefit from an anomaly detection perspective. Scarcity of abnormal data is something that promotes the use of anomaly detection as a candidate solution, since anomaly detection methods rely only on normal (as apposed to abnormal or anomalous) samples at training phase. In this domain, it is also important to increase the accuracy, recall and precision of the detection process to ensure the applicability of the method for complex real life

problems in industrial settings. So methods that avoid dependency on abundant defect samples whilst improving the detection performance are of particular interest.

Recently, a specific form of neural networks, namely an *Autoencoder (AE)* [8], has attracted the attention of researchers from different domains, due to its ability to learn an efficient, compressed representation of its input via its encoder part and reconstruct this input via its decoder part (see for instance [9] and [10]). The primary motivation for those researchers has been to exploit this capability to learn the features characterizing the normal samples and later use the error generated by the difference between the input and the reconstructed output to identify the anomalous samples. AEs, come in different flavors, such as Convolutional AE (CAE), Variational AE (VAE) and Robust AE (RAE) to name a few, each introducing an additional capability on top the central ability mentioned above. A variant of AEs called Robust Convolutional Autoencoder(RCAE) [11, 12] appears to be a particularly promising solution for two reasons:

- First, Convolutional Autoencoders(CAE), in general, combine the good aspects of AEs and Convolutional Neural Networks (CNN). As stated above, an AE is known for its talent to learn a low level compressed representation of a normal class via minimization of the reconstruction error through its encoding and decoding layers. A CNN on the other hand preserves the spatial locality [13] of important features and this is very important for 2D images since defects are locally positioned in an image.
- Second, *Robust* AEs, in particular, separate the input image into a *Signal(L)* and a *Noise(S)* part which, intuitively speaking, roughly corresponds to a more stable background scene (L) and an undesired anomaly (or defect) (S) for image based applications. This property of the method provides a convenient theoretical basis for divorcing intermittent anomalies that happen to clutter a relatively consistent background image. There is a large family of automated visual quality inspection applications that can benefit from such anomaly detection capability.

As described by [11], use of the *Robust AE* by feeding normal and anomalous samples during training and letting the RAE differentiate the anomalies as noise at the end of an iterative process, is relatively straightforward. However, this requires the availability of abnormal samples at training time and does not accommodate subsequent (post-training) anomalous instance identification. Training the RAE using only normal samples to specify a threshold based on a learned reconstruction score of the normal samples and later use that threshold to detect unseen anomalies is what we are interested in. This process is named as *inductive anomaly detection* by [12] and can be challenging when RAE struggles to learn the distribution of the normal data to a sufficient degree that it can differentiate out of distribution (OOD) samples correctly. This may be due to two problems:

1. the normal samples used for training may not be sufficiently representative (both in terms of quality and number), or
2. the training architecture and process may not be vigorous enough to cater for signal-noise separation in the absence of some representative abnormal samples (i.e. qualitative noise).

In this paper, we primarily target the improvement of the latter of the problems mentioned above. Our overall contribution is twofold:

1. We illustrate the viability of *Robust Deep Convolutional AEs* as an efficient solution for surface defect detection utilizing two relatively recent industrial quality datasets.
2. We propose a method to further enhance this solution, by introducing *synthetic simulated defects (or structured noise)* to the training process in a novel way, so that a safe discriminating threshold can be determined by relying on a more robust convergence criteria during training iterations, in the absence of true (real life) abnormal samples. As such, we offer a specific form of *pseudo-supervised* training process as a well defined mechanism to alleviate the dependency of RAEs to true abnormal samples.

The rest of the paper is organized as follows: Section 2 provides some theoretical background and related work, Section 3 introduces the details of our methodology, Section 4 provides information on

the structure of the experiments, Section 5 presents results and discussions and finally Section 6 provides some concluding remarks.

## 2. Background and Related Work

The problem of defect detection in textured surfaces is a good example of a research domain that stands to benefit from an anomaly detection perspective and therefore it is selected as the target case study for our method. Anomaly detection context, re-casts the defect identification problem into the problem of recognizing the *divergence from the normal* with reference to distinct features characterizing the normal. As such, relatively regular and uniform patterns that characterize a non-defected (normal) surface, provide a convenient referential basis for learning-based anomaly detection approaches. Traditionally, automated defect detection literature tend to categorize the solution methods into several groups including structural, statistical, spectral, model-based, learning-based and hybrid methods. [14, 15] and [16], for instance, provide a comprehensive account of these methods including a comparative study of their detection and classification performance. The most important aspect of the classical methods is that they apply a processing pipeline to an image containing the surface to be inspected, that starts with low level image processing techniques such as filtering, transformation, distribution identification etc., continues with feature extraction and defect detection, and finally terminates with defect classification. More recent surveys such as [17] focusing on textile domain and [18] offering a more diverse view of industrial surface inspection applications, place greater emphasis on learning-based approaches and in particular on deep-learning. Both acknowledge that deep learning methods simplify the aforementioned processing pipeline since they automate feature extraction to a greater extent, but also note that they may require abundant and quality data samples for effective training of the classifiers. This latest trend in fabric defect detection research towards learning-based methods, suggests that the techniques and methods that are mostly developed by the machine learning community, are well placed to capitalize on.

In that respect, there are several recent techniques and methods in the literature that are notably promising for anomaly detection cases that suffer from scarcity of abnormal samples. These are AEs [8], One-Class Support Vector Machines (OC-SVM) [19], Isolation Forests (IF) [20, 21], One-Class Support Vector Data Description (OC-SVDD) [22, 23] and One-Class Neural Networks (OC-NN) [24]. Earlier anomaly detection methods such as the One-Class SVM (OC-SVM) or Kernel Density Estimation (KDE) [25] are known to rely on tractable feature spaces with moderate dimensions and are prone to failure in cases involving large scale, complex data manipulation due to curse of dimensionality. More novel neural network based solutions such as Deep Convolutional Autoencoders(DCAE) [26, 27] and Deep One-Class Neural Networks [28, 29] have been on the agenda of contemporary research efforts, introducing some improvements that alleviate the deficiencies of the earlier methods.

As stated in the introduction, Robust Convolutional Autoencoders(RCAE), in particular, exhibit distinctly useful behaviors since they combine the ability to learn a highly efficient, locality preserving and non-linear representation of their input, with the ability to progressively learn to separate signal (normal surface) from the noise (defects). These convenient properties form the rationale for our adoption of the RCAE as a viable solution. From a methodological point of view, the most relevant work in the literature to the work presented in this paper is that of [11]. The authors augment an AE with a filter layer that culls out the anomalous parts of the input data,  $X$ , that are difficult to reconstruct. They then propose that the remaining portion of the data,  $S$  can be represented by the low-dimensional hidden layer,  $L_D$ , with small reconstruction error. The problem of finding anomalies is then cast into the following optimization problem:

$$\text{Min}_{\theta, S} = \| L_D - D_{\theta} (E_{\theta} (L_D)) \|_2 + \lambda \| S \|_{2,1}; \quad s. t. X - L_D - S = 0 \quad (1)$$

Here the input data  $X$  is split into two parts,  $L_D$  and  $S$ .  $L_D$  is the input to an AE  $D_{\theta} (E_{\theta} (L_D))$  and the AE is trained by minimizing the reconstruction error  $\| L_D - D_{\theta} (E_{\theta} (L_D)) \|_2$  through back-



propagation.  $S$ , on the other hand, contains outliers which are difficult to represent using the AE. We use this general framework for the formulation of the problem at hand but there are some differences between their approach and ours, and also some improvements provided by our work that deserve mentioning:

- They provide two distinct regularization methods one targeting denoising and the other targeting anomaly detection. We only aim at anomaly detection (or more specifically defect detection), so only  $l_{2,1}$  regularization is applied (as opposed  $l_1$  regularization which is used for denoising).
- They require true anomalous samples (i.e. real samples labeled as defects) for tuning the hyperparameter controlling noise-signal separation (i.e.  $\lambda$  in the equation above). As such, part of the training has to be done in a semi-supervised manner. On the contrary, we perform the same hyper parameter tuning using synthetic defects leading to a pseudo-supervised training procedure. This divorces the training process from the dependency on labeled, true anomalous samples.
- Their convergence control logic relies on the use of real defected samples and depend on the result of two conditional inquiry: 1 - check if  $X - (L_D + S) < \epsilon$  and 2 - check if  $L_D$  and  $S$  have converged to a fixed point (i.e. there is no a significant change any more). Whereas, we use synthetic defects in our training process and check the Area Under ROC Curve (AUC) score obtained by testing the AE performance in separating the (synthetic) defected samples from the normal samples. When the training ends we also obtain an outlier-threshold based on the reconstruction error that characterizes the normal samples (i.e. during actual testing, instances that produce a reconstruction error below that threshold are identified as normal).

Before delving into the details of our methodology in the next section, we proceed by providing some background information on DCAE and RAE in the following subsections.

## 2.1 Deep Convolutional Autoencoders (DCAE)

An AE is known for its ability to compress its input into an efficient feature representation via its encoding part and then reconstruct it via its decoding part. In the middle of the two parts lies its bottleneck layer (also known as latent space) where the input is encoded into an efficient, much lower dimensional feature space. The encoder and decoder parts can be defined as transitions  $E$  and  $D$ , such that:

$$\begin{aligned} E: \mathcal{X} &\rightarrow \mathcal{F} \\ D: \mathcal{F} &\rightarrow \mathcal{X} \\ \mathbf{x}' &= \|\mathbf{x} - D(E(\mathbf{x}))\| \end{aligned} \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^d = \mathcal{X}$  refers to an input in the  $\mathcal{X}$  domain and  $\mathbf{x}'$  denotes the reconstructed input. The hidden bottleneck layer, then, can be represented by  $\mathbf{z} = E(\mathbf{x}) \in \mathbb{R}^p = \mathcal{F}$  in the  $\mathcal{F}$  domain. In the most popular form of AEs,  $D$  and  $E$  are neural networks. In the special case that  $D$  and  $E$  are linear operations, we get a linear AE, where we would achieve the same latent representation as Principal Component Analysis (PCA) [30]. Therefore, an AE is in fact a generalization of PCA, where instead of finding a low dimensional hyperplane in which the data lies, it is able to learn a non-linear manifold [8]. In particular, an AE can be viewed as a solution to the following optimization problem:

$$\min_{D,E} \|\mathbf{x} - D(E(\mathbf{x}))\| \quad (3)$$

Where  $\|\cdot\|$  is usually the  $l_2$  norm. CAEs differ from conventional AEs in that their architecture contains an encoder part with convolutional and pooling layers, and an analogous decoder part with deconvolutional and upsampling layers. As such, recalling that  $\mathbf{x}'$  denotes the reconstruction, the encoder and decoder processes can be expanded as:

$$\begin{aligned} \mathbf{z} &= E(W \circ \mathbf{x} + b) \\ \mathbf{x}' &= D(W' \circ \mathbf{z} + b') \end{aligned} \quad (4)$$

where " $\circ$ " is the convolution process;  $W$  and  $W'$  are the weight matrices;  $b$  and  $b'$  are the bias vectors for the encoder and decoder, respectively; and  $E$  and  $D$  are the nonlinear mapping processes,

specifically, the convolutional, pooling, deconvolutional, and upsampling processes. Particularly, the pooling and upsampling processes are usually conducted in the form of max pooling and max unpooling.

## 2.2 Robust Autoencoders (RAE)

Robust AEs are built on a theoretical basis borrowed from Robust Principle Component Analysis (RPCA) [31, 32]. Specifically, RPCA splits a data matrix  $X$  into a low-rank matrix  $L$  and a sparse matrix  $S$  such that;

$$X = L + S \quad (5)$$

where the matrix  $L$  contains a low-dimensional representation of  $X$  and the matrix  $S$  consists of element-wise outliers, which can not be efficiently captured by the low-dimensional representation  $L$ . Similar to RPCA, a Robust Deep Autoencoder also splits input data  $X$  into two parts;

$$X = L_D + S \quad (6)$$

where  $L_D$  represents the part of the input data that is well represented by the hidden layer of the AE, and  $S$  contains noise and outliers which are difficult to reconstruct. So the idea is that, just as in RPCA, by iteratively separating out the noise and outliers from  $X$  into  $S$ , the remaining data  $L_D$  can be accurately reconstructed by an AE. As such, RAE combines non-linear representation capabilities of AEs with the anomaly detection capabilities of RPCA. The peculiar behavior of the AE that is conveniently exploited in reconstruction-based anomaly detection in a general context is that noise and outliers are essentially difficult to compress and therefore cannot effectively be projected to a low-dimensional hidden layer. So, if those outliers could be incorporated into the AE loss function in an appropriate way, then the low-dimensional hidden layer could provide accurate reconstruction, except for those few outliers [11].

## 3. Methodology

Our method introduces synthetic simulated defects (or structured noise) to the training process, so that a safe discriminating threshold can be determined by relying on a more robust convergence criteria during training iterations, in the absence of true (real life) abnormal samples. As such, we offer a pseudo-supervised training process to devise a well-defined mechanism for deciding that the defect-normal discrimination capability of the AE has reached to an acceptable point at training time.

Using noisy inputs in the training of the robust AEs has been adopted by other researchers. For instance in [33] white Gaussian random noise is used to simulate anomalous samples so that the AE can learn the distribution generating normal samples more efficiently. The main difference of our method is to use a more complex model (i.e. structured noise) for anomalies. Another example is the use of random noise in denoising AEs [34]. A denoising AE is a stochastic extension to classic AE where the AE is forced to learn the reconstruction of input given its noisy version, usually using a stochastic corruption process to randomly set some of the inputs to zero. It is important to note that, the use of structured noise in our case is categorically different from this type of noise utilization. In contrast to denoising AEs, we force the AE to separate common, stable features from the anomalous ones. As it will be elaborated on in Section 5.1, our findings show that, for such cases, the incorporation of structured noise (or synthetic defects) produces better results compared to injecting random noise into some of the normal samples. A similar approach, in terms of incorporating structured synthetic noise at training time is used in more recent works (see for instance [35] and [36]). Of these two work, [35] use an autoencoder equipped with skip connections and train it to reconstruct a clean image out of a synthetically corrupted version of it. To corrupt the training images they introduce a synthetic model, named Stain, that adds an irregular elliptic structure of variable color and size to the input image. Their main motivation for using synthetic anomalies is to alleviate a vulnerability of their skipped architecture that causes the network to perform identity mapping on uncorrupted clean images. In the work reported by [36], on the other hand, synthetic anomalies are generated using a sort of “confetti noise”, a simple noise model that inserts colored blobs into images and reflects the local nature of anomalies. Since their approach is semi-supervised, essentially they utilize the synthetic defects to model the incorporation of few known anomalies into the training process (in the absence of true anomalies), an approach which they report to be effective. Compared to our method, there are some differences, however, in the way both of these

approaches generate the synthetic defects as well as the exact purpose of and the final benefits obtained from using those synthetic defects:

1. Our defect generation model is based on a more complex, Julia set-based algorithm that facilitates modeling of pseudo-topological patterns that are well suited to structured fabric surfaces. The merits of modeling the defects in a structured way, compared to simple statistical noise, has been explored in this paper. However exploring the impact of modeling structured synthetic defects relatively more faithfully compared to other structured modeling approaches would require a thorough experimentation and analysis using different types of datasets, something we do not intend to endeavor in this particular paper. Intuitively speaking, our approach may stand to benefit from increased defect modeling fidelity for certain datasets by causing AE to learn normal/abnormal separation earlier and more effectively. A more extensive comparative analysis remains as an interesting future work.
2. The use of synthetic defects in our work is an integral part of the optimization process, both in terms of constituting an important ingredient of the noise component  $S$  in Equation 6 and also being an important facilitator for deciding on the outlier threshold during the training phase. Determining a threshold at training time with good discriminating power for testing phase is an important success factor for reconstruction based approaches and synthetic defects in our method play a critical role in managing this process.

Note that, in our case, synthetic defects are not used to train a classifier for defect classification, so extensive defect modeling is not needed here. Structured noise is used only to better exploit the signal-noise separation capability of the robust auto encoder for efficient and effective *defect detection*, not to ensure accurate *classification* of different defect types. In the following three subsections, the formulation of the optimization problem, the algorithm used to generate synthetic defects and the optimization algorithm are explained in more detail.

### 3.1 Formulation of the Optimization Problem

The optimization problem is formulated along similar lines to the one given by [11] for anomaly detection. Specifically;

$$\begin{aligned} \text{Min}_{\theta, S} &= \|L_D - D_{\theta}(E_{\theta}(L_D))\|_2 + \lambda \|S^T\|_{2,1} \\ \text{s. t. } &X - L_D - S = 0 \end{aligned} \quad (7)$$

where  $E_{\theta}(\cdot)$  denotes an encoder,  $D_{\theta}(\cdot)$  denotes a decoder, and  $S$  captures the anomalous data,  $L_D$  is a low dimension manifold and  $\lambda$  is a parameter that tunes the level of sparsity in  $S$ . Here the loss function for a given input  $X$  could be thought of as the ‘grouped  $l_{2,1}$  norm of  $S$ , balanced against the reconstruction error of  $L_D$ . The  $l_{2,1}$  norm of any  $X$  is defined as:

$$\|X\|_{2,1} = \sum_{j=1}^n \|x_j\|_2 = \sum_{j=1}^n \left( \sum_{i=1}^m |x_{ij}|^2 \right)^{\frac{1}{2}} \quad (8)$$

and, it can be viewed as inducing a 2 norm regularizer over members of each group and then a 1 norm regularizer between groups. In equation 7,  $\lambda$  plays an essential role in the defect and background separation. In particular, a small  $\lambda$  will encourage much of the data to be isolated into  $S$  as noise or outliers, and therefore minimize the reconstruction error for the AE. Similarly, a large  $\lambda$  will discourage data from being isolated into  $S$  as noise or outliers, and therefore increase the reconstruction error for the AE.

Note that here the  $l_{2,1}$  norm minimization problem can be implemented efficiently as a proximal problem as defined by [37] and adopted by [11], where the proximal operator is a block-wise soft-thresholding function.

### 3.2 Synthetic Defect Modeling

Algorithm 1 Pseudo Code for Julia Set Defect Generator

```

1  Procedure GenerateDefect ( $\alpha, \beta, \Delta x, \Delta y, cx, cy, zf, w, h$ )
2       $R =$  escape radius //such that  $R > 0, R^2 - R > \sqrt{cx^2 + cy^2}$ 
3      For each pixel  $(x, y)$  on the image of size  $(w, h)$ 
4
5           $zx = \frac{\alpha \left(x - \frac{w}{2}\right)}{0.5 * zf * w} + \Delta x$ 
6
7           $zy = \frac{\beta \left(y - \frac{h}{2}\right)}{0.5 * zf * h} + \Delta y$ 
8
9          max_iter = 255
10         i = max_iter
11         While  $zx^2 + zy^2 < R^2$  and  $i > \text{max\_iter}$ 
12              $zy = 2zx * zy + cy$ 
13              $zx = zx^2 + zy^2 + cx$ 
14              $i = i - 1$ 
15         End While
16         image[x,y] = i
17     End For
18 End Procedure

```

As mentioned earlier we are not using the synthetic defects for high fidelity defect simulation, so decoration of the normal fabric surface with defect-like structures are deemed sufficient. To this aim, we employ Julia set fractals [38]. Note that, other fractal pattern generation mechanisms can potentially be used for this purpose. However, Julia sets have a feature known as centro-symmetry [39] that facilitates modeling of some pseudo-topological patterns. As such, with proper parameter tuning it is possible to create certain Julia Set patterns that mimic some common defects in textured fabric surfaces, such as yarn tails, thick bars, holes, stains etc. Julia set fractals can be obtained by using a complex number  $z = x + yi$  where  $i^2 = -1$  and  $x$  and  $y$  are image pixel coordinates. The fractal is generated by repeatedly updating  $z$  using the formula  $z = z^2 + c$ , where  $c$  is another complex number that gives a specific Julia set. After numerous iterations, if the magnitude of  $z$  is less than a certain escape radius we say that pixel is in the Julia set and color it to generate desired patterns. Performing this calculation for a whole grid of pixels gives a fractal image.

We employ a parametric algorithm to generate different defect patterns. Algorithm 1 given above is controlled by nine parameters, where  $\alpha$  and  $\beta$  determine the extent (i.e. length) and alignment of the defect (i.e. horizontally extended, vertically extended, point-like),  $\Delta x, \Delta y$  are used as position offsets with respect to the center of the image;  $cx, cy$  are coefficients that determine the shape of the fractal pattern,  $zf$  is the zoom factor to determine coverage area of the defect, and finally  $w, h$  are width and height of the image to be generated.

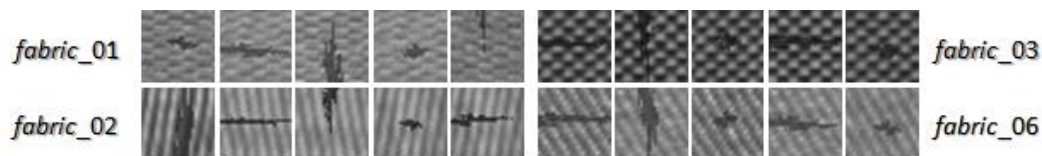


Figure 1 Examples of synthetic defects generated by decorating normal fabric samples. Various different types of defects (i.e. horizontal, vertical, point-like etc.) are shown using four different fabric type.

Figure 1 illustrates examples generated using four different fabric types in our dataset. The  $32 \times 32$  image patches are first extracted from non-defect fabric images and then they are decorated using the above algorithm. The original non-defect image patch and the  $32 \times 32$  matrix returned by the defect generator algorithm are blended by averaging the weighted pixel values to obtain a smoother overlay.

### 3.3 Optimization Algorithm

Algorithm 2 The pseudo code for our optimization procedure

1	Procedure Optimize ( $\alpha, \beta, \Delta x, \Delta y, cx, cy, zf, w, h$ )
2	Input $X$ // $X$ is a set of input images
3	$L_D = 0$ //init $L_D$ to 0 (same size as $X$ )
4	while upper-limit not reached
5	$L_D = X - S$ //Remove $S$ from $X$ , use $L_D$ to train the AE
6	$trainer.train(D(E(.)), L_D, epochnum)$ //Minimize recons. error using ADAM
7	$L_D = D(E(L_D))$ //Set $L_D$ to reconstruction from trained AE
8	$S = X - L_D$ //Set $S$ to be the difference between $X$ and $L_D$
9	$prox_{\lambda, l_{2,1}}(S^T)$ //Optimize $S^T$ using a proximal operator
10	$trainer.test(D(E(.)), X)$ //test $X$ for defect/non-defect separation
11	If AUC > 0.999 then
12	Break //convergence criteria met, so break
13	End If
14	End While
15	Return $L_D, S$ and outlier_treshold
16	End Procedure

Our optimization procedure (given in Algorithm 2) starts with reading the input  $X$  and initializing  $S$  and  $L_D$  to zero.  $X$  contains all the images used for training (i.e normal images and images with synthetic defects). Note that synthetic defects and normal samples are explicitly labeled (-1 and 0 respectively).  $X$  is a 4 dimensional tensor, so if the training set size is  $t$ , dimensions of  $X$  would be  $(t, w, h, c)$ , where  $w$  and  $h$  are width and height of the image in pixels; and  $c$  is the number of color channels of the images. The optimization loop starts by first setting  $L_D$  to  $X - S$ . Thus, in the first iteration  $L_D$  becomes equal to  $X$  since  $S$  is zero. Then the AE ( $D(E(.))$ ) is trained with the objective of minimizing the reconstruction error via the ADAM [40] optimization algorithm. The next step is to use the trained AE to get the reconstructed  $L_D$  set and assigning it onto itself, and later to set  $S$  to be the difference between  $X$  and  $L_D$ . So the purpose of lines 3.3 and 3.3 is to capture features that are easy to reconstruct in  $L_D$  and isolate the difference of the reconstructed and the original (which are supposed to be the features that are hard to reconstruct, and hence the noisy parts) in  $S$ . Then the second part of our optimization formulation given in Equation 7 (i.e.  $\lambda \| S^T \|_{2,1}$ ) is handled using a proximal operator [37, 11]. Finally we test for defect/ non-defect separation by checking the Area Under ROC Curve (AUC) score. If AUC is greater than 99.9%, then the algorithm returns  $L_D, S$  and an outlier\_treshold to be used for further defect identification. The outlier\_treshold is obtained by finding the highest 99.9% quantile of the set containing the reconstruction errors of non-defect (normal) samples. [11] use a different convergence criteria in their method using actual abnormal samples. They calculate the sum of  $L_D$  and  $S$  to see if the sum is close to the input  $X$ , and also they check if  $L_D$  and  $S$  have converged to a fixed point (i.e. there is no a significant change any more). Our AUC score calculation serves a similar purpose, but prioritizes the learning level of the AE. In both approaches it is necessary to put an upper limit on the iterations to avoid cases leading to futile convergence.

## 4. Experiments

### 4.1 The Experimental Setup

The architecture of our AE follows the style of a LeNet type CNN [41], where each convolutional module consists of a convolutional layer followed by leaky ReLU activation and  $2 \times 2$  max-pooling. Our CNN contains three modules,  $32 \times (5 \times 5 \times 3)$ -filters,  $64 \times (5 \times 5 \times 3)$ -filters, and  $128 \times (5 \times 5 \times 3)$ -filters, followed by a final dense layer of 256 units. This final dense layer implements the compressed latent representation. The batch size used in the experiments was 128 and weight decay hyper-parameter was set to  $\gamma = 10^{-6}$ . The workstation used had an i5 CPU with 6 cores and an NVIDIA GTX 1060 graphics card with 6GB RAM and 1280 GPU cores. We used PyTorch [42] as our main

machine learning framework. All of the neural networks and associated optimizers of our method are implemented in PyTorch with CUDA option enabled, to exploit the multi core capability of our GPU. We also utilized the popular python machine learning library scikit-learn [43] for generating the results of other methods we endorsed for comparison (see Section 5.3).

#### 4.2 The Datasets Used

We used two main datasets for the experiments: 1 - The AITEX Fabric Dataset is a recently introduced industrial quality image dataset targeted for fabric defect detection [44]. The dataset consists of 245 images of  $4096 \times 256$  pixels captured by the system of seven different fabric structures. The fabrics in the dataset are mainly plain, which is very convenient for illustrating the utility of our method, yet covers a reasonable range of fabric types. There are 140 defect free images in the database, sampled from 20 different types of fabric. The remaining 105 images are defected, containing 12 different types of fabric defects which commonly appear in the textile industry. 2 - The Kolektor Surface Defect Dataset 2 (KSDD2) is yet another recently introduced dataset that is constructed from images of defected production items that were provided and annotated by Kolektor Group d.o.o. [45]. The images were captured in a controlled industrial environment. The dataset consists of 356 images with visible defects and 2979 images without any defect, with image sizes of approximately  $230 \times 630$  pixels. The defected images contain several different types of defects (scratches, minor spots, surface imperfections, etc.). Both of the datasets contain a number of *mask images* each of which corresponds to a unique image containing one or more defects. The mask image is a black-and-white image depicting the exact pixel wise location of a defect, in an unambiguous way. This enabled us to generate labels for defected and defect-free patches of arbitrary sizes, in an automated and deterministic way.

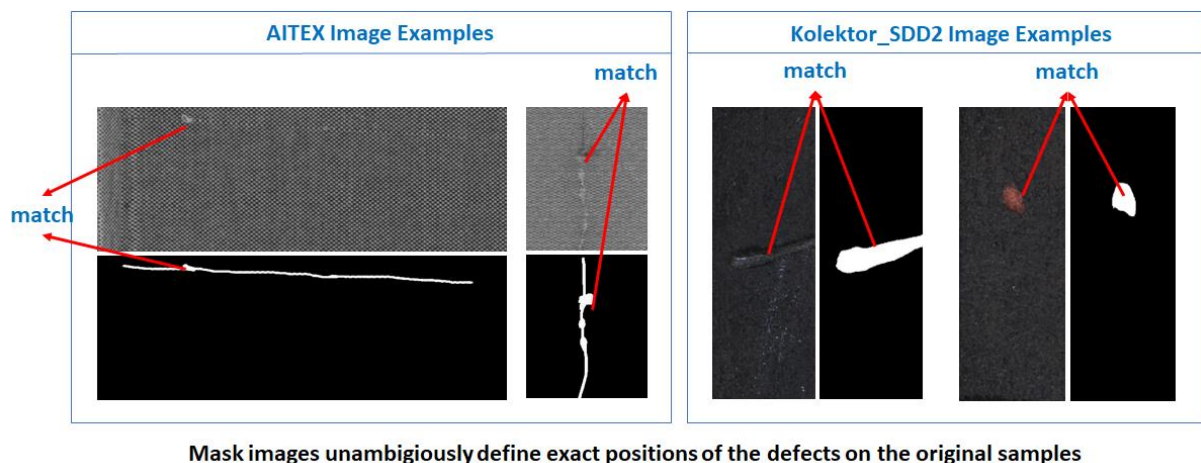


Figure 2 Examples of original samples and mask images from AITEX and KSDD2 datasets. Note that mask images pinpoint the position of defects

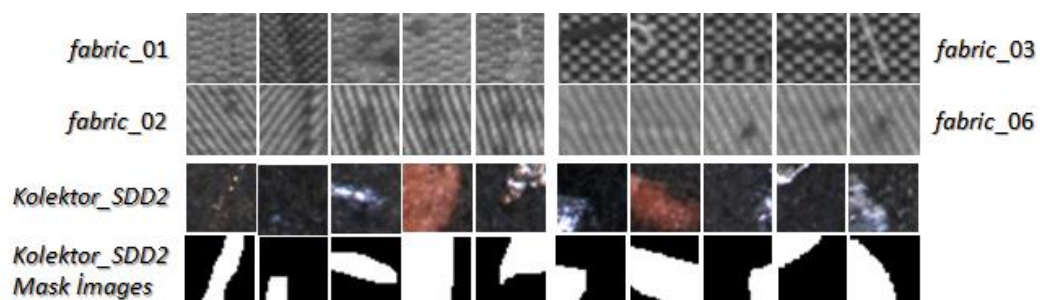


Figure 3 Examples of actual defect samples of size  $32 \times 32$  pixel each, generated using both defect images and corresponding mask images provided by the AITEX and KSDD2 datasets.

Thus, we implemented a custom sample generator class that can be configured to generate defect free and defected patches of different sizes (e.g.  $32 \times 32$ ,  $64 \times 64$  pixels etc) using the original images contained in both of the datasets. The custom generator is also able to decorate the samples with synthetic structural defects or random noise (e.g. Gaussian, Poisson etc.) with a pre-defined ratio of the total training set. We assume that in a typical industrial setting an image scan camera would send image frames to a processing computer at a certain rate, and that the processing unit would generate small patches ( $32 \times 32$  pixel in our case) to be fed into the defect detector to facilitate easier localization of defects. So an input image to our neural network is a  $32x \times 32 \times 3$  matrix (3 at the end is for RGB color channels). Figure 2 illustrates the original and the mask image examples from both datasets, and Figure 3 shows  $32 \times 32$  patches generated from images containing defects, and some corresponding mask image patches.

### 4.3 Performance Evaluation Method

Table 1 AUC, F1 Score and Accuracy results for different lambda values.

Dataset	$\lambda$	Noise Ratio	Noise Type	RDCAE AUC	F1 Score	Accuracy
<i>fabric_00</i>	1.0	3	<i>struct.</i>	$0.993 \pm 0.005$	$0.208 \pm 0.006$	$0.135 \pm 0.032$
	1.55	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.994 \pm 0.009$	$0.999 \pm 0.002$
	1.6	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.998 \pm 0.004$	$1.000 \pm 0.001$
	1.65	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.991 \pm 0.008$	$0.998 \pm 0.002$
	2.5	3	<i>struct.</i>	$0.992 \pm 0.011$	$0.936 \pm 0.069$	$0.987 \pm 0.013$
<i>fabric_01</i>	1.0	3	<i>struct.</i>	$0.999 \pm 0.001$	$0.715 \pm 0.028$	$0.576 \pm 0.054$
	1.55	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.000$	$1.000 \pm 0.000$
	1.6	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.000$	$1.000 \pm 0.000$
	1.65	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.001$	$1.000 \pm 0.001$
	2.5	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.990 \pm 0.006$	$0.990 \pm 0.006$
<i>fabric_02</i>	1.0	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.918 \pm 0.112$	$0.940 \pm 0.090$
	1.55	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.999 \pm 0.002$	$1.000 \pm 0.001$
	1.6	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.999 \pm 0.002$	$1.000 \pm 0.001$
	1.65	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.999 \pm 0.002$	$0.999 \pm 0.001$
	2.5	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.995 \pm 0.004$	$0.997 \pm 0.002$
<i>fabric_03</i>	1.0	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.865 \pm 0.011$	$0.795 \pm 0.019$
	1.55	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.000$	$1.000 \pm 0.000$
	1.6	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.000$	$1.000 \pm 0.000$
	1.65	3	<i>struct.</i>	$1.000 \pm 0.000$	$1.000 \pm 0.000$	$1.000 \pm 0.000$
	2.5	3	<i>struct.</i>	$1.000 \pm 0.001$	$0.987 \pm 0.011$	$0.983 \pm 0.014$
<i>fabric_04</i>	1.0	3	<i>struct.</i>	$0.995 \pm 0.004$	$0.186 \pm 0.022$	$0.230 \pm 0.106$
	1.55	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.994 \pm 0.007$	$0.999 \pm 0.001$
	1.6	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.997 \pm 0.006$	$0.999 \pm 0.001$
	1.65	3	<i>struct.</i>	$1.000 \pm 0.000$	$0.992 \pm 0.017$	$0.998 \pm 0.003$
	2.5	3	<i>struct.</i>	$0.998 \pm 0.002$	$0.934 \pm 0.109$	$0.991 \pm 0.015$
<i>fabric_06</i>	1.0	3	<i>struct.</i>	$0.929 \pm 0.041$	$0.260 \pm 0.167$	$0.950 \pm 0.035$
	1.55	3	<i>struct.</i>	$0.951 \pm 0.040$	$0.323 \pm 0.249$	$0.977 \pm 0.007$
	1.6	3	<i>struct.</i>	$0.951 \pm 0.040$	$0.287 \pm 0.191$	$0.976 \pm 0.005$
	1.65	3	<i>struct.</i>	$0.951 \pm 0.040$	$0.287 \pm 0.191$	$0.976 \pm 0.005$
	2.5	3	<i>struct.</i>	$0.953 \pm 0.041$	$0.210 \pm 0.128$	$0.974 \pm 0.003$
<i>KSDD2</i>	1.0	3	<i>struct.</i>	$0.863 \pm 0.047$	$0.903 \pm 0.003$	$0.824 \pm 0.004$
	1.55	3	<i>struct.</i>	$0.939 \pm 0.041$	$0.903 \pm 0.056$	$0.856 \pm 0.078$
	1.6	3	<i>struct.</i>	$0.976 \pm 0.012$	$0.960 \pm 0.015$	$0.936 \pm 0.024$
	1.65	3	<i>struct.</i>	$0.938 \pm 0.038$	$0.903 \pm 0.053$	$0.855 \pm 0.074$
	2.5	3	<i>struct.</i>	$0.838 \pm 0.033$	$0.759 \pm 0.032$	$0.674 \pm 0.034$

We conducted the experiments using 1600 defect-free patches. 70% of these defect-free patches are used for training and 30% are left for testing. The number of defect samples for testing depends on the

number of defects available in each of the dataset. For instance, some fabric types in the AITEX dataset include a wider range of defect types and so a higher number of defects. For each dataset type we generated synthetic defects by a pre-determined percentage of normal samples using Julia set fractals as described in Section 3.2. The metrics used for performance evaluation are Area Under ROC Curve (AUC), F1 Score and Accuracy. We resort to these metrics selectively depending on their utility. For instance to compare the performance of our method with respect to the use of different noise types (i.e. random or structured), or to assess the effect of  $\lambda$  parameter we employ all of the metrics above. Whereas to compare our method to other state of the art methods we use AUC only.

## 5. Results and Discussions

### 5.1 Tuning the Lambda parameter and comparison of noise types

As indicated in Section 3.1 the value of the  $\lambda$  parameter is critical to ensure that the signal and noise (normal background and defect) separation is optimal. This requires many experiments for tuning. We experimented extensively to search for an optimal value. Although dataset type seem to have an effect on the optimal value to a certain degree,  $\lambda = 1.6$  offers a reasonable compromise for the whole dataset range. This is illustrated in Table 1. So in subsequent runs we set  $\lambda = 1.6$ .

Table 2 AUC, F1 Score and Accuracy results for different image patch sizes.

Dataset	$\lambda$	patch size	RCAE AUC	F1 Score	Accuracy
<i>fabric_00</i>	1.6	32 × 32	<b>1.000 ± 0.000</b>	<b>0.918 ± 0.050</b>	<b>0.979 ± 0.014</b>
	6.7	64 × 64	0.132 ± 0.012	0.169 ± 0.000	0.092 ± 0.000
<i>fabric_01</i>	1.6	32 × 32	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>
	6.7	64 × 64	0.955 ± 0.012	0.931 ± 0.016	0.934 ± 0.015
<i>fabric_02</i>	1.6	32 × 32	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>
	6.7	64 × 64	0.999 ± 0.001	0.933 ± 0.008	0.950 ± 0.007
<i>fabric_03</i>	1.6	32 × 32	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>	<b>1.000 ± 0.000</b>
	6.7	64 × 64	0.491 ± 0.012	0.800 ± 0.000	0.667 ± 0.000
<i>fabric_04</i>	1.6	32 × 32	<b>1.000 ± 0.000</b>	<b>0.993 ± 0.007</b>	<b>0.999 ± 0.001</b>
	6.7	64 × 64	0.929 ± 0.000	0.854 ± 0.004	0.965 ± 0.000
<i>fabric_06</i>	1.6	32 × 32	<b>0.993 ± 0.003</b>	<b>0.955 ± 0.045</b>	<b>0.997 ± 0.003</b>
	6.7	64 × 64	0.510 ± 0.006	0.192 ± 0.058	0.921 ± 0.003
<i>KSSD2</i>	1.6	32 × 32	<b>0.976 ± 0.012</b>	<b>0.960 ± 0.015</b>	<b>0.936 ± 0.024</b>
	3.3	64 × 64	0.814 ± 0.023	0.919 ± 0.008	0.855 ± 0.014

Another important observation during our experiments was the effect of the image patch size (i.e. the size of the input image given to the autoencoder in pixels) on the detection performance of the autoencoder. As indicated in Table 2, smaller patch size leads to much better performance. In fact, smaller patch sizes are also better for more precise defect localization especially for small defects. One drawback of small patch sizes, however, may emerge when the defect (on the actual surface being inspected) is much larger compared to the patch size, which would entail a certain level of additional processing for higher level interpretation and classification of the defect. We also investigated the effect of using synthetic defects as opposed to random noise for pseudo supervised training. The results are illustrated in Table 3. Both in Table 2 and in Table 3 *noise ratio* refers to the percentage of the noisy samples with respect to the total number of samples in the training dataset. Note that using different defect types (i.e. structured or random noise) leads to different behavior of the AE during the optimization loop, which in turn causes our conversion logic to result in varying iteration numbers for the training. Depending on the fabric type, this may have dramatic effects on the performance of RAE during the test phase. To ensure fair comparison of the two types of defect modeling, we fixed the number of training loops to a total of 630 (including regularization iterations as well as AE training epochs) for this particular exercise. Notice that, by observing accuracy and F1 scores in addition to AUC scores, it can be concluded that for all dataset types synthetic defects perform better, and that for some fabric types (e.g. *fabric\_00*, *fabric\_04*, *fabric\_06* and *KSSD2*) the improvement induced by using

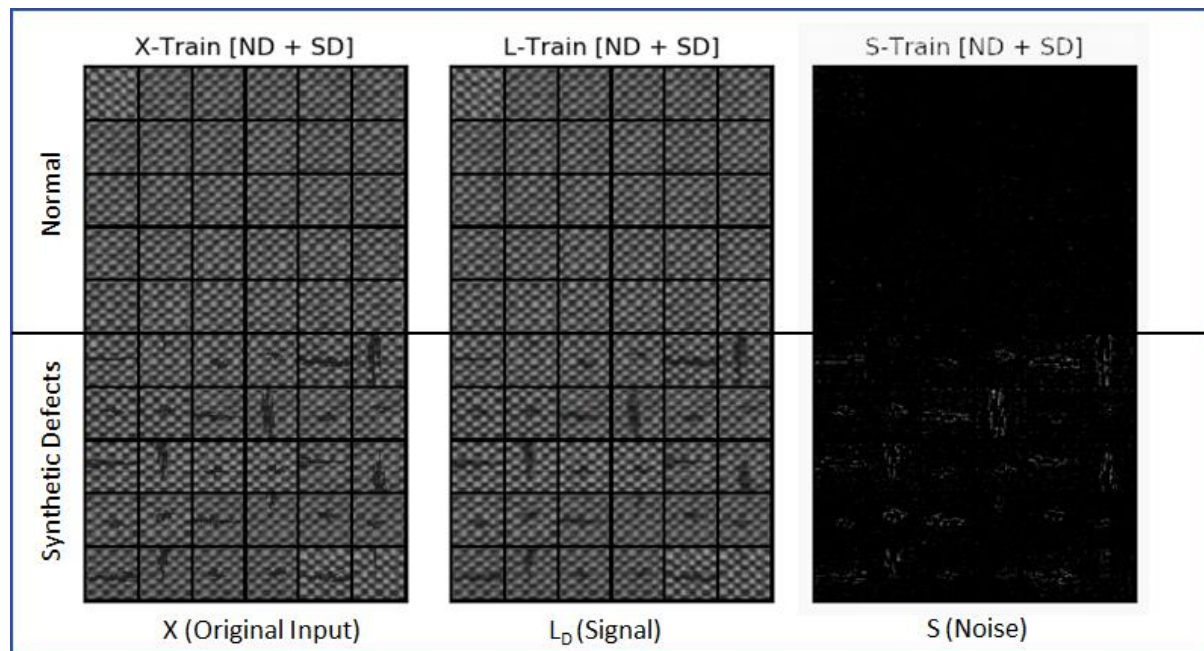


synthetic structured defects as opposed to random noise is more significant. In all experiments the convergence criteria used to terminate the iterations is the same, as explained in Section 3.1.

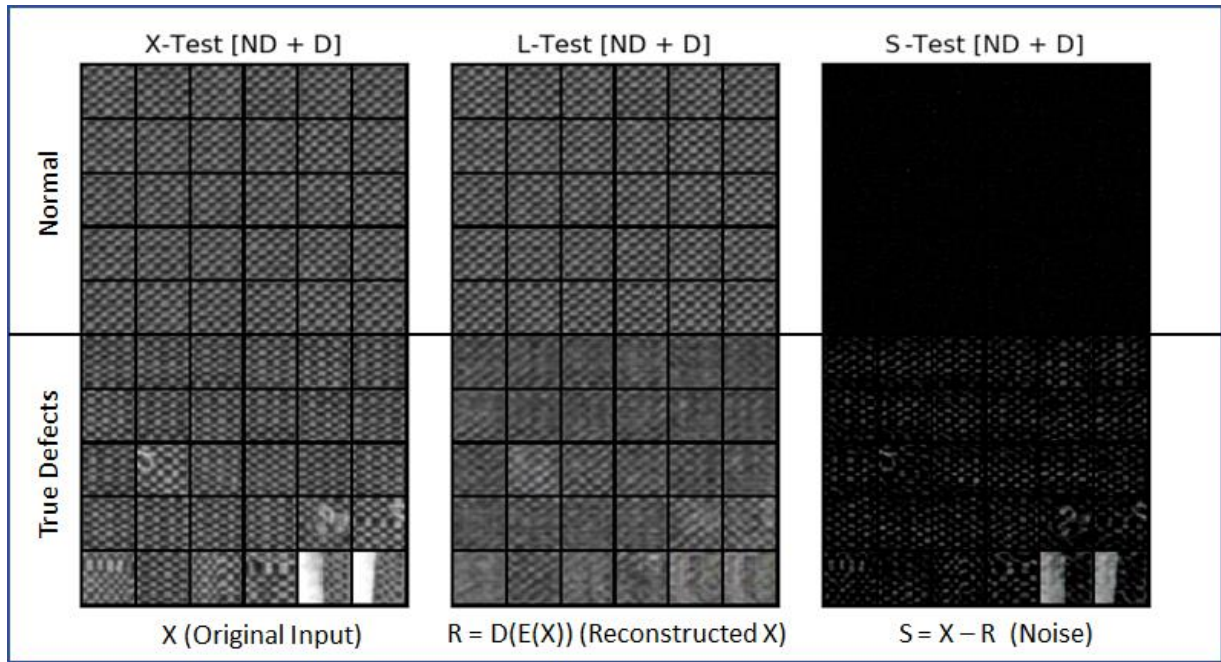
Table 3 Comparison of the Results When Using synthetic Defects (Structured Noise) vs. Random (Gaussian) Noise.

Dataset	Noise Type	Noise Ratio	RDCAE RAUC	F1 score	Accuracy
<i>fabric_00</i>	struct.	5	<b>0.991 ± 0.009</b>	<b>0.701 ± 0.278</b>	<b>0.846 ± 0.149</b>
	random	5	0.618 ± 0.137	0.228 ± 0.024	0.228 ± 0.114
<i>fabric_01</i>	struct.	5	<b>1.000 ± 0.000</b>	<b>0.996 ± 0.004</b>	<b>0.995 ± 0.005</b>
	random	5	0.999 ± 0.000	0.963 ± 0.025	0.959 ± 0.028
<i>fabric_02</i>	struct.	5	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
	random	5	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
<i>fabric_03</i>	struct.	5	<b>1.000 ± 0.000</b>	<b>0.998 ± 0.002</b>	<b>0.997 ± 0.003</b>
	random	5	0.996 ± 0.002	0.883 ± 0.034	0.863 ± 0.036
<i>fabric_04</i>	struct.	5	<b>1.000 ± 0.000</b>	<b>0.763 ± 0.223</b>	<b>0.925 ± 0.072</b>
	random	5	0.976 ± 0.004	0.333 ± 0.004	0.659 ± 0.001
<i>fabric_06</i>	struct.	5	<b>0.999 ± 0.000</b>	<b>0.840 ± 0.017</b>	<b>0.992 ± 0.000</b>
	random	5	0.983 ± 0.005	0.414 ± 0.014	0.977 ± 0.001
<i>KSDD2</i>	struct.	5	<b>0.953 ± 0.032</b>	<b>0.924 ± 0.047</b>	<b>0.885 ± 0.067</b>
	random	5	0.943 ± 0.006	0.813 ± 0.008	0.739 ± 0.010

## 5.2 Illustration of the Outputs



(a) Training samples



(b) Test samples

Figure 4 An example set of L (signal) and S (noise) matrices of 32x32 patches corresponding to fabric\_03, illustrating samples extracted from both training (a) and test (b) sets.

We further exemplify the performance of the RDCAE, in Figure 4, by illustrating the visual outputs of our method for signal (background) and noise (defect) samples of one of the datasets (i.e. fabric\_03). Figure 4 (a) depicts the original synthetic defects, the L (signal) part and S (noise) part corresponding to the same patches at the end of the training process. Note that the L matrices preserve the overall background pattern except where the defects were located. In contrast, the S matrices capture the defect areas. The figure includes both normal and defect images to highlight the difference. Observe that S images corresponding to the normal samples do not contain any pattern or pixels, indicating that there was no defect to separate out. Figure 4 (b) illustrates the same matrices obtained after the training of the AE, this time using the *test samples* that include the actual *real world* defects. This time instead of  $L_D$  we have the reconstructed input (i.e.  $R = D(E(X))$ ) in the middle of the Figure. Notice how the background (R) of the defect images is distorted, and the defects are reflected in the S images.

Figure 5 on the other hand presents the final reconstruction error scores of our RDCAE plotted against a Structural SIMilarity (SSIM) index [46] calculated by comparing the input and output of the AE. The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. In contrast to RDCAE reconstruction error scores, with SSIM we expect to have high scores (close to 1) for normal images, and relatively low scores for defect images. For the SSIM algorithm implementation we used the `compare_ssim` function provided by the `scikit-image` [47] library. Note that the normal samples of both training and test datasets are clustered closely in a relatively dense area and that the defect samples included in test datasets (i.e. actual real-world defects) are distributed away from the non-defect samples. Also, it can be seen that the synthetic defects are noticeably separated away from non-defect samples of both training and test datasets. This indicates that the RDCAE has learned an efficient representation of the normal samples exceptionally well, such that the RDCAE reconstruction error now behaves as a sound discriminator.

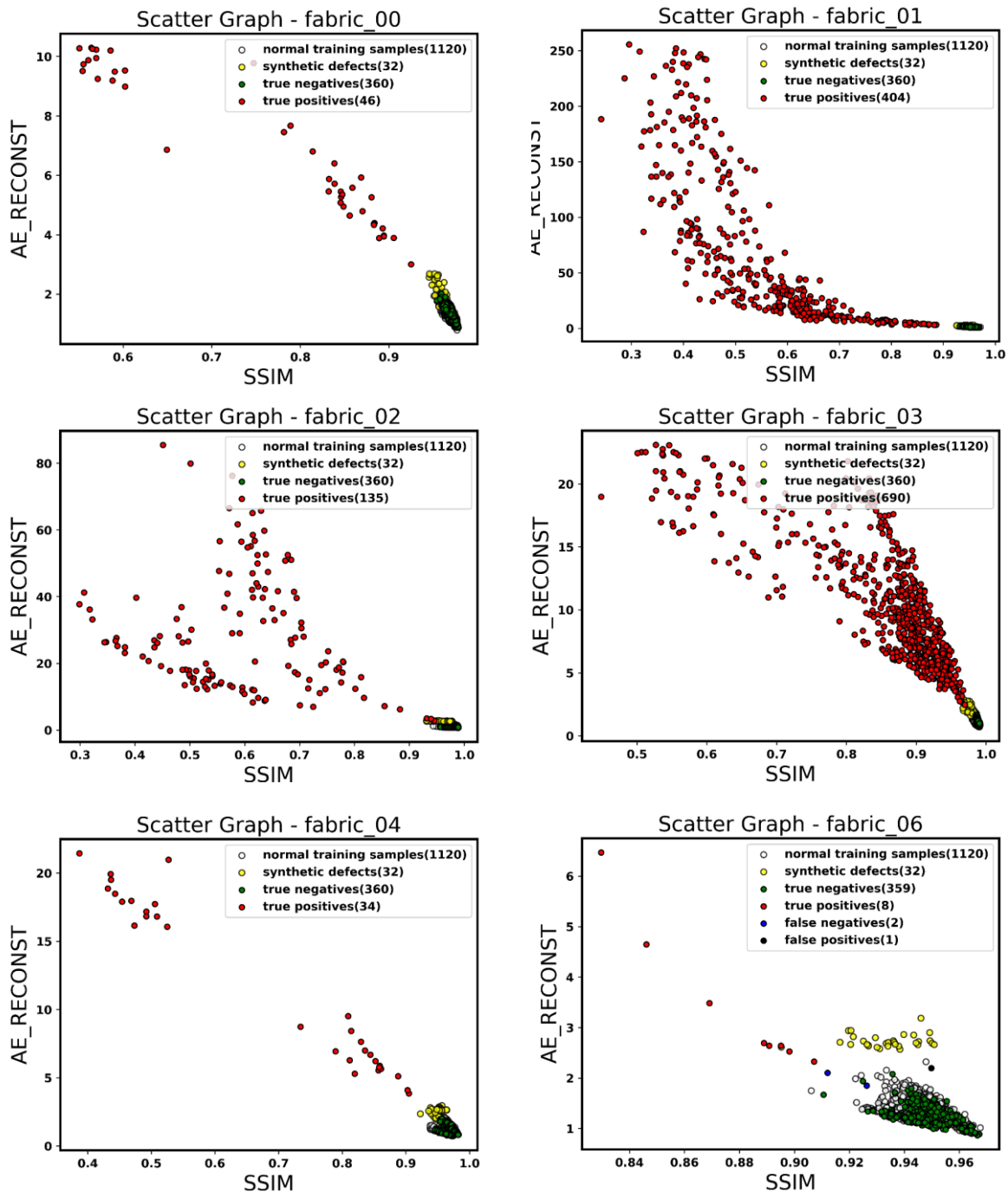


Figure 5 Scatter graphs illustrating a 2D distribution of RDCAE reconstruction errors plotted against structural similarity scores, corresponding to anomalous samples and normal samples of test datasets. Normal samples and synthetic defects used during training

Each graph includes a legend indicating the number of normal training samples, synthetic defects, true positives, true negatives, false positives (if any) and false negatives (if any). The results show that our RDCAE has a remarkable performance for detecting defects in regard with fabric types provided in an industrial grade dataset. Note that throughout the performance tests reported up to this point, the dataset *fabric\_06* performs relatively poorly. This is likely to be due to its challenging features with regard to two aspects:

1. The normal samples are relatively difficult to differentiate from the anomalous ones since they share more common features compared to other datasets. We note that this could be regarded as an illustrative example of challenges that may arise in real world industrial applications.

- The number of anomalous samples is much less compared to other datasets (i.e. only 10 samples, compared to 46 for *fabric\_00*, 404 for *fabric\_01*, 135 for *fabric\_02*, 690 for *fabric\_03* and 34 for *fabric\_04*). Therefore, metric scores (F1 in particular) are much more sensitive to the number of false positives and false negatives.

Note also that, there are some other elements that lead to exceptionally good results for other datasets. First, each of our datasets contains samples from a specific fabric type. So we perform training for a particular fabric type, and then carry out defect detection test for the same fabric type. This implies that in a real world industrial application we would need to train the visual inspection system for each fabric type. This should not be an important predicament, since training times are not huge (in the order of a couple of minutes) for even modest hardware specification we used, and using embedded software would dramatically improve the performance. Moreover, large-volume fabric production lines can tolerate such initial setup phases. However, this is something that should be recognized when interpreting the results. Second, combining AE optimization with pixel wise regularization in a single framework to achieve signal/noise separation and feature learning in a progressive way, seem to work particularly well for fabric defect detection.

### 5.3 Comparison to Other Anomaly Detection Methods

Table 4 Comparison of Area Under ROC Curve (AUC) Metric Results of Robust Deep Convolutional Autoencoder (RDCAE) to That of Other Methods

Dataset	Robust DCAE	DCAE	Deep SVDD	OC-SVM-LREP	OC-SVM-AERE	ISOF-LREP	ISOF-AERE
<i>fabric_00</i>	<b>1.000 ± 0.000</b>	0.973 ± 0.026	0.944 ± 0.030	0.676 ± 0.134	0.828 ± 0.028	0.669 ± 0.148	0.825 ± 0.042
<i>fabric_01</i>	<b>1.000 ± 0.000</b>	0.994 ± 0.002	0.842 ± 0.111	0.762 ± 0.015	0.911 ± 0.011	0.781 ± 0.027	0.902 ± 0.017
<i>fabric_02</i>	<b>1.000 ± 0.000</b>	1.000 ± 0.000	0.982 ± 0.012	0.665 ± 0.028	0.857 ± 0.009	0.664 ± 0.021	0.886 ± 0.008
<i>fabric_03</i>	<b>1.000 ± 0.000</b>	1.000 ± 0.001	0.818 ± 0.092	0.476 ± 0.057	0.875 ± 0.007	0.505 ± 0.031	0.873 ± 0.011
<i>fabric_04</i>	<b>1.000 ± 0.000</b>	0.978 ± 0.006	0.774 ± 0.109	0.488 ± 0.065	0.865 ± 0.019	0.485 ± 0.024	0.857 ± 0.019
<i>fabric_06</i>	<b>0.993 ± 0.003</b>	0.955 ± 0.019	0.739 ± 0.127	0.384 ± 0.008	0.737 ± 0.063	0.468 ± 0.003	0.777 ± 0.074
<i>KSSD2</i>	<b>0.972 ± 0.012</b>	0.787 ± 0.013	0.807 ± 0.002	0.704 ± 0.010	0.712 ± 0.003	0.661 ± 0.017	0.722 ± 0.003

We compare the performance of our method with four different state of the art methods:

- Deep Convolutional Autoencoders (DCAE)** [26, 27]. We employed exactly the same neural network architecture as our Robust DCAE except that the optimization method is based only on minimizing the reconstruction error as in standard AEs.
- Isolation Forests (IF)** [20]. We used the latest stable version of the widely-used python machine learning library scikit-learn [43]. The IsolationForest function provided by the library is an implementation of the algorithm presented in [20].
- One-Class SVM** [19]. We employ the implementation provided by scikit-learn [43] as we did for the Isolation Forests. This is an implementation of the algorithm presented in [19]. Once again, we used the latent representation of the DCAE above as input to the algorithm.
- One-Class Support Vector Data Description (OC-SVDD)**. We adapt the method provided by [23]. We used the PyTorch implementation available from their github repository at <https://github.com/lukasruff/Deep-SVDD-PyTorch>. To facilitate a meaningful comparison, we ensured that our RDCAE architecture and the AE architecture used in their method are exactly the same, and also used the encoder part in the corresponding one-class SVDD network as suggested by their method.

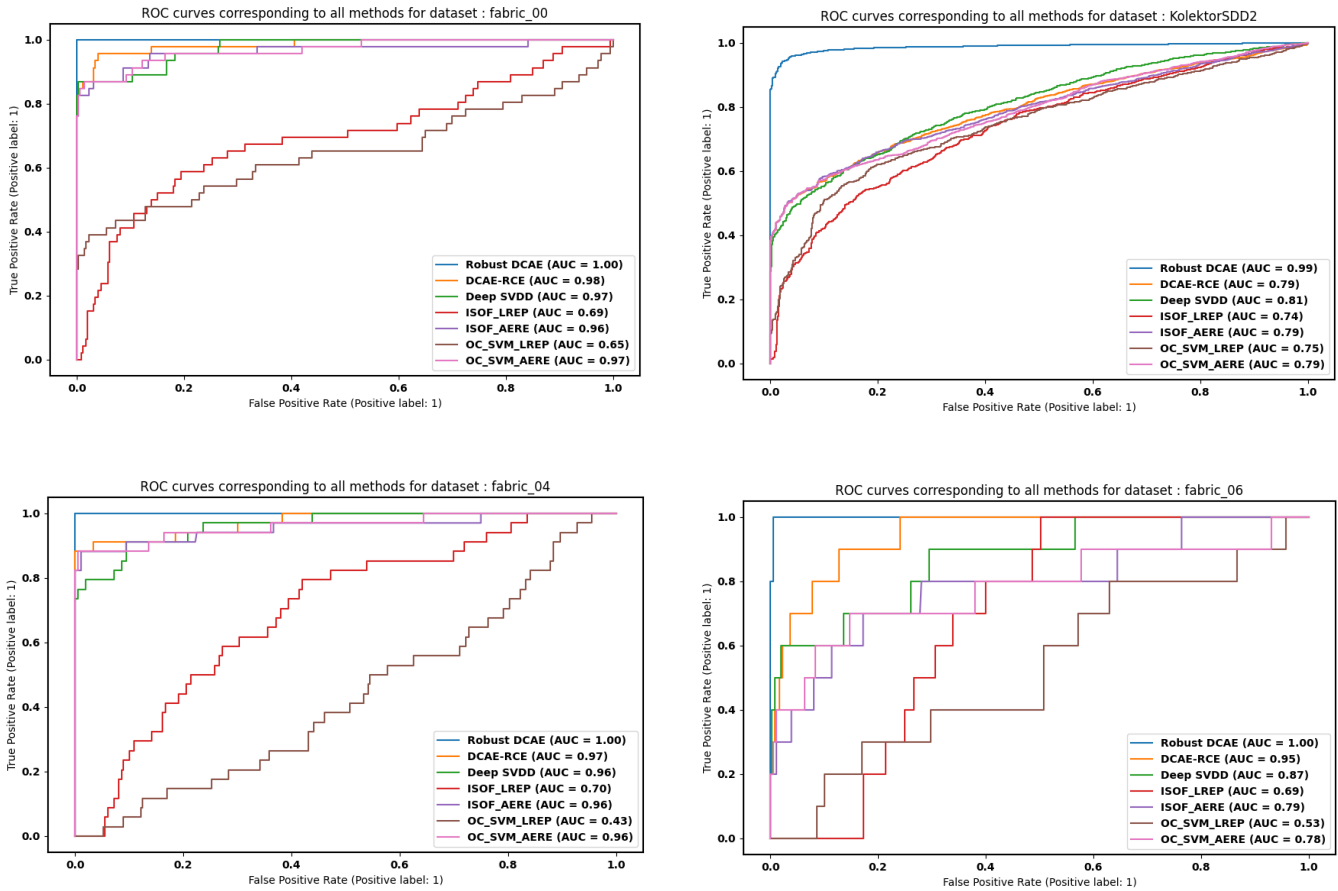


Figure 6 The graphs above illustrate the Receiver Operating Characteristic (ROC) curves generated by all the methods listed in Table 4, pertaining to datasets that are relatively more challenging.

For DCAE and Deep SVDD we trained the AEs using a comparable epoch number. As our Robust DCAE uses a specific convergence criteria, the number of total training iterations is not fixed. However, for DCAE and Deep SVDD a pre-determined epoch number has to be set. So, for a fair comparison we used the average training epoch number obtained from our replicated runs and used that number as the epoch number for DCAE and Deep SVDD. For OC-SVM and Isolation Forest we used the recommended hyper-parameters in their documentation. To establish a well-founded basis for comparison with these two methods we used two different inputs to their algorithm: 1 - the latent representation of the AE (in DCAE) mentioned above which is a feature set of size 256 for each image. This is indicated by the suffix "-LREP" at the end of corresponding column label in Table 4 (e.g. OC-SVM-LREP), 2 - the reconstruction error obtained from the AE, which is a single figure. This is indicated by the suffix "-AERE" at the end of corresponding column label (e.g. ISOF-AERE). In addition to AUC scores given in Table 4, we provide Receiver Operating Characteristic (ROC) curves generated by those methods in Figure 6, to illustrate a more comprehensive and discernible performance comparison of the methods. The ROC curve shows the trade-off between sensitivity (or True Positive Rate (TPR)) and specificity ( $1 - \text{False Positive Rate (FPR)}$ ). For all methods, normalized raw scores are used (rather than predicted labels) to obtain smoother and more indicative curves. The four graphs in the figure pertain to relatively more challenging datasets, to further articulate the level of improvement achieved for those datasets. It can be observed that our method, Robust DCAE, outperforms all of the methods, in some cases with a significant margin. This can be attributed to the ability of the robust convolutional AEs to efficiently learn the stable representative features, through the separation of signal and noise during its training.

## 6. Concluding Remarks

In this paper, we illustrated the use of Robust Deep Convolutional Autoencoders (RDCAE) for defect detection via two recently introduced industrial quality datasets and we presented some improvements to the training process of RDCAE, that enable us to more reliably manage the convergence of the training. We have illustrated the use of synthetic simulated defects (or structured noise), so that a robust convergence criteria can be settled without compromising the detection performance of the method. We believe that this introduces a plausible and efficient solution to the defect detection process *in the absence of true (real life) abnormal samples*.

Our experiment results are a clear manifestation of the theoretical argument stating the competency of robust deep convolutional AEs in signal-noise separation and thus their ability to more effectively learn the common, stable features as opposed to subtle and inconsistent features that are exhibited by outliers. In other words, compared to plain AEs (e.g DCAE) and many other anomaly detection methods, robust AEs are more adept to learn the boundary between the normal and abnormal samples. There are many application areas, such as automated visual surface inspection systems, that can benefit from the strengths of this method.

It is also worth noting that the clear separation of the noise (i.e. defects) into a separate image (e.g. S images in Figure 4) makes the method more amenable to either further image processing (as a post processing step), or various custom neural network architecture designs for further feature extraction to enable defect identification and classification.

## 7. Acknowledgments

1. This work was partially supported by TUBİTAK BİLGEM we are grateful for that support.
2. The author declares that he has no known competing interests or personal relationships that could have appeared to influence the work reported in this paper.

## 8. Availability of data and materials

There are two datasets used in the experiments conducted during our work:

1. The first one is AITEX Fabric Image Database which is publicly available from The Textile Industry Research Association - AITEX of Spain, and is downloadable from <https://www.aitex.es/afid/>
2. The second one is the Kolektor Surface-Defect Dataset 2 (KolektorSDD2/KSDD2) which is publicly available from The Visual Cognitive Systems Laboratory of Slovenia, and can be downloaded via <https://www.vicos.si/resources/kolektorsdd2/>

## References



- [1] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, *Artificial Intelligence Review* 22 (2) (2004) 85–126. doi:10.1007/s10462-004-4304-y .
- [2] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM Comput. Surv.* 41 (3) (Jul. 2009).
- [3] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, *Signal Processing* 99 (2014) 215–249. doi:10.1016/j.sigpro.2013.12.026 .
- [4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (2015) 436–44. doi:10.1038/nature14539 .
- [5] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition (2015). arXiv:1512.03385 .
- [6] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, D. Song, Anomalous instance detection in deep learning: A survey (2020). arXiv:2003.06979 .

- [7] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey (2019). arXiv:1901.03407 .
- [8] D. Bank, N. Koenigstein, R. Giryes, Autoencoders (2020). arXiv:2003.05991 .
- [9] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, B. Reimer, Detection of anomalies in large scale accounting data using deep autoencoder networks, CoRR abs/1709.05254 (2017). arXiv:1709.05254 .
- [10] D. Zimmerer, F. Isensee, J. Petersen, S. Kohl, K. Maier-Hein, Unsupervised anomaly localization using variational auto-encoders, in: D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, A. Khan (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, Springer International Publishing, Cham, 2019, pp. 289–297.
- [11] C. Zhou, R. C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, ACM, New York, NY, USA, 2017, pp. 665–674. doi:10.1145/3097983.3098052 .
- [12] R. Chalapathy, A. K. Menon, S. Chawla, Robust, deep and inductive anomaly detection, in: M. Ceci, J. Hollmén, L. Todorovski, C. Vens, S. Džeroski (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2017, pp. 36–51.
- [13] M. Ribeiro, A. E. Lazzaretti, H. S. Lopes, A study of deep convolutional auto-encoders for anomaly detection in videos , Pattern Recognition Letters 105 (2018) 13 – 22, machine Learning and Applications in Artificial Intelligence. doi:https://doi.org/10.1016/j.patrec.2017.07.016 .
- [14] H. Y. Ngan, G. K. Pang, N. H. Yung, Automated fabric defect detection? a review , Image and Vision Computing 29 (7) (2011) 442–458. doi:https://doi.org/10.1016/j.imavis.2011.02.002 .
- [15] K. Hanbay, M. F. Talu, Ömer Faruk Özgüven, Fabric defect detection systems and methods? a systematic literature review, Optik 127 (24) (2016) 11960–11973.
- [16] M. F. Nisha, P. Vasuki, S. M. M. Roomi, Survey on various defect detection and classification methods in fabric images, Journal of Environmental Nano Technology (JENT) 6 (2) (2017) 20–29.
- [17] C. Li, J. Li, Y. Li, L. He, X. Fu, J. Chen, Fabric defect detection in textile manufacturing: A survey of the state of the art, Security and Communication Networks 2021 (2021) 9948808.
- [18] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, P. Dario, Visual-based defect detection and classification approaches for industrial applications? a survey, Sensors 20 (5) (2020).
- [19] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.
- [20] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data 6 (1) (Mar. 2012). doi:10.1145/2133360.2133363 .
- [21] S. Hariri, M. Carrasco Kind, R. J. Brunner, Extended isolation forest, IEEE Transactions on Knowledge and Data Engineering (2019) 1–1 doi:10.1109/TKDE.2019.2947676 .
- [22] D. M. Tax, R. P. Duin, Support vector data description, Machine Learning 54 (1) (2004) 45–66.
- [23] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proceedings of the 35th International Conference on Machine Learning, Vol. 80, 2018, pp. 4393–4402.
- [24] R. Chalapathy, A. K. Menon, S. Chawla, Anomaly detection using one-class neural networks (2018). arXiv:1802.06360 .
- [25] E. Parzen, On estimation of a probability density function and mode , The Annals of Mathematical Statistics 33 (3) (1962) 1065–1076. URL <http://www.jstor.org/stable/2237880>
- [26] D. Gong, L. Liu, V. Le, B. Saha, M. Mansour, S. Venkatesh, A. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection (10 2019). doi:10.1109/ICCV.2019.00179 .
- [27] S. Edwards, M. S. Lee, Using convolutional neural network autoencoders to understand unlabeled data, in: T. Pham (Ed.), Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Vol. 11006, International Society for Optics and Photonics, SPIE, 2019, pp. 444 – 458. doi:10.1117/12.2518459 .
- [28] P. Wu, J. Liu, F. Shen, A deep one-class neural network for anomalous event detection in complex scenes, IEEE Transactions on Neural Networks and Learning Systems 31 (7) (2020) 2609–2622.

- [29] P. Schlachter, Y. Liao, B. Yang, Deep one-class classification using intra-class splitting , 2019 IEEE Data Science Workshop (DSW) (Jun 2019). doi:10.1109/dsw.2019.8755576 .
- [30] E. Plaut, From principal subspaces to principal components with linear autoencoders (2018). arXiv:1804.10253 .
- [31] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis, *J. ACM* 58 (3) (Jun. 2011).
- [32] F. De la Torre, M. J. Black, Robust principal component analysis for computer vision, in: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 1, 2001*, pp. 362–369 vol.1.
- [33] H. Akrami, A. A. Joshi, J. Li, S. Aydoore, R. M. Leahy, Robust variational autoencoder (2019). arXiv:1905.09961 .
- [34] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders , in: *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, Association for Computing Machinery, New York, NY, USA, 2008, p. 1096–1103. doi:10.1145/1390156.1390294 .
- [35] A. Collin, C. D. Vleeschouwer, Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise, 2020 25th International Conference on Pattern Recognition (ICPR) (2021) 7915–7922.
- [36] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, K.-R. Müller, Explainable deep one-class classification , in: *International Conference on Learning Representations, 2021*. URL <https://openreview.net/forum?id=A5VV3UyIQz>
- [37] S. Mosci, L. Rosasco, M. Santoro, A. Verri, S. Villa, Solving structured sparsity regularization with proximal methods, in: J. L. Balcázar, F. Bonchi, A. Gionis, M. Sebag (Eds.), *Machine Learning and Knowledge Discovery in Databases*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 418–433.
- [38] A. Douady, *Julia Sets and the Mandelbrot Set*, Springer, Berlin, Heidelberg, 1986, pp. 161–174. doi:10.1007/978-3-642-61717-1\_13 .
- [39] G.-L. Zhang, M.-Q. Cai, X.-L. He, X.-Z. Gao, M.-D. Zhao, D. Wang, Y. Li, C. Tu, H.-T. Wangmark, Pseudo-topological property of julia fractal vector optical fields, *Opt Express* 27 (9) (2019) 13263–13279.
- [40] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2017). arXiv:1412.6980 .
- [41] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324. doi:10.1109/5.726791 .
- [42] A. Paszke, et al. Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Álché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.
- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [44] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, J. Moreno, A public fabric database for defect detection methods and results, *Autex Research Journal* 19 (4) (2019) 363 – 374. doi:<https://doi.org/10.2478/aut-2019-0035> .
- [45] J. Božič, D. Tabernik, D. Skočaj, Mixed supervision for surface-defect detection: from weakly to fully supervised learning, *Computers in Industry* (2021).
- [46] Zhou Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing* 13 (4) (2004) 600–612.
- [47] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in python, *PeerJ* 2 (2014) e453.



# Simulation of Cargo Unloading Problem: A Case Study on Estimating the Optimal Number of Trucks and Cranes

 Waseem Hamdoon<sup>1</sup>,  Ahmet Zengin<sup>2</sup>

<sup>1</sup>Corresponding Author; Sakarya University, Department Of Computer And Information Engineering; waseem.algburi@ogr.sakarya.edu.tr

<sup>2</sup>Sakarya University, Department Of Computer And Information Engineering; azengin@sakarya.edu.tr

Received 11 September 2022; Revised 18 October 2022; Accepted 22 November 2022; Published online 31 December 2022

## Abstract

Unloading and loading activities consume high operational expenses at cargo exchange terminals; for instance, the costs of these activities are approximately half of the total costs of the port. On the other hand, interest in modeling and simulation tools has increasingly grown to analyze operational and organizational systems. Where it is possible to build many systems and study their behavior, which saves a lot of effort, time, and cost by using some frameworks to implement modeling and simulation using the computer. In this paper, discrete events DEVS-Suite is used to implement a simulation of the cargo unloading problem, which represents a study to estimate the optimal number of trucks and cranes required in the process of unloading cargo according to some parameters; and the simulation duration is one month, which is equivalent to 43200 minutes. Based on the performance measures adopted in this study, the optimal number of trucks and cranes is 5 of three assumptions which are 3, 5, and 10, where the work will be in a permanent working condition, with high productivity and low cost.

**Keywords:** Modeling and Simulation, Model, System, DEVS-Suite, Cargo Problem

## 1. Introduction

In cargo exchange terminals, unloading and loading activities suffer from high operational costs; for example, loading and unloading operations account for nearly half of the total costs of the port [1]. As a result, there is a growing interest in increasing the efficiency of unloading and loading activities while reducing operational costs.

On the other hand, Researchers typically use simulation software to analyze the interoperability of different peripheral areas or to examine the operation of a sub-system in greater detail to save effort, costs, and time with a clear visualization of system implementation [2]. Many analysts, project managers, and those engaged in research and development must use modeling and simulation techniques [3]. In addition, Modeling and Simulation (M&S) systems have become widely used in a wide range of application areas as computing technology advances at a rapid pace, allowing for the production of much faster computers every day [4], it has the potential to improve modeling and simulation performance significantly.

Simulating a real-world process or system over time is referred to as simulation. Simulations require models; the model represents the major features or behaviors of the chosen system or process, while the simulation depicts the model's evolution across time. The relationship between modeling and simulation can be described as shown in Figure 1.

Simulation is utilized in a variety of situations, including technology simulation for performance tuning or optimization, complex manufacturing systems [5], safety engineering, education, training, testing, video games, networks, and human movement [6],[7],[8]. Simulation may be used to demonstrate the real-world consequences of certain situations and actions. Simulation is employed when the real system cannot be utilized because it is not accessible, unsafe, or inappropriate to use, or created but not yet built, or does not exist. In economics, simulation is utilized alongside scientific modeling of natural or human systems to obtain insight into how they work [9]. Furthermore, simulation modeling offers

significant and distinctive capabilities for analyzing and designing service-oriented computing systems that must satisfy various and conflicting quality of service (QoS) standards [10].

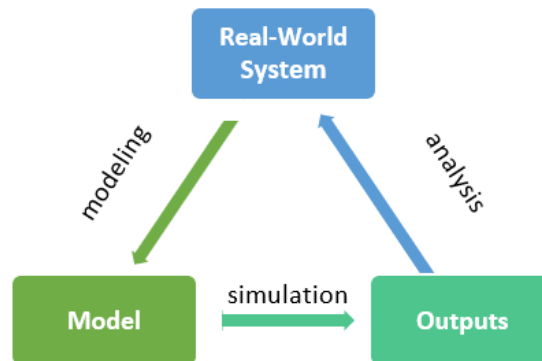


Figure 1 Relationship between modeling and simulation [17]

Several frameworks may be utilized to implement modeling and simulation concepts; in this study, the DEVS-Suite is used to implement the problem [16]. The DEVS-Suite simulator is one of the most commonly used modeling and simulation tools for parallel DEVS formalism [18]. DEVS describes a hierarchy structure for models using a few fundamental components. The coupling and decoupling ideas are supported by this framework, which is also very scalable [19]. The Discrete Event System Specification (DEVS) formalism allows a mathematical entity known as a system to be specified. A system has a time basis, inputs, states, and outputs, as well as functions that predict future states and outputs from current states and inputs [11],[12],[13],[20].

The case study in this paper is a simulation application of the problem of unloading goods that represents an investigation in determining the ideal number of trucks and cranes needed during the process of unloading products according to specific parameters. According to the performance metrics used in this study, the best number of trucks and cranes is 5 out of three assumptions of 3, 5, and 10, where the task will be in a steady state of work and good productivity.

## 2. Problem Description

This section contains a detailed description of the problem at hand, estimating the optimal number of trucks and cranes required for unloading goods shipments for a company. In this system, one container at a time can be handled by a cargo crane, and one container can typically be transported by a truck [14]. Trucks loaded with goods arrive randomly 24 hours a day, seven days a week, and are stored in the company's warehouses. To correctly complete the simulation process, a set of parameters is assumed. It is expected that these parameters are actual and obtained from a relevant company, but it is considered here for study [15]. They are as follows:

- The company has an equal number of truck and forklift drivers, referred to as C.
- The wages of each driver working for the company is 10 Turkish Liras per hour, in addition to a fixed salary of 20 Turkish Liras per day.
- The company adopts the principle of unloading the truck upon its arrival to avoid the significant fines imposed on it if it causes delays in the trucks.
- The unloading rate for each crane driver is fixed and equal to 7 tons per hour.
- The following cases were neglected:
  - The occurrence of malfunctions in the cranes owned by the company.
  - Absence or illness of any driver working for the company.
  - Filling the stores, as it was assumed that the company's stores are enormous and cannot be filled.

### 3. System Description

#### 3.1 System Analysis

To understand the system well and then analyze it accurately, it is necessary to refer to the following:

- System type: Queue system.
- Simulation type: Discrete Event
- Performance measures: Total unloading costs.
- System entities: Generate Truckload, Truck\_Crane, Unloading management
- Events:
  - The arrival of a truck
  - Start unloading the truck
  - Finish unloading the truck
- Relationships:
  - The service time (truck unloading time) is directly proportional to the number of cargo trucks, assuming that the unloading rate of trucks is fixed.
- System goal:  
Determine the optimum number of truck drivers and cranes to obtain the lowest total unloading costs.

The number of drivers should not be more than required because this causes additional costs to the company without any benefit. The number of drivers should not be less than the optimum number, which may cause delays in unloading the load, and consequently, the company may incur significant delay penalties.

#### 3.2 Statistical distributions of the system

The statistical distributions of random system inputs can be described as follows:

- The times between the arrival of the trucks follow an exponential distribution.
- Truck load weights are random, follow a uniform distribution, and are close to each other.

So, suppose that:

- Average time between the arrival of trucks is 140 (adhere to the exponential distribution)
- Truckload weights between 20 and 40 (stick to the exponential distribution).

Since the unloading rate for all drivers is fixed and is seven tones, the service time (the time required to unload any truckload) is directly proportional to the weights of truck loads. It is random and follows the same uniform distribution, and is specified during a certain period that can be calculated as follows:

The smallest period of time =  $20 / 7 = 2.85$  Hour \* 60 minutes = 171.4

The largest period of time =  $40 / 7 = 5.7$  Hour \* 60 minutes = 342.8

### 4. Modeling Components of Cargo Unloading

This section shows the proposed scenario design for the problem described according to the system described in the previous section.

#### 4.1 Cargo Coordinator model

This model is of multi-server coordinator type and is responsible for receiving trucks and distributing them to unloading stations, consisting of a truck and a crane for each truck with their drivers. This model is always active and represents a transit area only, so the truck arrival time does not affect its condition (See Figure 2).

#### 4.2 Cargo Coordinator model

This model is an atomic model that represents the process of unloading the truckload of each truck using a crane, as it takes time (uniform distribution between (171.4 and 342.8)) as mentioned earlier. This model also has two states: passive and busy. Passive implies that this station can accept a truck to unload its cargo if its sequence is in the queue, while busy means that the station is currently unloading a previous cargo. After completing the unloading process, a notification of the completion of the unloading process will be returned to the transducer model.

#### 4.3 Experimental Frame

The model's experimental frame is a coupled model. Consisting of two basic models: Generator and Transducer.

#### 4.4 Generator Truckload model

This is an atomic model responsible for generating truckloads to observe their behavior within the proposed model. Also, the input ports in this model are (in, start, stop) through the port (in), so it is possible to inject input data for testing this model. As for the output, there is one output port (out) (See Figure 2).

#### 4.5 Transducer model

This model is an atomic model responsible for measuring performance indices such as “turnaround time” and “throughput” for the truckloads processed by an unloading station model during a specified period (See Figure 2). Also, the input ports in this model are (ariv, solved, in) so that the “ariv” port for receiving a copy of the truckload generated by Generate Truckload model, solved for receiving the truckload processed number, In addition, (TA, Thru, out) are the outports, with "TA" standing for "turnaround time" and "Thru" for "throughput."

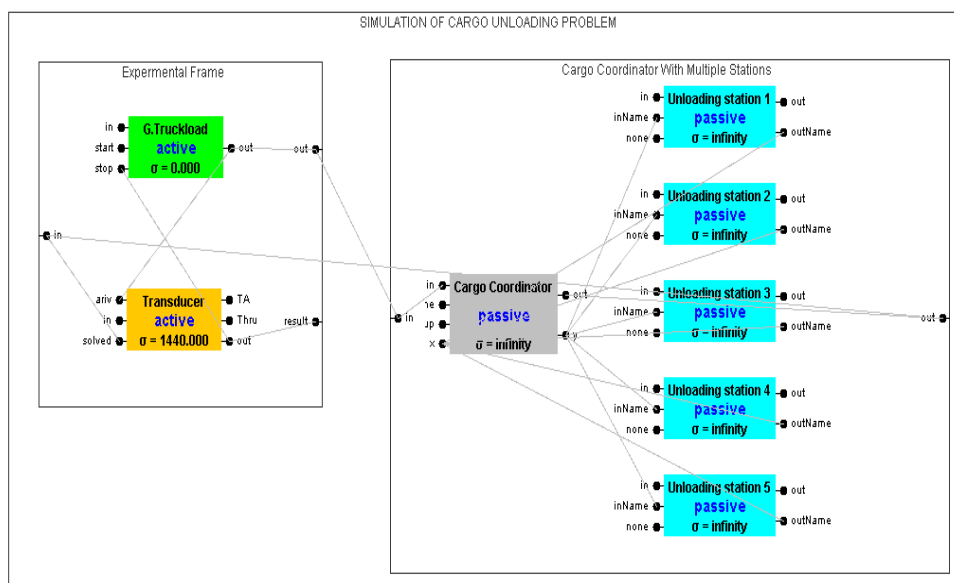


Figure 2 Simulation model of cargo unloading problem with five unloading stations

## 5. Simulation Experiments

In this section, the parameters of the built model will be placed with the parameters of the proposed system as follows:

- The state of Generate Truckload and Transducer models are always active.
- Sigma value of Generate Truckload model equals exponential distribution of (140) between each generating process.
- Sigma value of the Transducer model equals one month in minutes, which means ( $30*24*60=43,200$ ) minutes, representing the total simulation time. Thus, this time will decrease after each step by processing time value.
- The sigma value of the Cargo Coordinator model is always infinity, and its state value can take these values:
  - passive: in one of the two cases:
    - First case: On the initial run of the simulator (See Figure 3).
    - Second case: sending the truckload on “y” output to all (Unloading station) models.
      - send\_y: when the generated truckload by (Generate truckload) model is received by “in” inport and sent to “y” output.
      - send\_out: When the truckload that has been processed is sent from output “outName” of one of the (unloading station) models to “x” inport.
- The sigma value of each (unloading station) model is (infinity) when start running the simulation; after that, it will be taken (uniform distribution between (171.4 and 342. 8)), and then it decreases gradually.
- On the other hand, the state will take “passive or busy.”

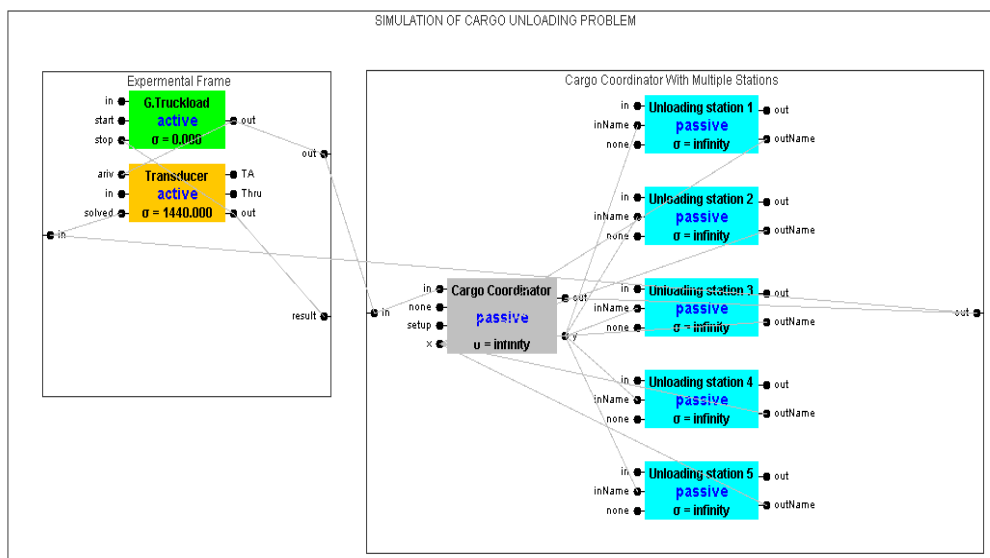


Figure 3 The initial run of the simulation

According to the previous parameters, the steps of the simulation will be as follows:

**Step 1:** Create a truckload by the (Generate Truckload) model, and send it to the (Cargo Coordinator) model (See Figure 4).

**Step 2:** Send the truckload to ALL (Unloading station) models for processing by one of them if its state is passive and its order equivalent to the next order in the queue, and change its state to “busy” also, decrease its sigma value by truckload processing time. (repeat this step until complete processing truckload) (See Figure 5).

**Step 3:** send the processed truckload information by the (unloading station) model to the (Cargo Coordinator) model (See Figure 6).

**Step 4:** send the processed truckload information from the (Cargo Coordinator) model to the (Transducer) model for computing some performance measures (See Figure 7).

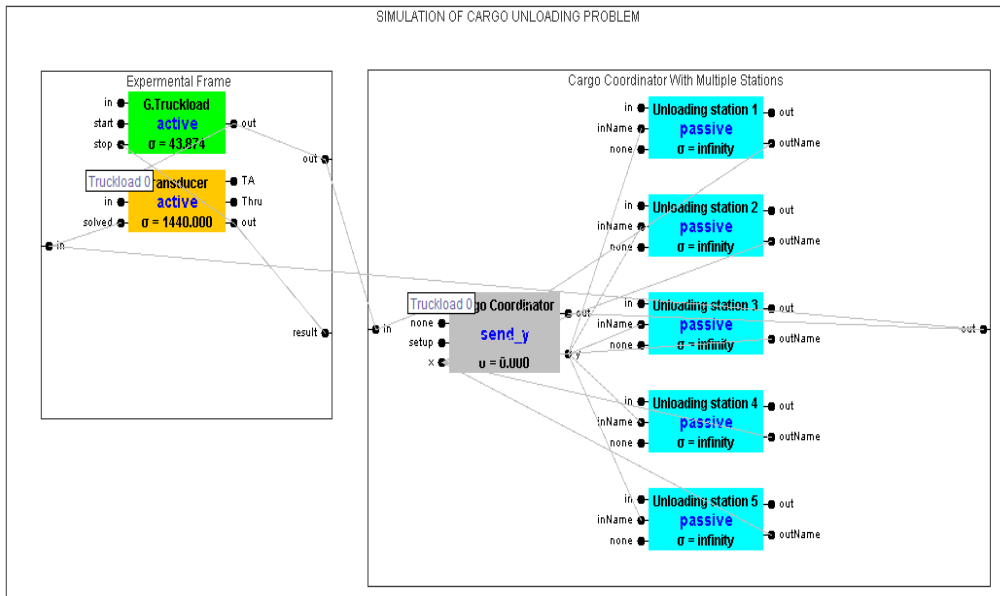


Figure 4 Step 1: Creating a truckload

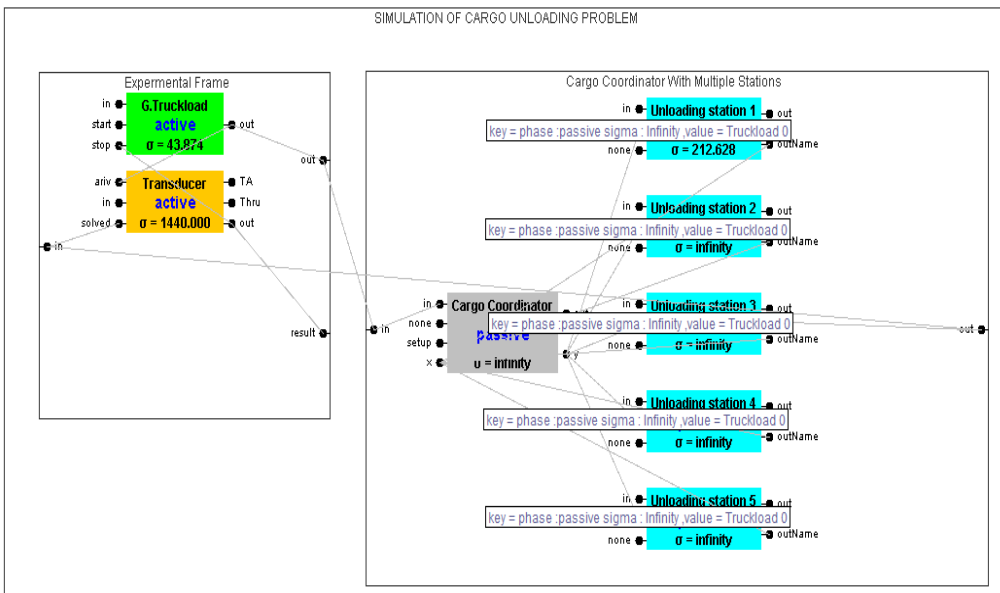


Figure 5 Step 2: Send the truckload to ALL (Unloading station) models

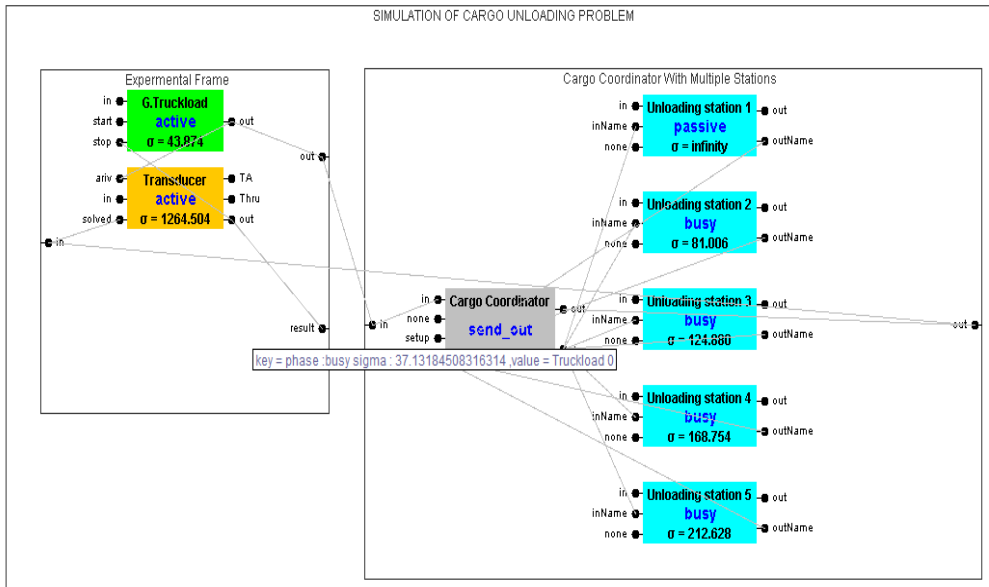


Figure 6 Step 3: send the processed truckload information to (Cargo Coordinator) model

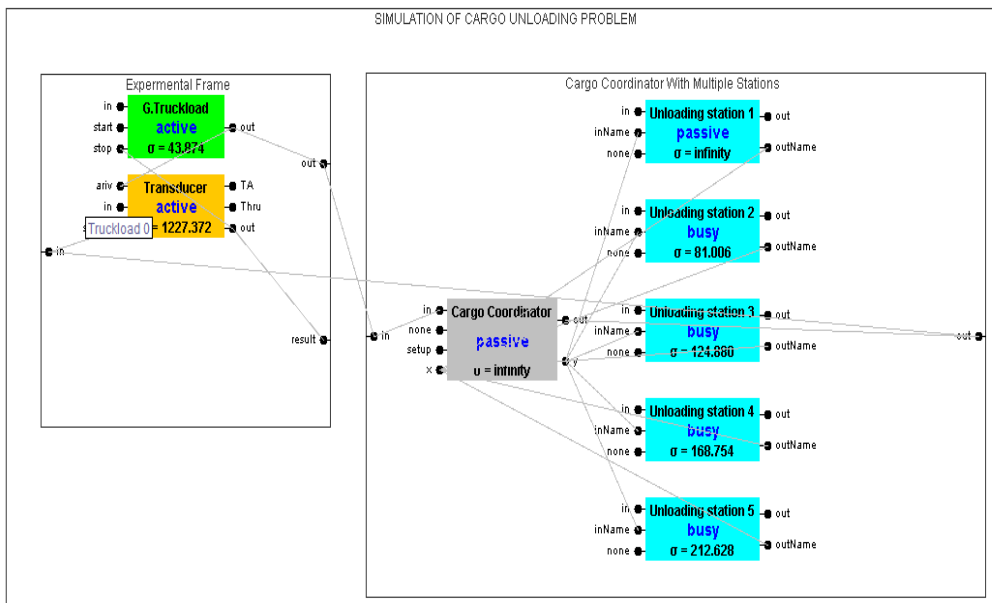


Figure 7 Step 4: send the processed truckload information from (Cargo Coordinator) model to (Transducer)

### 5.1 The metrics used in the simulation

It is assumed that this goal is achieved by finding the optimal number of trucks and cranes required to unload cargo. All stations must be in use to ensure that there are no idle drivers so that if drivers stop working, there is a loss when paid work wages and salaries for them without work. Accordingly, to find the value of this measure by simulation, it is assumed that the optimal number of trucks and cranes required is (10), then running the simulation for a month and extracting the available calculations (turnaround time, throughput), then repeating the same scenario if the number is (5) and finally compare the results obtained from the simulation of the three models.

## 6. Results And Discussion

This section contains simulation results for three models according to the number of unloading stations, with charts showing each model's throughput.

### 6.1 Simulation results of a model with ten (unloading stations)

Table 1 Simulation results with ten unloading stations

Metric	Value
The total truckload arrived	986
The total truckload solved	986
Average turnaround time (TA)	212.6279416460899
Throughput	0.02270396289958388
Iteration	3946
Time	43428.5417202682

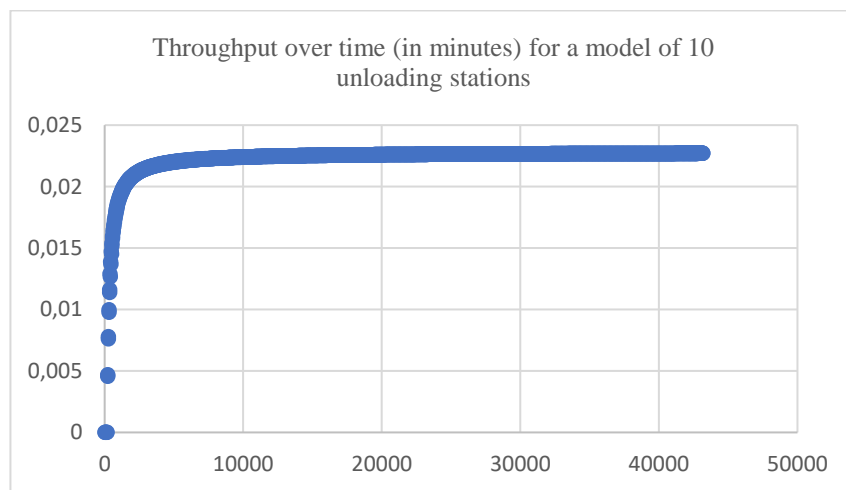


Figure 8 Throughput over time (in minutes) for a model of 10 unloading stations

As shown in Figure 8. We can see that the result gives high throughput on the 0.015 clocks even if it's good throughput, but in this case, we have wasted the workers. We will get at least four workers as idle workers. Results are summarized in Table 1.

### 6.2 Simulation results of a model with five (unloading stations)

Table 2 Simulation results with five unloading stations

Metric	Value
The total truckload arrived	986
The total truckload solved	986
Average turnaround time (TA)	212.6279416460899
Throughput	0.02270396289958388
Iteration	3946
Time	43428.5417202682



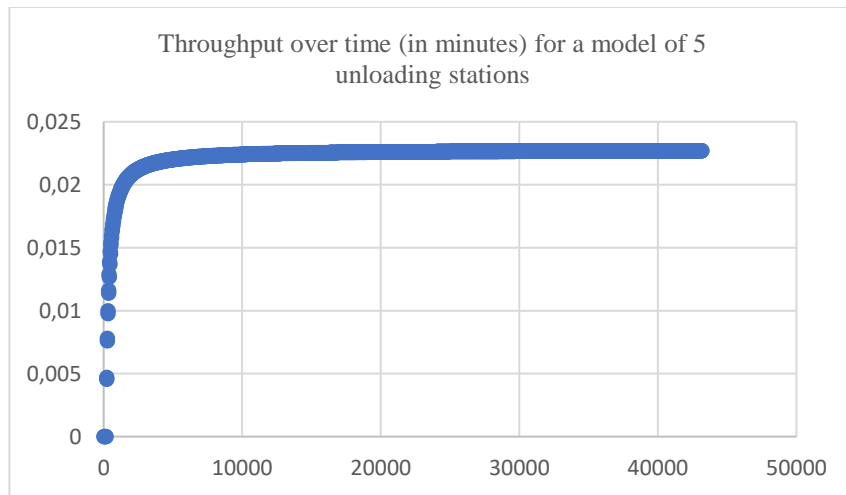


Figure 9 Throughput over time (in minutes) for a model of 5 unloading stations

Figure 9 shows that the time is very close to the ten unloadings because our simulation depends on the milliseconds. When we put the real-time, it is possible to recognize the real difference. Compared to the previous simulation, this case (5 unloadings) is better and closer to the optimum condition of the project because we always have one standby unloaders to cover the difference between each two unloading of a total of 5 cars. The results of this case are summarized in Table 2.

### 6.3 Simulation results of a model with three (unloading stations)

Table 3 Simulation results with three unloading stations

Metric	Value
The total truckload arrived	986
The total truckload solved	986
Average turnaround time (TA)	212.62794164609028
Throughput	0.013631588272366794
Iteration	2764
Time	43428.5417202682

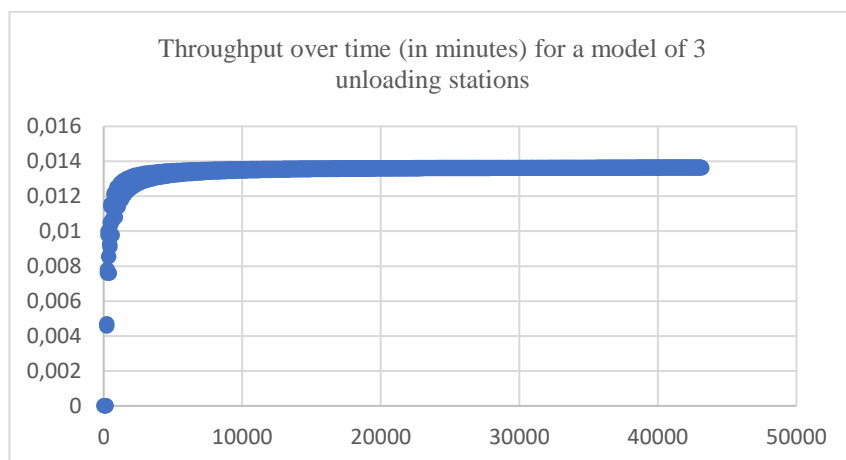


Figure 10 Throughput over time (in minutes) for a model of 3 unloading stations

Figure 10 shows that the throughput it's more intense in the clock 0.01 than the Figures 8. and 9. This is because the simulation iteration was less than the simulation iteration for the cases 10 and 5 unloadings, which means the longest simulation time shows us more accurate results than the shorter simulation time. Table 3 gives a summary of the results in this case.

## 7. Conclusion

According to the charts and tables above, if ten trucks and cranes are assigned to each truck with its driver, about half of the trucks would be idle, which will result in a loss of pay for five drivers who have no or very little work. Additionally, if three trucks and cranes exist, productivity will be reduced. Therefore, based on the simulation metric that was proposed, the model with five unloading stations will be in a state of permanent work and high productivity, which means that it is the optimal number of trucks and cranes according to the parameters specified in this study is (5) trucks and cranes for each truck with their drivers.

The contribution of this study is the possibility of building an application based on a DEVS-suite simulator to simulate the process of unloading cargo so that this application was used to calculate the optimal number of cranes and trucks required for the operation of unloading cargo by applying multiple scenarios and finding what achieves the goal, which is here a permanent state of work and high productivity without loss of wages.




In future work, the ideal number and type of trucks will be investigated in case there are various truck sizes (big, medium, and small) with a fixed crane size.

## References

- [1] F. Ramadhan, A. Imran, A. Rizana and L. Okdinawati, "Agent-based modeling and simulation for ship unloading processes: determining the number of trucks and container cranes," *International Journal for Simulation and Multidisciplinary Design Optimization*, vol. 11, no. 13, 2020.
- [2] M. Stojaković and E. Twrdy, "A Simulation Approach to the Definition of the Subsystems Parameters in Small Container Terminals," *Journal of Marine Science and Engineering*, vol. 9, no. 9, pp. 1023, 2021.
- [3] W. Menner, "Introduction to modeling and simulation". *Johns Hopkins APL Technical Digest*, pp. 6-17, 1995.
- [4] S. Temizer, "The state of the art and the future of modeling and simulation systems," *Journal of Aeronautics and Space technologies*, pp. 41-50, 2007.
- [5] O. Benedettini and B. Tjahjono, "Towards an improved tool to facilitate simulation modeling of complex manufacturing systems", *International Journal of Advanced Manufacturing Technology*, vol. 43, no. 1, pp. 191-199, 2009.
- [6] A. Zengin and H. Sarjoughian, "DEVS-Suite simulator: A tool teaching network protocols," *Proceedings - Winter Simulation Conference*, pp. 2947-2957, 2010.
- [7] M. Pandey, "Computer modeling and simulation of human movement," *Annual review of biomedical engineering*, vol. 3, no. 1, pp. 245-273, 2001.
- [8] C. Brigas, "Modeling and simulation in an educational context: Teaching and learning sciences," *Research in Social Sciences and Technology*, vol. 4, no. 2, pp. 1-12, 2019.
- [9] J. Gray and B. Rumpel, "Models for the digital transformation," *Software & Systems Modeling*, vol. 16, no. 2, pp. 307-308, 2017.
- [10] S. Kim, "Simulator for Service-based Software Systems: Design and Implementation with DEVS-Suite," *Doctoral dissertation, Arizona State University*, 2008.
- [11] B. Zeigler and S. Hessam, "Introduction to devs modeling and simulation with java: Developing component-based simulation models," *Technical Document, University of Arizona*, 2003.
- [12] H. Sarjoughian, C. Zhang and G. Scherer, "DEVS-Suite Simulator Guide: TestFrame and Database," 2020.
- [13] J. Banks, J. Carson, B. Nelson, and D. Nicol, "Discrete-event system simulation – fifth edition," *Pearson*, 2010.

- [14] M. Stojaković and E. Twrđy, "Determining the optimal number of yard trucks in smaller container terminals," *European Transport Research Review*, vol. 13, no. 1, pp. 1-12, 2021.
- [15] O. Elrajubi, "Study Case of a Simulation for the Transportation System Problem," in Arabic, *Libya, Misrata*. 2010.
- [16] H. Sarjoughian and S. Gholami, "Action-level real-time DEVS modeling and simulation," *Simulation*, vol. 91, no.10, pp. 869-887, 2015.
- [17] J. Pérez and Á. Alejandro. "*Computer Modeling & Simulation*", Ed. FUOC. ISBN: 978-84-692-9597-7, 2016.
- [18] E. Ahmad and H. Sarjoughian, "An AADL-DEVS Framework for Cyber-Physical Systems Modeling and Simulation Supported with an Integrated OSATE and DEVS-Suite Tools," 2020.
- [19] A. Markid, "Modeling And Simulation of Genetic Algorithm Using Meta-Models," *Journal of Hyperstructures*, vol. 7, no. 1, 2019.
- [20] S. Kara, S. Hizal and A. Zengin, "Design and Implementation of A Devs-Based Cyber-Attack Simulator for Cyber Security," *International Journal of Simulation Modelling (IJSIMM)*, vol. 21, no. 1, pp. 53-64, 2022.

# Sector-Based Stock Price Prediction with Machine Learning Models

 Doğangün Kocaoğlu<sup>1</sup>,  Korhan Turgut<sup>2</sup>,  Mehmet Zeki Konyar<sup>3</sup>

<sup>1</sup>Kocaeli University, Department of Computer Engineering; dogangun82@gmail.com

<sup>2</sup>Kocaeli University, Department of Computer Engineering; ateshan@yahoo.com

<sup>3</sup>Corresponding Author; Kocaeli University, Department of Software Engineering; mzeki.konyar@kocaeli.edu.tr

Received 7 November 2022; Accepted 30 November 2022; Published online 31 December 2022

## Abstract

Stock price prediction is an important topic for investors and companies. The increasing effect of machine learning methods in every field also applies to stock forecasting. In this study, it is aimed to predict the future prices of the stocks of companies in different sectors traded on the Borsa Istanbul (BIST) 30 Index. For the study, the data of two companies selected as examples from each of the holding, white goods, petrochemical, iron and steel, transportation and communication sectors were analyzed. In the study, in addition to the share analysis of the sectors, the price prediction performances of the machine learning algorithm on a sectoral basis were examined. For these tests, XGBoost, Support Vector Machines (SVM), K-nearest neighbors (KNN) and Random Forest (RF) algorithms were used. The obtained results were analyzed with mean absolute error (MAE), mean absolute percent error (MAPE), mean squared error (MSE), and  $R^2$  correlation metrics. The best estimations on a sectoral basis were made for companies in the Iron and Steel and Petroleum field. One of the most important innovations in the study is the examination of the effect of current macro changes on the forecasting model. As an example, the effect of the changes in the Central Bank Governors, which took place three times in the 5-year period, on the forecast was investigated. The results showed that the unpredictable effects on the policies after the change of Governors also negatively affected the forecast performance.

**Keywords:** Borsa Istanbul, machine learning, stock price prediction,

## 1. Introduction

Persons and institutions investing in the capital market should know and follow the market in which they invest. Therefore, all individual and institutional investors are required to make market forecasts by providing accurate and fast all economic and financial information about the general economy, sectors and the institutions they invest in. However, the difficulty of predicting people's feelings and expectations reduces the chances of any analysis system that can be considered fully successful. In addition, the fact that the people who set the prices (market professionals, institutional investors, speculators, manipulators) have different cultural, educational and knowledge structures make the situation even more difficult. There are different methods for stock price analysis in the literature. Fundamental analysis, technical analysis and statistical forecasting methods are the most frequently used methods [1].

In fundamental analysis, which is the most comprehensive method used in the evaluation of stocks, the actual value of the stock is tried to be calculated by considering all possible factors that may affect the value of stocks. The factors that can affect the value of the stock can be grouped under three main titles: economic analysis, sector analysis and firm analysis. As a result of these analyzes, the risk and return relationship of the stock is revealed and its real value is calculated with the help of various methods. If the calculated real value is higher than the market value, the share is purchased; otherwise, it will not be traded. In fundamental analysis, there are stages such as economic analysis, sector analysis, firm analysis, risk and return estimation, respectively. The first stage of fundamental analysis is economic analysis. With this analysis, it is checked whether the general economic conjuncture is suitable for investing in stocks affected by general economic conditions. When a positive result is obtained from the economic analysis for stock investment, the second stage of the fundamental analysis, the industry analysis, is started. At this stage, it is tried to determine which of the many sectors in the economy should be invested in.

In technical analysis, it is aimed to determine the direction of the market and stock prices by using certain market data. Market data used in technical analysis consists of stock market index or price transaction volume information for stocks. Technical analysts try to predict the future direction of stock prices based on past market data. With statistical forecasting methods, time series analysis is used to predict the future, and it is tried to determine the attitudes of the series towards the future outside the forecast period. The traditional time series analyzes past data and tries to calculate its future approximation in the form of linear combinations of this historical data. In other words, a model is tried to be established in relation to the past values of the nonlinear values of a variable [2].

Professional knowledge and skills are very important for analyzes made with traditional methods. It is necessary to evaluate many parameters together and to read the behavior of the market from past to present. So, in recent years, it has become very popular to use artificial intelligence methods to make stock analysis and forecasting processes faster and easier. Models trained with the features in the datasets make successful predictions in the face of a situation they have never seen before. Modeling of stocks has become easier thanks to machine learning and deep learning algorithms [3],[4].

In this study, it is aimed to predict stock prices by machine learning method. In the literature, it has been seen that machine learning and deep learning algorithms such as Artificial Neural Networks, Random Forest, XGBoost, SVM, KNN and Long Short-Term Memory are used for prediction [5]. In this study, Random Forest, XGBoost, SVM and KNN, which are the most used machine learning algorithms, were used. It has been observed that the existing studies in the literature generally predict BIST-30 or BIST-100 index, and when stock-based prediction are made, stocks are randomly selected. The most important contribution of this study proposed in this article to the literature is to make a sector-based estimation. Another important contribution of our study is the analysis of the effect of some specific periods, such as the change of the Central Bank governors, on the stock price prediction with the machine learning.

The rest of the paper is organized as follows; section 2 the similar literature studies are summarized. Details of the machine learning methods are given in section 3. The proposed methods and experimental results are given in section 4. In the last section the conclusions are summarized.

## 2. The Related Studies

The increasing effect of machine learning methods in every field has also become important for stock forecasting. In this section, some of the existing machine learning and deep learning studies in the literature are summarized.

In the [6], the price estimates of the stocks of 5 Turkish Banks were made according to the stock market. By using various indicator values of the shares, estimation was made with decision tree, multiple regression, and random forest machine learning models. The success of the estimation results obtained was evaluated with the  $R^2$  metric and values between 0.95-0.98 were reached. In the [7], the direction of change of the BIST 50 index was estimated with artificial neural networks (ANN), KNN, Naive Bayes and C4.5 decision tree models. The success of the estimation results obtained was evaluated with the classification accuracy, and the highest value was obtained as 92.71% with the C4.5 decision tree model. In the [8], the SP500 stock market index was estimated using a CNN-based forecasting model. In the study, an answer was sought on how to use the convolutional neural networks (CNN) model in stock market forecasting and how to optimize it. For the estimation process, 4 different CNN models based on different parameters were used. The obtained results were compared with the support vector machine (SVM) model and artificial neural network (ANN) model. In [9], daily returns of stocks in the Macedonian Stock Exchange were tried to be estimated based on linear regression and correlation analysis. The daily statistical forecast values of the stocks were evaluated over the  $R^2$  metric.

In the study of [10], a forecasting system is proposed for stocks in NYSE, NASDAQ and NYSE MKT stock exchanges using deep learning models LSTM (Long Short-Term Memory), Gated Recurrent Unit (GRU) and Bidirectional LSTM. Experimental results were obtained with the BLSTM model with an accuracy of 63.54%. According to [11] a study was conducted on Coca-Cola Company shares. The study aimed to determine whether SVM is more accurate than Linear Regression. The estimation results were evaluated using Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute

Percent Error (MAPE), Mean Square Error (MSE) and Correlation Regression ( $R^2$ ) and nonlinear multiple correlation factor evaluation criteria. As a result of the analysis, it was seen that SVM achieved more accurate results than LR. In the method of the [12], it is aimed to overcome the difficulty of forecasting crude oil prices due to the chaotic behavior of the time series. In the study, a wrapper-based feature selection approach using the multi-objective optimization technique and a support vector regression (SVR)-based prediction model are proposed. In the proposed model, features based on technical indicators such as simple moving average (SMA), exponential moving average (EMA) and Kaufman's adaptive moving average (KAMA) are used.

A data set was created by collecting the financial values of 22 companies traded on BIST-30 between the years 2010-2019 in the [13]. The stock prices of the companies were estimated using Artificial Neural Networks (ANN), Random Forest (RF) and XGBoost algorithms. The estimation results were compared over the  $R^2$  metric, and the highest value was obtained with XGBoost as 0.758. In the study of [14], price prediction was made on the trading data on the Bitcoin stock market. The Linear Regression model was used for the estimation process and the results were evaluated over the  $R^2$  metric. In the study of [15], stock price prediction was made by using machine learning and deep learning methods. Polynomial Regression, Random Forest Regression, Recurrent Neural Networks (RNN) and Long-Short-Term Memory (LSTM) methods were used for estimation. The estimation results were evaluated with RMSE, MAE, MSE metrics. The lowest error value was obtained with the Random Forest Regression model and the highest error value was obtained with the Polynomial Regression model. In addition to financial values, the authors of the [16] predicted stock market trends by making use of gold and oil prices. LSTM and CNN models were used to classify the data of SP500 index and compare investment returns. In experimental studies, the accuracy rate of the model increased up to 67%. In addition, it has been determined that this method will provide a return of around 13% with the investment. A 2-dimensional CNN-based forecasting approach is proposed for stocks in the Dow30 index proposed in the [17]. In the created rule-based model, the next day's buying, selling, or holding position of the stock is tried to be estimated.

Various machine learning algorithms were used for empirical asset pricing on the Chinese stock market in [18]. In the study, a comprehensive set of return estimation factors was created and analyzed. On the dataset, least absolute shrinkage and selection operator (LASSO), ordinary least squares (OLS) regression, partial least squares regression (PLS), gradient boosted regression trees (GBRT), elastic net (Enet), random forest (RF), variable subsampling Estimation aggregation (VASA), and neural networks methods based predictions were made. The prediction performance of the models was examined through the  $R^2$  metric in the predictions. In addition, predictability between different sub-samples was evaluated. In the [19], it is aimed both to predict the price of the stock and to compare the results obtained with Kalman filters, XGBoost and Auto Regressive Integrated Moving Average (ARIMA) models. We also compare the results of four models, including a hybrid model combining Kalman filters and XGBoost, to predict the price of New York Stock Exchange (NYSE) and National Stock Exchange (NSE) stocks. In the comparison, the lowest accuracy was obtained with the NYSE data set of the Kalman filter model as 64.96%. The highest accuracy was found to be 90.11% with the NYSE dataset of the XGBoost model.

### 3. The Machine Learning Methods

In this section, the machine learning methods used in the article will be briefly explained. In the article, four different regression methods were used for the stock analysis process. For the regression of the stock price prediction Support Vector Machines (SVM), K-nearest neighbors (KNN), XGBoost and Random Forest (RF) based algorithms were used.

#### 3.1 Support Vector Machine (SVM) Regression

Support Vector Regression is a regression model which uses the SVM approach that supports both linear and nonlinear regression operations. SVM based machine learning algorithm is used for both regression and classification problems. In the SVM, each data element is plotted as a point in the corresponding space such that the value of each attribute is a certain value in the coordinate system. Then, the

classification is made by obtaining the hyperplane that best separates the two classes. Support Vector Regression works according to the SVM principle with minor differences [20]. The points given in the data are used to find the regression curve. Since the process is done with a regression algorithm, the curve obtained is not used as a decision limit, but to obtain the match between the vector and the curve. In normal regression, the aim is to minimize the error rate. In SVR, it is aimed to fit the error within a threshold. Therefore, the SVR model is used to estimate the best value for data in a given range.

### 3.2 K-Nearest Neighborhood (KNN) Regression

Although linear regression cannot provide very precise estimates of time, it is very useful for some critical problems. Thanks to the linear regression approach, many alternative models such as KNN have emerged that can be used in the field of machine learning. The KNN algorithm is a supervised learning algorithm in which a target variable is estimated using one or more independent variables [21]. Regression is the construction of a predictive function in which the target variable is numeric. Some algorithms can only classify, some can only regress, and some can do both classification and regression. The KNN algorithm adapts seamlessly to both classification and regression.

### 3.3 Extreme Gradient Boosting (XGBoost) Regression

XGBoost as a community learning method has been showing significant results recently. Relying on the results of a single machine learning model is not enough for some critical prediction and classification operations. Community learning offers more valuable results for combining the predictive power of multiple students. This result is achieved with a single model that consists of several models and gives a collective output. In the strengthening phase of the model, trees are created sequentially, so each subsequent tree aims to reduce the errors of the previous tree. Each tree learns new information from its predecessors, reducing existing errors. Therefore, a tree in a queue will learn from an updated version of what remains. The basic learners used for support are weak learners and their power for prediction is slightly better than random prediction. Each of weak learners contributes some critical information for prediction. Despite the weaknesses of these learners, by combining them effectively, the reinforcement technique reveals a powerful learning method. Parameters such as the number of trees or iterations, the depth of the tree can be optimally selected through validation techniques such as the learning rate of gradient boosting and k-fold cross validation [22].

### 3.4 Random Forest (RF) Regression

This regression is a collection of prediction trees based on the RF classifier. Each tree has a similar distribution to other trees in the random forest and depends on independently sampled random vectors. Random forest is an RF-based modeling technique used in behavior analysis and predictions. It contains several decision trees that represent a different classification example of random forest data entry. The random forest technique considers the samples one by one and takes the most voted sample as the selected prediction. Each tree in the classifications receives its inputs from the samples in the first dataset. Then, randomly selected features are used to grow the tree at each node. Each of the trees in the forest should not be pruned until the end of the exercise until the estimate is reached with certainty. In this way, the random forest ensures that any classifier with weak correlations will be a strong classifier. The random forest technique can also process big data with thousands of variables. A class can automatically balance datasets when data is less sparse than other classes[23].

## 4. The Proposed Method and Experimental Studies

The flow chart of the proposed method of this study for the analysis of stocks of different sectors given in Figure 1. First, the data set was collected and prepared by examining the data of two companies selected as examples from the Holding, White Goods, Petrochemical, Iron and Steel, Air Transport and Communication sectors. In Table 1, the company shares, and the sectors of the companies used for the experimental studies are given. The end-of-day closing prices of the stocks, which are the output of the

models used in this study, are taken from the publicly available data on the website of IS Investment (<https://www.isyatirim.com.tr>) [24] for the last five-year period (01.10.2016 – 30.09.2021).

Table 1 Stock dataset information

Stock Name	Sector
SAHOL	Holding
KCHOL	Holding
EREGL	Iron and Steel
KRDMR	Iron and Steel
TUPRS	Petrochemical
PETKM	Petrochemical
ARCLK	White Goods
VESTL	White Goods
TRCELL	Communication
TTKOM	Communication
THYAO	Air Transport
PGSUS	Air Transport

The financial statement data used in the technical analysis of the companies were obtained from the website of IS Investment. Various firm ratios (current ratio, acid-test ratio, cash ratio, net profit margin) revealed by these data are used as inputs in our model. Macro data such as policy rate, inflation, and USD/TL parity used in the fundamental analysis of the companies were obtained retrospectively from institutional websites such as the Central Bank of the Republic of Turkey (TCMB) and the Turkish Statistical Institute (TUIK).

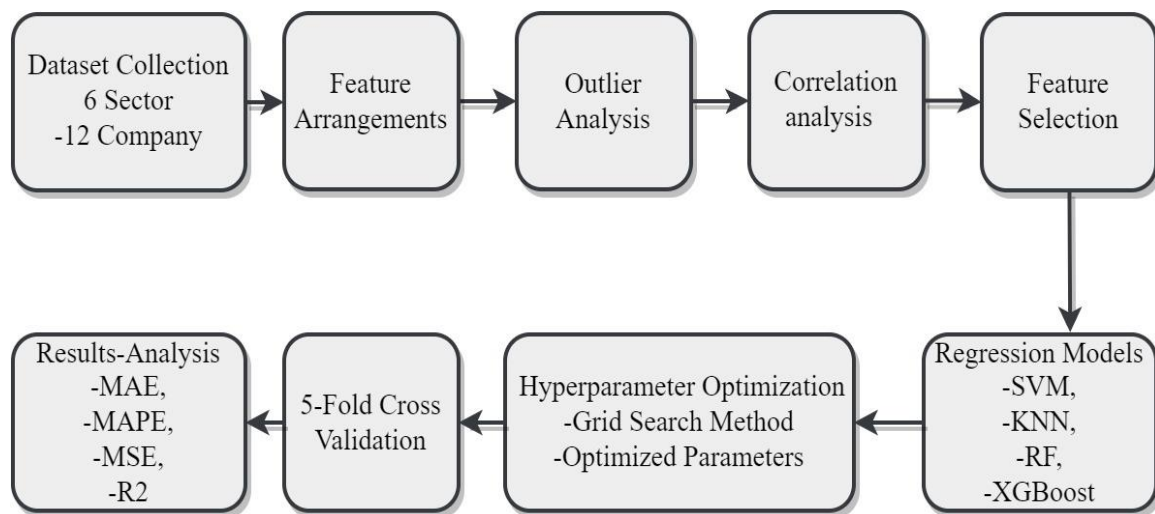


Figure 1 Flow chart of the proposed method

The frequency with which data is disclosed to supply the forecast model varies. Therefore, an imbalance arises in the training of machine learning models. To eliminate this imbalance, certain arrangements have been made to convert the data to the ones with the highest frequency. Since the interest rate data are shared weekly by the TCMB, the announced weekly rate has been retrospectively accepted as the same every day for seven days. Inflation rate data is shared monthly by TUIK, and the announced monthly rate has been accepted as the same every day for thirty days retrospectively. Since the USD/TL exchange rate data is announced by the TCMB every weekday, the daily data is used without any changes. The financial data of the companies are published quarterly, and the announced quarterly data has been retrospectively accepted as the same every day for ninety days. All the data studied within the scope of the article are for the time interval of 01.10.2016 – 30.09.2021 and weekday data were used. The characteristics and explanations of the data used in the study are given in Table 2. The data used are 10 types for each day and include values for a total of 1256 days.



Table 2 Features used for stock analysis

Feature Name	Description
Date	Transaction day
Closing Value (TL)	The closing price of the stock
Volume (TL)	Multiplying the total amount of trading in the relevant stock during the day with the price of the stock at the time of the transaction.
Market Value (mn TL)	The value found by multiplying the share price of the company with the total number of shares
Current Rate	The ratio of current assets of the company to short-term liabilities
Acid Test Ratio	The ratio of the value resulting from the deduction of company stocks from current assets of the company to short-term liabilities
Cash Ratio	Ratio of cash and cash equivalents of the company to short-term liabilities
Net Profit Margin	The ratio of the company's net profit to its net sales
Dollar exchange rate	USD/TL rates announced by the TCMB every weekday
Interest rates	Interest rates announced weekly by the TCMB
Inflation Value	Inflation rates announced monthly by TUIK

Some operations were performed to select the best features to use in the prediction model. First, the 'Date' variable has been removed since the period that the data represents is now known. Data for all features and all companies were examined in detail, and missing value and outlier values were checked. Values that are far outside the general limits of the data are considered outliers. Outlier analysis on our data set in the proposed study was evaluated with the box plot approach, and some results are given in Figure 2. As a result of the examinations made on both the features in the data set and the figures, it has been determined that all the features are numerical and there are no missing or outlier values.

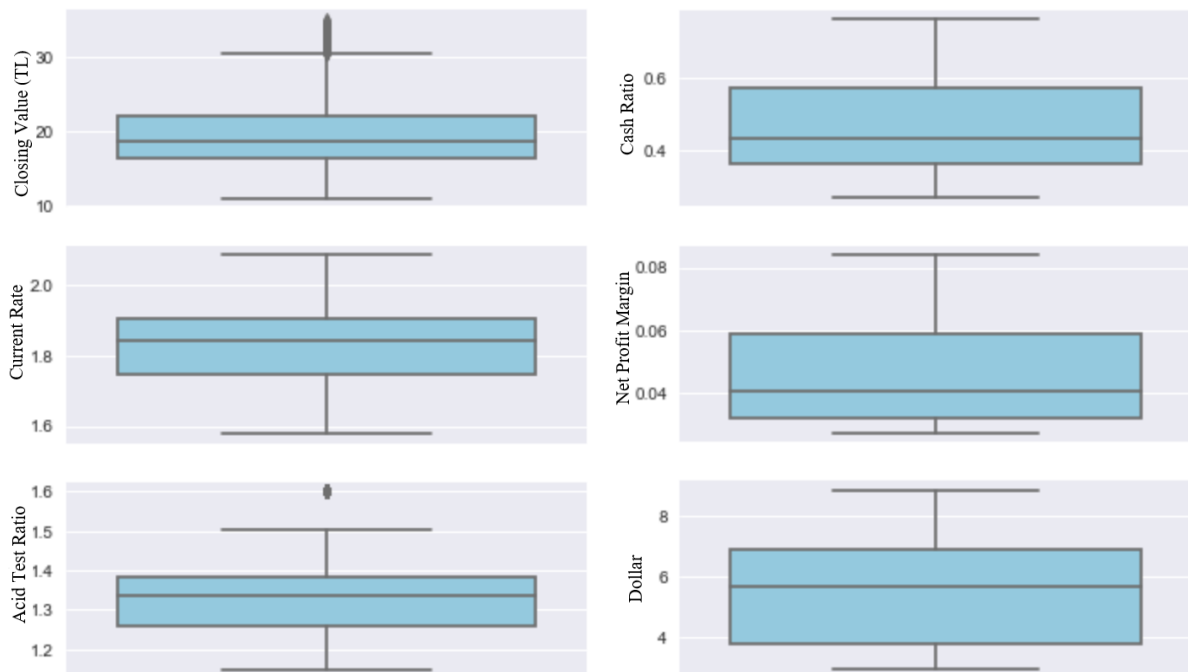


Figure 2 Outlier analysis with boxplot of the features

In order to prevent the machine learning model from being exposed to the multicollinearity problem caused by similar features, correlation analysis between features was performed. The results of the correlation analysis are given in Figure 3. Variables with more than 80% correlation in Figure 3 were excluded from the analysis in order not to cause multicollinearity problems. According to Figure 3, there is a 99% correlation between the closing prices and the Market value. There is a 91% correlation between the current rate and the Acid test rate. There is an 87% correlation between the inflation rate and the

interest rates. Market value, Acid test rate and Inflation rate data were removed from the data to eliminate the multicollinearity problem.

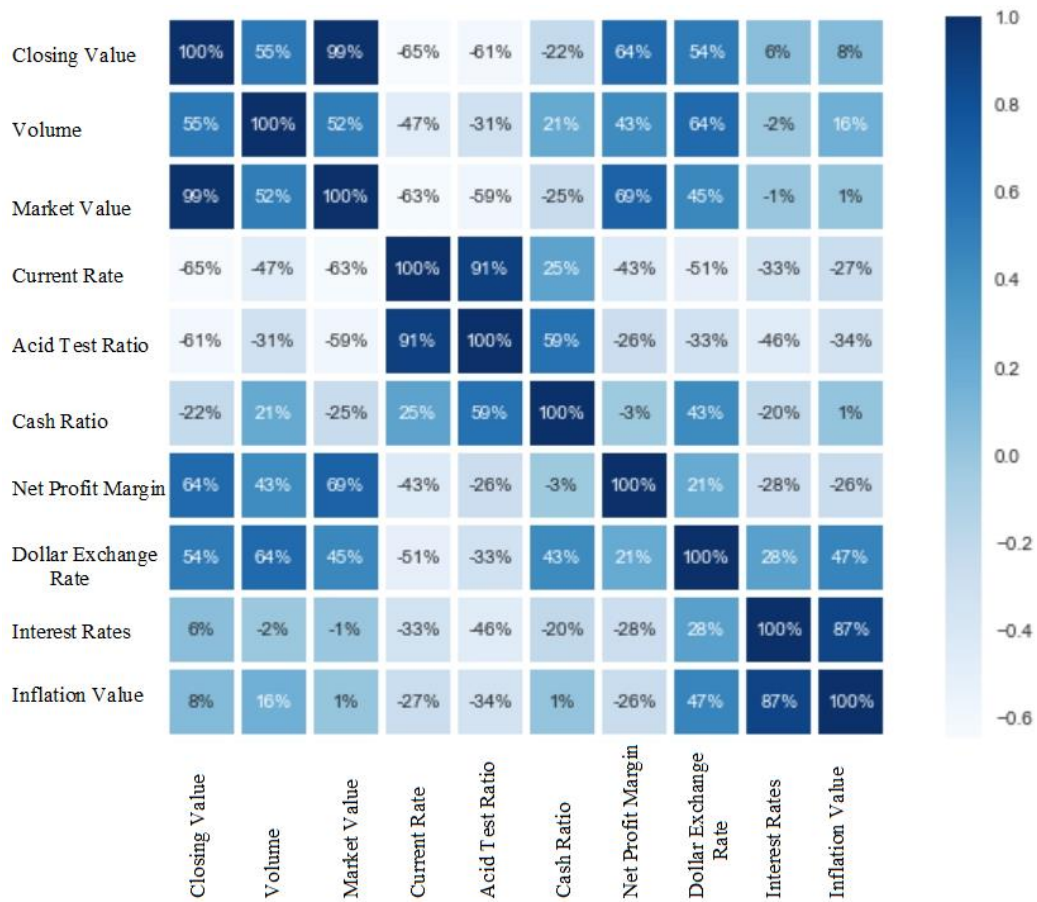


Figure 3 Correlation analysis results of the features

To obtain the experimental results in the proposed study, all algorithms are coded in Python language. The data obtained were separated as training and test data, and it was tried to determine which of them was more successful in which sector by using various algorithms. For the prediction model to classify the data it will see for the first time more successfully, 5-fold cross validation is used on the training set.

Table 3 The hyperparameter space for the machine learning models

Model	Hyperparameter Space
SVM	C: [0.1, 0.5, 1, 1.5] epsilon: [0.01, 0.1, 1] gamma: ['auto'] kernel: ['linear', 'poly', 'rbf'] degree: [2,3,5]
KNN	n_neighbors: [4-20] p: [1,2,3]
XGBoost	learning_rate: [0.01, 0.02, 0.09] The maximum depth: [2, 3, 4, 5, 6] Number of estimators: [100, 200, 500, 2000]
RF	The maximum depth: [80, 90, 100, 110] max_features: [2, 3] min_samples_leaf: [3, 4, 5] min_samples_split: [8, 10, 12] Number of estimators:[100, 200, 300, 1000]

In this approach, the training set is divided into 5 equal parts, each time 4 of these parts are used for model training and one for validation [25][26]. In this study, the dataset is divided into two parts as 70% training and validation set, and 30% test set. After the training process was over, the final success of the model was tested on the test set that aside and never encountered in the training process. SVM, KNN, XGBoost and RF regression models were used to predict future prices of stocks. The performance of regression models is directly related to the hyperparameters selected for the model to run. For the optimization of algorithms, it is very important to find the most optimal parameters instead of directly giving values. In this study, to find the optimum parameters, the parameters with the highest accuracy were selected automatically from the hyperparameter space in Table 3 with the grid search approach.

$$MAE = \frac{1}{N} \sum_{k=1}^N |y_k - \hat{y}_k| \quad (1)$$

$$MAPE = \frac{1}{N} \sum_{k=1}^N \frac{|y_k - \hat{y}_k|}{\max(\epsilon, |y_k|)} \quad (2)$$

$$MSE = \frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2 \quad (3)$$

$$R^2 = 1 - \frac{\sum_{k=1}^N |y_k - \hat{y}_k|}{\sum_{k=1}^N |y_k - \bar{y}_k|} \quad (4)$$

$$\bar{y}_k = \frac{1}{N} \sum_{k=1}^N y_k \quad (5)$$

For the performance measurement of the experimental results, the real values of the stocks and the values predicted by the model were compared. Mean absolute error (MAE), mean absolute percent error (MAPE), mean square error (MSE), and  $R^2$  metrics were used for comparisons.  $y_k$  k. actual value,  $\hat{y}_k$  k. The metrics used to be the predicted value are given in Equation 1 to Equation 4 for test set size N. A small positive number  $\epsilon$  is defined to prevent the value of the MAPE score from going to infinity where  $y_k$  is zero.

A lower value in the MAE, MAPE, and MSE metrics indicates a better estimate. The value of the  $R^2$  score, which is the most popular metric in linear regression models, is usually between 0 and 1, although sometimes it can be negative. At the highest regression fit, the value of the  $R^2$  score approaches 1. The  $\bar{y}_k$  used in calculating  $R^2$  represents the average of all real values as shown in Equation 5.

Table 4 MAPE results for all models of stock price prediction algorithms

Stock Name	Prediction Models			
	SVM	KNN	XGBoost	RF
SAHOL	0,035	0,027	0,023	0,023
KCHOL	0,052	0,031	0,023	0,025
EREGL	0,054	0,035	0,029	0,032
KRDMR	0,061	0,041	0,037	0,043
TUPRS	0,078	0,028	0,025	0,026
PETKM	0,047	0,029	0,024	0,027
ARCLK	0,053	0,027	0,022	0,024
VESTL	0,084	0,045	0,043	0,044
TRCELL	0,037	0,024	0,022	0,023
TTKOM	0,046	0,031	0,024	0,028
THYAO	0,081	0,046	0,028	0,031
PGSUS	0,155	0,049	0,046	0,047
<b>Average</b>	<b>0,065</b>	<b>0,034</b>	<b>0,029</b>	<b>0,031</b>

The experimental results obtained in the tests carried out based on companies and sectors are given below. First, MAPE error values for all forecasting models are given in Table 4. A low MAPE value

indicates that the best estimation is made with the relevant regression model. According to Table 4, when the average MAPE values were examined, the lowest average was obtained as 0.029 for XGBoost. The second-best performance was obtained with RF as 0.031, while the lowest estimation results were obtained with SVM as 0.065. Although the estimation values on a share basis are close to each other, the shares of TRCELL and ARCLK companies reached the best estimation with MAPE values of 0.022.

Secondly, in the experimental results, all prediction models were evaluated in terms of the  $R^2$  metric. In Table 5, the values of the  $R^2$  metric obtained by estimating the shares for four different models are given. The high value of  $R^2$  indicates that the best estimation is made with the relevant regression model. When the average  $R^2$  values are examined according to Table 5, the best value was obtained as 0.989 for XGBoost. The second-best performance was obtained with RF as 0.987, while the lowest predictive value was obtained with SVM as 0.946. While the shares of seven companies have  $R^2$  values of 0.990 and above, the shares of EREGL, KRDMR and VESTL have reached  $R^2$  values of 0.995 and above. When the results in Table 5 are analyzed by sector, it is seen that the models used make the best estimation for the iron and steel sector and the white goods sector.

Table 5  $R^2$  results for all models of stock price prediction algorithms

Stock Name	Prediction Models			
	SVM	KNN	XGBoost	RF
SAHOL	0,920	0,949	0,965	0,966
KCHOL	0,902	0,961	0,981	0,976
EREGL	0,987	0,992	0,996	0,995
KRDMR	0,987	0,991	0,995	0,993
TUPRS	0,888	0,985	0,988	0,988
PETKM	0,971	0,985	0,993	0,991
ARCLK	0,970	0,990	0,994	0,993
VESTL	0,978	0,993	0,995	0,994
TRCELL	0,968	0,985	0,989	0,988
TTKOM	0,951	0,976	0,988	0,983
THYAO	0,911	0,972	0,990	0,987
PGSUS	0,917	0,986	0,992	0,991
<b>Average</b>	<b>0,946</b>	<b>0,980</b>	<b>0,989</b>	<b>0,987</b>

Among the forecasting models, the best results were obtained with the XGBoost model, both on a sectoral and company basis. Therefore, XGBoost has come to the fore as the most successful model. In Table 6, the estimation results of all companies' stocks with the XGBoost model are evaluated in terms of MAE, MAPE, MSE and  $R^2$  metrics. MAE, MAPE and MSE values are preferred to be low as they indicate estimation errors. According to Table 6, MAE and MSE values of TUPRS and PGSUS stocks are quite high compared to other stocks.

Table 6 Results of forecasting stock prices with the XGBoost model for all metrics

Stock Name	Metrics			
	MAE	MAPE	MSE	$R^2$
SAHOL	0,190	0,023	0,069	0,965
KCHOL	0,366	0,023	0,270	0,981
EREGL	0,217	0,029	0,113	0,996
KRDMR	0,116	0,037	0,032	0,995
TUPRS	2,469	0,025	11,153	0,988
PETKM	0,086	0,024	0,016	0,993
ARCLK	0,424	0,022	0,352	0,994
VESTL	0,386	0,043	0,314	0,995
TRCELL	0,242	0,022	0,106	0,989
TTKOM	0,133	0,024	0,032	0,988
THYAO	0,341	0,028	0,233	0,990
PGSUS	1,722	0,046	6,996	0,992
<b>Average</b>	<b>0,558</b>	<b>0,029</b>	<b>1,640</b>	<b>0,989</b>

The lowest MAE values were obtained for PETKM, KRDMR and TTKOM shares as 0.086, 0.116 and 0.133, respectively. The lowest MSE values were also 0.016, 0.032 and 0.032 for the same shares,

respectively. Considering all metrics and evaluated on a share basis, the best estimation is made in KRDMR shares. According to error metrics, the worst estimation is for PGSUS stocks. For the sector-based evaluation, the best estimations were made for the companies in the Iron-Steel and Petroleum field.

One of the most important contributions of the proposed study is to examine the effect of current macro changes on the forecasting model. As an example, the effect of the changes in the Central Bank Governors, which took place three times in the 5-year period, on the forecast was investigated. To include this variable in the model, it is assumed that the effect on the market will continue for three months from the date of the change of Governors. A new synthetic variable is used in addition to the previous features for the relevant 90-day period. The results after adding this variable to the model as an independent variable are shown in Table 7. As stated above, since the most successful model is XGBoost, this model has been compared. For all sectors, the effect on share prices before and after the change of Governor was evaluated over the  $R^2$  value.

As can be seen from the table, a decrease is observed in the success of stock forecasting in most companies. While the forecast success of PETKM and ARCLK stocks increased insignificantly, success decreased in SAHOL, KCHOL and TUPRS stocks. This is because the unpredictable impact of the chairman change on policies also affects forecast performance. From an economic point of view, it is thought that the information of companies with foreign exchange deficit and surplus will be important. Looking at the markets after the Central Bank chairman changes, it is seen that there was an increase in the dollar exchange rate and a decrease in stock prices.

Table 7 The effect of the Central Bank's governors changes on the stock price prediction

Stock Name	$R^2$ Before Governor Change	$R^2$ After Governor Change	Difference
SAHOL	0,9650	0,8990	-0,0660
KCHOL	0,9810	0,9345	-0,0465
EREGL	0,9960	0,9926	-0,0034
KRDMR	0,9950	0,9793	-0,0157
TUPRS	0,9880	0,9057	-0,0823
PETKM	0,9930	0,9951	0,0021
ARCLK	0,9939	0,9943	0,0005
VESTL	0,9953	0,9864	-0,0089
TRCELL	0,9892	0,9901	0,0009
TTKOM	0,9880	0,9851	-0,0030
THYAO	0,9905	0,9682	-0,0223
PGSUS	0,9922	0,9875	-0,0047

Although the increase in the dollar exchange rate seems to be a positive development in terms of balance sheet for companies with excess foreign exchange position, these stocks also decline with the effect of the general decline in the stock market index, and thus, the explanation success of the model decreases due to the share price, which is negatively affected because of the positive development. For companies with short foreign exchange position, the increase in the dollar exchange rate is a negative development in terms of the balance sheet, and these stocks also decline with the effect of the general decline in the stock market index, and thus, the share price is adversely affected because of the negative development.

#### 4. Conclusions

In this study, the prices of the stocks of companies in different sectors traded in the BIST 30 Index were examined. The data of two companies selected as examples from Holding, White Goods, Petrochemical, Iron and Steel, Air Transport and Communication sectors were examined. The study has shown that the Xgboost algorithm is the most successful algorithm based on both sectors and companies. While the RF algorithm has the second-best performance on the basis of industry and company, the worst performance belongs to the SVM algorithm. In the current literature studies, BIST30 and BIST100 index estimations are made. When stock-based forecasting was made, stocks were randomly selected. In the proposed

study of this article, the effects of some specific periods such as the fact that a sector-based analysis was made and the change of the central bank governor effect on the sectors were examined by machine learning methods. As a result of the studies, it has been determined that the machine learning-based estimation of stocks in the iron and steel, petrochemical and communication sectors has achieved more successful results.

## Acknowledgments

This study was partially carried out in the Software Technologies Research Laboratory (STAR Lab) of the Kocaeli University Software Engineering Department.

## References

- [1] I. K. Nti, A. F. Adekoya and B. A. Weyori, "A systematic review of fundamental and technical analysis of stock market predictions," *Artificial Intelligence Review*, 53(4), pp. 3007-3057, 2020.
- [2] H. Dağlı, "Sermaye Piyasası ve Portföy Analizi," 3<sup>rd</sup> Ed., *Derya Kitabevi*, Trabzon, 2009.
- [3] S. Tekin, "Destek vektör makineleri yöntemi ile İMKB 100 endeksi hareket yönü tahmini" *Uşak University Social Sciences Institute*, Master Thesis, Uşak, 2013.
- [4] U Demirel, "Hisse senedi fiyatlarının makine öğrenmesi yöntemleri ve derin öğrenme algoritmaları ile tahmini", *Giresun University Social Sciences Institute*, Master Thesis, 2019
- [5] P. Chhajer, M. Shah and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction," *Decision Analytics Journal*, 2, 100015, 2022.
- [6] Z. D. Akşehir and E. Kılıç, "Prediction of Bank Stocks Price with Machine Learning Techniques", *TBV Journal of Computer Science and Engineering*, 12 (2) , pp. 30-39, 2019.
- [7] E. Filiz, H. A. Karaboğa and S. Akoğul, "Bist-50 index change values classification using machine learning methods and artificial neural networks", *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 26(1), pp. 231-241, 2017.
- [8] H. S. Sim, H. I. Kim and J. J. Ahn, "Is Deep Learning for Image Recognition Applicable to Stock Market Prediction", *Complexity*, 4324878, 2019.
- [9] Z. Ivanovski, N. Ivanovska and Z. Narasanov, "The regression analysis of stock returns at MSE", *Journal of Modern Accounting and Auditing*, 12(4), pp. 217-224, 2016.
- [10] G. Şişmanoğlu, F. Koçer, M. Önde and O. K. Sahingoz, " Price Forecasting in Stock Exchange with Deep Learning Methods ", *BEU Journal of Science*, 9(1), pp. 434-445, 2020.
- [11] V. Gururaj, V.R. Shriya, and K. Ashwini, "Stock market prediction using linear regression and support vector machines", *Int J Appl Eng Res*, 14(8), 1931-1934, 2019.
- [12] S. Karasu, A. Altan, S. Bekiros and W. Ahmad, "A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series", *Energy*, 212, 118750, 2020.
- [13] N. K. Ustalı, N. Tosun, and Ö. Tosun, "Stock Price Forecasting Using Machine Learning Techniques", *Eskişehir Osmangazi University Journal of Economics and Administrative Sciences*, 16(1), pp. 1-16, 2021.
- [14] M. E. Arslan and P. Kırıcı, "Stock Market Analysis with Machine Learning". *European Journal of Science and Technology*, (28), pp. 1117-1120, 2021.
- [15] S. Arslankaya and Ş. Toprak, "Using Machine Learning and Deep Learning Algorithms for Stock Price Prediction", *International Journal of Engineering Research and Development*, 13(1), 178-192, 2021.
- [16] Y. C. Chen and W. C. Huang, "Constructing a stock-price forecast CNN model with gold and crude oil indicators", *Applied Soft Computing*, 112, 107760, 2021.
- [17] Z. D. Akşehir and E. Kılıç, "A new rule-based approach for encountered data imbalance problem in stock prediction and 2D-CNN model", *TBV Journal of Computer Science and Engineering*, 15 (1), pp. 6-13, 2022.
- [18] M. Leippold, Q. Wang and W. Zhou, "Machine learning in the Chinese stock market", *Journal of Financial Economics*, 145(2), pp. 64-82, 2022.

- [19] V. V. Prasad, S. Gumparathi, L.Y. Venkataramana, S. Srinethe, R. M. Sruthi Sree and K. Nishanthi, "Prediction of Stock Prices Using Statistical and Machine Learning Models: A Comparative Analysis", *The Computer Journal*, 65(5), 1338-1351, 2022.
- [20] V. Vapnik, S. Golowich and A. Smola, "Support vector method for function approximation, regression estimation and signal processing", *Advances in neural information processing systems*, 9, 1996.
- [21] S.B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background", *Internat. J. Eng. Res. Appl.* 3(5), 605-610, 2013.
- [22] L. Huang, Y. Li, S. Chen, Q. Zhang, Y. Song, J. Zhang and M. Wang, "Building safety monitoring based on extreme gradient boosting in distributed optical fiber sensing", *Optical Fiber Technol.*, 55, 102149, 2020.
- [23] S. Obata, C. J. Cieszewski, R. C. Lowe III, and P. Bettinger, "Random Forest Regression Model for Estimation of the Growing Stock Volumes in Georgia, USA, Using Dense Landsat Time Series and FIA Dataset" *Remote Sensing*, 13(2), 218.
- [24] IS Investment, , "Market Data," 2022. [Online]. Available: <https://www.isyatirim.com.tr>. [Accessed: 24-May-2022].
- [25] F. Alareqi and M. Z. Konyar , "High Accuracy Classification of Covid-19 from CT Images Using Transfer Learning Architectures", *Dicle University Journal of Engineering*, 13(3), pp. 457-466, 2022
- [26] F. Al-Areqi and M. Z. Konyar, "Effectiveness evaluation of different feature extraction methods for classification of covid-19 from computed tomography images: A high accuracy classification study", *Biomedical Signal Processing and Control*, 76, 103662, 2022.

# A Deep Transfer Learning-Based Comparative Study for Detection of Malaria Disease

 Emel Soylu<sup>1</sup>

<sup>1</sup>Corresponding Author; Samsun University, Faculty of Engineering, Department of Software Engineering, Samsun/TURKEY; emel.soylu@samsun.edu.tr

Received 31 October 2022; Revised 8 November 2022; Accepted 30 November 2022; Published online 31 December 2022

## Abstract

Malaria is a disease caused by a parasite. The parasite is transmitted to humans through the bite of infected mosquitoes. Thousands of people die every year due to malaria. When this disease is diagnosed early, it can be fully treated with medication. Diagnosis of malaria can be made according to the presence of parasites in the blood taken from the patient. In this study, malaria detection and diagnosis study were performed using The Malaria dataset containing a total of 27,558 cell images with samples of equally parasitized and uninfected cells from thin blood smear slide images of segmented cells. It is possible to detect malaria from microscopic blood smear images via modern deep learning techniques. In this study, 5 of the popular convolutional neural network architectures for malaria detection from cell images were retrained to find the best combination of architecture and learning algorithm. AlexNet, GoogLeNet, ResNet-50, MobileNet-v2, VGG-16 architectures from pre-trained networks were used, their hyperparameters were adjusted and their performances were compared. In this study, a maximum 96.53% accuracy rate was achieved with MobileNet-v2 architecture using the adam learning algorithm.

**Keywords:** malaria detection, deep transfer learning, Matlab, Convolutional Neural Network

## 1. Introduction

Malaria; is a disease that is transmitted to humans by the bite of a mosquito that carries parasites, can be fatal if not treated in time, and causes fever and chills in seizures. Anemia and jaundice may develop in cases where diagnosis and treatment are delayed. In some types of parasites that cause malaria, if treatment is not started within 24 hours, it can progress and lead to death. Malaria is a disease that can be treated with drugs. If the disease is diagnosed early and treated appropriately, patients can fully recover [1].

According to The World Health Organization's report, globally 229 million malaria cases were estimated. It is more common, especially in the Africa region. In 2019, 409 thousand people died from malaria disease. In the 2000s, this number was 736 thousand. Between 2000 and 2019, there were 1.5 billion cases of malaria globally, and 7.6 million people died from this cause [2]. Because malaria causes so much illness and death, the disease is a huge burden for many national economies. Most of the countries with malaria are already among the poor countries, and the economy of these countries is badly affected by the disease. The malaria parasite resides in the red blood cells of an infected person. Malaria can also be transmitted through blood transfusions, organ transplants, or the shared use of needles or syringes. Malaria can also be transmitted to an unborn baby [3].

Artificial intelligence techniques are effective methods that can produce solutions to optimization, prediction, fault diagnosis, image processing problems [4]–[6]. Artificial intelligence has entered a new phase with the start of running multi-layer neural networks on graphics cards. Thus, it became easier to solve problems such as image processing where too much data is processed. Multilayer deep convolutional neural networks produce very successful results for image processing problems [7]–[10]. In classical image processing, feature extraction is required to determine representation from the image. In contrast to this situation, raw pixel values are used in the CNN (Convolutional Neural Network) model. Deep learning techniques are also successfully used in the diagnosis of diseases in the field of health [11].



In the last decades, great advances have been made in the diagnosis of disease from medical images. Researchers have developed techniques that produce highly accurate results with various image processing operations. Examples of these techniques are artificial neural networks, machine learning, and deep learning techniques. According to the type of disease, images can be obtained and analyzed from sources such as microscopic, x-ray, MR, and ECG, and computer-assisted disease diagnosis can be made [12]–[17].

Training a deep network from scratch takes a lot of time. Retraining a previously trained model saves time. Re-training of a previously trained class network with a new data set is called DTL (Deep Transfer Learning). Using DTL provides a great advantage. There are many deep network models currently developed. These networks have been obtained as a result of days-long training on very powerful computers using millions of data [18]. Fig. 1. shows how DTL works [19].

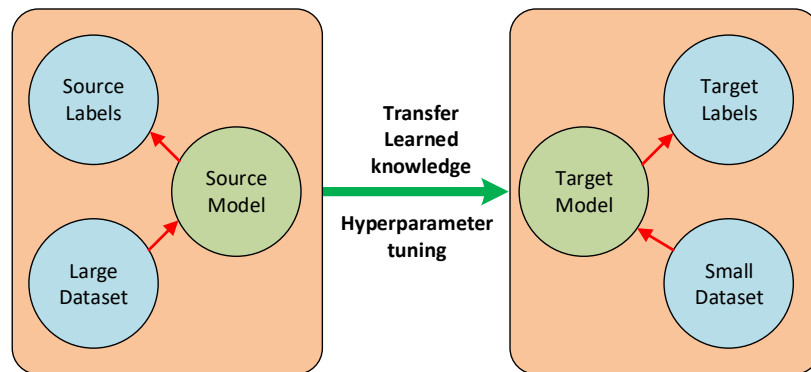


Figure 1 Concept of DTL

In this study, five of the deep network models available in the Matlab environment were retrained for the malaria data set. These are AlexNet, GoogLeNet, ResNet-50, MobileNet-v2, VGG-16 models. When previous studies were examined, no study was found to compare these five architectures. In this study, it has been seen that high performance can be achieved with DTL when hyperparameters are adjusted appropriately.

Information about the data set and deep learning architectures used in the rest of the study was given. After the hyperparameter settings are made, the training and comparison results are given.

## 2. Relevant Work

Scientists have done many studies around the world on the detection of the deadly malaria disease. With the technological improvements in computer hardware, software styles based on running parallel programs on graphics cards, great progress has been made in image processing studies. Image processing techniques have become frequently used in the field of health sciences, especially in disease detection. Deep artificial neural network techniques with a high number of neurons and layers are the technique with the highest performance today. Unlike machine learning, deep learning architectures with millions of parameters do not have feature extraction. The fact that high-performance results without a laborious process such as feature extraction increases the preference rate of convolutional neural networks by people.

Existing deep learning approaches applied to Malaria detection are given in Table 1. Vijayalakshmi et al. Retrained the Visual Geometry Group (VGG) by replacing some of its layers with the Support Vector Machine (SVM) and achieved 93.1% accuracy in malaria detection [20]. Dong et al. In their study on Identification of Malaria Infected Cells, they used transfer-based deep learning techniques and achieved a success rate of over 95% [21]. Yang et al. created a dataset with 1819 thick smear images collected from 150 patients. They reached a maximum of 94.33% accuracy in their malaria detection study using deep learning techniques on this data set [22]. According to Pan et al., It demonstrated that the deep convolutional network based on LeNet-5 can achieve very high classification accuracies for automatic malaria diagnosis. They analyzed the performance effect of the dataset by running their method on

datasets containing different numbers of images [23]. Reddy et al. reached a 95.91% accuracy rate with ResNet50 architecture for malaria detection. They used a dataset containing 27558 images [24]. Fuhad et al. used a variety of techniques including knowledge distillation, data augmentation, autoencoder, feature extraction by a CNN model, and classified by Support Vector Machine (SVM) or K-Nearest Neighbors (KNN). They reached a 99.23% accuracy rate for malaria detection problems with training the network using  $32 \times 32$  images.

Table 1 Existing deep learning approaches applied to Malaria detection

Authors	Methods	Year	Images	Best Accuracy rate (%)
Reddy et al. [24]	ResNet50	2019	27558	95,91
Fuhad et al. [25]	CNN-SVM, CNN-KNN	2020	26161	99,23
Vijayalakshmi et al. [20]	Visual Geometry Group (VGG) network and Support Vector Machine (SVM)	2019	2550	93,1
Dong et al. [21]	LeNet, AlexNet and GoogLeNet	2017	2565	98,13
Yang et al. [22]	ResNet50, VGG19, AlexNet, CNN	2020	1443	93,46
Pan et al. [23]	LeNet-5	2018	800	99

In this study, different from the others, 5 types of methods were compared according to the learning algorithm. High performance has been achieved without applying pre-image processing techniques in the dataset. The data set was applied directly to the input of the networks. Each architecture has been tested for 3 types of learning algorithms for two different initial learning rates (lr). The effect of learning algorithm selection on success was investigated. After 2x15 re-network training, it was observed that the success rates were between 50% and 96.53%.

### 3. Materials and Methods

The technical features of the computer used in this study are as follows:

- GPU: NVIDIA GeForce GTX 1070 8GB
- CPU: Intel I7 3.4 GHz
- Ram: 12 GB
- Operating System: Windows 10, 64 bit

The entire study has been done in the Matlab development environment using Deep Network Designer application. The last layer of the models is updated according to the number of categories, parameter settings are made and training of the network is carried out.

One of the most popular types of deep neural networks is the Convolutional Neural Network (CNN). CNN has a very good performance especially when a lot of images need to be processed. CNN has more than one layer. These are the convolutional layer, non-linearity layer, pooling layer, fully connected layer [26].

The first layer is the convolutional layer. In this layer, the image is passed through more than one parallel convolutional filter. These filters act as feature extractors. The output of the filters is a feature map [27]. The nonlinear transform layer normalizes between nearby feature maps [28]. With the pooling layer, the number of parameters and dimensions of the network is reduced. In the fully connected layer, data from previous layers are combined by weighting, and thanks to a loss function, the optimal weight to be given to neurons during training is found. Usually, the softmax activation function is used in this layer, and classification is made as probabilistic [29].

Three different optimizers are available in Matlab deep network designer tool as Stochastic Gradient Descent with Momentum (sgdm), Adaptive Moment Estimation (adam), and Root Mean Square Propagation (rmsprop). In this study their performance according to network models are compared. Sgdm is the preferred optimization method for many large-scale learning problems. Adam is a form of optimization that can be used instead of stochastic gradient descent. Quickly achieving good results on net weights makes this method popular. Rmsprop divides the learning rate by an exponentially

decreasing square gradient average. Rmsprop produces effective results for online and unstable problems [30].

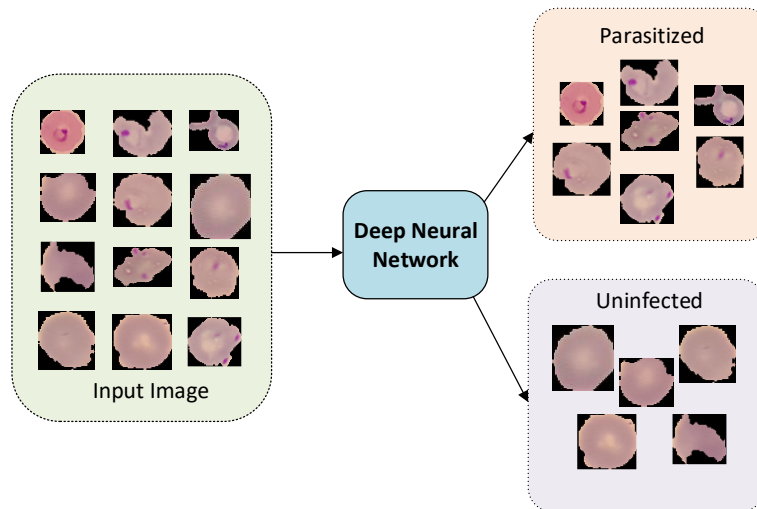


Figure 2 Block Diagram of the System

In this study, pre-trained networks are used to detect malaria disease. In deep learning, retraining a previously trained model for another problem is called transfer-based deep learning. Transfer learning is an approach in deep learning where knowledge is transferred from one model to another. The information obtained from the pre-trained model that was previously trained with a large-scale data can be used in a new model. Using a pre-trained network, especially for applications where the number of data is low, produces satisfactory results in the field of deep learning. When transfer-based learning is used, it is necessary to make changes in the last layers according to the number of classes to be classified, and to retrain the model after making the hyperparameter settings [31].

The block diagram of the system is given in Fig. 2. Deep network classifies input image as infected from parasite or not. The used networks and their properties are given in Table 2. In this study there are two classes, so the last classification layers of network models are modified to classify new images. For comparison when setting training parameters, batch size and epoch number set the same.

Table 2 Properties of network models

Network	Year of development	Input Image Size	Depth	Number of parameters	Number of categories
AlexNet	2012	224x224x3	8	61 million	1000
GoogLeNet	2014	224x224x3	22	7 million	1000
VGG-16	2014	224x224x3	16	138 million	1000
ResNet-50	2015	224x224x3	50	25,6 million	1000
MobileNet-v2	2018	224x224x3	53	3,5 million	1000

### 3.1. AlexNet

AlexNet is a convolutional neural network with 8-layer deep CNN. It has been trained with 1.2 million images in the ImageNet dataset and can classify 1000 objects [32]. AlexNet is designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton [33]. The detailed configuration of AlexNet model for this study is given in Table 3.

Table 3 The architecture of AlexNet model

No	Layer	Properties	No	Layer	Properties
1	Image Input	227×227×3	14	2-D Grouped Conv.	2 groups of 128 3×3×192
2	2-D Conv.	96 11×11×3	15	ReLU	ReLU
3	ReLU	ReLU	16	2-D Max Pooling	3×3
4	Cross Channel Norm.	5 channels per element	17	Fully Connected	4096
5	2-D Max Pooling	3×3	18	ReLU	ReLU
6	2-D Grouped Conv.	2 groups of 128 5×5×48	19	Dropout	50% dropout
7	ReLU	ReLU	20	Fully Connected	4096
8	Cross Channel Norm.	5 channels per element	21	ReLU	ReLU
9	2-D Max Pooling	3×3	22	Dropout	50% dropout
10	2-D Conv.	384 3×3×256	23	Fully Connected	2
11	ReLU	ReLU	24	Softmax	softmax
12	2-D Grouped Conv.	2 groups of 192 3×3×192	25	Classification Output	2 classes
13	ReLU	ReLU			

### 3.2. GoogLeNet

GoogLeNet is developed by researchers working at Google. GoogLeNet was the winner of ILSVRC 2014 competition [34]. GoogLeNet's other name is Inception block. It has a 22-layer deep CNN and 7 million parameters. Pre-trained network can classify 1000 objects. The detailed configuration of GoogLeNet model for this study is given in Table 4.

Table 4 The architecture of GoogLeNet model

No	Layer	Properties	No	Layer	Properties	No	Layer	Properties
1	Image Input	224×224×3	49	2-D Convolution	48 5×5×16	97	2-D Convolution	256 1×1×528
2	2-D Convolution	64 7×7×3	50	ReLU	ReLU	98	ReLU	ReLU
3	ReLU	ReLU	51	2-D Max Pooling	3×3	99	2-D Convolution	160 1×1×528
4	2-D Max Pooling	3×3	52	2-D Convolution	64 1×1×480	100	ReLU	ReLU
5	Cross Channel Norm.	channels per element	53	ReLU	ReLU	101	2-D Convolution	320 3×3×160
6	2-D Convolution	64 1×1×64	54	Depth concatenation	4 inputs	102	ReLU	ReLU
7	ReLU	ReLU	55	2-D Convolution	160 1×1×512	103	2-D Convolution	32 1×1×528
8	2-D Convolution	192 3×3×64	56	ReLU	ReLU	104	ReLU	ReLU
9	ReLU	ReLU	57	2-D Convolution	112 1×1×512	105	2-D Convolution	128 5×5×32
10	Cross Channel Norm.	5 channels per element	58	ReLU	ReLU	106	ReLU	ReLU
11	2-D Max Pooling	3×3	59	2-D Convolution	224 3×3×112	107	2-D Max Pooling	3×3

12	2-D Convolution	64 1×1×192	60	ReLU	ReLU	10 8	2-D Convolution	128 1×1×528
13	ReLU	ReLU	61	2-D Convolution	24 1×1×512	10 9	ReLU	ReLU
14	2-D Convolution	96 1×1×192	62	ReLU	ReLU	11 0	Depth concatenation	4 inputs
15	ReLU	ReLU	63	2-D Convolution	64 5×5×24	11 1	2-D Max Pooling	3×3
16	2-D Convolution	128 3×3×96	64	ReLU	ReLU	11 2	2-D Convolution	256 1×1×832
17	ReLU	ReLU	65	2-D Max Pooling	3×3	11 3	ReLU	ReLU
18	2-D Convolution	16 1×1×192	66	2-D Convolution	64 1×1×512	11 4	2-D Convolution	160 1×1×832
19	ReLU	ReLU	67	ReLU	ReLU	11 5	ReLU	ReLU
20	2-D Convolution	32 5×5×16	68	Depth concatenation	4 inputs	11 6	2-D Convolution	320 3×3×160
21	ReLU	ReLU	69	2-D Convolution	128 1×1×512	11 7	ReLU	ReLU
22	2-D Max Pooling	3×3	70	ReLU	ReLU	11 8	2-D Convolution	32 1×1×832
23	2-D Convolution	32 1×1×192	71	2-D Convolution	128 1×1×512	11 9	ReLU	ReLU
24	ReLU	ReLU	72	ReLU	ReLU	12 0	2-D Convolution	128 5×5×32
25	Depth concatenation	4 inputs	73	2-D Convolution	256 3×3×128	12 1	ReLU	ReLU
26	2-D Convolution	128 1×1×256	74	ReLU	ReLU	12 2	2-D Max Pooling	3×3
27	ReLU	ReLU	75	2-D Convolution	24 1×1×512	12 3	2-D Convolution	128 1×1×832
28	2-D Convolution	128 1×1×256	76	ReLU	ReLU	12 4	ReLU	ReLU
29	ReLU	ReLU	77	2-D Convolution	64 5×5×24	12 5	Depth concatenation	4 inputs
30	2-D Convolution	192 3×3×128	78	ReLU	ReLU	12 6	2-D Convolution	384 1×1×832
31	ReLU	ReLU	79	2-D Max Pooling	3×3	12 7	ReLU	ReLU
32	2-D Convolution	32 1×1×256	80	2-D Convolution	64 1×1×512	12 8	2-D Convolution	192 1×1×832
33	ReLU	ReLU	81	ReLU	ReLU	12 9	ReLU	ReLU
34	2-D Convolution	96 5×5×32	82	Depth concatenation	4 inputs	13 0	2-D Convolution	384 3×3×192
35	ReLU	ReLU	83	2-D Convolution	112 1×1×512	13 1	ReLU	ReLU
36	2-D Max Pooling	3×3	84	ReLU	ReLU	13 2	2-D Convolution	48 1×1×832
37	2-D Convolution	64 1×1×256	85	2-D Convolution	144 1×1×512	13 3	ReLU	ReLU
38	ReLU	ReLU	86	ReLU	ReLU	13 4	2-D Convolution	128 5×5×48

39	Depth concatenation	4 inputs	87	2-D Convolution	288 3×3×144	13 5	ReLU	ReLU
40	2-D Max Pooling	3×3	88	ReLU	ReLU	13 6	2-D Max Pooling	3×3
41	2-D Convolution	192 1×1×480	89	2-D Convolution	32 1×1×512	13 7	2-D Convolution	128 1×1×832
42	ReLU	ReLU	90	ReLU	ReLU	13 8	ReLU	ReLU
43	2-D Convolution	96 1×1×480	91	2-D Convolution	64 5×5×32	13 9	Depth concatenation	4 inputs
44	ReLU	ReLU	92	ReLU	ReLU	14 0	2-D Global Avg. Pooling	2-D
45	2-D Convolution	208 3×3×96	93	2-D Max Pooling	3×3	14 1	Dropout	40% dropout
46	ReLU	ReLU	94	2-D Convolution	64 1×1×512	14 2	Fully Connected	2
47	2-D Convolution	16 1×1×480	95	ReLU	ReLU	14 3	Softmax	softmax
48	ReLU	ReLU	96	Depth concatenation	4 inputs	14 4	Classification Output	2 classes

### 3.3. ResNet-50

ResNet stands for Residual Network introduced in the 2015 p by He Kaiming et. al.[35] ResNet50 is a CNN architecture with 50-layer deep CNN. Pre-trained network can classify into 1000 categories. The architecture has 25.6 million parameters. The detailed configuration of ResNet-50 model for this study is given in Table 5.

Table 5 The architecture of ResNet-50 model

No	Layer	Properties	No	Layer	Properties	No	Layer	Properties
1	Image Input	224×224×3	60	2-D Conv.	128 1×1×512	119	Batch Norm.	1024 channels
2	2-D Conv.	64 7×7×3	61	Batch Norm.	128 channels	120	Addition	2 inputs
3	Batch Norm.	64 channels	62	ReLU	ReLU	121	ReLU	ReLU
4	ReLU	ReLU	63	2-D Conv.	128 3×3×128	122	2-D Conv.	256 1×1×1024
5	2-D Max Pooling	3×3	64	Batch Norm.	128 channels	123	Batch Norm.	256 channels
6	2-D Conv.	64 1×1×64	65	ReLU	ReLU	124	ReLU	ReLU
7	Batch Norm.	64 channels	66	2-D Conv.	512 1×1×128	125	2-D Conv.	256 3×3×256
8	ReLU	ReLU	67	Batch Norm.	512 channels	126	Batch Norm.	256 channels
9	2-D Conv.	64 3×3×64	68	Addition	2 inputs	127	ReLU	ReLU
10	Batch Norm.	64 channels	69	ReLU	ReLU	128	2-D Conv.	1024 1×1×256
11	ReLU	ReLU	70	2-D Conv.	128 1×1×512	129	Batch Norm.	1024 channels
12	2-D Conv.	256 1×1×64	71	Batch Norm.	128 channels	130	Addition	2 inputs
13	2-D Conv.	256 1×1×64	72	ReLU	ReLU	131	ReLU	ReLU

14	Batch Norm.	256 channels	73	2-D Conv.	128 3×3×128	132	2-D Conv.	256 1×1×1024
15	Batch Norm.	256 channels	74	Batch Norm.	128 channels	133	Batch Norm.	256 channels
16	Addition	2 inputs	75	ReLU	ReLU	134	ReLU	ReLU
17	ReLU	ReLU	76	2-D Conv.	512 1×1×128	135	2-D Conv.	256 3×3×256
18	2-D Conv.	64 1×1×256	77	Batch Norm.	512 channels	136	Batch Norm.	256 channels
19	Batch Norm.	64 channels	78	Addition	2 inputs	137	ReLU	ReLU
20	ReLU	ReLU	79	ReLU	ReLU	138	2-D Conv.	1024 1×1×256
21	2-D Conv.	64 3×3×64	80	2-D Conv.	256 1×1×512	139	Batch Norm.	1024 channels
22	Batch Norm.	64 channels	81	Batch Norm.	256 channels	140	Addition	2 inputs
23	ReLU	ReLU	82	ReLU	ReLU	141	ReLU	ReLU
24	2-D Conv.	256 1×1×64	83	2-D Conv.	256 3×3×256	142	2-D Conv.	512 1×1×1024
25	Batch Norm.	256 channels	84	Batch Norm.	256 channels	143	Batch Norm.	512 channels
26	Addition	2 inputs	85	ReLU	ReLU	144	ReLU	ReLU
27	ReLU	ReLU	86	2-D Conv.	1024 1×1×256	145	2-D Conv.	512 3×3×512
28	2-D Conv.	64 1×1×256	87	2-D Conv.	1024 1×1×512	146	Batch Norm.	512 channels
29	Batch Norm.	64 channels	88	Batch Norm.	1024 channels	147	ReLU	ReLU
30	ReLU	ReLU	89	Batch Norm.	1024 channels	148	2-D Conv.	2048 1×1×512
31	2-D Conv.	64 3×3×64	90	Addition	2 inputs	149	2-D Conv.	2048 1×1×1024
32	Batch Norm.	64 channels	91	ReLU	ReLU	150	Batch Norm.	2048 channels
33	ReLU	ReLU	92	2-D Conv.	256 1×1×1024	151	Batch Norm.	2048 channels
34	2-D Conv.	256 1×1×64	93	Batch Norm.	256 channels	152	Addition	2 inputs
35	Batch Norm.	256 channels	94	ReLU	ReLU	153	ReLU	ReLU
36	Addition	2 inputs	95	2-D Conv.	256 3×3×256	154	2-D Conv.	512 1×1×2048
37	ReLU	ReLU	96	Batch Norm.	256 channels	155	Batch Norm.	512 channels
38	2-D Conv.	128 1×1×256	97	ReLU	ReLU	156	ReLU	ReLU
39	Batch Norm.	128 channels	98	2-D Conv.	1024 1×1×256	157	2-D Conv.	512 3×3×512
40	ReLU	ReLU	99	Batch Norm.	1024 channels	158	Batch Norm.	512 channels
41	2-D Conv.	128 3×3×128	100	Addition	2 inputs	159	ReLU	ReLU
42	Batch Norm.	128 channels	101	ReLU	ReLU	160	2-D Conv.	2048 1×1×512
43	ReLU	ReLU	102	2-D Conv.	256 1×1×1024	161	Batch Norm.	2048 channels

44	2-D Conv.	512 1×1×128	103	Batch Norm.	256 channels	162	Addition	2 inputs
45	2-D Conv.	512 1×1×256	104	ReLU	ReLU	163	ReLU	ReLU
46	Batch Norm.	512 channels	105	2-D Conv.	256 3×3×256	164	2-D Conv.	512 1×1×2048
47	Batch Norm.	512 channels	106	Batch Norm.	256 channels	165	Batch Norm.	512 channels
48	Addition	2 inputs	107	ReLU	ReLU	166	ReLU	ReLU
49	ReLU	ReLU	108	2-D Conv.	1024 1×1×256	167	2-D Conv.	512 3×3×512
50	2-D Conv.	128 1×1×512	109	Batch Norm.	1024 channels	168	Batch Norm.	512 channels
51	Batch Norm.	128 channels	110	Addition	2 inputs	169	ReLU	ReLU
52	ReLU	ReLU	111	ReLU	ReLU	170	2-D Conv.	2048 1×1×512
53	2-D Conv.	128 3×3×128	112	2-D Conv.	256 1×1×1024	171	Batch Norm.	2048 channels
54	Batch Norm.	128 channels	113	Batch Norm.	256 channels	172	Addition	2 inputs
55	ReLU	ReLU	114	ReLU	ReLU	173	ReLU	ReLU
56	2-D Conv.	512 1×1×128	115	2-D Conv.	256 3×3×256	174	2-D Global Average Pooling	2-D
57	Batch Norm.	512 channels	116	Batch Norm.	256 channels	175	Fully Connected	Fully Connected
58	Addition	2 inputs	117	ReLU	ReLU	176	Softmax	softmax
59	ReLU	ReLU	118	2-D Conv.	1024 1×1×256	177	Classification Output	2 classes

### 3.4. MobileNet-v2

MobileNet-v2 has an architecture designed to be used mostly on mobile devices. With 3.5 million parameters, it has fewer parameters than other architectures. It has 53-layer deep CNN. It is trained with over a million data from the ImageNet dataset. The pre-trained network can classify into 1000 categories. The low number of parameters also reduces the training time. The detailed configuration of MobileNet-v2 model for this study is given in Table 6.

Table 6 The architecture of MobileNet-v2 model

Layer	Properties	No	Layer	Properties	No	Layer	Properties
Image Input	224×224×3	53	2-D Conv.	192 1×1×32	105	2-D Conv.	576 1×1×96
2-D Conv.	32 3×3×3	54	Batch Norm.	192 channels	106	Batch Norm.	576 channels
Batch Norm.	32 channels	55	Clipped ReLU	ceiling 6	107	Clipped ReLU	ceiling 6
Clipped ReLU	ceiling 6	56	2-D Grouped Conv.	192 groups of 1 3×3×1	108	2-D Grouped Conv.	576 groups
2-D Grouped Conv.	32 groups	57	Batch Norm.	192 channels	109	Batch Norm.	576 channels
Batch Norm.	32 channels	58	Clipped ReLU	ceiling 6	110	Clipped ReLU	ceiling 6
Clipped ReLU	ceiling 6	59	2-D Conv.	64 1×1×192	111	2-D Conv.	96 1×1×576
2-D Conv.	16 1×1×32	60	Batch Norm.	64 channels	112	Batch Norm.	96 channels



Batch Norm.	16 channels	61	2-D Conv.	384 1×1×64	113	Addition	2 inputs
2-D Conv.	96 1×1×16	62	Batch Norm.	384 channels	114	2-D Conv.	576 1×1×96
Batch Norm.	96 channels	63	Clipped ReLU	ceiling 6	115	Batch Norm.	576 channels
Clipped ReLU	ceiling 6	64	2-D Grouped Conv.	384 groups	116	Clipped ReLU	ceiling 6
2-D Grouped Conv.	96 groups of 1 3×3×1	65	Batch Norm.	384 channels	117	2-D Grouped Conv.	576 groups of 1 3×3×1
Batch Norm.	96 channels	66	Clipped ReLU	ceiling 6	118	Batch Norm.	576 channels
Clipped ReLU	ceiling 6	67	2-D Conv.	64 1×1×384	119	Clipped ReLU	ceiling 6
2-D Conv.	24 1×1×96	68	Batch Norm.	64 channels	120	2-D Conv.	160 1×1×576
Batch Norm.	24 channels	69	Addition	2 inputs	121	Batch Norm.	160 channels
2-D Conv.	144 1×1×24	70	2-D Conv.	384 1×1×64	122	2-D Conv.	960 1×1×160
Batch Norm.	144 channels	71	Batch Norm.	384 channels	123	Batch Norm.	960 channels
Clipped ReLU	ceiling 6	72	Clipped ReLU	ceiling 6	124	Clipped ReLU	ceiling 6
2-D Grouped Conv.	144 groups	73	2-D Grouped Conv.	384 groups	125	2-D Grouped Conv.	960 groups
Batch Norm.	144 channels	74	Batch Norm.	384 channels	126	Batch Norm.	960 channels
Clipped ReLU	ceiling 6	75	Clipped ReLU	ceiling 6	127	Clipped ReLU	ceiling 6
2-D Conv.	24 1×1×144	76	2-D Conv.	64 1×1×384	128	2-D Conv.	160 1×1×960
Batch Norm.	24 channels	77	Batch Norm.	64 channels	129	Batch Norm.	160 channels
Addition	2 inputs	78	Addition	2 inputs	130	Addition	2 inputs
2-D Conv.	144 1×1×24	79	2-D Conv.	384 1×1×64	131	2-D Conv.	960 1×1×160
Batch Norm.	144 channels	80	Batch Norm.	384 channels	132	Batch Norm.	960 channels
Clipped ReLU	ceiling 6	81	Clipped ReLU	ceiling 6	133	Clipped ReLU	ceiling 6
2-D Grouped Conv.	144 groups of 1 3×3×1	82	2-D Grouped Conv.	384 groups	134	2-D Grouped Conv.	960 groups
Batch Norm.	144 channels	83	Batch Norm.	384 channels	135	Batch Norm.	960 channels
Clipped ReLU	ceiling 6	84	Clipped ReLU	ceiling 6	136	Clipped ReLU	ceiling 6
2-D Conv.	32 1×1×144	85	2-D Conv.	64 1×1×384	137	2-D Conv.	160 1×1×960
Batch Norm.	32 channels	86	Batch Norm.	64 channels	138	Batch Norm.	160 channels
2-D Conv.	192 1×1×32	87	Addition	2 inputs	139	Addition	2 inputs
Batch Norm.	192 channels	88	2-D Conv.	384 1×1×64	140	2-D Conv.	960 1×1×160

Clipped ReLU	ceiling 6	89	Batch Norm.	384 channels	141	Batch Norm.	960 channels
2-D Grouped Conv.	192 groups	90	Clipped ReLU	ceiling 6	142	Clipped ReLU	ceiling 6
Batch Norm.	192 channels	91	2-D Grouped Conv.	384 groups	143	2-D Grouped Conv.	960 groups
Clipped ReLU	ceiling 6	92	Batch Norm.	384 channels	144	Batch Norm.	960 channels
2-D Conv.	32 1×1×192	93	Clipped ReLU	ceiling 6	145	Clipped ReLU	ceiling 6
Batch Norm.	32 channels	94	2-D Conv.	96 1×1×384	146	2-D Conv.	320 1×1×960
Addition	2 inputs	95	Batch Norm.	96 channels	147	Batch Norm.	320 channels
2-D Conv.	192 1×1×32	96	2-D Conv.	576 1×1×96	148	2-D Conv.	1280 1×1×320
Batch Norm.	192 channels	97	Batch Norm.	576 channels	149	Batch Norm.	1280 channels
Clipped ReLU	ceiling 6	98	Clipped ReLU	ceiling 6	150	Clipped ReLU	ceiling 6
2-D Grouped Conv.	192 groups	99	2-D Grouped Conv.	576 groups	151	2-D Global Average Pooling	2-D global average pooling
Batch Norm.	192 channels	100	Batch Norm.	576 channels	152	Fully Connected	Fully connected
Clipped ReLU	ceiling 6	101	Clipped ReLU	ceiling 6	153	Softmax	softmax
2-D Conv.	32 1×1×192	102	2-D Conv.	96 1×1×576	154	Classification Output	2 classes
Batch Norm.	32 channels	103	Batch Norm.	96 channels			
Addition	2 inputs	104	Addition	2 inputs			

### 3.5. VGG-16

It is trained with more than 14 million data in the VGG-16 ImageNet dataset. It's training took weeks. It has 41 layers. With 138 million parameters, it is the architecture with the most parameters among those used in this study. The pre-trained network can classify into 1000 categories. The detailed configuration of VGG-16 model for this study is given in Table 7.

Table 7 The architecture of VGG-16 model

No	Layer	Properties	No	Layer	Properties	No	Layer	Properties
1	Image Input	224x224x3	15	ReLU	ReLU	29	ReLU	ReLU
2	Convolution	64 3x3x3	16	Convolution	256 3x3x256	30	Convolution	512 3x3x512
3	ReLU	ReLU	17	ReLU	ReLU	31	ReLU	ReLU
4	Convolution	64 3x3x64	18	Max Pooling	2x2	32	Max Pooling	2x2
5	ReLU	ReLU	19	Convolution	512 3x3x256	33	Fully Connected	4096
6	Max Pooling	2x2	20	ReLU	ReLU	34	ReLU	ReLU
7	Convolution	128 3x3x64	21	Convolution	512 3x3x512	35	Dropout	50% dropout
8	ReLU	ReLU	22	ReLU	ReLU	36	Fully Connected	4096

9	Convolution	128 3x3x128	23	Convolution	512 3x3x512	37	ReLU	ReLU
10	ReLU	ReLU	24	ReLU	ReLU	38	Dropout	50% dropout
11	Max Pooling	2x2	25	Max Pooling	2x2	39	Fully Connected	2
12	Convolution	256 3x3x128	26	Convolution	512 3x3x512	40	Softmax	softmax
13	ReLU	ReLU	27	ReLU	ReLU	41	Classification Output	2 classes
14	Convolution	256 3x3x256	28	Convolution	512 3x3x512			

### 3.6. Dataset

The data set used in this study was created with a mobile application developed to take microscopic images and samples taken from patients and non-sick individuals in Mahidol-Oxford Tropical Medicine Research Unit in Bangkok [36]. The data set was shared on the internet available to researchers. It is possible to reach the data set from many different links. In this study, the data obtained from the Kaggle platform was used [37]. The Malaria dataset contains a total of 27,558 cell images with samples of equally parasitized and uninfected cells from thin blood smear slide images of segmented cells. Sample images from dataset is given in Figure 3. Parasitized cells contain Plasmodium in different sizes and shapes.

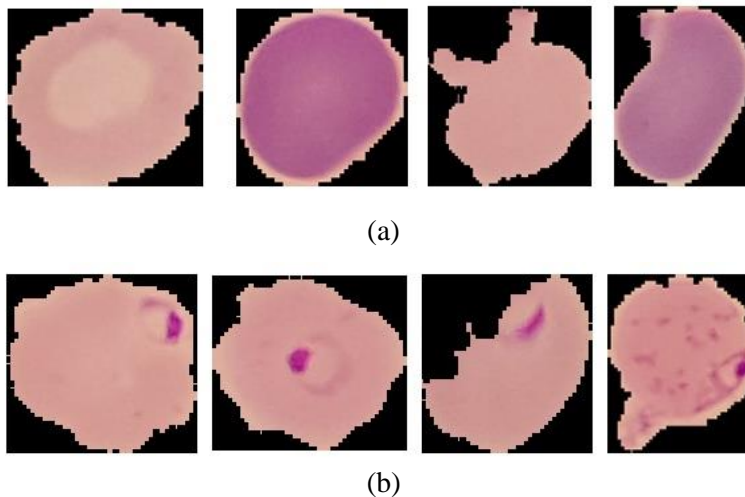


Figure 3 Sample dataset images (a) uninfected (b) parasitized

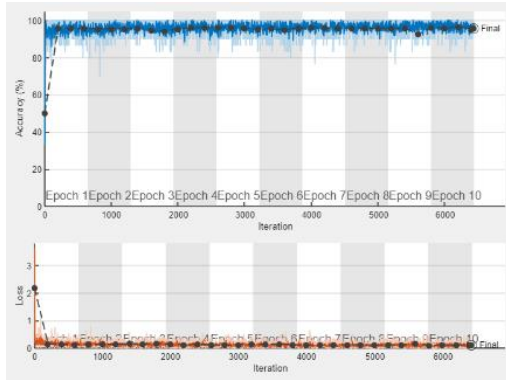
The image sizes in the data set are resized according to the input sizes of the network to be used. In this study data augmentation was not applied. 70% of the data were used as training data and 30% as test data. Number of images for training is 19290 and number of images for validation is 8268. Number of parasitized and uninfected images are equal.

## 4. Training of models

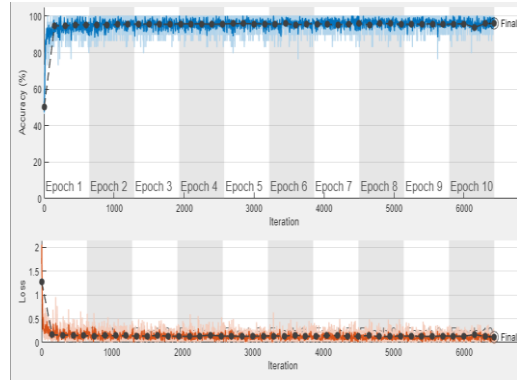
Learning curves that showing the progress over the experience during the training of a machine learning models are just a mathematical representation of the learning process. We observe accuracy and loss performances from plots according to validation data. In this section training progress of models are given.

Screenshots of the training window for AlexNet are given in Fig. 4, Fig. 5, and Fig.6 respectively. Accuracy and loss rates according to iteration are shown in these graphs. The validation accuracy is obtained 95,9% with sgd optimizer, 50% with adam optimizer, 95,22% with rmsprop optimizer at

learning rate of 0.001 and 96.08% with sgdm optimizer, 94.19% with adam optimizer, 95.85% with rmsprop optimizer learning rate of 0.0001.

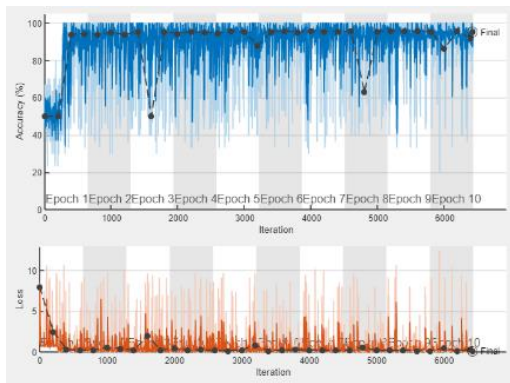


(a)

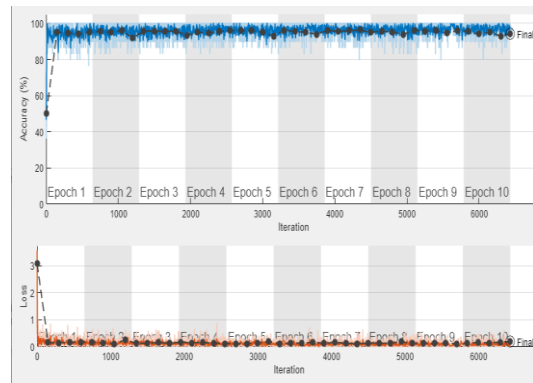


(b)

Figure 4 Re-training of AlexNet Network Model with sgdm Optimizer (a)lr=0.001 (b) lr=0.0001

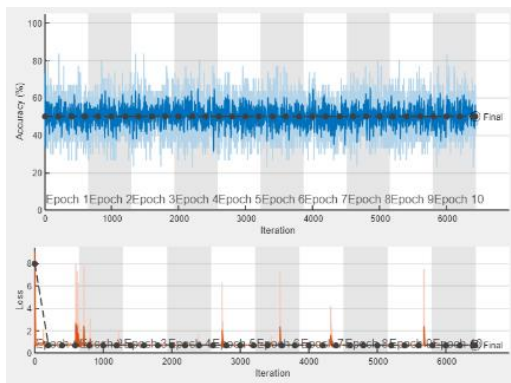


(a)

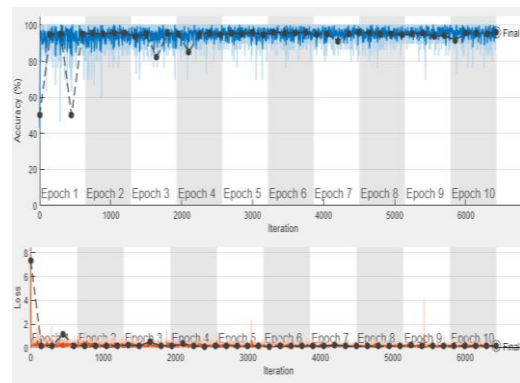


(b)

Figure 5 Re-training of AlexNet Network Model with adam Optimizer (a)lr=0.001 (b) lr=0.0001



(a)



(b)

Figure 6 Re-training of AlexNet Network Model with rmsprop Optimizer (a)lr=0.001 (b) lr=0.0001

Screenshots of the training window for GoogLeNet are given in Fig. 7, Fig. 8, and Fig.9 respectively. Accuracy and loss rates according to iteration are shown in these graphs. Accuracy and loss rates according to iteration are shown in these graphs. The validation accuracy is obtained 95,22% with sgdm optimizer, 95,46% with adam optimizer, 95,54% with rmsprop optimizer and 96.07% with sgdm optimizer, 96.26% with adam optimizer, 96.73% with rmsprop optimizer learning rate of 0.0001.

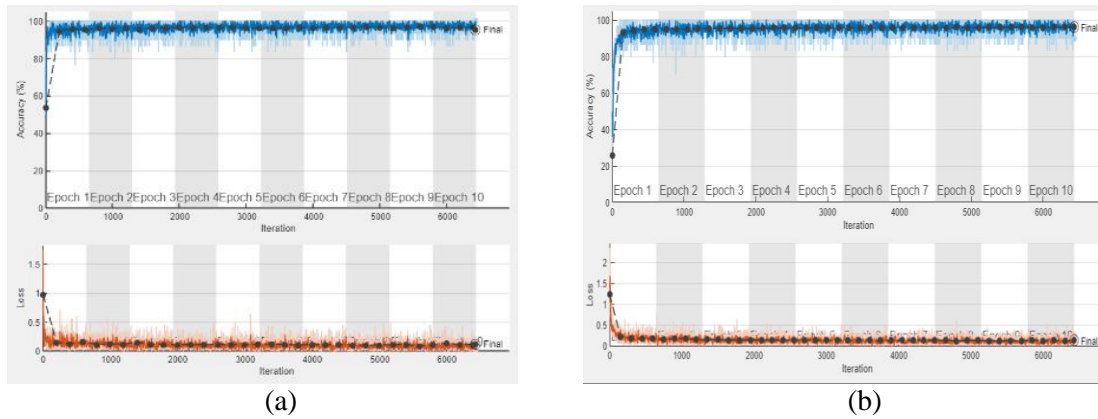


Figure 7 Re-training of GoogLeNet Network Model with sgd Optimizer (a)lr=0.001 (b) lr=0.0001

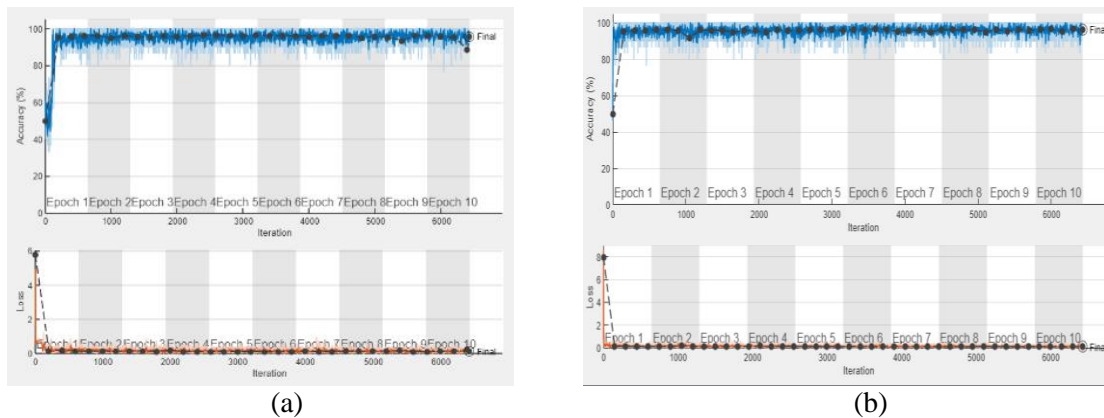


Figure 8 Re-training of GoogLeNet Network Model with adam Optimizer (a)lr=0.001 (b) lr=0.0001

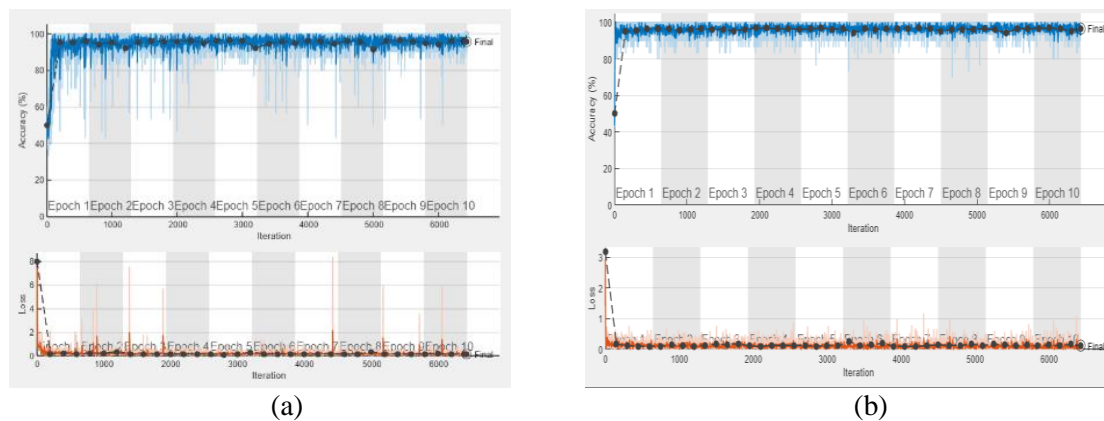
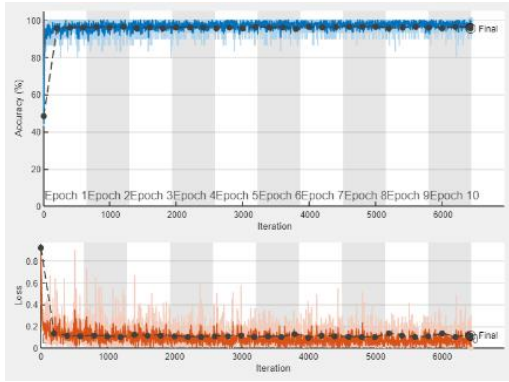
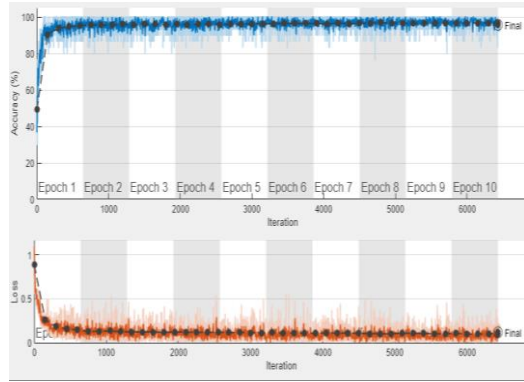


Figure 9 Re-training of GoogLeNet Network Model with rmsprop Optimizer (a)lr=0.001 (b) lr=0.0001

Screenshots of the training window for ResNet-50 are given in Fig. 10, Fig. 11, and Fig.12 respectively. Accuracy and loss rates according to iteration are shown in these graphs. The validation accuracy is obtained 95,57% with sgd optimizer, 95,66% with adam optimizer, 95,05% with rmsprop optimizer at learning rate of 0.001 and 95.65% with sgd optimizer, 96.76% with adam optimizer, 96.07% with rmsprop optimizer learning rate of 0.0001.

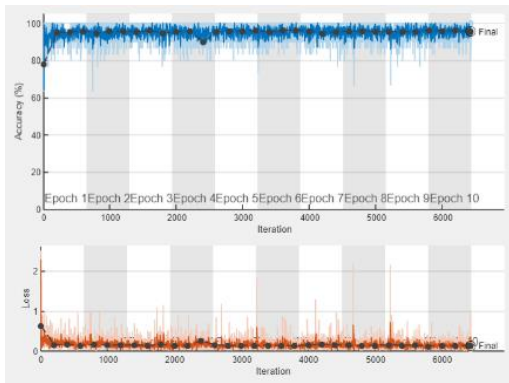


(a)

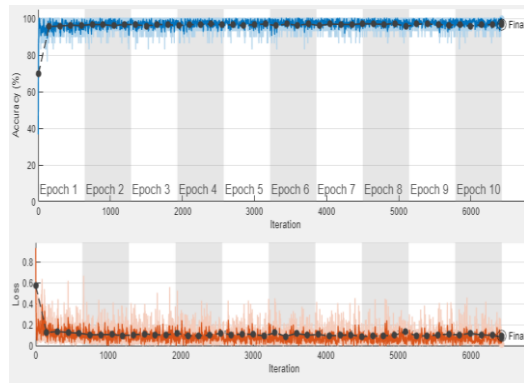


(b)

Figure 10 Re-training of ResNet-50 Network Model with sgd Optimizer (a)lr=0.001 (b) lr=0.0001

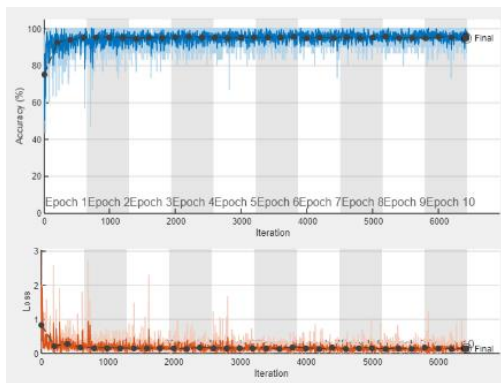


(a)

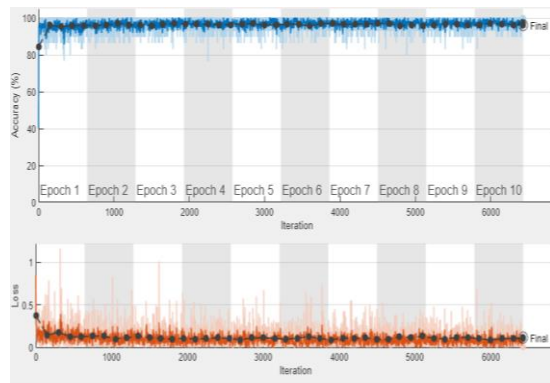


(b)

Figure 3 Re-training of ResNet-50 Network Model with adam Optimizer (a)lr=0.001 (b) lr=0.0001



(a)



(b)

Figure 4 Re-training of ResNet-50 Network Model with rmsprop Optimizer (a)lr=0.001 (b) lr=0.0001

Screenshots of the training window for MobileNet-v2 are given in Fig. 13, Fig. 14, and Fig.15 respectively. Accuracy and loss rates according to iteration are shown in these graphs. The validation accuracy is obtained 95,09% with sgd optimizer, 96,53% with adam optimizer, 96,31% with rmsprop optimizer at learning rate of 0.001 and 95.72% with sgd optimizer, 95.63% with adam optimizer, 96.13% with rmsprop optimizer learning rate of 0.0001.

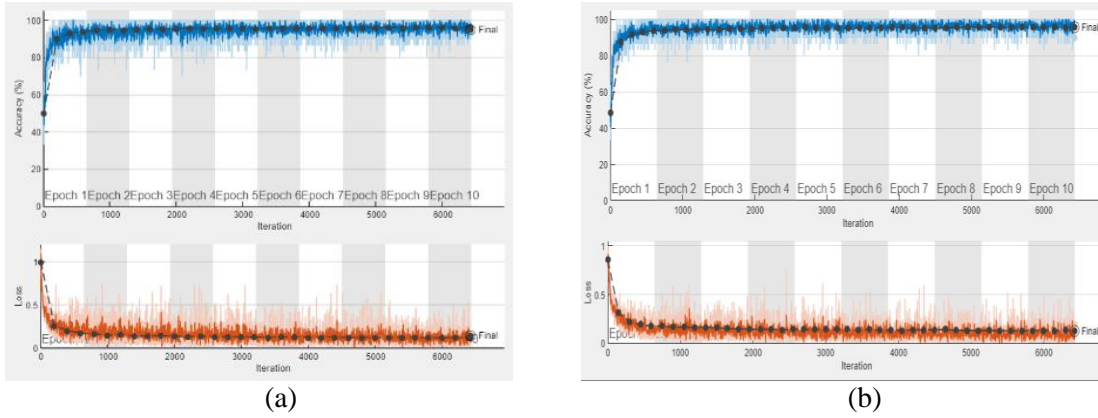


Figure 5 Re-training of MobileNet-v2 Network Model with sgd Optimizer (a)lr=0.001 (b) lr=0.0001

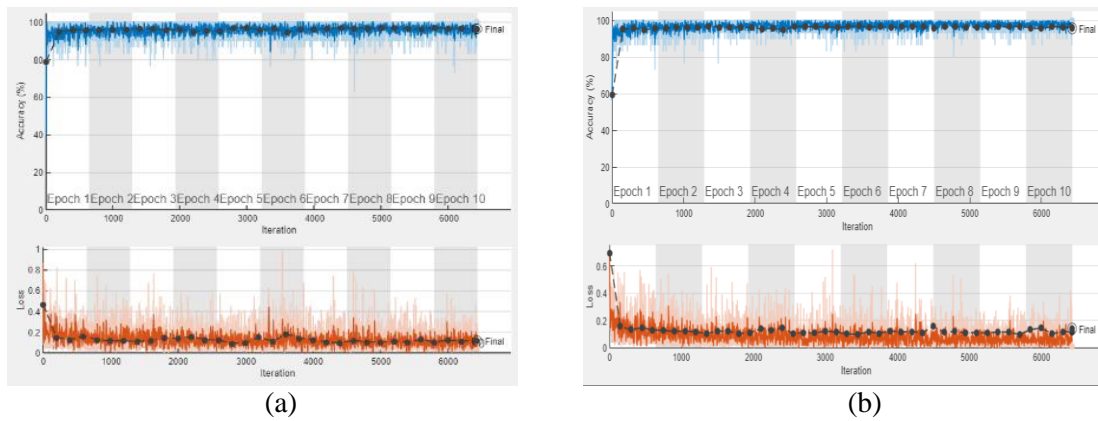


Figure 6 Re-training of MobileNet-v2 Network Model with adam Optimizer (a)lr=0.001 (b) lr=0.0001

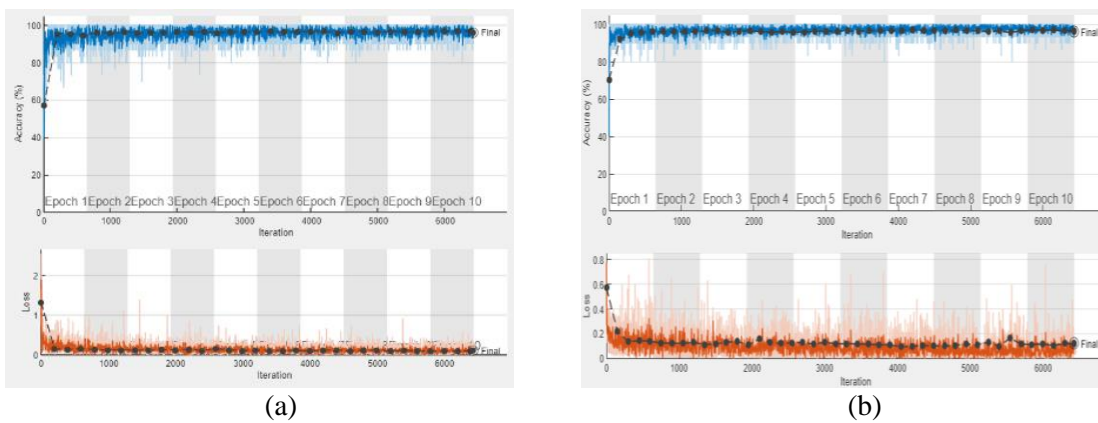


Figure 7 Re-training of MobileNet-v2 Network Model with rmsprop Optimizer (a)lr=0.001 (b) lr=0.0001

A large number of parameters also affects the retraining speed. Among the architectures used in this study, the longest training period belongs to this architecture. Screenshots of training window for VGG-16 are given in Fig. 16, Fig. 17, and Fig.18 respectively. The validation accuracy is obtained 93,89% with sgd optimizer, 50% with adam optimizer, 50% with rmsprop optimizer at learning rate of 0.001 and 95.42% with sgd optimizer, 96.46% with adam optimizer, 95.52% with rmsprop optimizer learning rate of 0.0001.

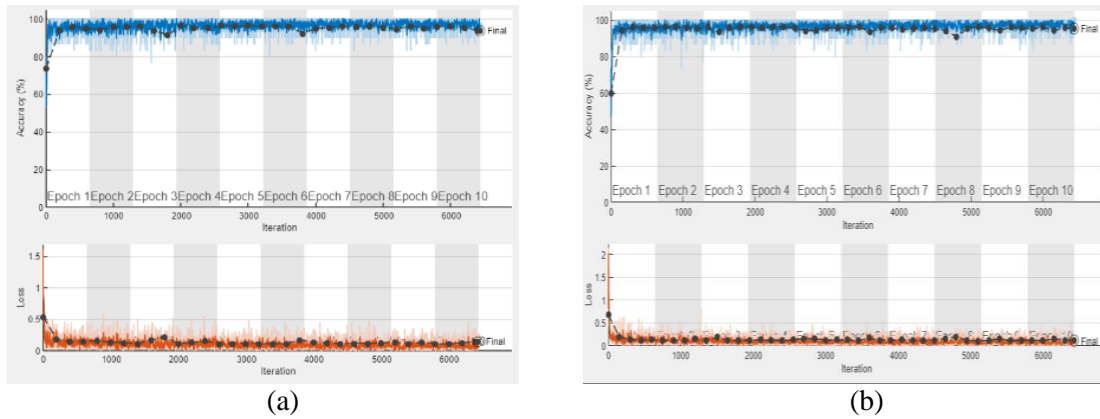


Figure 8 Re-training of the VGG-16 Network Model with sgd Optimizer (a)lr=0.001 (b) lr=0.0001

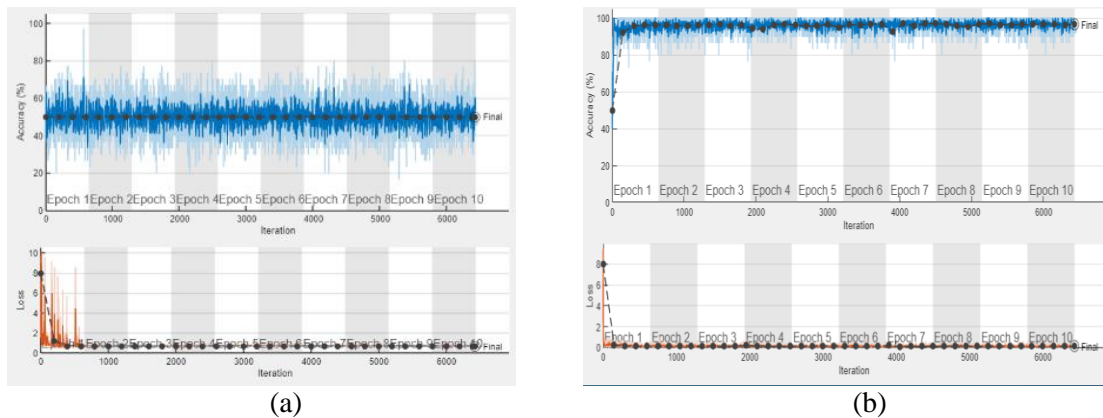


Figure 9 Re-training of the VGG-16 Network Model with adam Optimizer (a)lr=0.001 (b) lr=0.0001

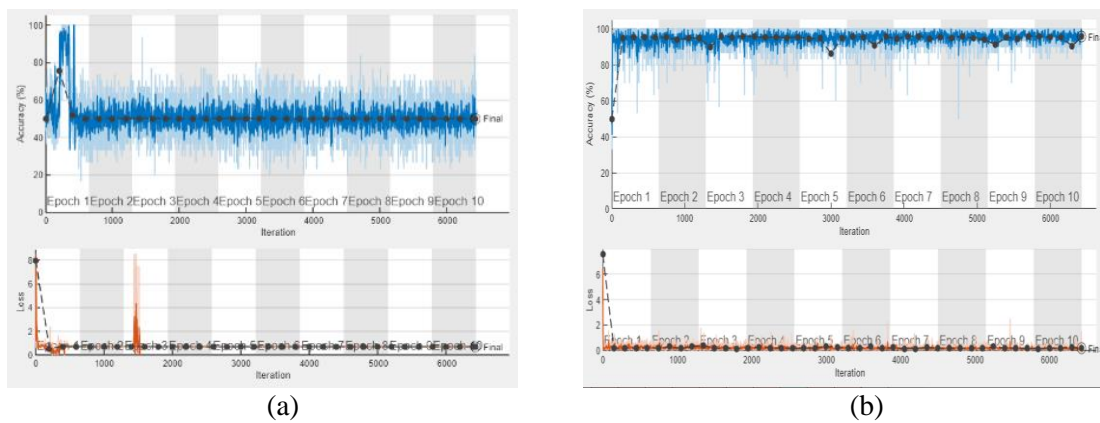


Figure 10 Re-Training of The VGG-16 Network Model with Rmsprop Optimizer (a)lr=0.001 (b) lr=0.0001

## 5. Results

Table 8. represents the entire training results for 0.001 initial learning rate, 30 batch size, and 10 epoch. The most successful results were obtained when the MobileNet-v2 network was trained using the adam optimizer. The network reached a 96,53% validation accuracy rate.



Table 8 Re-training results of network models at 0.001 learning rate

No	Architecture	Learning Algorithm	Learning Rate	Batch Size	Validation Accuracy
1	AlexNet	sgdm	0.001	30	95.9
2	AlexNet	adam	0.001	30	50
3	AlexNet	rmsprop	0.001	30	95.22
4	GoogLeNet	sgdm	0.001	30	95.46
5	GoogLeNet	adam	0.001	30	95.75
6	GoogLeNet	rmsprop	0.001	30	95.54
7	ResNet-50	sgdm	0.001	30	95.57
8	ResNet-50	adam	0.001	30	95.66
9	ResNet-50	rmsprop	0.001	30	95.05
10	MobileNet-v2	sgdm	0.001	30	95.09
11	MobileNet-v2	adam	0.001	30	96.53
12	MobileNet-v2	rmsprop	0.001	30	96.31
13	VGG-16	sgdm	0.001	30	93.89
14	VGG-16	adam	0.001	30	50
15	VGG-16	rmsprop	0.001	30	50

Performance rates from highest to lowest at 0.001 learning rate are given in Figure 19. According to the experimental results, the best results were obtained from the combination of MobileNet-v2 architecture, adam learning algorithm. Goodfits are obtained except three experiments. Combinations VGG16 -sgdm, VGG16-adam, AlexNet-adam failed with this problem.

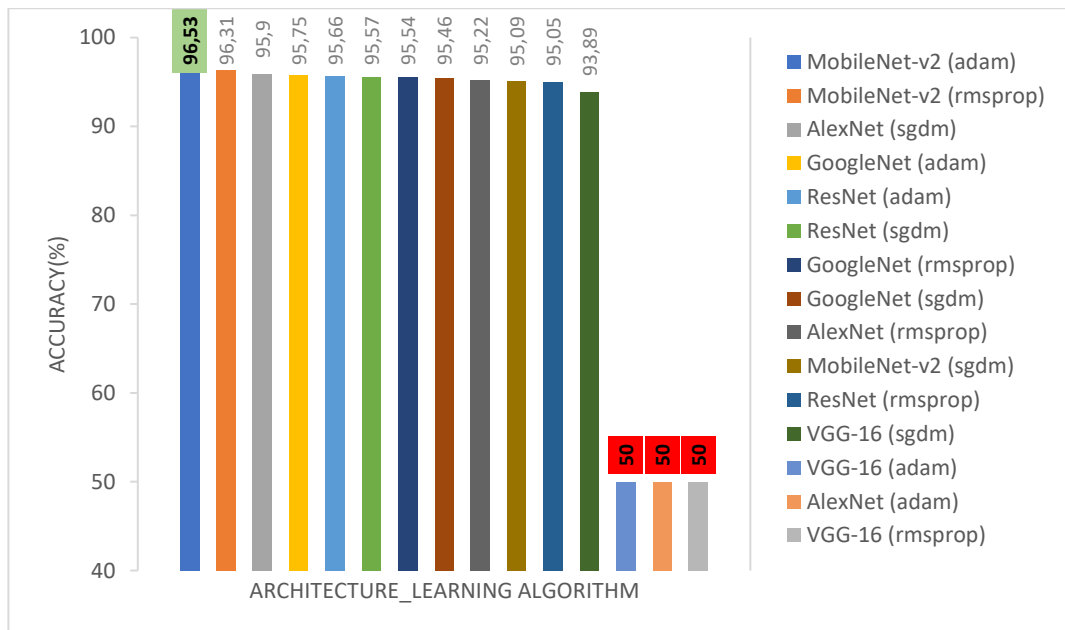


Figure 11 Success Rates of Models at 0.001 learning rate

Table 9. represents the entire training results for 0.0001 initial learning rate, 30 batch size, and 10 epoch. The most successful results were obtained when the ResNet-50 network was trained using the adam optimizer. The network reached a 96,76 % validation accuracy rate. The optimizer type setting is important when using a low learning rate.

Table 9 Re-training results of network models at 0.0001 learning rate

No	Architecture	Learning Algorithm	Learning Rate	Batch Size	Validation Accuracy
1	AlexNet	sgdm	0.0001	30	96.08
2	AlexNet	adam	0.0001	30	94.19
3	AlexNet	rmsprop	0.0001	30	95.85
4	GoogleNet	sgdm	0.0001	30	96.07
5	GoogleNet	adam	0.0001	30	96.26
6	GoogleNet	rmsprop	0.0001	30	96.73
7	ResNet-50	sgdm	0.0001	30	95.65
8	ResNet-50	adam	0.0001	30	96.76
9	ResNet-50	rmsprop	0.0001	30	96.07
10	MobileNet-v2	sgdm	0.0001	30	95.72
11	MobileNet-v2	adam	0.0001	30	95.63
12	MobileNet-v2	rmsprop	0.0001	30	96.13
13	VGG-16	sgdm	0.0001	30	95.42
14	VGG-16	adam	0.0001	30	96.46
15	VGG-16	rmsprop	0.0001	30	95.52

Performance rates from highest to lowest at 0.0001 learning rate are given in Figure 20. According to the experimental results, the best results were obtained from the combination of ResNet architecture, adam learning algorithm.

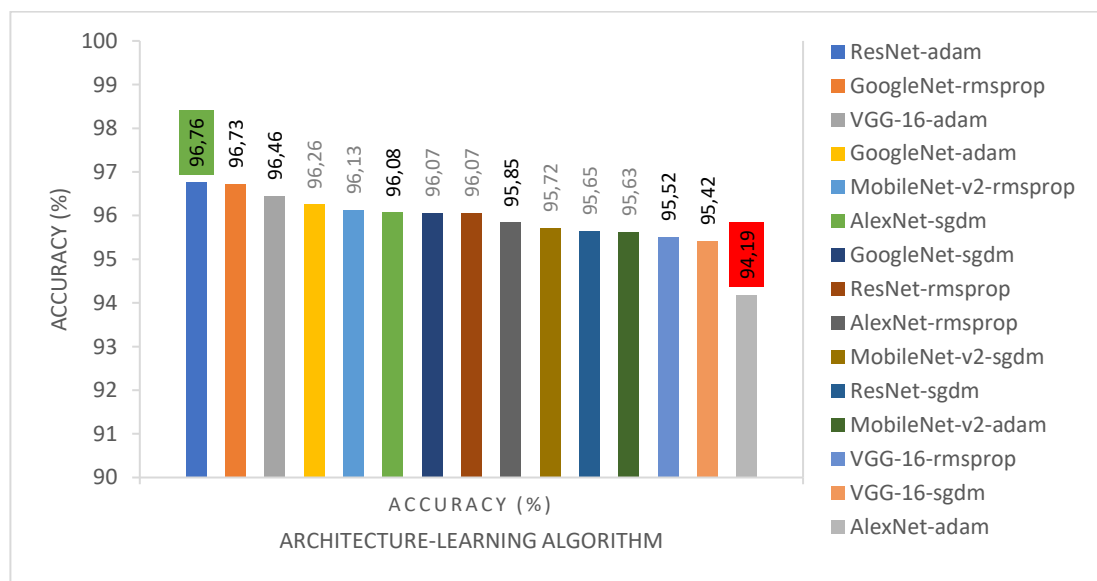


Figure 20 Success Rates of Models at 0.0001 learning rate

In general, the training results were good when the learning rate value was set to 0.0001. Goodfits have been obtained. There is not much difference between the performance rates of the models. For situations where the performance ratios are close to each other, it would be logical to choose the architecture with less number of parameters. In this way, the processing load is less and the result is calculated faster.

## 6. Conclusions

Malaria is a type of disease that kills when left untreated. Thousands of people die each year due to this disease. When treated, there is full recovery. Malaria can be diagnosed by looking for the malaria




parasite in the red blood cell. Deep learning techniques are frequently used in disease detection. Deep learning techniques are very successful in classification problems. Using transfer-based deep learning techniques provides fast and high-performance solutions in image classification. Pre-trained networks are trained using millions of data sets and have proven architectures. In this study, the effect of 3 types of learning algorithms on the performance of 5 types of pre-trained networks at two different learning rate values was investigated. The disease was diagnosed by classifying the red blood cells as having or not having malaria parasites. The duration of the re-trainings, the success rates, and the effects of the learning algorithm on the success was interpreted. When the learning value is set to 0.0001 with the ResNet-50 model and adam optimizer, the maximum success rate of 96.76% has been reached.

## References

- [1] "Sıtma." [Online]. Available: <https://hsgm.saglik.gov.tr/tr/zoontikvektorel-sitma/detay.html>.
- [2] WHO, "World malaria report 2020- WHO," 2020. [Online]. Available: <https://www.who.int/publications/i/item/9789240015791>.
- [3] "What is malaria?," *Global Health, Division of Parasitic Diseases and Malaria*, 2021. [Online]. Available: <https://www.cdc.gov/>.
- [4] E. Soylu, T. Soylu, and R. Bayir, "Design and implementation of SOC prediction for a Li-Ion battery pack in an electric car with an embedded system," *Entropy*, vol. 19, no. 4, 2017.
- [5] Y. Karabacak and A. Uysal, "Fuzzy logic controlled brushless direct current motor drive design and application for regenerative braking," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017, pp. 1–7.
- [6] A. Uysal, S. Gokay, E. Soylu, T. Soylu, and S. Çaşka, "Fuzzy proportional-integral speed control of switched reluctance motor with MATLAB/Simulink and programmable logic controller communication," *Meas. Control (United Kingdom)*, vol. 52, no. 7–8, 2019.
- [7] L. V. Selby, W. R. Narain, A. Russo, V. E. Strong, and P. Stetson, "Autonomous detection, grading, and reporting of postoperative complications using natural language processing," *Surg. (United States)*, vol. 164, no. 6, pp. 1300–1305, 2018.
- [8] A. Shustanov and P. Yakimov, "CNN Design for Real-Time Traffic Sign Recognition," *Procedia Eng.*, vol. 201, pp. 718–725, 2017.
- [9] Y. LeCun *et al.*, "Comparison of learning algorithms for handwritten digit recognition," in *International conference on artificial neural networks*, 1995, vol. 60, pp. 53–60.
- [10] Philipp Seeböck, "Deep Learning in Medical Image Analysis," *Vienna University of Technology Faculty of Informatics, Master Thesis*, 2015.
- [11] U. Kaya, A. Yılmaz, and Y. Dikmen, "Sağlık Alanında Kullanılan Derin Öğrenme Yöntemleri," *Eur. J. Sci. Technol.*, no. 16, pp. 792–808, 2019.
- [12] V. B. Kumar, S. S. Kumar, and V. Saboo, "Dermatological Disease Detection Using Image Processing and Machine Learning," *2016 3rd Int. Conf. Artif. Intell. Pattern Recognition, AIPR 2016*, pp. 88–93, 2016.
- [13] S. Jain, V. Jagtap, and N. Pise, "Computer aided melanoma skin cancer detection using image processing," in *Procedia Computer Science, International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015)*, 2015, vol. 48, no. C, pp. 735–740.
- [14] A. Chaudhary and S. S. Singh, "Lung cancer detection on CT images by using image processing," *Proc. Turing 100 - Int. Conf. Comput. Sci. ICCS 2012*, pp. 142–146, 2012.
- [15] P. Kumar Mallick, S. H. Ryu, S. K. Satapathy, S. Mishra, G. N. Nguyen, and P. Tiwari, "Brain MRI Image Classification for Cancer Detection Using Deep Wavelet Autoencoder-Based Deep Neural Network," *IEEE Access*, vol. 7, pp. 46278–46287, 2019.
- [16] M. J. Horry *et al.*, "COVID-19 Detection through Transfer Learning Using Multimodal Imaging Data," *IEEE Access*, vol. 8, pp. 149808–149824, 2020.
- [17] M. Toğaçar, B. Ergen, and Z. Cömert, "Tumor type detection in brain MR images of the deep model developed using hypercolumn technique, attention modules, and residual blocks," *Med. Biol. Eng. Comput.*, vol. 59, no. 1, pp. 57–70, 2021.
- [18] A. A. Abbasi *et al.*, "Detecting prostate cancer using deep learning convolution neural network with transfer learning approach," *Cogn. Neurodyn.*, vol. 14, no. 4, pp. 523–533, 2020.

- [19] T. Rahman *et al.*, “Transfer learning with deep Convolutional Neural Network (CNN) for pneumonia detection using chest X-ray,” *Appl. Sci.*, vol. 10, no. 9, 2020.
- [20] Vijayalakshmi A and Rajesh Kanna B, “Deep learning approach to detect malaria from microscopic images,” *Multimed. Tools Appl.*, vol. 79, no. 21–22, pp. 15297–15317, 2020.
- [21] Y. Dong *et al.*, “Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells,” *2017 IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2017*, pp. 101–104, 2017.
- [22] F. Yang *et al.*, “Deep Learning for Smartphone-Based Malaria Parasite Detection in Thick Blood Smears,” *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 5, pp. 1427–1438, 2020.
- [23] W. D. Pan, Y. Dong, and D. Wu, “Classification of Malaria-Infected Cells Using Deep Convolutional Neural Networks,” in *Machine Learning - Advanced Techniques and Emerging Applications*, 2018, pp. 159–173.
- [24] A. Sai Bharadwaj Reddy and D. Sujitha Juliet, “Transfer learning with RESNET-50 for malaria cell-image classification,” *Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019*, pp. 945–949, 2019.
- [25] K. M. F. Fuhad, J. F. Tuba, M. R. A. Sarker, S. Momen, N. Mohammed, and T. Rahman, “Deep learning based automatic malaria parasite detection from blood smear and its smartphone based application,” *Diagnostics*, vol. 10, no. 5, 2020.
- [26] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2018.
- [27] B. Bayar and M. C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer,” *IH MMSec 2016 - Proc. 2016 ACM Inf. Hiding Multimed. Secur. Work.*, pp. 5–10, 2016.
- [28] D. Miao, W. Pedrycz, D. Ślezak, G. Peters, Q. Hu, and R. Wang, “Mixed Pooling for Convolutional Neural Networks,” in *International Conference on Rough Sets and Knowledge Technology*, 2014, vol. 8818, pp. 364–375.
- [29] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, “Learning Pooling for Convolutional Neural Network,” *Neurocomputing*, vol. 224, no. April 2016, pp. 96–104, 2017.
- [30] S. Postalcıoğlu, “Performance Analysis of Different Optimizers for Deep Learning-Based Image Recognition,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 2, 2020.
- [31] H. Chen *et al.*, “Deep Transfer Learning for Person Re-Identification,” *2018 IEEE 4th Int. Conf. Multimed. Big Data, BigMM 2018*, 2018.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “2012 AlexNet,” *Adv. Neural Inf. Process. Syst.*, 2012. [Online]. Available: <https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [34] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [36] S. Rajaraman *et al.*, “Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images,” *PeerJ*, vol. 6, p. e4568, 2018.
- [37] “Malaria Cell Images Dataset.” [Online]. Available: <https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria>.

# A Decision Support System For Detecting Stage In Hodgkin Lymphoma Patients Using Artificial Neural Network and Optimization Algorithms

 Fatma Akalın<sup>1</sup>,  Mehmet Fatih Orhan<sup>2</sup>,  Mustafa Büyükavcı<sup>3</sup>

<sup>1</sup>Corresponding Author; Information System Engineering Department, Faculty of Computer and Information Sciences, Sakarya University, Sakarya, Türkiye; fatmaakalin@sakarya.edu.tr

<sup>2</sup>Department of Pediatric Hematology and Oncology; Sakarya University Faculty of Medicine; Sakarya, Türkiye; forhan@sakarya.edu.tr

<sup>3</sup>Department of Internal Medical Sciences; Meram Faculty of Medicine; Necmettin Erbakan University; Konya; Türkiye; buyukavci@hotmail.com

Received 27 November 2022; Revised 03 December 2022; Accepted 06 December 2022; Published 31 December 2022

## Abstract

Hodgkin-type lymphoma is a disease with unique histological, immunophenotypic, and clinical features. This disease occurs in nearly 30% of all lymphomas. Its treatable is high. However, the treatment plan is specified after the stage and risk status are determined. For this reason, it is an important process for doctors to decide on the stage of the disease correctly. Some of the data used for this decision are the patient's history, detailed physical examination, laboratory findings, imaging methods and bone marrow biopsy results. Hybrid FDG-PET is an important imaging method used in the medical world. This method is used in diagnosis, evaluation of response given to treatment, staging and restaging process. However, it is radiation-based. Therefore it has the possibility of producing undesirable results in the future. In this study, an artificial intelligence-based computer-assisted decision support system is done to reduce the number of used medical methods and radiation exposure. Data were obtained from the NCBI-GEO dataset. The evaluation of these data, which contains missing values, is handled in two ways. Firstly, samples with missing values in the initial evaluation are deleted from the dataset. Then, these data are trained with “trainlm” function in artificial neural network architecture. However, reducing the error value of the estimates is important. For this, the artificial neural network architecture is retrained with the artificial bee colony algorithm, particle swarm optimization algorithm and invasive weed algorithm, respectively. Secondly, the same operations are performed again on the dataset containing missing values. As a result of the training, the maximum performance was obtained for invasive weed and particle swarm optimization algorithms with 1,45547E+14 and 1,23103E+14 average error rates, respectively.

**Keywords:** staging in hodgkin lymphoma, artificial neural networks, particle swarm optimization algorithm, invasive weed optimization algorithm, constructing hybrid structure for decision support system

## 1. Introduction

Hodgkin-type lymphoma was described by Thomas Hodgkin in 1832. This disease occurs from the B cell lineage. It has unique histological, immunophenotypic and clinical features. It consists of nearly 30% of all lymphomas. Hodgkin lymphoma is classified into two groups according to the WHO (World Health Organization) guide. The first group is Nodular Lymphocyte Predominant Hodgkin Lymphoma (NLP-HL), which constitutes 5% of Hodgkin lymphomas. The second group is Classical Hodgkin Lymphoma, which constitutes 95% of Hodgkin lymphomas. Classical Hodgkin Lymphoma group is also evaluated in 4 subclasses that are Nodular Sclerosis, Mixed Cellularity, Lymphocyte Rich and Lymphocyte Depleted. In this scale that constitutes 95% of Hodgkin lymphomas, the incidence rate of these subclasses is stated as 70%, 20%, 5% and 5%, respectively. The symptoms of Hodgkin lymphoma patients are manifested by painless enlargement of lymph nodes, spleen, and other immune tissues. Fever, night sweats, itchy skin, weight loss, loss of appetite and fatigue are other possible symptoms [1].

Hodgkin lymphoma is a treatable disease. The survival rate for 5 years is 81%. However, there is a possibility of the relapsing of the disease to different regions. This possibility is seen in nearly 30% of patients. Hodgkin lymphoma consists of 4 stages. Stage 1 is the involvement of a single lymph node

region or lymphoid structure. Stage 2 is the involvement of two or more lymph node regions for the same side of the diaphragm. Stage 3 is the involvement of lymph node regions or structures for both regions of the diaphragm. Stage 4 is the diffuse involvement of one or more extra lymphatic organs [1].

A treatment plan is made after determining the stage and risk group for HL patients. Characterizations for early-stage good risk, early-stage bad risk and advanced-stage are made. These characterizations are (Stage I-II, no adverse factor), (Stage I-II, any of the negative factors), (Stage III-IV disease) respectively [2].

The decision to be made for the stage of the disease by the doctors is important. Detailed physical examination, laboratory findings, imaging methods (chest x-ray, chest and CT scan), and bone marrow biopsy results are used for this decision. However, pathological examinations performed in NLP Hodgkin Lymphoma and Classical Hodgkin Lymphoma are monitored for different target cells and a sufficient sample is an important parameter in pathological examinations. On the other hand, biopsy evaluations containing insufficient malignant cells do not produce clear results. At the same time, it is stated that CT (Computed tomography), which is a radiological-based imaging method, is insufficient in demonstrating spleen involvement. In addition, nearly 15% of HL patients, which were described as an early stage in the staging process with CT, were shown to be advanced stage with PET-CT (Positron Emission Tomography-Computed Tomography). For this reason, the patient's history, detailed physical examination, laboratory tests, bone marrow biopsy and imaging approaches is a critical issues in order to decide on the correct diagnosis and stage [1][2].

Positron Emission Tomography is an approach that reaches the patient from vascular after the radioactive labelling of glucose sugar. The distribution of this substance in the body is examined using a scanner. Glucose is a substance used a lot, especially by lymphoma cells. Therefore, as a result of this process, the body's glucose metabolism changes and sick areas appear [2]. The PET-CT method is accepted in the medical world as a standard for the detection of tumour cells [2]. In the scope of this hybrid structure, while PET provides information on the distribution of glucose in the body, CT provides anatomical details of normal and pathological tissues in the body. Hybrid PET-CT is also used in diagnosis, evaluation of response given to treatment, staging and restaging [2][3]. FDG-PET/CT has a higher sensitivity than bone marrow biopsy [4]. Since 2014, the response given to treatment is evaluated according to CT and PET-CT [2]. Today, the most widely used PET radiopharmaceutical is fluorodeoxyglucose (FDG), a glucose analog labeled with Fluor-18 (F-18) [5]. After it is given to the body, it accumulates in the bladder. This results in an increase in the rate of radiation in the bladder and its membrane. On the other hand, the late effects of radiation are another important issue. Standard dose protocol or dose protocol varying according to weight is an important criterion to be considered in PET-CT procedures to be applied to the patient. However, it is stated that the rate of PET-CT facilities, where patient weight is taken into account, is 44% [3]. In addition, if there is a suspicion of pregnancy or if the breastfeeding process continues, it should be done under the control of a doctor within the scope of certain instructions [5]. Also, positron emission tomography is a medical method with high economic costs [4].

FDG-PET, which is also included in the post-treatment evaluation process, may produce false positive results for some conditions. This situation causes the to be misdirected in the evaluation process due to post-treatment infections. As a result of misdirections; unnecessary radiation exposure, biopsies and patient anxieties occur [4].

In normal conditions, the prognosis of Hodgkin lymphoma is good. Especially if the disease is in its early stage and there are no undesirable factors, the healthy life expectancy for 5 years is nearly 90%. Even in the advanced stage of the disease, this rate is in the period of 70%-90%. For HL patients with a high probability of success, the main goals of future-oriented are to prevent the treatment's side effects and to make an early diagnosis. Depending on the late treatment, conditions such as breast cancer, thyroid cancer, GIS cancer, leukemia, soft tissue sarcoma, lung cancer and other different cancers, cardiovascular diseases, and organ failure may occur. Therefore, early diagnosis and early prediction of alternative treatment options are important in terms of preventing complications and unnecessary treatments [6].

In the literature, decision support systems using different methods have been constructed to evaluate the diagnosis, stage and response given to treatment of lymphoma disease. Because it is important to choose the correct diagnosis and staging for the treatment to be effective. In this context, in the [7] study a multiclass classification of non-hodgkin lymphomas was made. The best success rate achieved in this article, in which morphological and non-morphological descriptors were extracted from cell nuclei, was obtained as 0.956 with the linear regression approach. In the [8] study is provided to distinguish cancer lesions from other structures with FDG PET/CT. The maximum performance criterion reached with the SVM classifier was obtained as 0.91 for the AUC value. In the [9] study is aimed segmentation and classification of lymphoma histological images. Firstly, the segmentation process is carried out with evolutionary algorithms. Then, classification is provided with the Support Vector Machine method using the texture and color features extracted from the images. The best average accuracy rate is obtained with Differential Evolution technique as 99.38%. In the [10] study is classified centroblast and non-centroblast cells for microscopic images obtained through follicular lymphoma tissue biopsy with Support Vector Regression and Radial Basis Kernel Function. The average detection accuracy is obtained as 97.44%. The feature extraction method is developed in the [11] study to classify 3 lymphoma types (mantle cell lymphoma, follicular lymphoma and chronic lymphocytic leukemia). Features are extracted from lymphoma images using this developed method. Then these attributes are classified by decision tree (DT), support vector machines (SVM), random forests (RaF), naive bayes (NB), K-star ( $K^*$ ) classifiers. The best result achieved in a result of the classification process is obtained with the random forest. On the test dataset, this ratio is evaluated with the cross validation method and the AUC ratio is found as 0.963. In the [12] study used stained tissue images; benign lesion, carcinoma, and lymphoma data is classified with the weighted KNN model. The average success rate is improved with CLAHE and PCA methods. The maximum success rate achieved is obtained as 85.5%. In the [13] study, features are extracted from histological images using a convolutional neural network for the classification of malignant lymphatic images. Then these feature vectors are given as inputs to Convolutional Neural Networks, Support Vector Machines (SVM) and Random-Forests classifiers. The convolutional neural network model produced the best result with a classification success of 93.27%. In the [14] study, lymphoma images belonging to 3 classes is classified. In the classification process, Resnet 50, modified Resnet50 and 5-layer CNN structures are trained with different optimization algorithms. The achieved maximum accuracy is 0.9990 for the KIMIA Path 960 dataset and 0.9813 for the NIA Curated dataset.

Studies in the literature were carried out to create a computer-based decision support system. Because the treatment process can be difficult and laborious for both patients and doctors. The oncologists, especially for patients with HL uses critical parameters such as anamnesis, detailed physical examination, laboratory findings, imaging modalities (X-ray, CT scan, and PET-CT) and bone marrow biopsy results for early diagnosis, evaluation of response given to treatment, staging or restaging. [15][16][17][18][19][20] studies indicate that PET-CT imaging approach offers more successful outcomes in the process of deciding the stage of lymphoma disease compared to other approaches. All these approaches applied for the diagnosis and treatment of patients diagnosed with HL may cause anxiety on patients or radiation-based methods may produce undesirable results in the long term. For this reason, a decision support system was built to reduce the number of methods used in staging and the radiation exposure.

In this study, initially, diagnostic biopsy samples taken from Hodgkin lymphoma patients were obtained from the NCBI-GEO dataset. These data consist of IPS score, age, gender information, albumin level, hemoglobin level, lymphocyte ratio and white blood cell count features. However, some patient samples contain missing values. Therefore, 2 different evaluations were made regarding the data set. In the first evaluation, patient samples containing missing values from the dataset were deleted. The remaining 99 data were given as input to the artificial neural network. This model was trained with the `trainlm` function updating the bias and weight values according to the Levenberg-Marquardt approach. In this model consisting of 5 neurons and 1 hidden layer, the `tansig` and `purelin` transfer functions were used. The achieved error rate was found as  $2.33185E+14$ . Then, the same processes were repeated for the dataset that contain missing values. The error rate reached for this evaluation was also found as  $4.89165E+13$ .

However, in order to increase the power of the predictions and to give more successful decisions for the 4 stages, the same neural network was retrained with ABC, PSO and IWO algorithms. As a result of 1000 iterations, the most successful output for the test dataset that does not contain missing values was obtained as the artificial neural network trained with the PSO algorithm. The minimum error value reached on the test data set was found as  $1.23103e+14$ . Then, the same operations performed with the optimization algorithms were performed again on 130 data that contain missing values. As a result of the training, the most successful output was obtained with the artificial neural network trained with the IWO algorithm. The minimum error rate reached for the second evaluation was found as  $1.45547E+14$ . An improvement has been made on the test datasets for both evaluations and the achieved error rates are close values. This shows that the data of patients diagnosed with HL containing missing values can be tolerated with the hybrid use of artificial intelligence and optimization algorithms. At the same time, real-world data is likely to contain missing values. Consequently, it is planned that the created the computer-aided decision support system will give an idea to the doctors.

## 2. Material and Method

This section presents an artificial intelligence-based study in order to correctly decide on the stage of patients diagnosed with lymphoma. In this study, firstly, an estimation process in which artificial neural networks are used in the training of the model takes place. Then, three different optimization algorithms named artificial bee colony, particle swarm and invasive weed are used to improve the obtained performance of predictions. The format of the dataset, classification method and optimization approaches are explained below.

### 2.1. Dataset

In this study, NCBI-GEO dataset is used. The used data has the id numbers in the GSM447610-GSM447739 range in the GSE17920 series and it is provided from the [21] site. The dataset consists of data containing values of diagnostic biopsy samples obtained from patients with Hodgkin lymphoma. In this study, estimation is done using IPS score, age, gender information, albumin level, hemoglobin level, lymphocyte ratio and white blood cell count values. However, there are missing values in the data obtained from 130 different individuals. For this reason, two separate evaluations are made. In the first evaluation, 130 different samples that all data is used. The second evaluation is carried out on 99 separate data that consists of not missing values. The IPS (International Prognostic Score) score, which is included in the features, is a feature that provides an evaluation of the prognosis on a certain scale. It is evaluated in the range of 0 to 7. An IPS value closer to 7 indicates increased risk [22]. 70% and %30 of the data used in this study are chosen randomly from all data as training and testing datasets. Then, these data are given as input to the artificial neural network architecture.

### 2.2. Used Neural Network Architecture and Optimization Algorithms

Artificial intelligence is a framework that enables the cognitive abilities of humans to be imitated by machines. It can be used in different fields such as mathematics, statistics, linguistics, computer science, neurobiology and psychology [23]. However, it also has a tendency to perform error-prone tasks that are studied using statistical methods and whose evaluation of human intelligence is impractical [24]. In particular, it provides an effective interpretation of the data created by people as a result of observation. It also has the power inferring for specific clinical diagnostic tasks from large and complex data stacks [24].

Artificial intelligence, which can be used as a decision support system in the medical field, performs the learning actions on the same type of data for a clinical diagnosis. Then it learns the interpretation function on the target data [24]. The interpretation task differs in the scope of the target problems, such as time series analysis, computer vision, or natural language processing [24].



Artificial intelligence is a discipline that contains machine learning, neural networks and deep learning [23]. The clinical data used in this study are analyzed with neural networks and then inferences are made.

Neural Networks are developed with inspiration from the biological nervous system. This structure consists of an input layer, an output layer and the hidden layer/layers. In the first stage of the structure, the properties of the data taken as input are automatically extracted. Then the weight values are adjusted and the activation function is applied. Weight values are updated to strengthen or weaken connections in the network to produce successful performance with mathematical functions defined by the system [23]. The activation function is used to teach the neural network nonlinear real-world problems that have a great impact on the performance of the network [25].

The neural network architecture used in this study is given in Figure 1.

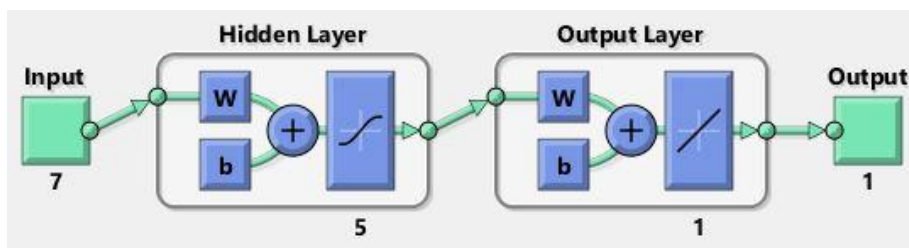


Figure 1 The neural network architecture used in this study

In this architecture adopting the feed-forward backpropagation approach, the inputs are transmitted along the next layers. The errors that occur in this process are fed back and a successful training model is obtained. The architecture used in the study consists of 1 hidden layer and it runs on 7 separate input features. The number of neurons in this hidden layer is defined as 5. The trainlm training function is selected for network's training. The trainlm function updating bias and weight values works according to Levenberg-Marquardt optimization [26]. Two different neural transfer functions are used to calculate the output from the input of each layer. These transfer functions defined as tansig and purelin are used in the hidden and output layers, respectively. These functions were selected as a result of fine-tuning. The graphical representation of the hyperbolic tangent sigmoid transfer function (tansig) and linear transfer function (purelin) is given in Figure 2 [26].

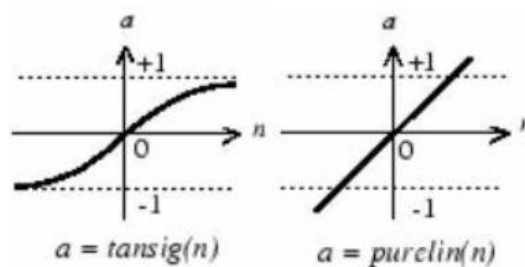


Figure 2 The transfer functions used in this study[26]

The one output is obtained about stage of disease after the classification. The maximum similarity between the real results and the results obtained from the training of the artificial neural network is achieved with the network structure expressed in Figure 1. However, it is important to increase the success of predictions. For this, optimization algorithms were used.

Optimization algorithms produce correct, stable and effective solutions for the available problems. Different optimization algorithms have been developed for real-life problems. Optimization algorithms are examined in two main frameworks as stochastic and deterministic optimization algorithms.

Stochastic optimization algorithms that contain randomness compared to deterministic optimization algorithms are evaluated in 2 parts heuristic and metaheuristic algorithms. In this study, metaheuristic algorithms are used. These algorithms are particle swarm optimization algorithm, artificial bee colony optimization algorithm and invasive weed optimization algorithm and it is based on colony intelligence [27].

The basic logic of the algorithm is the gathering of entities with limited capabilities in order to achieve the targeted purpose. It is inspired by communities that can easily find answers to difficult problems [27][28]. This indicates the spreading behaviour of bird colonies organized for foraging purposes, honey bees in which foraging behaviour is simulated, and weeds invading the field for the particle swarm optimization algorithm, artificial bee colony algorithm and invasive weed algorithm, respectively [27].

Optimization algorithms are methods that optimally solve the relevant problem under certain conditions for complex and difficult targets. It plays an important role in improving performance [27]. For this reason, 3 different optimization algorithms were used in the training of artificial neural networks and the evaluations were made again.

### 3. The Research Findings and Discussion

The hyperparameters of the artificial neural network model adopting the feedforward backpropagation approach are fine-tuned. Then, the artificial neural network model is trained. The outputs of classification produced by the trained model on the training and testing dataset that do not contain missing values are presented in Figure 3.

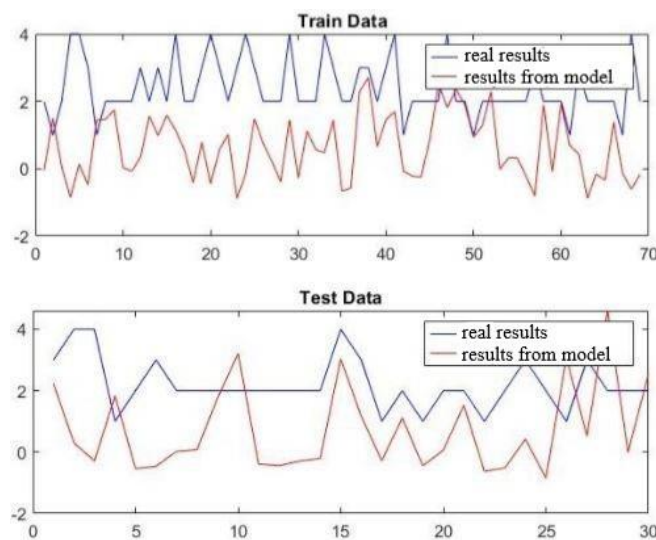


Figure 3 The outputs produced by the artificial neural network architecture trained with the trainlm function for the dataset that does not contain missing values

When Figure 3 is examined, the predictive power of the outputs obtained even after the fine-tuning process is not found enough. For this reason, the training of artificial neural networks was carried out with optimization algorithms instead of trainlm function. Thus, the predictive power of artificial neural networks were increased with the 3 different optimization algorithms (artificial bee colony optimization algorithm, particle swarm optimization algorithm and invasive weed optimization algorithm). Then a comparison was made via alternative solutions in the staging and restaging of patients diagnosed with lymphoma on 99 data that do not contain missing values. The outputs of classification produced in the training and test datasets for the 4 stages of patients diagnosed with HL are given in Figures 4, 5 and 6, respectively.

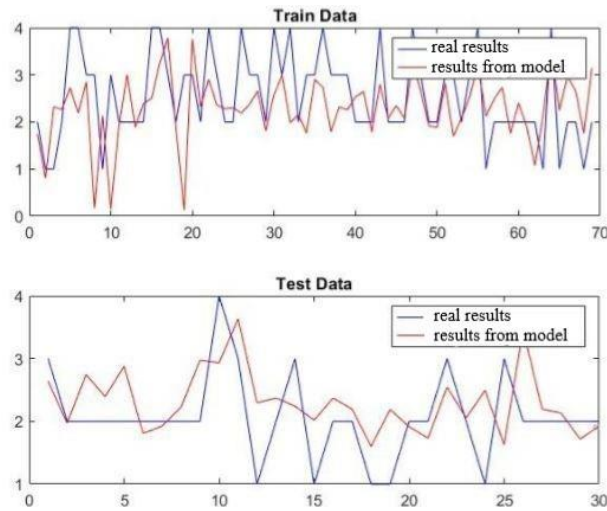


Figure 4 The outputs produced by the artificial neural network architecture trained with the ABC optimization algorithm for the dataset that does not contain missing values

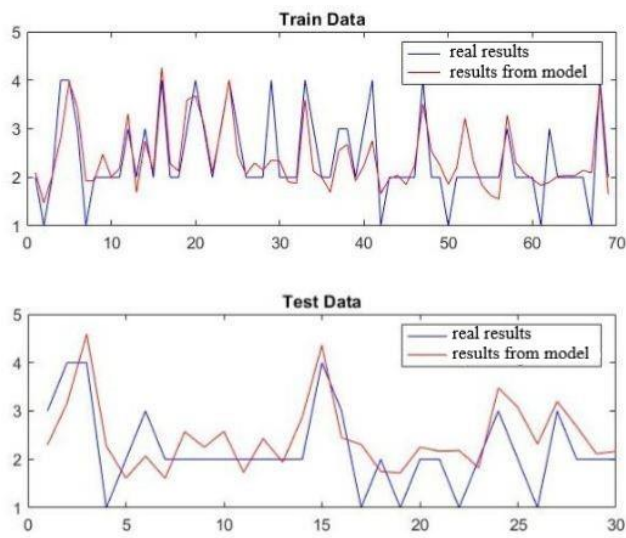


Figure 5 The outputs produced by the artificial neural network architecture trained with the PSO optimization algorithm for the dataset that does not contain missing values

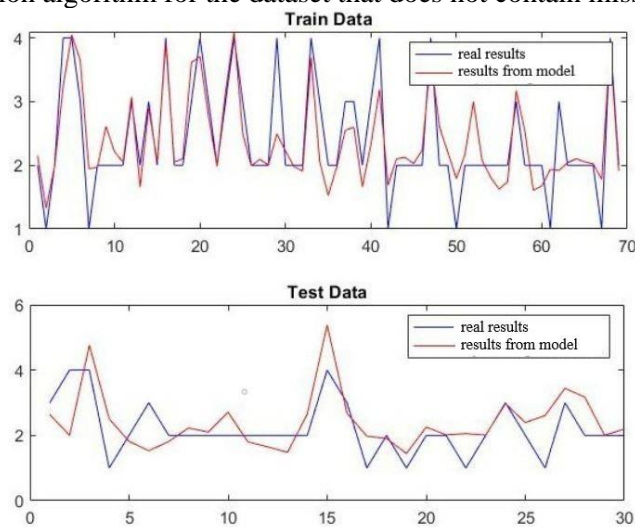


Figure 6 The outputs produced by the artificial neural network architecture trained with the IWO optimization algorithm for the dataset that does not contain missing values

The same evaluation is repeated for 130 different samples, among which there were missing data. The results of classification obtained using 3 different optimization algorithms are given in Figures 7.8 and 9, respectively.

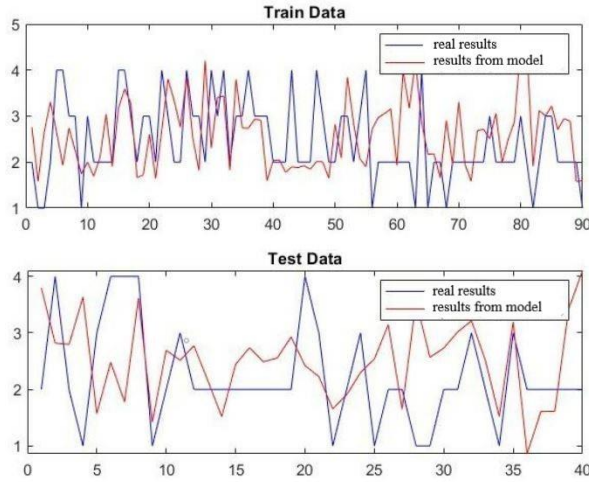


Figure 7 The outputs produced by the artificial neural network architecture trained with the ABC optimization algorithm for the dataset that contains missing values

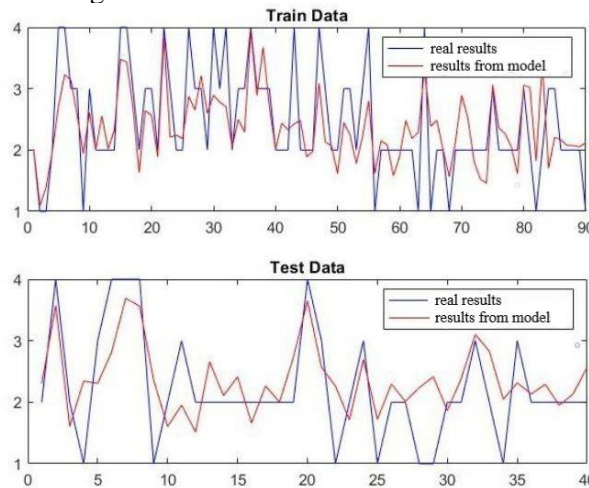


Figure 8 The outputs produced by the artificial neural network architecture trained with the PSO optimization algorithm for the dataset that contains missing values

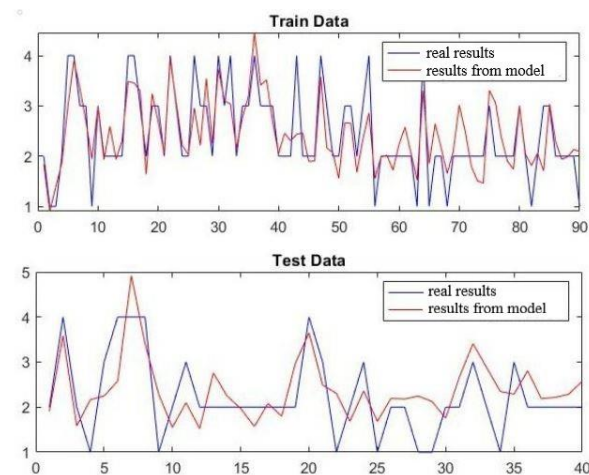


Figure 9 The outputs produced by the artificial neural network architecture trained with the IWO optimization algorithm for the dataset that contains missing values

The outputs produced in the dataset with and without missing data are fuzzy values. These results are produced to give an idea to doctors. The output will be interpreted according to the evaluation process of the doctors. However, a statistical evaluation process is also carried out in this study. In this evaluation process, the target fuzzy value will be characterized according to the phase to which it is closest mathematically. The outputs obtained statistically in the dataset with and without missing data are given in Table 1 and Table 2.

Table 1 Evaluation criteria reached for the dataset with missing data

FOR MISS. DATA	Stage 1			Stage 2			Stage 3			Stage 4		
	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.
YSA	0.14	0.60	0.23	0.5	0.13	0.21	0	0	0	0	0	0
YSA ABC	0	0	0	0.59	0.76	0.66	0.14	0.2	0.16	0	0	0
YSA PSO	0	0	0	0.6	0.705	0.64	0.25	0.4	0.307	1	0.66	0.79
YSA IWO	0.50	0.2	0.285	0.76	0.76	0.76	0.44	0.80	0.57	1	0.66	0.79

Table 2 Evaluation criteria reached for the dataset without missing data

FOR FULL DATA	Stage 1			Stage 2			Stage 3			Stage 4		
	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.	Prec.	Sens.	F scr.
YSA	0	0	0	0.64	1	0.78	0.20	0.16	0	0	0	0
YSA ABC	0.50	0.14	0.22	0.50	0.36	0.42	0.18	0.50	0.27	0.16	0.20	0.18
YSA PSO	0	0	0	0.64	0.81	0.71	0.37	0.50	0.42	1	0.80	0.88
YSA IWO	0	0	0	0.57	0.72	0.63	0.11	0.16	0.13	1	0.60	0.75

The sensitivity criterion given in Table 1 and Table 2 gives the rate at which the target data is estimated correctly among all the predictions belonging to the same category. The Precision criterion gives the rate at which the target data is predicted correctly in all categories. The F criterion is the harmonic mean of the sensitivity and precision criteria. This criterion provides that outliers are taken into account [29].

However, there is a disadvantage to this evaluation process. For an example where the real stage is 1, the model can produce a fuzzy value of 1.51. In this case, the stage predicted by the computer-aided system will be 2. For this reason, the performance produced by the same fuzzy outputs will change with different evaluation approaches to be produced by computer aided systems. For this, it is aimed that the evaluation process is interpreted by the doctors and evaluated as a new parameter. For this reason, the improvement of the methods applied in this study on the outputs is explained with the mean error value.

The classification outputs produced by the artificial neural network model trained with 3 different optimization algorithms during 1000 iterations on the training and testing dataset are examined. The average error rates obtained according to the optimization algorithms are given in Table 3.

Table 3 Average error values trained according to models

Structures trained on the training and test dataset	Average error for 130 data	Average error for 99 data
YSA (TRAIN DATASET)	4,81727E+13	2,40011E+14
YSA (TEST DATASET)	4,89165E+13	2,33185E+14
ABC_YSA (TRAIN DATASET)	4,47807E+13	1,53627E+14
ABC_YSA (TEST DATASET)	6,21566E+13	1,27148E+14
PSO_YSA (TRAIN DATASET)	1,67974E+13	<b>1,25067E+14</b>
PSO_YSA (TEST DATASET)	2,4747E+13	<b>1,23103E+14</b>
IWO_YSA (TRAIN DATASET)	<b>1,29518E+14</b>	1,29518E+14
IWO_YSA (TEST DATASET)	<b>1,45547E+14</b>	1,45547E+14

Table 3 shows the average error data produced by the artificial neural networks trained with the trainlm training function, ABC, PSO and IWO algorithms on the dataset with and without missing values, respectively. Results obtained from Table 3 show that the artificial neural network architecture is not successful enough in deciding the stage of patients diagnosed with HL. Therefore, the same artificial neural network is retrained with ABC, PSO and IWO algorithms to improve the error rate and general performance. The most successful training on the test dataset that contains missing values that they did not see during the training is the artificial neural network trained with the IW optimization algorithm. However, the most successful training on the test dataset that does not contain missing values is the artificial neural network trained with the PSO optimization algorithm. At the same time, the results produced on the dataset that contains missing values are close to the results produced on the dataset that does not contain missing values. It has been experimentally proven that artificial neural network architecture can tolerate this situation in datasets that contain missing values in patients diagnosed with HL. Because real-world data is likely to contain missing values. This hybrid structure produces a result that can increase the final performance even on missing data.

Additionally trained these architectures are not good enough to predict samples that are stage 1. The reason for this situation is the scarcity of data belonging to stage 1. The model cannot produce successful predictions because it cannot learn the patterns related to the first stage sufficiently. To compensate for such a situation, the number of samples belonging to the first stage should be increased. Also, the addition of different clinical data on the diagnosis of HL is an advantage for increasing successful predictions.

Different methods are used for lymphoma disease in the literature. Table 4 shows some studies in the literature.

Table 4 includes some of the studies done in the scope of lymphoma disease from the past to the present. These studies usually involve an estimation process that is used image processing, classification algorithms or statistical methods. On the other hand, this study decides on the staging or restaging process of individuals diagnosed with HL on 7 different clinical data. A different and hybrid structure is used in the study. It has been experimentally proven that this approach can also be used to decide the stage of data with missing values. It is thought that this study including a different approach compared to the studies in the literature will contribute to the literature.

Table 4 Studies related to patients diagnosed with HL

References	Aims & Methods	Evaluation Results
[30]	This study makes a classification based on histological grades of Follicular lymphoma images with MBIR approach.	Classification accuracy in identifying for histological grades of grade 1,2 and 3 is obtained as %74.9, %84.6 and %95.0.
[31]	This study provides a prediction for the subtypes of main malignant lymphoma with SVM and RF classifiers.	The sensitivity and precision rate is obtained as %97.0 and %94.1 respectively.
[32]	This study realizes a prognostic information for HL and NHL patients with CTTA using Kaplan-Meier and Cox regression methods.	This pilot study shows that complementary prognostic information for interim FDG-PET is provided.
[33]	This study presents a classification for histological images of non-Hodgkin lymphoma with SWT and ANOVA approaches.	Best result is obtained with ANOVA approach as %100 accuracy.
[34]	This study applied a method to classify for three types of lymphoid cells with Fuzzy C-Means clustering algorithm.	Maximum classification performance has at HCL cells. Rate of HCL cells taking place in group 3 is %98.
[35]	Detection of centroblast cells on H&E stained Follicular lymphoma tissue samples with the computer-aided system that has 2 steps for specifying staging.	It has %80.7 detection accuracy.
[36]	Classification of 3 types of malignant lymphoma is provided with two-stage approach.	The best signal was obtained as %98-%99 for unseen images.

## Conclusion

This study was done to decide on the staging or restaging process of patients diagnosed with hodgkin lymphoma. For this reason, firstly, data containing the values of diagnostic biopsy samples were obtained from the NCBI-GEO dataset. However, the dataset contains missing values. For this reason, the data were evaluated as 2 separate datasets with and without missing values. In the first stage, the data was trained with the trainlm function of the artificial neural network approach. Then, the same artificial neural network architecture was retrained with ABC, PSO and IWO algorithms in order to reduce the error prediction rate and produce more successful predictions. Optimization algorithms achieved an improvement in the error rates produced as a result of the training. The most successful training on the test dataset with missing values was realized with the IW optimization algorithm and the most successful training on the test dataset without missing values was with the PSO optimization algorithm. For these optimization algorithms in which artificial neural networks are trained, the average error rates achieved for both complete and incomplete datasets are close. Therefore, this hybrid approach has proven its usability on real-world data that may contain missing values for patients diagnosed with HL. This situation will cause a decrease in the use of PET-CT, which is costly. On the other hand, the final performance tends to increase with this hybrid structure having a different target. For this reason, it is expected that the performance of the study will increase with the addition of new clinical data to the dataset in the future. Also, it can be preferred for diseases that are difficult to diagnose. Consequently, it is thought that the present study will contribute to the literature. Finally, since our approach can tolerate the lack of data, it is thought to be a contribution, especially for studies where data collection is difficult and costly.

## References

- [1] A. W. MD, A. Q. MD, A. Dasgupta ‘Hodgkin lymphoma - Chapter 14’, *Hematology and Coagulation (Second Edition)*, pp. 217–225, 2020.
- [2] Z. Abbasov, ‘Hodgkin Hastalığı Tanılı Hastaların Klinik, Laboratuar Bulguları ve Tedavi Sonuçlarının Değerlendirilmesi’, *Uzmanlık Tezi, İstanbul Üniversitesi*, 2017.
- [3] T. Şahmaran and M. Bayburt, ‘Pozitron Emisyon Tomografi-Bilgisayar Tomografi (PET-BT) Uygulamalarında Hastanın Aldığı Radyasyon Dozunun Belirlenmesi’, *Kafkas Univ. Inst. Nat. Appl. Sci. J.*, vol. 13, no. 1, pp. 58–63, 2020.
- [4] S. Y. Aksoy and M. Halac, ‘Pediatrik Hodgkin lenfomalarda FDG PET/BT’, *Turk Onkol. Derg.*, vol. 30, no. 4, pp. 240–251, 2015, doi: 10.5505/tjoncol.2015.1218.
- [5] Ç. Soydal et al., ‘F-18 FDG PET/CT Practice Guideline in Oncology’, *Nucl. Med. Semin.*, vol. 6, pp. 339–357, 2020, doi: 10.4274/nts.galenos.2020.0028.
- [6] P. Ö. Kara, ‘Pediatrik Lenfomalarda PET\_BT Görüntüleme’, *Turkiye Klin. J Nucl Med-Special Top.*, vol. 3, no. 1, pp. 93–99, 2017.
- [7] T. P. De Faria, M. Z. Do Nascimento, and L. G. A. Martins, ‘Understanding the multiclass classification of lymphomas from simple descriptors’, *Proc. - 2021 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2021*, pp. 1202–1208, 2021, doi: 10.1109/CSCI54926.2021.00250.
- [8] C. Lartizien, M. Rogez, E. Niaf, and F. Ricard, ‘Computer-aided staging of lymphoma patients with FDG PET/CT imaging based on textural information’, *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 3, pp. 946–955, 2014, doi: 10.1109/JBHI.2013.2283658.
- [9] T. A. A. Tosta, M. Z. Do Nascimento, P. R. De Faria, and L. A. Neves, ‘Application of Evolutionary Algorithms on Unsupervised Segmentation of Lymphoma Histological Images’, *Proc. - IEEE Symp. Comput. Med. Syst.*, pp. 89–94, 2017, doi: 10.1109/CBMS.2017.69.
- [10] E. Michail, K. Dimitropoulos, T. Koletsa, I. Kostopoulos, and N. Grammalidis, ‘Morphological and textural analysis of centroblasts in low-thickness sliced tissue biopsies of follicular lymphoma’, *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 3374–3377, 2014, doi: 10.1109/EMBC.2014.6944346.
- [11] M. Goncalves Ribeiro, L. Alves Neves, G. Freire Roberto, T. A. A. Tosta, A. S. Martins, and M. Z. Do Nascimento, ‘Analysis of the Influence of Color Normalization in the Classification of Non-Hodgkin Lymphoma Images’, *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 369–376, 2018, doi: 10.1109/SIBGRAPI.2018.00054.
- [12] A. E. Nugroho, W. D. Lukito, I. Anshori, W. Adiprawita, H. A. Usman, and O. Husain, ‘CLAHE Performance on Histogram-Based Features for Lymphoma Classification using KNN Algorithm’, *Proceeding 15th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2021*, 2021, doi: 10.1109/TSSA52866.2021.9768221.
- [13] N. Hatipoglu and G. Bilgin, ‘Classification of Malignant Lymphoma Types Using Convolutional Neural Network’, *2020 Med. Technol. Congr.*, 2020.
- [14] A. Ganguly, R. Das, and S. K. Setua, ‘Histopathological Image and Lymphoma Image Classification using customized Deep Learning models and different optimization algorithms’, *2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020*, 2020, doi: 10.1109/ICCCNT49239.2020.9225616.
- [15] A. I. Kamel, T. F. Taha Ali, and M. A. Tawab, ‘Potential impact of PET/CT on the initial staging of lymphoma’, *Egypt. J. Radiol. Nucl. Med.*, vol. 44, no. 2, pp. 331–338, 2013, doi: 10.1016/j.ejrnm.2012.12.008.
- [16] N. H. E. D. Behairy, T. A. Rafaat, A. S. E. D. El Noyal, and M. I. Bassiouny, ‘PET/CT in initial staging and therapy response assessment of early mediastinal lymphoma’, *Egypt. J. Radiol. Nucl.*



- Med.*, vol. 45, no. 1, pp. 61–67, 2014, doi: 10.1016/j.ejrn.2013.11.009.
- [17] A. Elsammak, ‘Clinical usefulness of PET-CT in staging, evaluation of treatment response and restaging of thoracic lymphoma’, *Egypt. J. Radiol. Nucl. Med.*, vol. 48, no. 4, pp. 1073–1081, 2017, doi: 10.1016/j.ejrn.2017.04.005.
- [18] R. A. Elshafey, N. Daabes, and S. Galal, ‘FDG-PET/CT in re-staging of patients with non Hodgkin lymphoma and monitory response to therapy in Egypt’, *Egypt. J. Radiol. Nucl. Med.*, vol. 49, no. 4, pp. 1076–1082, 2018, doi: 10.1016/j.ejrn.2018.06.003.
- [19] M. Panebianco *et al.*, ‘Comparison of 18F FDG PET-CT AND CECT in pretreatment staging of adults with Hodgkin’s lymphoma’, *Leuk. Res.*, vol. 76, pp. 48–52, 2019, doi: 10.1016/j.leukres.2018.11.018.
- [20] D. Albano *et al.*, ‘Diagnostic and Clinical Impact of Staging 18F-FDG PET/CT in Mantle-Cell Lymphoma: A Two-Center Experience’, *Clin. Lymphoma, Myeloma Leuk.*, vol. 19, no. 8, pp. e457–e464, 2019, doi: 10.1016/j.clml.2019.04.016.
- [21] ‘NCBI Gene Expression Omnibus’. <https://www.ncbi.nlm.nih.gov/geo>.
- [22] M. D. Christian Steidl, et al., ‘Tumor-Associated Macrophages and Survival in Classic Hodgkin’s Lymphoma’, *N. Engl. J. Med.*, vol. 362, no. 10, pp. 875–885, 2010.
- [23] D. A. Hashimoto, T. M. Ward, and O. R. Meireles, ‘The Role of Artificial Intelligence in Surgery’, *Adv. Surg.*, vol. 54, pp. 89–101, 2020, doi: 10.1016/j.yasu.2020.05.010.
- [24] R. Dias and A. Torkamani, ‘Artificial intelligence in clinical and genomic diagnostics’, *Genome Med.*, vol. 11, pp. 1–12, 2019, doi: 10.1186/s13073-019-0689-8.
- [25] S. Hayou, A. Doucet, and J. Rousseau, ‘On the impact of the activation function on deep neural networks training’, *Arxiv*, 2019.
- [26] ‘MathWorks-Help Center’. <https://www.mathworks.com/help/>.
- [27] C. Doğan, ‘Balina Optimizasyon Algoritması ve Gri Kurt Optimizasyonu Algoritmaları Kullanılarak Yeni Hibrit Optimizasyon Algoritmalarının Geliştirilmesi’, 2019.
- [28] E. G. Dada, S. B. Joseph, D. O. Oyewola, A. A. Fadele, H. Chiroma, and S. M. Abdulhamid, ‘Application of Grey Wolf Optimization Algorithm: Recent Trends, Issues, and Possible Horizons’, *Gazi Univ. J. Sci.*, vol. 35, no. 2, pp. 485–504, 2022, doi: 10.35378/gujs.820885.
- [29] F. Akalın and N. Yumuşak, ‘DNA genom dizilimi üzerinde dijital sinyal işleme teknikleri kullanılarak elde edilen ekson ve intron bölgelerinin EfficientNetB7 mimarisi ile sınıflandırılması’, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Derg.*, vol. 37, no. 3, pp. 1355–1371, 2022, doi: 10.17341/gazimmfd.900987.
- [30] O. Sertel, J. Kong, U. V. Catalyurek, G. Lozanski, J. H. Saltz, and M. N. Gurcan, ‘Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading’, *J. Signal Process. Syst.*, vol. 55, pp. 169–183, 2009, doi: 10.1007/s11265-008-0201-y.
- [31] M. Lippi *et al.*, ‘Texture analysis and multiple-instance learning for the classification of malignant lymphomas’, *Comput. Methods Programs Biomed.*, vol. 185, 2020, doi: 10.1016/j.cmpb.2019.105153.
- [32] B. Ganeshan *et al.*, ‘CT-based texture analysis potentially provides prognostic information complementary to interim fdg-pet for patients with hodgkin’s and aggressive non-hodgkin’s lymphomas’, *Eur. Radiol.*, vol. 27, pp. 1012–1020, 2017, doi: 10.1007/s00330-016-4470-8.
- [33] M. Z. Nascimento, L. Neves, S. C. Duarte, Y. A. S. Duarte, and V. R. Batista, ‘Classification of histological images based on the stationary wavelet transform’, *J. Phys. Conf. Ser.*, vol. 574, 2015, doi: 10.1088/1742-6596/574/1/012133.
- [34] E. S. Alférez, A. Merino, L. E. Mújica, M. Ruiz, L. Bigorra, and J. Rodellar, ‘Digital Blood

- Image Processing and Fuzzy Clustering for Detection and Classification of Atypical Lymphoid B cells', *Jornades Recer. Euetib 2013*, pp. 1–12, 2013.
- [35] O. Sertel, G. Lozanski, A. Shanáah, and M. N. Gurcan, 'Computer-aided detection of centroblasts for follicular lymphoma grading using adaptive likelihood-based cell segmentation', *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2613–2616, 2010, doi: 10.1109/TBME.2010.2055058.
- [36] N. V. Orlov *et al.*, 'Automatic classification of lymphoma images with transform-based global features', *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 4, pp. 1003–1013, 2010, doi: 10.1109/TITB.2010.2050695.

# Automatic Classification of White Blood Cells Using Pre-Trained Deep Models

 Oguzhan Katar<sup>1</sup>,  İlhan Firat Kilincer<sup>2</sup>

<sup>1</sup>Corresponding Author; Firat University, Department of Software Engineering; okatar@firat.edu.tr; 0000-0002-5628-3543; +90 424 607 81 07

<sup>2</sup>Firat University, Department of Informatics; ifkilincer@firat.edu.tr; 0000-0001-8090-4998

Received 31 October 2022; Revised 1 November 2022; Accepted 22 December 2022; Published online 31 December 2022

## Abstract

White blood cells (WBCs), which are a crucial component of the immune system, help our body defend against infections and other diseases. Some diseases may cause our body to produce fewer WBCs than it requires. Therefore, WBCs are of great importance in medical imaging. Artificial intelligence-based computer systems can assist experts in analyzing WBCs. In this study, we proposed an approach for the automatic classification of WBCs into five different classes using a pre-trained model. We trained ResNet-50, VGG-19, and MobileNet-V3-Small pre-trained models with ImageNet weights. For the training, validation, and testing processes of the models, we used a public dataset containing 16,633 images with an uneven class distribution. While the ResNet-50 model achieved an accuracy of 98.79%, the VGG-19 model achieved an accuracy of 98.19%, and the MobileNet-V3-Small model achieved the highest accuracy rate at 98.86%. When examining the predictions of the MobileNet-V3-Small model, we observed that it was not affected by class dominance and was able to correctly classify even the least sampled class images in the dataset. In addition to the high accuracy achieved in the classification of WBCs using the proposed pre-trained deep learning models, we also applied the Grad-CAM method to further understand and interpret the model's predictions.

**Keywords:** white blood cells, classification, pre-trained models, artificial intelligence, Grad-CAM

## 1. Introduction

Blood is a vital fluid that helps to nourish the body, maintain acid-base balance, transport hormones, and maintain salt and water balance. Blood consists of three types of cells: erythrocytes, platelets, and leukocytes [1].

Erythrocytes, the most abundant type of blood cell, contain a substance called hemoglobin, which is responsible for transporting oxygen in the body [2]. Oxygen, inhaled into the lungs through respiration and then entering the blood, can be transported to all body tissues with the help of hemoglobin in erythrocytes. Adequate oxygen access to each cell in the body depends on the sufficient number and function of erythrocytes in the blood. Erythrocytes, which are reddish in color and therefore also referred to as red blood cells, obtain their color from the iron mineral in the structure of hemoglobin [3].

Platelets are cell fragments that are formed by the disintegration of cells called megakaryocytes in the bone marrow tissue located in the center of our bones after they mature and enter the blood [4]. Platelets play a vital role in regulating certain chemical reactions that occur in the blood due to the biochemical substances they contain [5]. However, their primary function is in the case of bleeding due to injury to blood vessels; they help to quickly close and repair the wounded area.

Leukocytes, also known as white blood cells (WBCs), are an important part of the immune system and a group of cells that protect the body against infections [6]. When the body encounters foreign organisms, they reproduce rapidly. The primary function of leukocytes is to identify and eliminate antigens such as bacteria, viruses, fungi, and poisonous toxins that have entered the body in various ways. Leukocytes consist of five different types of WBCs, each with its own specific functions:

- Basophils, which are the least common type of leukocyte in the body, fight infections and parasitic infections. By releasing histamine during allergic reactions, basophils enable the body

to produce an antibody called immunoglobulin E. Additionally, by secreting heparin, they increase the fluidity of the blood [7].

- Eosinophils produce enzymes that destroy parasites that cause inflammatory and allergic reactions in the body [8].
- Monocytes are produced in the bone marrow and then enter the bloodstream. These cells are called monocytes when in the bloodstream, but within a few hours, they leave the circulatory system and enter the tissues. The monocyte cells that reach the tissue are called macrophages. They eliminate microorganisms that cause infections and clean up dead cells [9].
- Lymphocyte cells, which are produced in the bone marrow and lymph tissue, secrete chemicals called lymphokines against foreign organisms in the body, stimulating other immune system cells and allowing them to attack the foreign organism [10].
- Neutrophils are the first precursor cells to reach foreign organisms that cause infections in the body. They release and digest chemical enzymes to combat foreign organisms [11].

Leukemia, anemia, cancer, and various other diseases can be diagnosed through the analysis of WBCs [12]. This analysis is often conducted using a peripheral blood smear, which is a common laboratory method. To obtain a sample, a healthcare provider draws blood from a patient's finger or toe using a sterile needle, and the sample is then examined in a laboratory to create a peripheral blood film [13]. This film is manually analyzed by a specialist to identify signs of disease. However, manual analysis can be time-consuming and laborious for experts. As a result, computer-aided systems have been developed to assist with the classification of WBCs. With the advancement of hardware technology, the use of artificial intelligence (AI) in this field has increased. AI-based systems, also known as decision support systems, are designed to minimize errors caused by human factors and are used in various sectors, including healthcare. For example, decision support systems have been successfully used to detect COVID-19 through chest computed tomography images and to detect brain tumors through brain magnetic resonance imaging without human intervention [14].

Many studies have been carried out for the automatic classification of WBCs by AI-based systems. In the study [15], researchers proposed a system that uses the DenseNet-121 model to classify different types of WBCs. A publicly available dataset including eosinophil, lymphocyte, monocyte, and neutrophil classes was used for model training. The dataset contains 12,444 different samples with a resolution of 320×240px. The normalization process was applied to the dataset samples to speed up model training. The number of dataset samples has been increased with data augmentation techniques such as flipping, rotation, brightness, and zooming. The dataset samples were resized to a resolution of 224×224px. After the pre-processing steps, 20,050 WBCs images were obtained, including synthetic images. The model is trained for 10 epochs with the help of the Adam optimizer. Four different training processes were performed and the batch size value was changed to 8, 16, 32, and 64 in each training. The model, which was trained with 8 batch sizes, achieved 98.84% accuracy, 99.33% precision, 98.85% sensitivity, and 99.61% specificity values during the test phase, and achieved more successful results compared to other models.

In the study [16], researchers proposed an approach that can classify WBCs from microscopic blood images. The researchers used a publicly available dataset of images with different values in resolutions ranging from 350×236px to 2592×1944px. AlexNet, ResNet-101, and GoogleNet models were trained to detect five different classes: basophil, eosinophil, lymphocyte, monocyte, and neutrophil. While the dataset samples are resized to 227×227px resolution for training the AlexNet model, this value is 224×224px for the training of the GoogleNet and ResNet-101 models. To compare the success of the pre-trained models in classifying WBCs, 178 test images were given to the relevant models as input. The AlexNet model achieved better results compared to other models with 96.63% accuracy, 97.85% specificity, and 89.18% sensitivity rates.

In one study [17], researchers designed a deep convolutional neural network (CNN) model to classify microscopic images of WBCs. They proposed a new data augmentation method based on feature concentration to enhance the dataset and address the small number of samples. The training, validation,

and testing processes for the CNN model, which was designed to automatically classify the neutrophil, lymphocyte, monocyte, eosinophil, and basophil classes, were carried out using a special dataset provided by Sichuan Meisheng Biotech Company. This dataset consists of 8600 leukocyte images with a resolution of 1024×768px collected from various individuals. These images were divided into 217×217px pieces, resulting in a total of 11,658 sub-images. 80% of the dataset samples were reserved for training, while the remaining 20% were used for validation. The proposed model achieved an average test accuracy of 97.6% in classifying the five different WBCs.

In another study [18], researchers proposed an approach for classifying WBCs in microscopic images. Samples from a publicly available dataset containing a total of 352 images were augmented using various image augmentation techniques, resulting in 12,444 images. The dataset included samples belonging to the eosinophil, lymphocyte, monocyte, and neutrophil classes. A seven-layer convolutional neural network with an input size of 120×160px was created to automatically classify these samples. To this end, all of the dataset samples were resized to 120×160px. The proposed model was subjected to two different training processes to examine its binary and multiclass classification performance. In binary classification, a mononuclear class was created using eosinophil and neutrophil samples, and a polynuclear class was created using lymphocyte and monocyte samples. The model achieved an accuracy of 96.30% in binary classification and 87.93% accuracy in multiclass classification.

In another study [19], the researchers proposed a system that can simultaneously detect and classify WBCs in an image. This system is based on the F-RCNN and YOLOv4 architectures. The models were trained on samples from the Blood Cell Count Dataset (BCCD), which includes samples of four different WBCs: neutrophils, eosinophils, monocytes, and lymphocytes. The F-RCNN model achieved an accuracy of 96.25% and the YOLOv4 model achieved 95.75% accuracy during the testing phase.

In yet another study [20], the researchers proposed a U-Net-based approach for WBCs segmentation. In the U-Net encoder network, ResNet-50 blocks were integrated instead of the default layers, and squeeze-and-excitation blocks were added to the decoder network. The training and testing stages of the model were conducted using samples from the BCISC and LISC datasets. Using various data augmentation techniques, the number of samples for each dataset was increased to 10,000. The dataset samples were divided into 80% for training, 10% for validation, and 10% for testing. The ResNet-50-based U-Net model was trained for 200 epochs with a batch size of 8 and Adam optimization. It was reported that the model achieved a Dice score of 98.13% and a mean Intersection over Union (mIoU) rate of 96.36% during the testing phase using the BISC dataset samples.

The primary objective of this study is to use deep learning to automatically detect WBCs from microscopic blood images, thereby assisting specialists in the early diagnosis of diseases related to WBCs counts. The main contributions of this study are as follows:

- Demonstrating the effectiveness of existing deep learning models on a new dataset.
- Achieving high performance on a non-uniformly distributed dataset without using data augmentation for WBCs classification.
- Visualizing, using Gradient-weighted Class Activation Mapping (Grad-CAM), which pixel areas the deep learning models focus on during the decision-making phase, thereby providing an explainable structure for pre-trained models.
- Reducing human errors and subjectivity by using deep learning structures to perform these tasks, which are currently carried out by experts visually.

The remainder of this paper is organized as follows: Section 2 presents the proposed method for this study, including the dataset used, pre-trained deep learning models, classification performance measures, and the Grad-CAM algorithm. Section 3 presents the parameters and environments used in the training phase, the numerical values of the model during the training phase, the test phase predictions, and performance values. The discussion and conclusion sections of the study are presented in Section 4 and Section 5, respectively.

## 2. Material and Methods

An approach has been proposed for the deep learning-based automated classification of WBCs from microscopic blood images. The block representation of the proposed method is given in Figure 1.

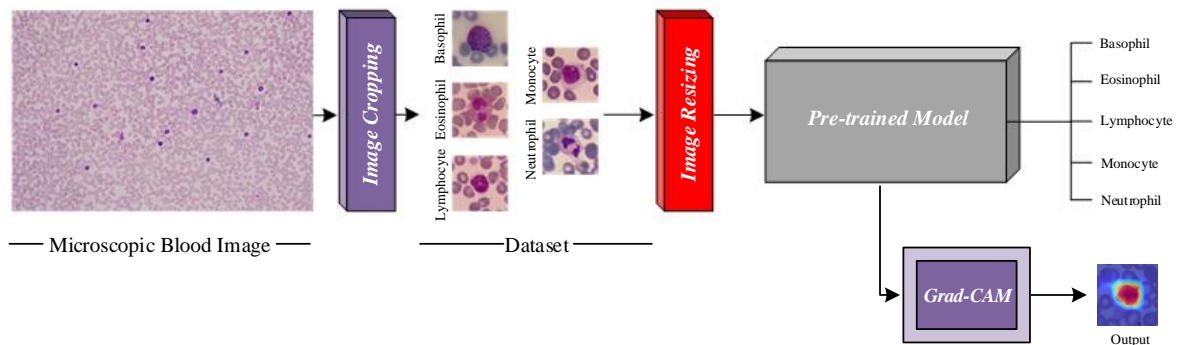


Figure 1 A Block Representation of The Proposed Method

In the proposed method, the image given as input to the deep learning model is classified as a basophil, eosinophil, lymphocyte, monocyte, or neutrophil at the model's output. The choice of dataset and model is critical for achieving high success rates in this classification process. The quality of the dataset directly affects the performance of the deep learning model, and therefore it is important that it is created or verified by experts. This can be a resource and time-intensive process. However, several researchers have created and publicly shared WBCs datasets, as listed in Table 1.

Table 1 Publicly Available WBCs Datasets

Dataset	Basophil	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Total
LISC [21]	54	42	59	55	56	266
BCCD [22]	3	86	33	19	208	349
MISP [23]	0	42	36	33	38	149
ALL-IDB [24]	1	2	60	3	18	84
Zheng et al. [25]	1	22	53	48	176	300
Raabin-WBC [26]	301	1066	3609	795	10,862	16,633

## 2.1 Dataset

In this study, the Raabin-WBC dataset [26] was used for the training, validation, and testing of the models. The Raabin-WBC dataset was created using 72 peripheral blood films collected from Shariati Hospital, which were examined using Olympus Cx18 and Zeiss microscopes. A total of 16,633 WBCs images with a resolution of  $575 \times 575$ px were obtained, and these images were labeled by two experts: 301 were labeled as basophils (Bas), 1066 as eosinophils (Eos), 3609 as lymphocytes (Lym), 795 as monocytes (Mon), and 10,862 as neutrophils (Neu). Samples of each class in the dataset are shown in Figure 2.

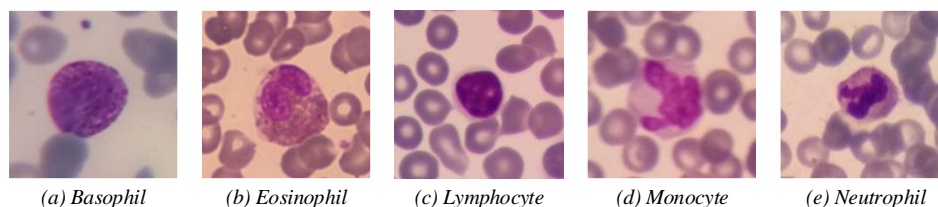


Figure 2 Dataset Samples [26]

Upon examination of the class-based distribution of samples in the Raabin-WBC dataset, it was observed that the Neu class is dominant. Data augmentation methods, which involve creating synthetic images, can be used to balance the distribution of classes. However, in this study, no data augmentation was performed in order to test the performance of pre-trained models under challenging conditions.

## 2.2 Pre-trained Models

Transfer learning is a machine learning technique that involves using the weights of a previously trained model as initial weights in the training phase of CNN. This allows the model, which was previously trained on a task, to be reused for different tasks. Transfer learning is highly effective for achieving good performance with a small amount of data. It is now a widely used method, especially for tasks related to image or natural language processing, as it allows researchers to use pre-trained models that have already learned how to classify images and have learned general features such as edges and shapes. Examples of pre-trained models that are often used as the basis for transfer learning include ResNet [27], VGG [28], and MobileNet [29], which were trained using the ImageNet [30] database. Pre-trained models can be grouped into three categories based on the number of parameters they contain: low (less than 15 M), medium (between 15 M - 70 M), and high (more than 70 M). Information on the pre-trained models is provided in Table 2 [31].

Table 2 Pre-trained Models Used in This Study [31]

Model	Default Input Size	Parameters (Million)	Category
ConvNeXtXLarge	224×224	350.1	High
ConvNeXtLarge	224×224	197.7	High
VGG-19	224×224	143.7	High
VGG-16	224×224	138.4	High
EfficientNetV2L	480×480	119	High
NASNetLarge	331×331	88.9	High
ConvNeXtBase	224×224	88.5	High
EfficientNetB7	600×600	66.7	Medium
ResNet152	224×224	60.4	Medium
ResNet152V2	224×224	60.4	Medium
InceptionResNetV2	299×299	55.9	Medium
EfficientNetV2M	480×480	54.4	Medium
ConvNeXtSmall	224×224	50.2	Medium
ResNet101	224×224	44.7	Medium
ResNet101V2	224×224	44.7	Medium
EfficientNetB6	528×528	43.3	Medium
EfficientNetB5	456×456	30.6	Medium
ConvNeXtTiny	224×224	28.6	Medium
ResNet50	224×224	25.6	Medium
ResNet50V2	224×224	25.6	Medium
InceptionV3	299×299	23.9	Medium
Xception	299×299	22.9	Medium
EfficientNetV2S	384×384	21.6	Medium
DenseNet201	224×224	20.2	Medium
EfficientNetB4	380×380	19.5	Medium
EfficientNetV2B3	300×300	14.5	Low
DenseNet169	224×224	14.3	Low
EfficientNetB3	300×300	12.3	Low
EfficientNetV2B2	260×260	10.2	Low
EfficientNetB2	260×260	9.2	Low
EfficientNetV2B1	240×240	8.2	Low
DenseNet121	224×224	8.1	Low
EfficientNetB1	240×240	7.9	Low
EfficientNetV2B0	224×224	7.2	Low
MobileNet_v3_large	224×224	5.4	Low
NASNetMobile	224×224	5.3	Low
EfficientNetB0	224×224	5.3	Low
MobileNet	224×224	4.3	Low
MobileNetV2	224×224	3.5	Low
MobileNet_v3_small	224×224	2.9	Low

To directly assess the effect of the number of parameters on model performance, three pre-trained models were randomly selected from the categories specifically created for this study: ResNet-50, VGG-19, and MobileNet-V3-Small.

### 2.3 Performance Metrics

Various metrics can be calculated using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) to evaluate the performance of models. In this study, four metrics were used to evaluate the models for each class. These metrics and their corresponding equations are as follows:

- Accuracy is a performance metric that measures the percentage of correct predictions made by a classification model. It is the most widely used performance metric, but it may not fully reflect the performance of a model and can sometimes be misleading. For example, in a dataset where some classes are more represented than others, accuracy may not be a sufficient metric.
- Precision measures the percentage of predictions made by a model that are correct. The main difference between precision and accuracy is that precision only considers correct predictions, while accuracy considers all predictions. Therefore, precision is often a more precise metric and is given greater consideration when evaluating the performance of classification models.
- Sensitivity is a performance metric that measures the success of a classification model. It shows the percentage of data that the model predicted correctly. The main difference with other metrics is that sensitivity only evaluates correct predictions. For example, a model may have low sensitivity even though it has high accuracy. In this case, most of the data that the model predicts correctly are misclassified data, indicating that the model is not performing well.
- The F-1 score is a combination of sensitivity and precision ratios, used to evaluate the performance of a classification model, especially for multi-label data. The advantage of the F-1 score is that it does not rely solely on accuracy values, allowing it to show whether the model has balanced performance for all classes.

$$Precision (P) = \frac{TP}{TP + FP} \quad (1)$$

$$Sensitivity (S) = \frac{TP}{TP + FN} \quad (2)$$

$$Accuracy (Acc) = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

$$F1 \text{ Score } (F1) = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \quad (4)$$

### 2.4 Grad-CAM Algorithm

Grad-CAM is a technique that visualizes the regions of an image that are most important for a CNN to make a prediction. It allows us to understand which parts of an image a CNN is using to make a decision, and can be used to generate heatmaps that highlight these regions [32]. Grad-CAM works by using the gradients of the output of the CNN with respect to the input image to produce a weighted sum of the feature maps in the final convolutional of the network. The resulting heatmap is then overlaid on the input image to show which regions had the greatest influence on CNN's prediction. The architecture of the Grad-CAM algorithm is depicted in Figure 3.



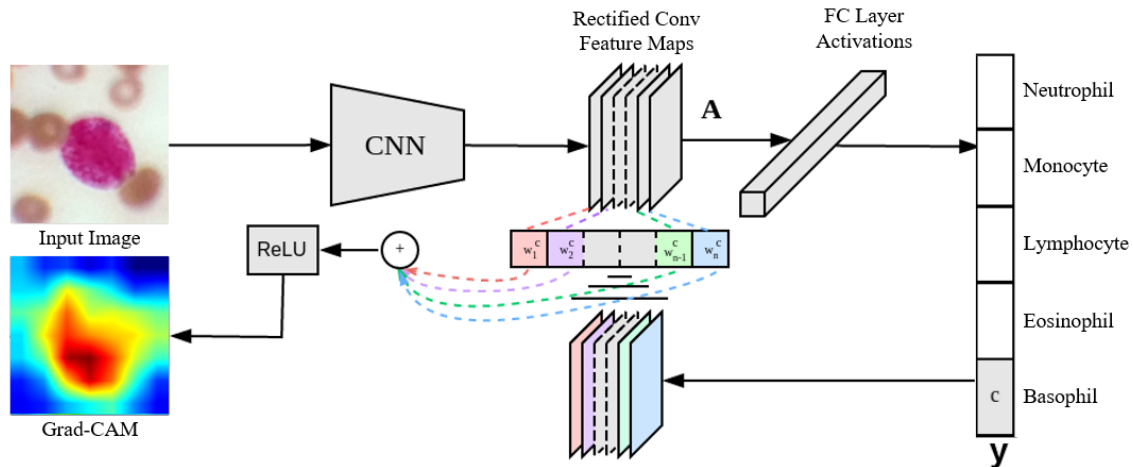


Figure 3 The architecture of the Grad-CAM [32]

The process for creating a Grad-CAM visualization for a pre-trained CNN is as follows:

1. Feed the input image through the CNN to generate a prediction.
2. Compute the gradients of the output of the CNN with respect to the feature maps in the final convolutional layer.
3. Take the weighted sum of the feature maps, using the gradients as weights.
4. Resize the resulting heatmap to the size of the input image.
5. Overlay the heatmap on the input image to highlight the regions that were most important for CNN's prediction.

Grad-CAM is relatively simple to implement and can be used with any CNN, regardless of its architecture. It is also an efficient method, as it only requires a single forward and backward pass through the network to generate the visualization. However, there are some limitations to Grad-CAM. For example, it can only provide visualizations for a single class at a time, and it is sensitive to the specific layer chosen for visualization. Additionally, the visualizations produced by Grad-CAM may not always align perfectly with human intuition, as they are based on the internal representation of the CNN rather than the visual features that a human might use to classify the image.

### 3. Experimental Results

The results of models trained to classify WBCs from microscopic blood images are presented in this section. In addition, an analysis of the experimental findings with performance metrics is shown in the following sections.

#### 3.1 Experimental Setups

The default input size of the ResNet-50, VGG-19, and MobileNet-V3-Small models used in this study is  $224 \times 224$ px, so all of the dataset samples were resized to this value. Before training the model, 70% of the resized dataset samples were randomly divided for use in the training, 20% for validation, and 10% for testing. The visual representation of these processes is shown in Figure 4.

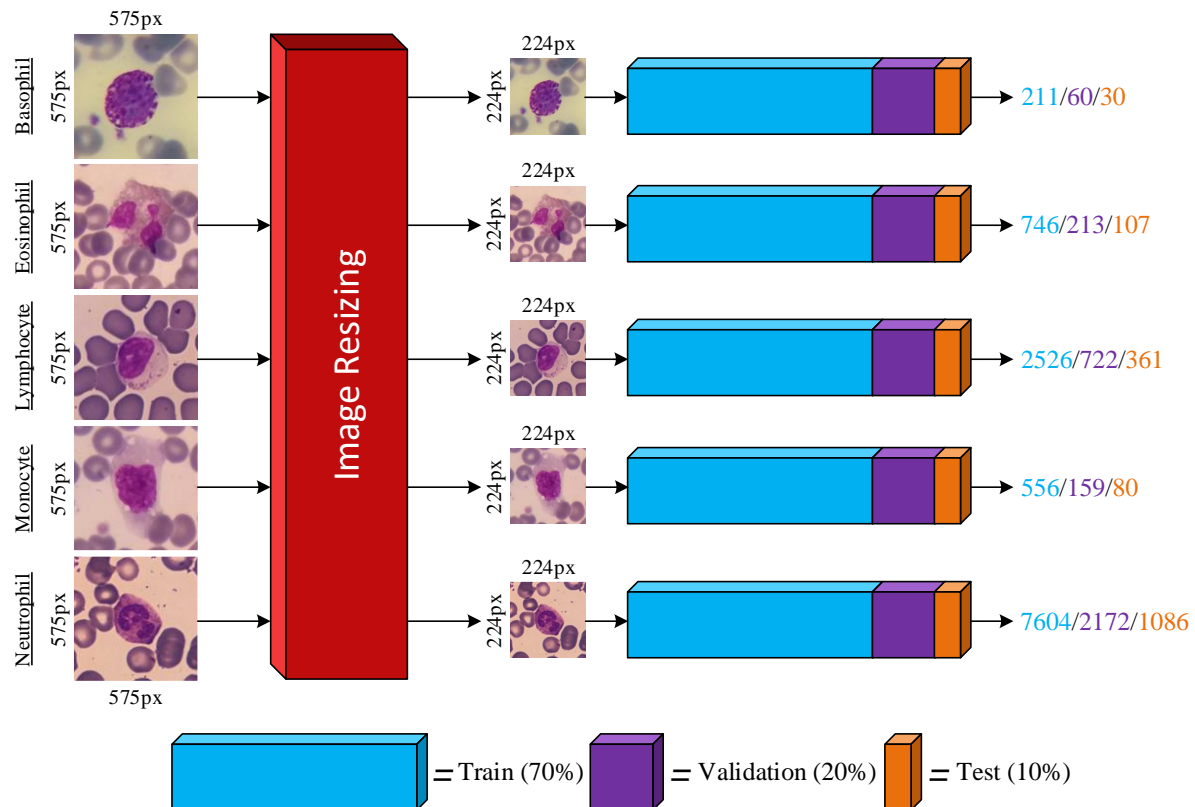
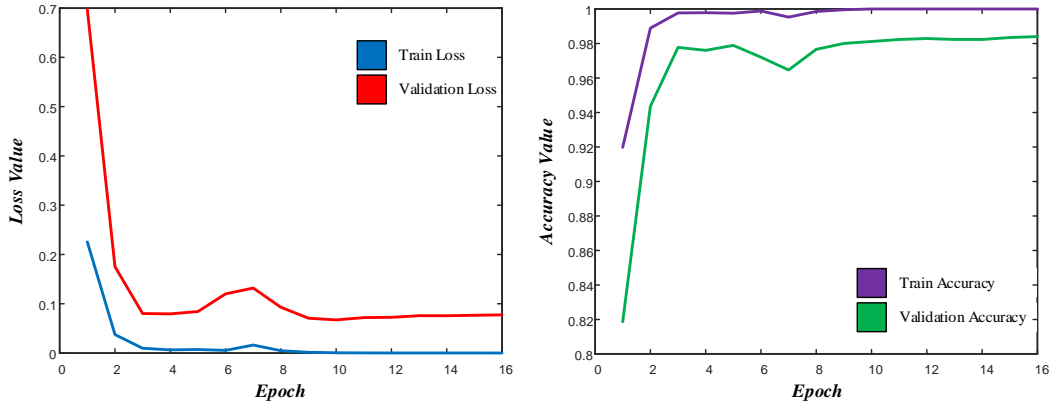


Figure 4 Image Resizing and Splitting Method

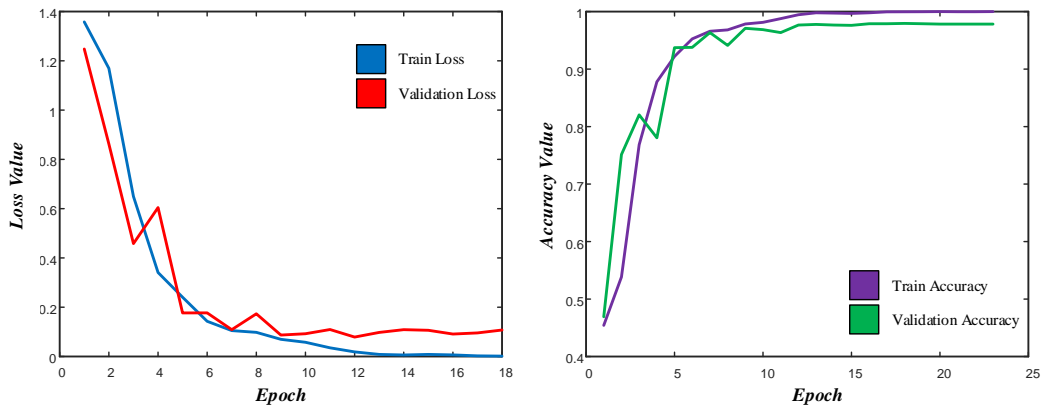
The pre-trained ResNet-50, VGG-19, and MobileNet-V3-Small models were included in the training using the Keras library. Since the models will only make predictions for five different classes, the Dense layers were revised and the softmax activation function was used. The models were compiled with an Adam optimizer and a learning rate of 0.0001. ImageNet weights were used instead of random initial weights for the training of the models. The models were trained with a constant batch size of 64, and training was carried out for a maximum of 50 epochs using the early stopping function. If the monitored validation accuracy value does not improve for five consecutive epochs, the early stopping function terminates the training phase and the weights of the epoch with the highest validation accuracy value are recorded in the '.h5' format. All of these processes were performed in the Google Colab environment.

### 3.2 Results

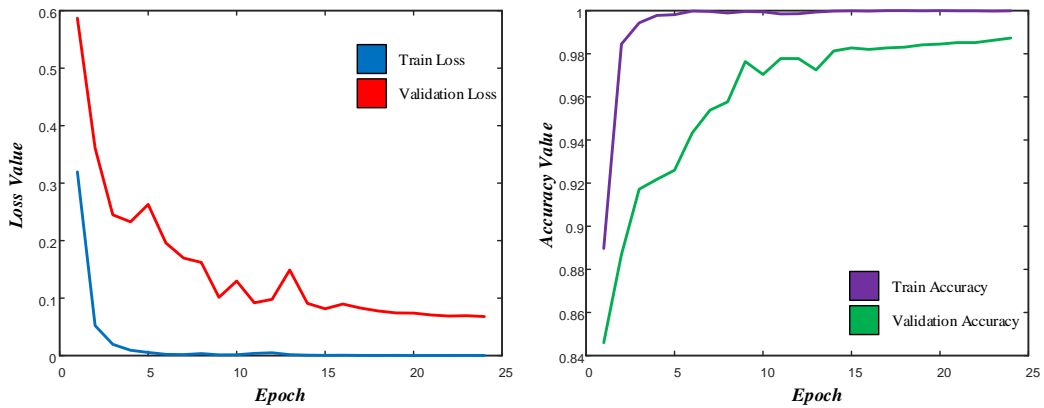
Three different deep-learning models were trained with the same parameters. The time required to complete the model training processes is directly proportional to the number of parameters and layers they have. The training stages of the models were carried out using the early stopping function, and the weights of the epoch that achieved the highest validation accuracy were recorded. ResNet-50 reached the highest validation accuracy after 16 epochs, VGG-19 reached the highest validation accuracy after 23 epochs, and MobileNet-V3-Small reached the highest validation accuracy after 24 epochs. The loss and accuracy graphs for the models during the training and validation phases are shown in Figure 5.



(a) ResNet-50



(b) VGG-19



(c) MobileNet-V3-Small

Figure 5 Loss and Accuracy Graphs

When the validation accuracy values were examined for the three models that completed training, it was observed that a rate of more than 95% could be achieved in less than 25 epochs. This is due in part to the fact that the models were trained using ImageNet weights instead of starting with random weights. Even though the models were trained with a dataset that is not evenly balanced, the lack of overfitting indicates the success of the pre-trained models. Performance metrics were used to compare the classification performance of the three different models trained to classify five different WBCs from microscopic blood images. For this, images that were not included in the training and validation phases but were reserved solely for use in the testing phase were given as input to each model. The confusion matrices generated by the predictions of the models for these inputs are shown in Figure 6.

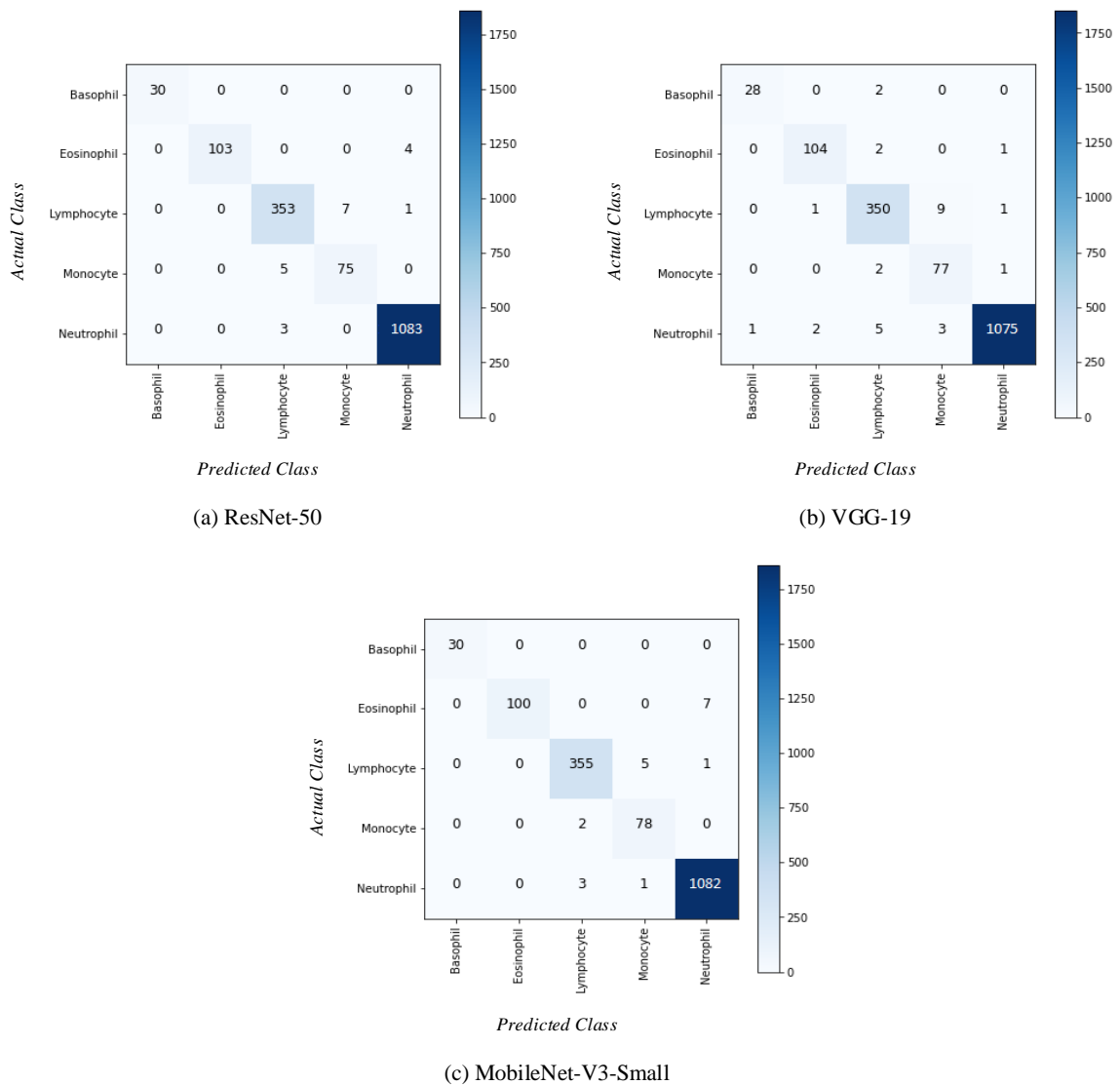


Figure 6 Confusion Matrices for Each Model

To evaluate the Grad-CAM outputs of the deep learning model, it is necessary to first assess the performance of the model during the training phase. This helps to understand the accuracy of the model's predictions and assess the reliability of the model. When the predictions are analyzed, it is apparent that the models have learned to classify WBCs. In Table 3, the performance metric values achieved by the relevant models during the testing phase are provided.

Table 3 The Results of The Pre-trained Models

Model	Basophil			Eosinophil			Lymphocyte			Monocyte			Neutrophil		
	P (%)	S (%)	F1 (%)	P (%)	S (%)	F1 (%)	P (%)	S (%)	F1 (%)	P (%)	S (%)	F1 (%)	P (%)	S (%)	F1 (%)
ResNet-50	100	100	100	100	96.26	98.09	97.78	97.78	97.78	91.46	93.75	92.59	99.54	99.72	99.62
VGG-19	96.55	93.33	94.91	97.19	97.19	97.19	96.95	96.95	96.95	86.51	96.25	91.12	99.72	98.98	99.34
MobileNet-V3-Small	100	100	100	100	93.45	96.61	98.61	98.33	98.46	92.85	97.50	95.11	99.26	99.63	99.44

Following the training phase, the model should be evaluated using the test data. The resulting outputs should be carefully analyzed to interpret how the Grad-CAM outputs describe the images and identify the features that the model considers important. Figure 7 presents the Grad-CAM outputs for a selection of randomly chosen images from the test set.

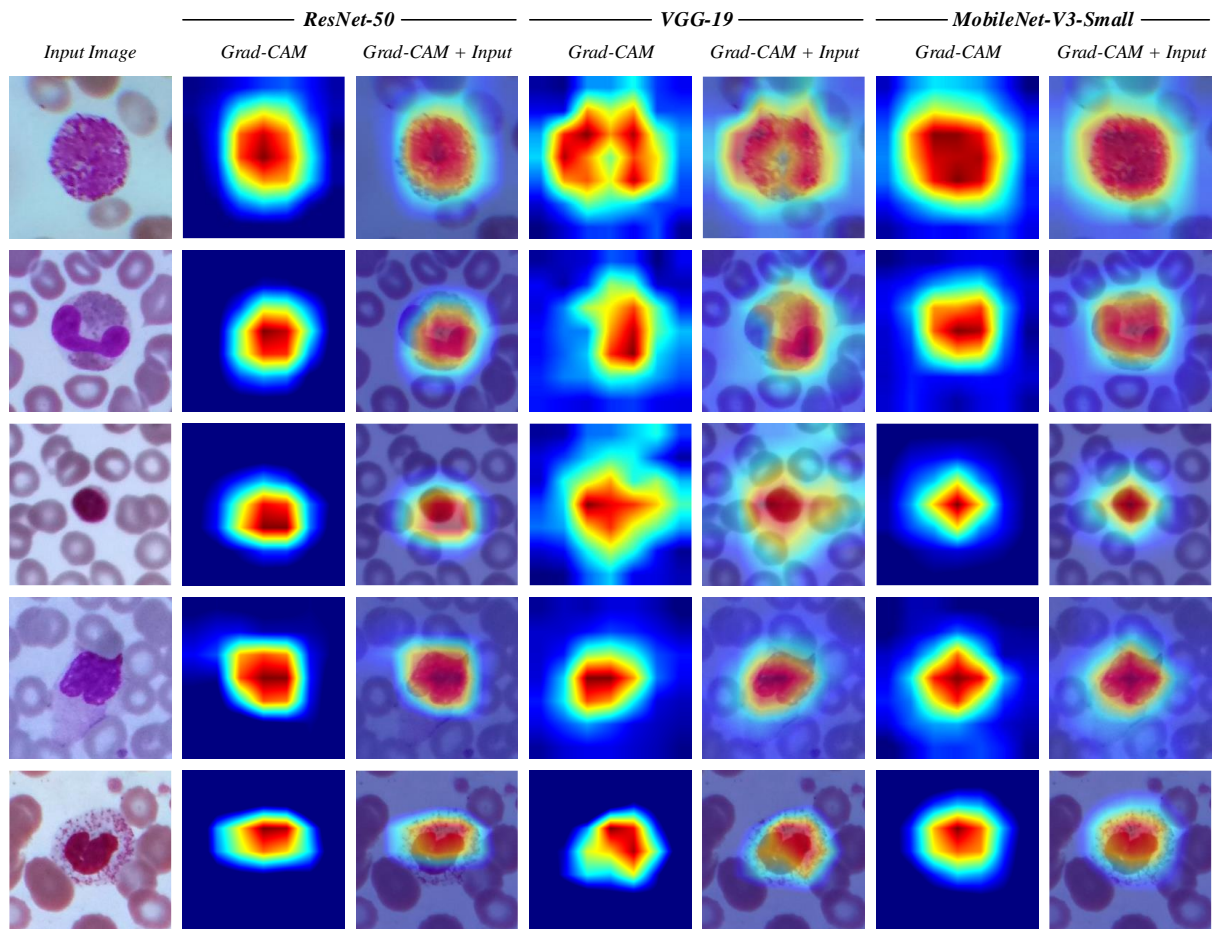


Figure 7 The Grad-CAM outputs

Upon examination of the Grad-CAM outputs, it was observed that all three models effectively identified the relevant features in the images with a high degree of accuracy. As the model performance is consistent with expectations, it is not necessary to further adjust the hyperparameters or layer configurations of the models.

#### 4. Discussion

The detection of WBCs using microscopic blood images is a topic of active research. Table 4 presents a selection of studies on this subject that have been curated by hand. Yildirim and Cinar [33] employed AlexNet, ResNet-50, DenseNet-201, and GoogleNet architectures on a dataset comprising 9,663 images. For each model, three different training stages were conducted using original data, data filtered with a Gaussian filter, and data filtered with a median filter. The highest accuracy rate of 83.44% was achieved by the DenseNet-201 model trained with Gaussian-filtered data. Ekiz et al. [34] classified 12,442 WBCs images using both a CNN model and a Con-SVM model, with the Con-SVM model found to be more accurate, achieving an accuracy rate of 85.96%, compared to the CNN model's accuracy rate of 83.91%. Sharma et al. [15] implemented a deep learning model based on the DenseNet121 architecture for the classification of various types of WBCs. The model was optimized with normalization and data augmentation and achieved an accuracy of 98.84%. Girdhar et al. [35] proposed a method that demonstrated the ability to accurately classify WBCs types in a shorter number of epochs/time compared to other approaches. The performance of the proposed method was evaluated using the Kaggle dataset, resulting in an overall accuracy of 98.55%. Nahzat et al. [36] aimed to develop a CNN-based model for the classification of WBCs. They used images of WBCs from the Kaggle dataset to train and evaluate their proposed model, testing it with various optimizers to determine the best performance. They also compared the performance of their model with four pre-trained CNN models

(MobileNetV2, DenseNet121, InceptionV3, and ResNet50) and found that the proposed model, despite having the lowest number of trainable parameters and training time, outperformed the others with an accuracy of 99.5%. Karakuş and Özbay [37] used CNN models and combined them with three different machine learning classifiers. They applied contrast-limited adaptive histogram equalization (CLAHE) and Gaussian filters to images from the Kaggle dataset, which were then reclassified using the three CNN networks. The results showed that the classification performance was higher when the images were preprocessed with these filters compared to the original data. Jung et al. [38] proposed W-Net, a CNN-based method for the classification of WBCs. To evaluate W-Net, they used a large-scale dataset of 6,562 real images of the five WBCs types, obtained from The Catholic University of Korea. The results showed that W-Net achieved an average accuracy of 97%. Wang et al. [39] proposed a deep CNN called WBC-AMNet for automatically classifying WBCs subtypes based on a focused attention mechanism. This method uses feature fusion strategies, combining Squeeze-and-Excitation and Gather-Excite modules, to obtain more localized attention from the CNN. The WBC-AMNet achieved an overall accuracy of 98.39. They also used Grad-CAM to visualize the attention heatmaps of different feature maps. Roy and Ameer [40] applied a semantic segmentation technique using a deep learning network to accurately segment WBCs from microscopic blood images. The proposed model employed the DeepLabv3+ architecture with a ResNet-50 network as the feature extractor. The model was evaluated on three different public datasets containing five categories of WBCs, using 10-fold cross-validation to assess its effectiveness. The average segmentation accuracy achieved by the proposed model was 96.1% IoU. Wu et al. [20] proposed a WBC image segmentation network based on U-Net that combines residual networks. The encoder structure of the network uses ResNet50 residual blocks as the main unit. The proposed model achieved 96.36% mIoU.

Table 4 Comparison of Our Work With Some State-of-the-art Studies

Study	Number of Class	Number of Images	Model	Explainability	Task	Performance
Yildirim and Cinar [33]	4 (Eos, Lym, Mon, Neu)	9,663	DenseNet-201	Black-box	Classification	Acc=83.44%
Ekiz et al. [34]	4 (Eos, Lym, Mon, Neu)	12,442	Con-SVM	Black-box	Classification	Acc=85.96%
Sharma et al. [15]	4 (Eos, Lym, Mon, Neu)	12,444	DenseNet-121	Black-box	Classification	Acc=98.84%
Girdhar et al. [35]	4 (Eos, Lym, Mon, Neu)	12,444	CNN	Black-box	Classification	Acc=98.55%
Nahzat et al. [36]	4 (Eos, Lym, Mon, Neu)	12,444	Hybrid CNN	Black-box	Classification	Acc=99.50%
Karakuş and Özbay [37]	4 (Eos, Lym, Mon, Neu)	12,444	CNN	Black-box	Classification	Acc=97.10%
Jung et al. [38]	5 (Bas, Eos, Lym, Mon, Neu)	6,562	W-Net	Black-box	Classification	Acc=97.00%
Wang et al. [39]	4 (Eos, Lym, Mon, Neu)	16,873	WBC-AMNet	Grad-CAM	Classification	Acc=98.39%
Roy and Ameer [40]	5 (Bas, Eos, Lym, Mon, Neu)	642	DeepLabv3+	Black-box	Segmentation	mIoU=96.10%
Wu et al. [20]	5 (Bas, Eos, Lym, Mon, Neu)	516	U-Net	Black-box	Segmentation	mIoU=96.36%
The proposed our study	5 (Bas, Eos, Lym, Mon, Neu)	16,633	MobileNet-V3-Small	Grad-CAM	Classification	Acc=98.86%

In this study, we employed a pre-trained MobileNet-V3-Small model for automated WBCs classification. Our results demonstrated a high accuracy of 98.86%, which is higher than the accuracy reported in most other studies. This suggests that the model in our study was able to accurately classify the images into the appropriate categories. Our study included a larger number of classes (5) compared to many other studies (which often have only 4 classes). This increased complexity made the task more challenging and required a more sophisticated model. Our dataset was also relatively large, with 16,663 images, which may have contributed to the robustness and generalizability of our model. We also employed Grad-CAM as an explainability method to provide insights into the model's decision-making process and identify any potential biases or weaknesses.

It is worth noting that some other studies have focused on image segmentation, a task distinct from classification. Image segmentation involves predicting a pixel-level mask for each class in the image, while classification simply involves predicting a single class label for the entire image. In this study, we employed the MobileNet-V3-Small model architecture, which may not be optimal for all tasks and datasets. Alternative model architectures may yield better performance in certain cases. Some other studies have utilized models with more layers and a greater number of parameters (e.g. DenseNet-201, DenseNet-121), which may improve performance but also require more computational resources and may be more prone to overfitting.

The limitations of this study are as follows:

- The dataset consists of only 16,633 images, which may not be sufficient to fully capture the variability and complexity of the WBCs being analyzed.
- Our study only evaluated the performance of three pre-trained models (ResNet-50, VGG-19, and MobileNet-V3-Small) on the WBCs classification task.
- The durability of models against changes due to variations in lighting, background, or other factors that may affect the appearance of WBCs in images has not been validated.
- As k-fold cross-validation was not employed, the model was only evaluated on a single split of the data.

In future research, it would be beneficial to augment the dataset with a larger number of images that have a more balanced distribution of classes. This would likely lead to more robust and accurate classifications. It would also be useful to evaluate the model on a range of different datasets to assess its generalizability and performance on diverse types of images. While the models in this study demonstrated high accuracy rates, there is always a potential for further improvement. Additional research could be conducted to optimize the models and enhance their performance. While the models in this study demonstrated high accuracy in classifying WBCs, it would be valuable to assess their performance in real-world settings. This might involve testing the models on images from actual medical cases or incorporating the models into existing medical imaging systems for use by healthcare professionals.

## 5. Conclusion

In recent years, advances in hardware technology have enabled the use of machine learning techniques in the field of healthcare, specifically in the automatic classification of WBCs using microscopic blood images. Accurate identification of WBCs is crucial for medical diagnosis and research. This study proposes a deep learning-based approach for the automatic classification of WBCs using microscopic blood images and investigates its effectiveness through experiments on a dataset of 16,633 different WBCs images. Several popular pre-trained models, including MobileNet-V3-Small, were employed for the deep learning models. The MobileNet-V3-Small model achieved the highest accuracy rate of 98.86%. To understand how the model was making its predictions, we employed a visualization technique called Grad-CAM to identify the pixel areas that the model was focusing on. The findings of this study suggest that deep learning may be a useful tool for the automated identification of WBCs in medical diagnosis and research. However, further research is needed to fully evaluate the robustness

and generalizability of these results, as well as to explore the potential for using deep learning in other aspects of medical diagnosis and treatment.

## References

- [1] C. J. Walsh and C. A. Luer, "Elasmobranch hematology: identification of cell types and practical applications," *The Elasmobranch Husbandry Manual: Captive Care of Sharks, Rays and their Relatives*, pp. 307-323, 2004.
- [2] A. Glenn and C. E. Armstrong, "Physiology of red and white blood cells," *Anaesthesia & Intensive Care Medicine*, vol. 20, no. 3, pp. 180-174, 2019.
- [3] R. Van Zwieten, A. J. Verhoeven and D. Roos, "Inborn defects in the antioxidant systems of human red blood cells," *Free Radical Biology and Medicine*, vol. 67, pp. 377-386, 2014.
- [4] I. Andia and N. Maffulli, "Platelet-rich plasma for managing pain and inflammation in osteoarthritis," *Nature Reviews Rheumatology*, vol. 9, no. 12, pp. 721-730, 2013.
- [5] B. Olas and B. Wachowicz, "Role of reactive nitrogen species in blood platelet functions," *Platelets*, vol. 18, no. 8, pp. 555-565, 2007.
- [6] M. Habibzadeh, M. Jannesari, Z. Rezaei, H. Baharvand and M. Totonchi, "Automatic white blood cell classification using pre-trained deep learning models: Resnet and inception," *Tenth international conference on machine vision*, vol. 10696, pp. 274-281, 2018.
- [7] A. L. Gillen and J. Conrad, "Our Impressive Immune System: More Than a Defense", *Faculty Publications and Presentations*, 135, 2014.
- [8] H. Kita and B. S. Bochner, "Biology of eosinophils", *Middleton's allergy principles and practice*, vol. 8 pp. 265-279, 2013.
- [9] F. Ginhoux and S. Jung, "Monocytes and macrophages: developmental pathways and tissue homeostasis", *Nature Reviews Immunology*, vol. 14, no. 6, pp. 392-404, 2014.
- [10] Y. Ueda, M. Kondo and G. Kelsoe, "Inflammation and the reciprocal production of granulocytes and lymphocytes in bone marrow", *The Journal of experimental medicine*, vol. 201, no. 11, pp. 1771-1780, 2005.
- [11] E. Bronze-da-Rocha and A. Santos-Silva, "Neutrophil elastase inhibitors and chronic kidney disease", *International journal of biological sciences*, vol. 14, no.10, pp. 1343-1360, 2018.
- [12] A. Shahzad, M. Raza, J. H. Shah, M. Sharif and R. S. Nayak, "Categorizing white blood cells by utilizing deep features of proposed 4B-AdditionNet-based CNN network with ant colony optimization," *Complex & Intelligent Systems*, vol. 8, no. 4, pp. 3143-3159, 2022.
- [13] L. B. Maedel and K. Doig, "Examination of the peripheral blood film and correlation with the complete blood count," *Hematology: clinical principles and applications*, pp. 192-209, 2013.
- [14] O. Katar and E. Duman, "Deep Learning Based Covid-19 Detection With A Novel CT Images Dataset: EFSC-19," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 29, pp. 150-155, 2021.
- [15] S. Sharma, S. Gupta, D. Gupta, S. Juneja, P. Gupta, G. Dhiman and S. Kautish, "Deep learning model for the automatic classification of white blood cells," *Computational Intelligence and Neuroscience*, 2022.
- [16] M. J. Macawile, V. V. Quiñones, A. Ballado, J. D. Cruz and M. V. Caya, "White blood cell classification and counting using convolutional neural network," *3rd International conference on control and robotics engineering (ICCRE)*, pp. 259-263, 2018.
- [17] Y. Wang and Y. Cao, "Human peripheral blood leukocyte classification method based on convolutional neural network and data augmentation," *Medical physics*, vol. 47, no. 1, pp. 142-151, 2020.
- [18] M. Sharma, A. Bhavne and R. R. Janghel, "White blood cell classification using convolutional neural network," *Soft Computing and Signal Processing*, pp. 135-143, 2019.
- [19] J. Yao et al., "High-efficiency classification of white blood cells based on object detection", *Journal of Healthcare Engineering*, 2021.
- [20] J. Wu et al., "WBC Image Segmentation Based on Residual Networks and Attentional Mechanisms," *Computational Intelligence and Neuroscience*, 2022.
- [21] S. H. Rezatofighi and H. Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," *Computerized Medical Imaging and Graphics*, vol. 35, no. 4, pp. 333-



- 343, 2011.
- [22] M. Mohamed, B. Far and A. Guaily, "An efficient technique for white blood cells nuclei automatic segmentation," *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 220-225, 2012.
- [23] O. Sarrafzadeh, H. Rabbani, A. Talebi and H. U. Banaem, "Selection of the best features for leukocytes classification in blood smear microscopic images," *Medical Imaging 2014: Digital Pathology*, vol. 9041, pp. 159-166, 2014.
- [24] R. D. Labati, V. Piuri and F. Scotti, "All-IDB: The acute lymphoblastic leukemia image database for image processing," *18th IEEE international conference on image processing*, pp. 2045-2048, 2011.
- [25] X. Zheng, Y. Wang, G. Wang and J. Liu, "Fast and robust segmentation of white blood cell images by self-supervised learning," *Micron*, vol. 107, pp. 55-71, 2018.
- [26] Z. M. Kouzehkanan et al., "A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm," *Scientific reports*, vol. 12, no. 1, pp. 1-14, 2022.
- [27] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016.
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [29] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [30] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE conference on computer vision and pattern recognition*, pp. 248-255, 2009.
- [31] [Online]. Available: <https://keras.io/api/applications/> [Accessed in December 2022].
- [32] R. R. Selvaraju et al., "Grad-cam: Visual explanations from deep networks via gradient-based localization," *IEEE international conference on computer vision*, pp. 618-626, 2017.
- [33] M. Yildirim and A. Çınar, "Classification of White Blood Cells by Deep Learning Methods for Diagnosing Disease," *Rev. d'Intelligence Artif.*, vol. 33, no. 5, pp. 335-340, 2019.
- [34] A. Ekiz, K. Kaplan and H. M. Ertunç, "Classification of white blood cells using CNN and Con-SVM," *29th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4, 2021.
- [35] A. Girdhar, H. Kapur and V. Kumar, "Classification of White blood cell using Convolution Neural Network," *Biomedical Signal Processing and Control*, vol. 71, 2022.
- [36] S. Nahzat, F. Bozkurt and M. Yağanoğlu, "White Blood Cell Classification Using Convolutional Neural Network. Journal Of Science," *Technology And Engineering Research*, vol. 3, no. 1, pp. 32-41, 2022.
- [37] M. Ö. Karakuş and E. Özbay, "Lökosit Tespiti İçin Beyaz Kan Hücrelerinin Esa Kullanılarak Sınıflandırılması," *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 9, no. 17, pp. 333-344, 2022.
- [38] C. Jung, M. Abuhamad, J. Alikhanov, A. Mohaisen, K. Han and D. Nyang, "W-net: a CNN-based architecture for white blood cells image classification," *arXiv preprint arXiv:1910.01091*, 2019.
- [39] Z. Wang, J. Xiao, J. Li, H. Li and L. Wang, "WBC-AMNet: Automatic classification of WBC images using deep feature fusion network based on focalized attention mechanism," *Plos one*, vol. 17, no. 1, 2022.
- [40] R. M. Roy and P. M. Ameer, "Segmentation of leukocyte by semantic segmentation model: A deep learning approach," *Biomedical Signal Processing and Control*, vol. 65, 102385, 2021.