

# Journal Of Computer And Information Sciences



SAKARYA UNIVERSITY

e-ISSN 2636-8129

VOLUME 5

ISSUE 1

APRIL 2022

*Terrorism in Cyberspace: A Critical Review of Dark Web Studies under the Terrorism Landscape*

*Augmented Artificial Neural Network Model for the COVID-19 Mortality Prediction: Preliminary Analysis of Vaccination in Turkey*

*Performance Analysis of Chaotic Neural Network and Chaotic Cat Map Based Image Encryption*

*Calculation of Driving Parameters for GOA4 Signaling System using Machine Learning Methods*

*A Comparative Study On COVID-19 Prediction Using Deep Learning And Machine Learning Algorithms: A Case Study On Performance Analysis*



www.saucis.sakarya.edu.tr

*Comparison of Different Machine Learning Algorithms to Predict the Diagnostic Accuracy Parameters of Celiac Serological Tests*

*A Novel Hybrid Binary Farmland Fertility Algorithm with Naïve Bayes for Diagnosis of Heart Disease*

*Classification of Imbalanced Offensive Dataset – Sentence Generation for Minority Class with LSTM*

*Detection of Heart Rate Variability from Photoplethysmography (PPG) Signals Obtained by Raspberry Pi Microcomputer*

*Using a Convolutional Neural Network as Feature Extractor for Different Machine Learning Classifiers to Diagnose Pneumonia*





# SAUCIS

**Sakarya University Journal of Computer and Information  
Sciences Volume: 5 – Issue No: 1 (April 2022)**  
<http://saucis.sakarya.edu.tr/issue/69696>

## 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, dduural@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, Fırat 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



# SAUUCIS

## 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: 1 (April 2022)  
<http://saucis.sakarya.edu.tr/issue/69696>

## Contents

Author(s), Paper Title	Pages
<i>Eda Sonmez, Keziban Seckin Codal</i> Terrorism in Cyberspace: A Critical Review of Dark Web Studies under the Terrorism Landscape	1-18
<i>Sena Kır, Elif Elçin Günay</i> Augmented Artificial Neural Network Model for the COVID-19 Mortality Prediction: Preliminary Analysis of Vaccination in Turkey	22-36
<i>Sefa Tunçer, Cihan Karakuzu</i> Performance Analysis of Chaotic Neural Network and Chaotic Cat Map Based Image Encryption	37-47
<i>Enes Ayan</i> Using a Convolutional Neural Network as Feature Extractor for Different Machine Learning Classifiers to Diagnose Pneumonia	48-61
<i>Mehmet Taciddin Akçay, Abdurrahim Akgündoğdu</i> Calculation of Driving Parameters for GOA4 Signaling System using Machine Learning Methods	62-70
<i>Hilal Arslan, Orhan Er</i> A Comparative Study On COVID-19 Prediction Using Deep Learning And Machine Learning Algorithms: A Case Study On Performance Analysis	71-83
<i>Özgül Özer, Nazlı Arda</i> Comparison of Different Machine Learning Algorithms to Predict the <i>Diagnostic Accuracy</i> Parameters of Celiac Serological <i>Tests</i>	84-89
<i>Vafa Radpour, Farhad Soleimani Gharehchopogh</i> A Novel Hybrid Binary Farmland Fertility Algorithm with Naïve Bayes for Diagnosis of Heart Disease	90-103
<i>Ziyet Pamuk, Ceren Kaya</i> Detection of Heart Rate Variability from Photoplethysmography (PPG) Signals Obtained by Raspberry Pi Microcomputer	104-120
<i>Ekin Ekinçi</i> Classification of Imbalanced Offensive Dataset – Sentence Generation for Minority Class with LSTM	121-133

# Terrorism in Cyberspace: A Critical Review of Dark Web Studies under the Terrorism Landscape

 Eda Sonmez<sup>1</sup>,  Keziban Seckin Codal<sup>2</sup>

<sup>1</sup>Corresponding Author; Ankara Yıldırım Beyazıt University; Department of Management Information Systems; edasonmez@uludag.edu.tr; 03123230134

<sup>2</sup> Ankara Yıldırım Beyazıt University; Department of Management Information Systems; kseckin@ybu.edu.tr;

Received 10 June 2021; Revised 25 August 2021; Accepted 8 November 2021; Published online 30 April 2022

## Abstract

Crime, terrorism, and other illegal activities are increasingly taking place in cyberspace. Crime in the dark web is one of the most critical challenges confronting governments around the world. Dark web makes it difficult to detect criminals and track activities, as it provides anonymity due to special tools such as TOR. Therefore, it has evolved into a platform that includes many illegal activities such as pornography, weapon trafficking, drug trafficking, fake documents, and more specially terrorism as in the context of this paper. Dark web studies are critical for designing successful counter-terrorism strategies. The aim of this research is to conduct a critical analysis of the literature and to demonstrate research efforts in dark web studies related to terrorism. According to result of the study, the scientific studies related to terrorism activities have been minimally conducted and the scientific methods used in detecting and combating them in dark web should be varied. Advanced artificial intelligence, image processing and classification by using machine learning, natural language processing methods, hash value analysis, and sock puppet techniques can be used to detect and predict terrorist incidents on the dark web.

**Keywords:** terrorism, cyberspace, dark web, deep web, anonymity

## 1. Introduction

In the twenty-first century, governments face a new security threats that has evolved as a result of globalization and the ever-accelerating pace of technological breakthroughs [1]. As technological advances intensify, abuse has also increased especially in cyberspace and traditional terrorist groups are increasingly expanding their activities into the cyberspace [2], [3]. The combination of cyberspace and terrorism has also revealed the concept of cyber terrorism. Following the 9/11 attacks, cyber terrorism became a prominent topic in security and terrorism discussions [4].

The increase of cyber terrorism reflects the Internet's rising popularity, the substantial number of malicious activities, and the development of sophisticated and high-tech dependent tools. Understanding the characteristics of Internet is the important step in dealing with the cyber terrorism. The internet has three different layers called surface web, deep web and dark web [5].

The Surface Web, called as a visible, indexable Web is a tier of the internet that is readily available to the general public [6]. Presently, there are roughly 4.66 billion Internet users and 5,54 billion indexed pages around the world [7], [8]. Since the late 1990s, terrorists have been active in the surface web and use various social media platforms such as YouTube, Twitter to communicate, recruit and propagate [9]. The surface web poses danger for terrorists due to easily followed; hence, terrorists have shifted their illegal activities to the deeper layer of the Internet, deep and dark web.

The deep web is a part of the internet that can't be reached by conventional search engines; it can only be searched through specific keywords and queries, and it is protected by safety precautions involving membership records, login IDs, passwords, and codes [10]. According to Bergman (2001), the most cited

researcher on the scale of the deep web, the deep web is 4,000-5,000 times larger than the surface web. The term "Dark Web" refers to a part of the deep web that is purposefully concealed and accessible only through specific software [11]. The best known special tool for accessing and surfing on the dark web is Onion Router (TOR) [12]. TOR is a free tool that uses the onion routing technique to provide anonymity [13]. It was originally developed to protect the classified data of the US Naval Research Laboratory, but it has since grown into an encryption tool for hiding users' activities and IP addresses [14]. Criminals come together on dark web platforms to perform illegal activities [15]. Pornography, gun trafficking, illegal drug trade, fake documents and counterfeit currency, and terrorism cover 57% of the dark web crime [16].

Terrorist activities can be carried out directly over the dark web, at the same time, the other dark web crimes can also aid in the spread of terrorism. Terrorist groups also widely conduct illegal actions such as weapons trade, drug trafficking, human smuggling, money laundering, to provide resources and finance for their organizations. Thus, other dark web crimes also become an element of terrorism [17]. Dark web terrorism is a worldwide problem that requires multilateral effort at the national, regional and global levels [18]. The researchers and law enforcement has tended to focus upon the variety of illicit activities in the dark web to examine and determine the necessary precautions. It is vital to consider existing research specifically related to the dark web terrorism in order to address how cyber environments are used for terrorist acts. The aim of this paper is to critically review current studies as well as to summarize research efforts in the dark web in the scope of terrorism.

The rest of the paper structured as follows: The next section discusses the dark web in pertaining to terrorism. Evaluating and justifying the methodological choices are explained in Section 3. Results and discussions are presented in section 4 and finally, there is the conclusion part in section 5.

## **2. Terrorism and the Dark Web**

Terrorism is typically described as violence that is designed to cause fear, is carried out for political, religious, or ideological purposes, intentionally ignore the protection of civilians. However, it does not have a generally accepted legal definition in the international area [19].

Terrorism damages stability and peace, creates violence in society and directly endangers the lives of people [20]. Due to terrorist attacks, the millions of innocent people are harmed, animals are also killed and millions of things are destroyed. The total number of deaths caused by terrorism between 2006 and 2019 is presented in Figure 1. The highest number of deaths was observed in 2014 and 2018. Additionally, it is noteworthy that the number of deaths has fluctuated in recent years.



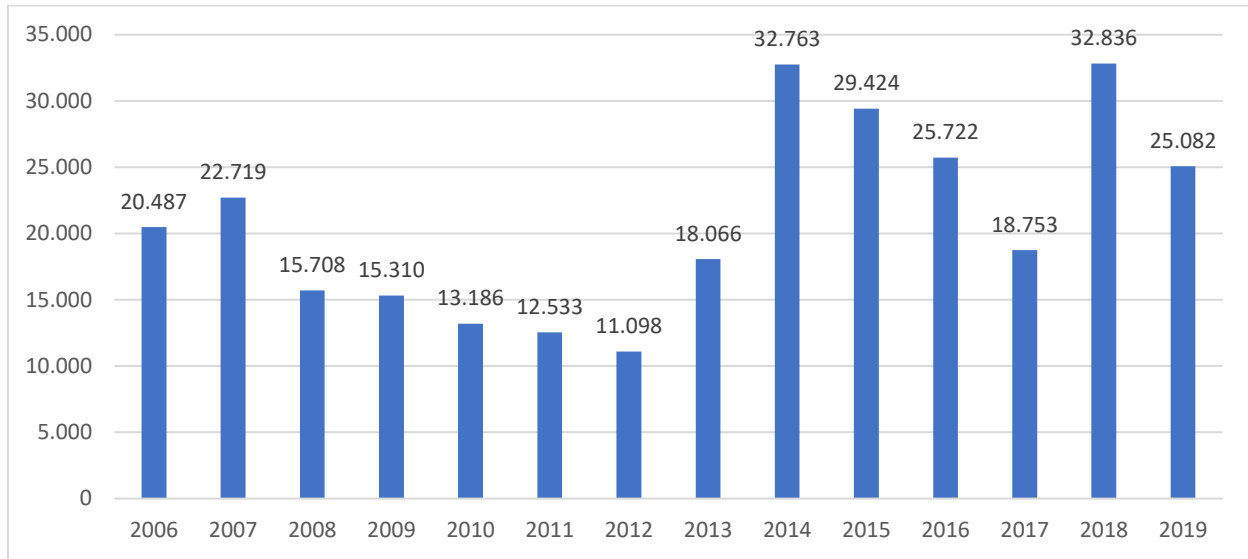


Figure 1 The total number of deaths caused by terrorism worldwide [21]

Moreover, terrorist incidents pose a serious threat to economic growth by damaging investment, tourism and consumption [22]. Targeting visitors and travel destinations, terrorism activities are one of the important factors that negatively affect the economy. The fear and anxiety created by terrorist attacks that harm public spaces and civilian population in particular lead to the loss of tourists, which is an important source of employment and currency in the economy [23]. The rates of foreign direct investment and portfolio investment have a very important share in economic development. However, companies and investors invest in countries where prosperity and security are developed, rather than in regions with high terrorism risk [24].

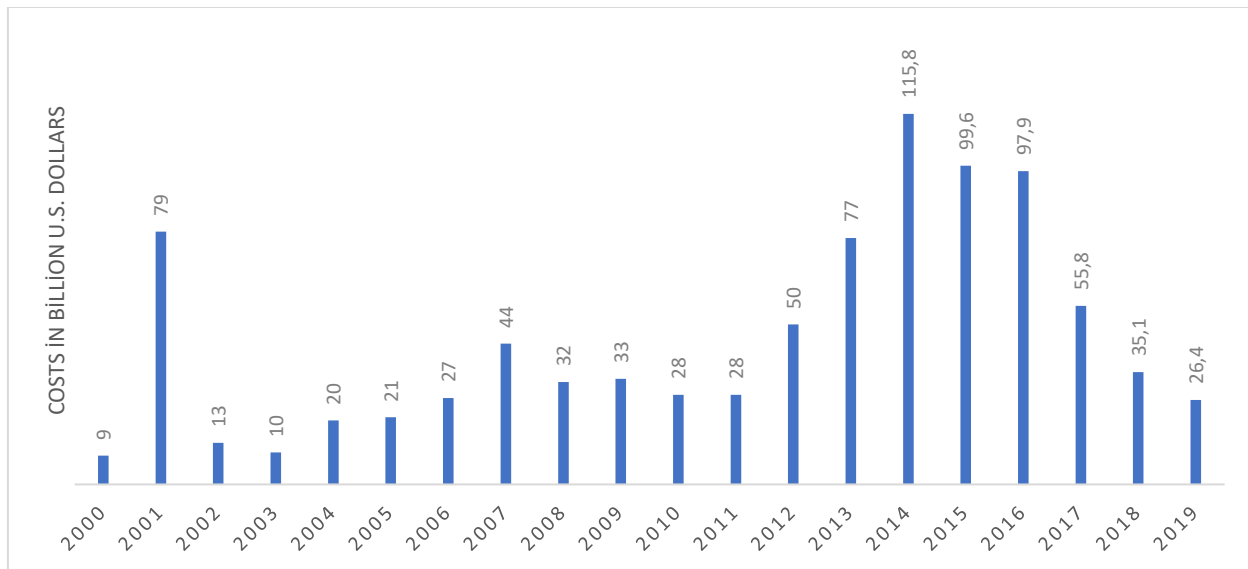


Figure 2 Global economic costs of terrorism 2000-2019 [19]

According to the Figure 2, terrorism cost totaled \$ 901.6 billion between 2000 and 2019. The highest economic costs, at 115.8 billion dollars, were recorded in 2014. It has been observed that the cost of terrorism has decreased since 2014. The September 11, 2001 attacks is one of the most damaging terrorist incidents to the economy with \$ 40.6 billion. The second most costly terrorist attack, the Sinjar massacre,



which resulted in the deaths of 104 members of the Yazidi community in Iraq by Islamic State groups in 2014, caused 104 million losses [22], [25].

Terrorist groups are global threats to the defense, infrastructure, and people of countries and communities around the world [26]. The most deadly terrorist groups in 2019 were the Taliban, Boko Haram, ISIL, and Al-Shabaab. They were responsible for 7.578 terrorism-related deaths in 2019, accounting for 55% of all terrorism-related deaths based on historical record [27].

Having no accepted universally definition [28], cybercrime is crimes that involve the use of a computer and hardware devices or network systems to inflict the vulnerable targets [29]. The computer or device can be the target as a perpetrator or facilitator, as well as the crime can be committed in other non-virtual places. Therefore, cybercrime can be classified into two different types. The first type of cybercrime is usually singular incidents from the perspective of the victim and it is more technical nature. It occurs when criminal software programs such as viruses, trojans, etc., infiltrate the user's computer through security vulnerabilities. The second type of cybercrime, on the other hand, usually involves repeated contacts or events from the user's point of view, and is often facilitated by programs that do not fit into the crimeware classification, such as instant messaging. Cyberstalking and harassment, child predation, extortion, blackmail, stock market manipulation, intricate corporate espionage, and planning or carrying out terrorist actions online are all examples of the second type of cybercrime [30]. As seen in Table 1, there are various types of cybercrime. In this study, cyberterrorism will be examined in detail.

Table 1 The characteristics of cyber crimes [30]

<b>Cybercrimes</b>	<b>Type</b>	<b>Software</b>
Phishing	I	Mail Client
Identity Theft	I	Keylogger, Trojan
Cyberstalking	II	Email Client, Messenger Clients
DDoS	I	Bots
Cyberterrorism (communication)	II	Steganography, Encryption, Chat Software

Cyber terrorism was primarily defined as a planned attack on data and computer systems by terrorists. All kinds of terrorist activities that use Internet as a tool are covered in the concept of cyber terrorism [31]. Phishing, identity theft, cyber stalking, DDoS, cyberterrorism (communication) have been successfully committed through the Internet and also substantially affected with each other.

The globalization and modern technology have strengthened the presence of terrorists in cyberspace as well as physical environments [32]. In the 1990s, terrorists were used Internet for only cyber attacks in order to damage critical infrastructures. Subsequently, their purpose changed with the 9/11 attacks and they use the Internet mainly for propaganda, data mining, operations coordination, recruiting and fundraising [33]. Indeed, the perpetrators of the 9/11 attacks frequently contacted al-Qaeda leaders over the Internet to plan their attacks [34], [35].

On the surface web, counterterrorism teams can detect terrorist activities or remove extremist contents. These interventions have led terrorist groups to escape repression and move towards more anonymous environments that are difficult to monitor and identify [5]. Beatrice Berton from the European Union Security Institute stated that ISIS tend to employ new safe online environments due to government interventions on jihadists' extremist content on the Internet [14].

The anonymity provided by the dark web enables encryption for communication, additionally, cryptocurrencies provide privacy in the financial environment for terrorist activities [17]. Donors secretly fund terrorist group [36] and terrorists collect donations by taking advantage of the anonymity of cryptocurrencies on the dark web [37].

There is a lot of evidence that terrorists operate on the dark web. The French Interior Minister Bernard Cazeneuve stated that the terrorist organizations responsible for terrorist attacks in Europe through communicate using dark web (Weimann, 2015). The research conducted by The Institute for National Security Studies (INSS) (2013), Al-Qaeda is also one of the terrorist groups communicating over the dark web. In 2015, after the closure of many websites of ISIS in the Operation Paris (OpParis), the information required for the transition to the Dark Web was published in the Al-Hayat Media Center (ISIS media) and ISIS officially proclaimed that it would continue its activities on the dark web [5]. Terrorists publish the books and manuals about the use of the dark web and TOR for their supporters [39], [40]. In addition to establishing connections, ISIS members are also using dark web marketplaces to obtain fake IDs and passports as legal regulations increase in border controls [14]. Terrorist organizations such as Aum Shinrikyo and Al-Qaeda are experimenting with various methods for access to adequate and efficient resources, equipment, and qualified experts in order to produce chemical, biological, radiological, and nuclear (CBRN) weapons [41]. In order to acquire CBRN weapons, terrorists utilize the Dark Web as a source. They purchase materials of these weapons from the darknet markets and they recruit chemists or other staff knowledgeable about CBRN weapons production [42]. Even more terrifying, there is strong evidence that the weapons used in the 2015 Paris attacks and 2016 Munich attack were supplied from the dark web [43], [44].

Terrorism in cyberspace is an important issue that needs to be investigated and prevented, as it is cheaper, more anonymous, more universal, and more effective than "offline" terror. Throughout one computer and necessary software, terrorist ideas and propaganda can easily be transported across borders, more supporters can be found, and then innocent people is affected due to terrorist acts in the dark web [45]. Therefore, to mitigate these wide-ranging effects, national and international efforts should be stepped up to tackle terrorism in the cyber environment.

### **3. Research Methodology**

The study aims to observe research efforts, detect the methodological gaps and gain further research opportunity of the dark web studies under the terrorism concept.

Our research questions (RQs) are;

RQ1: What is the main focus of the researchers?

RQ2: What types of data source are used to uncover criminals on the Dark Web?

RQ3: What systematic methods are implemented for detecting dark web crimes?

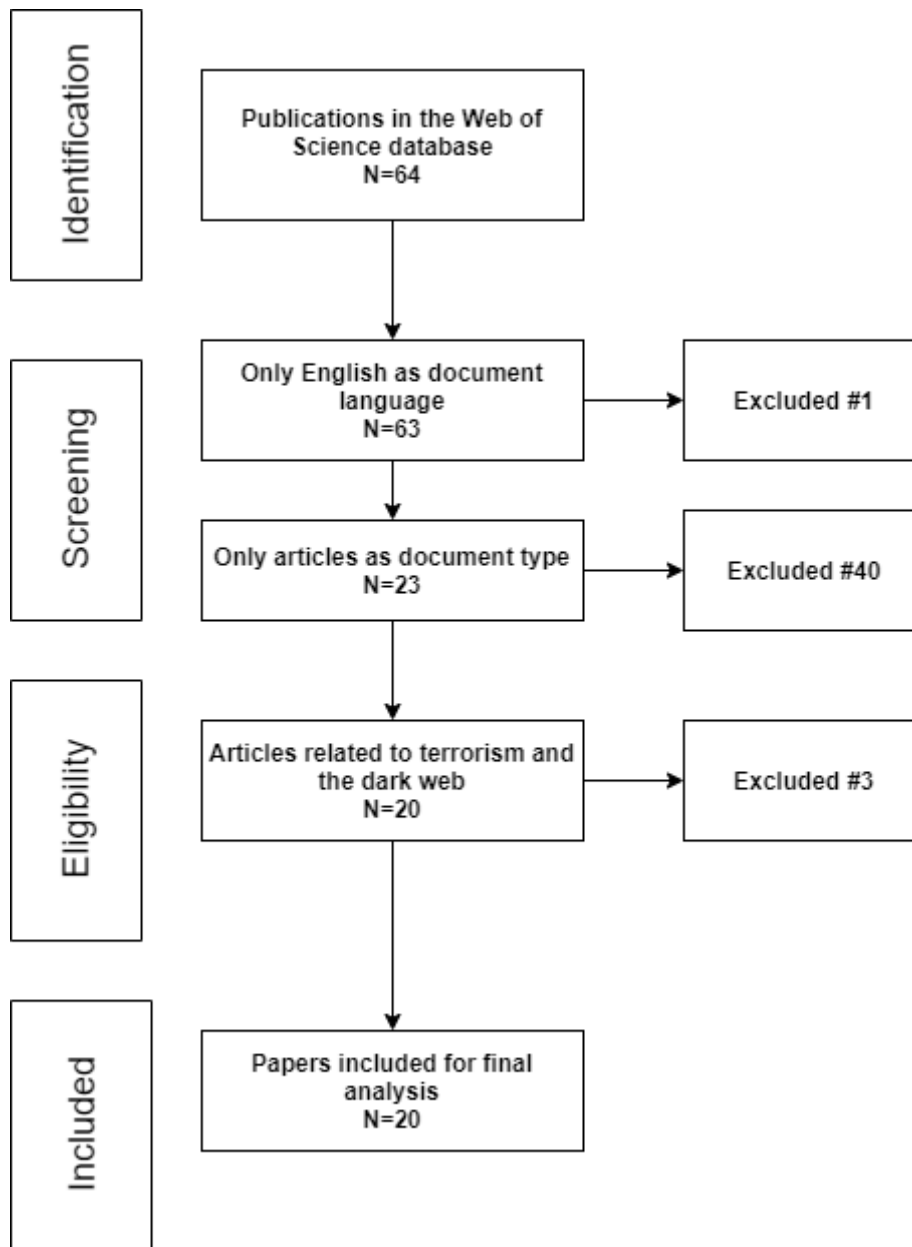


Figure 3 The flow diagram for the database search of publications for literature review

To address the research questions, a critical literature review method are applied to dark web studies within the scope of terrorism. A critical literature review is the assessment and overview of the ideas and information in manuscripts. There are two main points in critical review. The first is to scan the relevant literature efficiently and the second is to evaluate the information in the documents. Moreover, the content and different components of the text are analyzed. These components can define method that has boundaries such as the main theme of the text, data source, discussions made by the author and so on.

The flow diagram for the database search for publications is given in Figure 3. Web of science was chosen as the database since it is the world's most trusted publisher-independent global citation database. It is also a comprehensive platform with over 171 million records and 1.9 billion cited reference[46]. To collect data for the review articles, the most relevant keywords were selected. The search keywords were ("dark web"

OR "TOR" OR "anonymous network" OR "darknet") AND ("terrorism" OR "terrorist groups" OR "terrorist organization" OR "terrorist" OR "ISIL" OR "ISIS" OR "Daesh" OR "The Islamic State of Iraq and the Levant" OR "Al-Qaeda" OR "Boko Haram" OR "Taliban" OR "jihad" OR "jihadist" OR "cyberterrorism" OR "international security" OR "international terrorism"))).

There were some searching criteria. There is no date and category limitation and only articles were selected as the document type, and only English was chosen as the studies' language as seen in Figure 3. As a result of searching, 23 studies were reached, but since three studies were found to be irrelevant to the topic, 20 studies were examined finally.

#### **4. Result and Discussion**

Determining the purposes of dark web studies under terrorism concept can be a clue for identifying emerging dark web threats and the focal points of acts. The foci, methods and data sources used to detect terrorist activities in the dark web studies will provide guide book to researchers with an overview of the latest methodologies used in combating dark web terrorism. The findings of the study are categorized in three different tables based on research design. Table 2 shows the detail of studies using quantitative research design, Table 3 figures the studies using qualitative research design, and also Table 4 represents the studies using mix research design.

According to Table 2, the motivations of studies using quantitative research were generally oriented identifying dark web contents, suggesting methods to determine crime pattern, detecting terrorist activities, and uncovering the illegal activities. In these studies, researchers generally collected the data in Dark websites which are primary data. Most of the studies using primary data are collaborative research. The co-authors may devote more resources and effort to data collection and analysis.

Table 2 The details of studies using Quantitative research design

<b>Name</b>	<b>Year</b>	<b>Purpose</b>	<b>Data Types</b>	<b>Dataset</b>	<b>Data Source</b>	<b>Sample Size</b>	<b>Data Period</b>	<b>Data analysis method</b>
[47]	2005	Discovery and analysis of the Dark Web content	Primary Data	Multimedia and multilingual Web contents	Dark Web	Not reported	until April 2004	The content and link analysis
[48]	2006	Suggesting a method for collecting Dark Web content and investigating terrorists' use of the Internet.	Primary Data	Multimedia Web documents	Dark Web	200.000 websites	until June 2004	Content Analysis
[49]	2006	Performing topological analyses of terrorist websites from various geographical region	Primary Data	Websites contents	Dark Web	311 Websites	until November 2004	Social network analysis
[35]	2007	Recommending an approach for collecting terrorist/extremist Web content on the Dark Web	Primary Data	Multimedia Web documents	Dark Web	200.000 Websites	until June 2004	Content Analysis
[50]	2015	Predicting the daily amount of violent extremist groups' cyber-recruitment activity	Secondary Data	Ansar1 forum posts	Dark Web Forum Portal	28.747 posts	December 2008- January - 2010	LDA and time series analysis
[51]	2016	Offering a hybridized term-weighting strategy for the detection of terrorist activities	Secondary Data	Arabic dark web pages and non- dark web pages	Dark Web Forum Portal and Open Source Arabic Corpora	1.000 Arabic dark web pages and non- dark web page	Not reported	Classification methods
[52]	2018	Examining internet users' views and perceptions about online hate speech and informing internet users and policy makers about cyberhate	Primary Data	Responses from survey participants	Internet users in Turkey and the USA	372 respondents	Not accessible	Survey

Table 2 The details of studies using Quantitative research design (continue)

[53]	2018	Examining the Tor structure and designing a strategy for quickly identifying places that may contain material of interest to law enforcement.	Primary Data	Website contents	Dark Web	232.792 sites	12 April 2016 -01 July 2016	Classification methods
[54]	2019	Developing automatic purchasing models to detect unauthorized firearm purchases by gathering and data from different discussion forums in the dark web	Secondary Data	Ansar Aljihad Network, Islamic Awakening, Gawaher, and Islamic Network forum sites	Dark Web Forum Portal	4,297,961 messages, 1,553,122 thread in the forums	2004-2012	Machine learning classification techniques (SVM, Boosting, Random Forest, GLMNET, Tree, and MAXEN)
[55]	2019	Creating a model that predicts a terrorist group's future lethality	Secondary Data	Information related to terrorist attacks	GTD and RAND	157 Terrorist Attacks	GTD-1970-2014 RAND-1968-2009	Regression analysis and simulation
[56]	2021	Proposing a link-based ranking approach for evaluating and identifying the hidden services in the Tor	Primary Data	Website contents	Dark Web	Not reported	Not reported	Link analysis

Qualitative research usually maintains the information about Dark web and terrorist activities as seen in Table 3. Some topics seem unrelated to terrorism such as Captagon -a psychostimulant drug- , animal trade, digital artifacts, however they are indirectly feed terrorism. Most of studies in qualitative research do not include method section and data sources. The methods section outlines included the research problem, specific procedures, process, and analysis of data relevant to understanding the problem provide the reader to critically assess the study's reliability and validity [57]. Therefore, the absence of a method section in these articles adversely affects the evaluation process. Table 3 displays all of the qualitative analysis using secondary data. Researchers focus on the literature, scientific publications, and various databases as data sources and the documentary analysis/review method is the main approach in qualitative research.

Table 3 The details of studies using Qualitative research design

Article	Year	Purpose	Data Types	Dataset	Data Source	Sample Size	Data Period	Data analysis method
[58]	2015	Examining studies that describe online data mining literature with a clear focus on law enforcement applications	Secondary Data	Scientific literature	IEEEExplore, The ACM Digital Library, Springer-Link, ScienceDirect	206 publication	December 2012 - January 2013	Systematic literature review
[59]	2016	Presenting an overview of the most common darknets and their related information and the perspective of Law Enforcement Agencies on Open Source Intelligence	Secondary Data	Documents and scientific publications	Literature	Not reported	Not reported	Documentary analysis / Review and Case Study
[33]	2016	Presenting general information about the dark web and dark web terrorist activity.	Secondary Data	Documents and scientific publications	Literature	Not reported	Not reported	Documentary analysis / Review
[60]	2016	Providing detailed information on wildlife smuggling via the dark web, as well as a map of the trafficking of animal parts	Secondary Data	Documents and scientific publications	Literature	Not reported	Not reported	Documentary analysis / Review
[61]	2016	Getting the most up-to-date information on Captagon e-commerce in the Middle East	Secondary Data	Websites, drug forums and other online resources in both English and Arabic, literature	Medical and paramedical databases, web ,Darkweb, and the Global Public Health Intelligence Network database	Not reported	October 2015- May 2016	Thematic analysis
[62]	2018	Investigating Dark Web networks that exploit digital artifacts and identify the hidden actors behind these operations	Secondary Data	Documents and scientific publications	Literature	Not reported	Not reported	Document analysis / Review
[63]	2019	Defining the risks posed by the use of Fentanyl and Fentanyl +, as well as the demographic at risk.	Secondary Data	Documents and scientific publications	Literature	Not reported	Not reported	Documentary analysis / Review



While many studies in the cyber terrorism literature addressed the quantitative and qualitative research design, two studies used mixed research design as seen in Table 4. They compile both data types and have sophisticated data analysis methods.

Table 4 The details of studies using mixed research design

<b>Article</b>	<b>Year</b>	<b>Purpose</b>	<b>Data Types</b>	<b>Dataset</b>	<b>Data Source</b>	<b>Sample Size</b>	<b>Data period</b>	<b>Data analysis method</b>
[64]	2008	Creating a modern approach to gathering and analyzing Dark Web data.	Primary Data	Web sites contents	Dark Web	94. 326 websites	until 2004	Web page clustering, classification, and case study
[65]	2011	Offering an explanation about new phenomenon called as "Terrorism Informatics"	Primary and Secondary Data	Books, terrorism research centers and resources, and international terrorist organizations	Think Tanks and Intelligence Resources, Terrorism Databases and Online Resources, Higher Education Research Institutes, and the Dark Web	10.000 website, 300 terrorist forums in the Dark Web	Not reported	Review, Social Network Analysis, Content Analysis, Web Metric Analysis, Sentiment and Affect Analysis, Authorship Analysis and Writeprint, Video Analysis

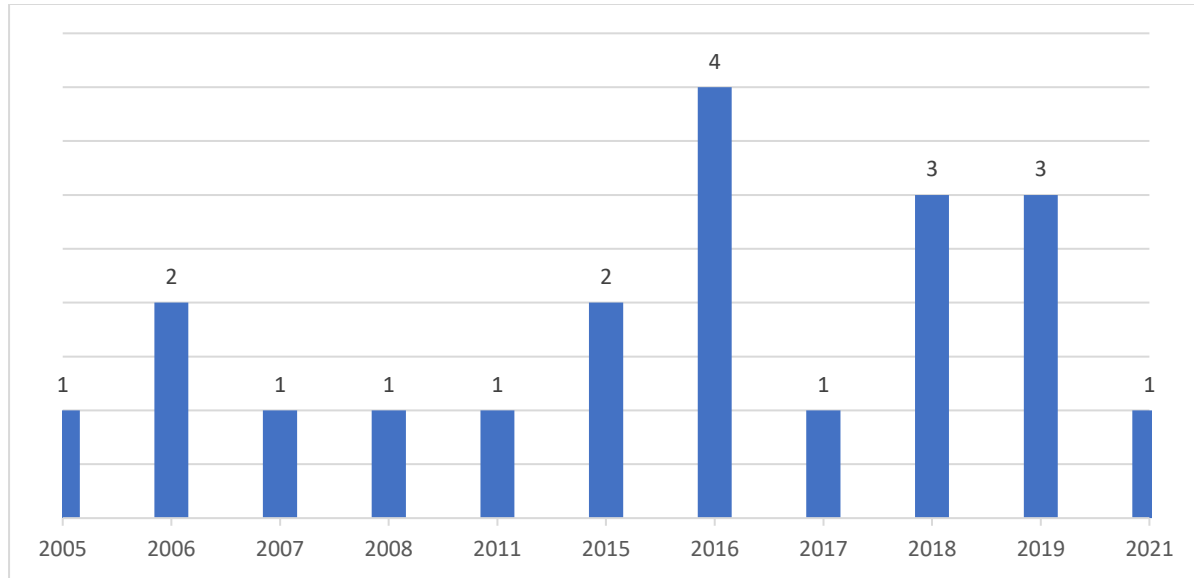


Figure 4 Period of published articles

Consequently, the number of studies is quite insufficient but these are still important topics. Although, there was no time and research area limitation, the interest and study efforts in this field are inadequate. As seen in Figure 4, the number of publications is excessive as an instance in 2016, 2018 and 2019 compare with the other timespan, there are no publications before 2005. The first use of the Dark Web phrase dates back to the 2000s [66], and also the first scientific study on terrorism activities in the dark web was published in 2005. A few articles were published in most years, and nearly 50% of the total publications were published in 2016, 2018 and 2019. In addition, there are no manuscripts in 2009, from 2012 to 2014 and in 2020 as well. This indicates that there is an academic gap in this field. The same authors mostly have their manuscripts since 2011. The articles [36] and [49] are the same in terms of purpose, dataset and method, but they have published as different articles.

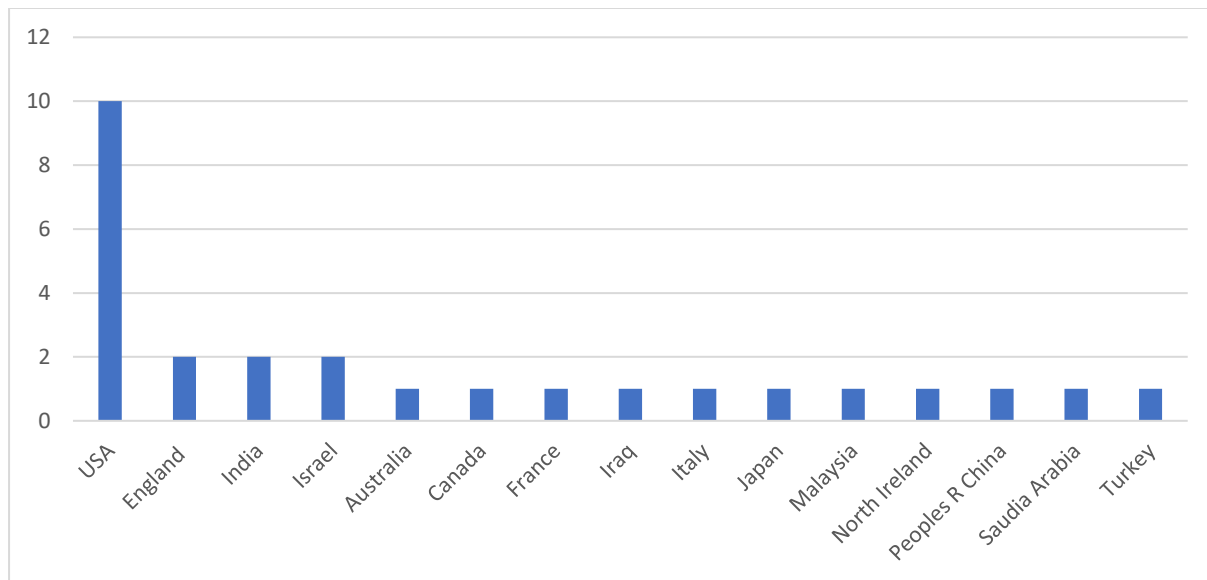


Figure 5 The distribution of articles by region

The majority of studies are from the United States, followed by England, India and Israel according to Figure 5. USA is one of the countries with the highest dark web usage [67], and accordingly, most of the studies

originated from the USA. Turkey, however, despite being among the countries with the highest usage of dark web, it is surprising that there is lack of the dark web studies in the context of terrorism. Therefore, researchers' efforts in this field should increase in Turkey.

Approximately 38% of the studies were supported by funding agencies. This rate may be a reason for the limited number of studies. Without funding, researchers are more reluctant to innovative approaches and may have to redirect their research areas to other activities that require less resources, time, and effort [68].

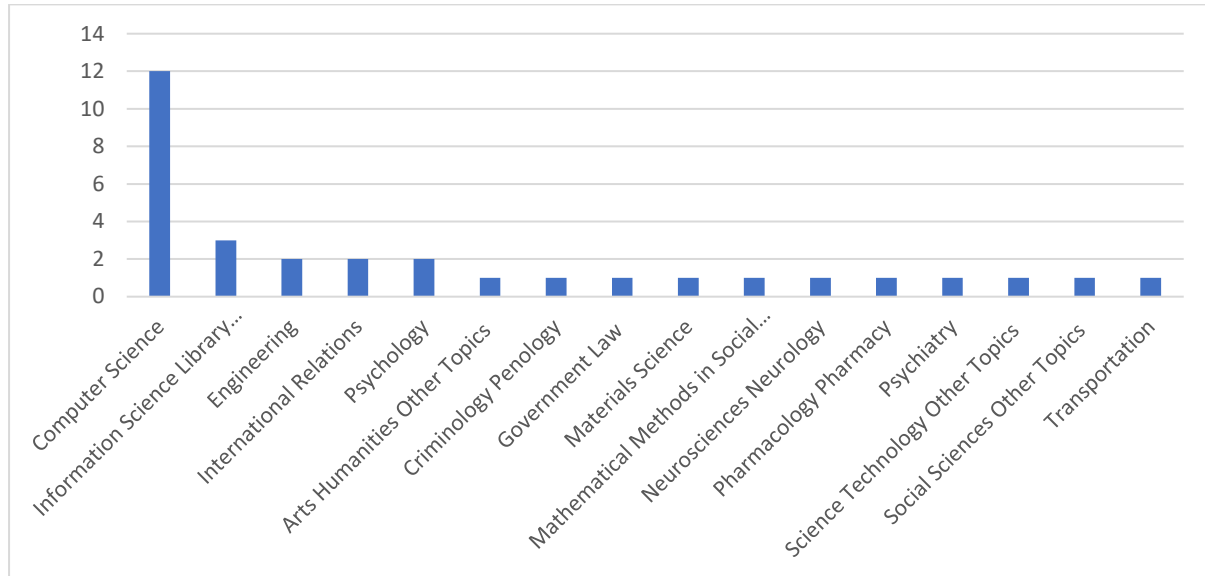


Figure 6 The distribution of articles according to web of science categories

The studies are concentrated in the category of Computer Science as seen in Figure 6. A little manuscript is in the international relations and social sciences interdisciplinary areas. Such research should be expanded in sociopolitical disciplines, as the topic also covers the notion of terrorism.

Researchers focus on providing a framework for understanding the concept of terrorism on the dark web and trying to identify trends and patterns in that concept. They frequently use descriptive analysis to learn general information about dark web and its components, comprehend structural characteristics, investigate terrorist incidents on the dark web, and establish the location of terrorist groups. They also utilized predictive analysis to make predictions about the future terrorist activities in the dark web.

The development of scientific approaches to explain and combat violent extremism is of worldwide interest [69]. The U.S. government alone spends half a trillion dollars every year investigating, fighting and reacting terrorism [70]. In terms of quantity, qualitative and quantitative methods utilized in dark web literature in the context of terrorism are comparable, but quantitative methods have a minor edge.

The principal methods of quantitative research performed on the dark web are social network analysis, link analysis and content analysis as seen in Table 2. Social network analysis is a method that uses various modeling techniques to examine the dynamics of social networks based on structure and interaction of communities [71]. In terrorist investigations, text mining techniques and social network analysis aid in the development of counter-terrorism approaches. Social network analysis can be useful for identifying key members of terrorist organizations, centrality measures for terrorism network, sub-groups detection, and so on [72].

The link analysis is known as the act of establishing networks of interconnected objects through relationships for discovering patterns and trends. It is commonly used to find central players and noteworthy patterns in dataset [73]. It is an effective method that can be used in dark web studies to destabilize the organization's activities by capturing some key figures in terrorist groups.

Analyzing the structure of terrorist networks can provide a technical understanding that can be used to prevent unlawful operations on the dark web [72]. Content analysis can be qualitative and quantitative techniques. The quantitative content analysis based on counting and measuring while qualitative content analysis based on interpreting and comprehending [74]. Quantitative content analysis is commonly used approach in dark web research to characterize the appearance of a variety of features in content, such as technical sophistication, media richness, and web interaction.

Classification methods are relatively implemented methods in dark web studies. The widespread term weighting methods, TF, DF, TF-IDF, Entropy and Glasgow are used to create structured data then analyze the web page. Many different effective techniques such as affect analysis, sentiment analysis, authorship analysis, latent Dirichlet allocation (LDA) were also applied. The sentiment and affect analysis allow for the identification of violent and extremist sites that pose serious threats [75]. The affect analysis of terrorists on the Dark Web reveals the dissemination of terror, hatred, and propaganda [72]. Another method, authorship analysis includes techniques for investigating the attributes of study in order to present conclusions on its authors [76]. Authorship analysis techniques are needed in cyber forensic to detect criminals on the Dark web who use services like TOR [72]. Topic based approach, LDA uncovers hidden topics from large document of corpus [77], and it can be used for specifying the topics discussed in the dark web.

Researchers tend to design research based on available data rather than collecting the data themselves. Because terrorists may disguise their identities and remove traces of their Internet actions, researchers and scholars have a tough time acquiring and analyzing Dark Web [64]. Other barriers of the collection are information overload and language barriers [78]. Therefore, researchers generally use secondary data instead of collecting their own dataset. The initial research relied on primary data due to young concept and the scarcity of secondary data. Spiders were used by the researchers to find the relevant terrorist groups based on reports from authoritative sources. The Dark Web Forum Portal (DWFP), which was created by collecting Dark Websites in 2004, was often used as a secondary dataset in later studies. The DWFP allows users to access vital foreign jihadist web forums through the Internet [79]. The forum sites named Gawaher, Islamic Awakening, Islamic Network, and Turn to Islam are closer to jihadist approaches, some studies used these forum sites' contents as data. The Global Terrorism Database (GTD) and RAND Database on Worldwide Terrorism Incidents (RAND) are another databases referenced. Some studies is not depicted the time interval of data collection, so it is unclear whether the studies yield up-to-date results.

Future studies will require advanced analytical methodologies and terrorism databases or data collection methods to emerge complex interactions and activities in the Dark Web. Scholars should adapt a wider range of data collection techniques. Increased utilization of primary data could help researchers build a more solid empirical foundation for understanding terrorism and counterterrorism. They can create automated approaches that scrape the dark web with web scraping techniques for produced content assisted lawmakers and law enforcement [72]. Moreover, advanced artificial intelligence techniques, image processing, and natural language processing techniques, hash value analysis, sock puppet detection techniques can be applied as well. Advanced artificial intelligence techniques can be quite effective for tracking and detecting malicious activities. Image classification is one of the effective methods to detect unwanted, harmful or criminal content on Web pages. Detecting people who have criminal tendency by analyzing posts on Dark Web forums can be provided with natural language processing methods. Hash value analysis is a strong technique in cryptography for proving the authenticity of digital evidence during an inquiry [72]. The connecting server's destination can be determined via hash value analysis at the onion routing's exit node layer [80], [81]. Sock puppet is the use of numerous usernames or fake identity to converse online [82]. Sock puppet detection plays a critical role in monitoring communication and screening in the Dark Web for terrorist tracking [72].

## **5. Conclusion**

This research identifies the existing research on dark web in the scope of terrorism and these studies are critically reviewed in terms of purpose, dataset, and applied method for identifying the criminals and crimes in the Dark Web.

According to the findings, the number of dark web research on terrorism was insufficient, indicating a gap in the literature. Thus, scholars and practitioners must continue to urge action in order to close the gap. The link analysis, web mining techniques, classification methods, affect analysis, sentiment analysis, authorship analysis, content analysis, and natural language processing techniques, documentary analysis/review are used in these studies. The dark web' anonymity gives drawback to discover crimes; hence, systematic methods for detection should be expanded. Advanced artificial intelligence, machine learning, deep learning, image processing, natural language processing techniques, hash value analysis, sock puppet detection techniques can be used to track and combat cyber-terrorism.

This study appears as one of the first papers that have mapped the literature exploring dark web studies in the scope of terrorism. New methods suggestions for improving dark web studies are offered through an overview of the academic publications. This constitutes a guideline for researchers and practitioners in the dark web field in the future. By using the critical review of existing literature, researchers can improve studies with new proposed methods and also contribute to the quantitative and qualitative research design in dark web studies, which are few in number.

This paper is limited by language restrictions. The only language of the reviewed studies was English. Moreover, only the Web of Science was used as a database. Another limitation is the low number of dark web studies in the context of terrorism. Sophisticated techniques to detect the main topics could not be used due to the scarcity of studies. Additionally, the lack of some details in some studies (areas indicated by "not reported" in table 2,3,4) is another point that challenges us during the analysis phase.

Future studies may focus on various databases, languages, and keywords to achieve more global results. The terrorism in the dark web can be examined with more advanced methods. Therefore, various text mining techniques, bibliometric analysis, and topic modeling approaches such as LDA can be applied on extensive literature. Furthermore, other evaluation criteria can be added in order to reach more comprehensive results.

## **References**

- [1] S. D. Keene, "Terrorism and the internet : a double-edged sword," *J. Money Laund. Control*, vol. 14, no. 4, pp. 359–370, Oct. 2011, doi: 10.1108/13685201111173839.
- [2] J. R. C. Nurse and M. Bada, "The Group Element of Cybercrime: Types, Dynamics, and Criminal Operations," Jan. 2019, doi: 10.1093/oxfordhb/9780198812746.013.36.
- [3] D. Bieda and L. Halawi, "Cyberspace: A Venue for Terrorism," 2015. Accessed: May 28, 2021. [Online]. Available: <https://commons.erau.edu/publication/304>.
- [4] G. Weimann, "Cyberterrorism How Real Is the Threat?," 2004. Accessed: May 28, 2021. [Online]. Available: [www.usip.org](http://www.usip.org).
- [5] G. Weimann, "Going Darker ? The Challenge of Dark Net Terrorism," 2015.
- [6] M. Chertoff and T. Simon, "The Impact of the Dark Web on Internet Governance and Cyber Security," *Glob. Comm. Internet Govrnance*, no. 6, pp. 6–8, 2015, [Online]. Available:

[https://www.cigionline.org/sites/default/files/gcig\\_paper\\_no6.pdf](https://www.cigionline.org/sites/default/files/gcig_paper_no6.pdf).

- [7] Statista, “Internet users in the world,” 2021. <https://www.statista.com/statistics/617136/digital-population-worldwide/> (accessed Dec. 22, 2020).
- [8] M. de Kunder, “The size of the World Wide Web (The Internet),” 2020. <https://www.worldwidewebsite.com/> (accessed Dec. 21, 2020).
- [9] G. Weimann, “Terrorist Migration to the Dark Web,” *Perspect. Terror.*, vol. 10, no. 3, pp. 40–44, 2016.
- [10] M. . Bergman, “The deep Web: Surfacing hidden value,” *J. Electron.*, vol. 7, no. 1, 2001.
- [11] K. Finklea, “Dark Web,” Taylor and Francis Inc., Mar. 2017. doi: 10.1080/1057610X.2015.1119546.
- [12] E. Jardine, “The Dark Web Dilemma: Tor, Anonymity and Online Policing,” 2015. Accessed: Jan. 21, 2021. [Online]. Available: <https://ssrn.com/abstract=2667711>.
- [13] R. Dingedine, N. Mathewson, and P. Syverson, “Tor: The Second-Generation Onion Router,” 2004.
- [14] B. Berton, “The dark side of the web : ISIL ’ s one-stop shop ?,” no. June, pp. 1–2, 2015, doi: 10.2815/454889.
- [15] N. Tavabi, N. Bartley, A. Abeliuk, S. Soni, E. Ferrara, and K. Lerman, “Characterizing activity on the deep and dark web,” *Web Conf. 2019 - Companion World Wide Web Conf. WWW 2019*, pp. 206–213, 2019, doi: 10.1145/3308560.3316502.
- [16] D. Moore and T. Rid, “Cryptopolitik and the darknet,” *Survival (Lond.)*, vol. 58, no. 1, pp. 7–38, Jan. 2016, doi: 10.1080/00396338.2016.1142085.
- [17] N. Malik, “Terror in the Dark,” London, 2018. [Online]. Available: <http://henryjacksonsociety.org/wp-content/uploads/2018/04/Terror-in-the-Dark.pdf>.
- [18] S. Alayda, N. A. Almowaysher, F. Alserhani, and M. Humayun, “Terrorism on Dark Web,” vol. 12, no. 10, pp. 3000–3005, 2021.
- [19] Statista, “Global economic costs of terrorism 2019,” 2021. <https://www.statista.com/statistics/489649/global-economic-costs-of-terrorism/> (accessed May 09, 2021).
- [20] U. N. H. C. for H. Rights, “Negative effects of terrorism on the enjoyment of all human rights and fundamental freedoms,” *Ge*, vol. 23159, no. December 2016, 2016, [Online]. Available:

<https://documents-dds-ny.un.org/doc/UNDOC/GEN/G16/444/16/PDF/G1644416.pdf?OpenElement>.

- [21] Statista, “Number of fatalities due to terrorist attacks worldwide between 2006 and 2019,” 2021. <https://www.statista.com/statistics/202871/number-of-fatalities-by-terrorist-attacks-worldwide/> (accessed May 09, 2021).
- [22] H. Bardwell and M. Iqbal, “The Economic Impact of Terrorism from 2000 to 2018,” *Peace Econ. Peace Sci. Public Policy*, 2020, doi: 10.1515/peps-2020-0031.
- [23] N. ÇELİK and M. KARAÇUKA, “Terör Saldırılarının Yerli ve Yabancı Turistlerin Destinasyon Tercihleri Üzerindeki Etkileri: Türkiye İBBS-II Bölgeleri’ne Yönelik Mekansal Bir Analiz,” *Akdeniz Üniversitesi İktisadi ve İdari Bilim. Fakültesi Derg.*, pp. 204–222, May 2019, doi: 10.25294/auibfd.559403.
- [24] Y. Özkaya and T. Şimşek, “THE RELATIONSHIP BETWEEN TERRORISM AND FINANCIAL STRUCTURE TERÖR VE FİNANSAL YAPI ARASINDAKİ İLİŞKİ ÖZ,” 2017.
- [25] BBC, “Iraq’s Yazidi community buries 104 victims of IS massacre,” Feb. 07, 2014. <https://www.bbc.com/news/world-middle-east-55968068> (accessed May 28, 2021).
- [26] RAND, “Terrorist Organizations,” 2021. <https://www.rand.org/topics/terrorist-organizations.html> (accessed May 09, 2021).
- [27] IEP, “GLOBAL TERRORISM INDEX 2020,” 2020.
- [28] E. C. Viano, “Cybercrime: Definition, Typology, and Criminalization Defining Cybercrime,” 2017, doi: 10.1007/978-3-319-44501-4\_1.
- [29] A. Chandra and M. J. Snowe, “A taxonomy of cybercrime: Theory and design,” *Int. J. Account. Inf. Syst.*, vol. 38, Sep. 2020, doi: 10.1016/J.ACCINF.2020.100467.
- [30] S. Gordon and R. Ford, “On the definition and classification of cybercrime,” *J. Comput. Virol.*, vol. 2, no. 1, pp. 13–20, Aug. 2006, doi: 10.1007/S11416-006-0015-Z.
- [31] C. Wu and J. Wang, “Analysis of Cyberterrorism and Online Social Media,” vol. 351, no. Mmetss, pp. 925–927, 2019.
- [32] G. Weimann, “Terror in Cyberspace,” 2009. [https://www.researchgate.net/publication/45380139\\_Terror\\_in\\_Cyberspace](https://www.researchgate.net/publication/45380139_Terror_in_Cyberspace) (accessed May 28, 2021).
- [33] G. Weimann, “Going Dark: Terrorism on the Dark Web,” *Stud. Confl. Terror.*, vol. 39, no. 3, pp. 195–206, 2016, doi: 10.1080/1057610X.2015.1119546.



- [34] CTIT, “Countering the Use of the Internet for Terrorist Purposes— Legal and Technical Aspects,” 2011. Accessed: May 09, 2021. [Online]. Available: [www.un.org/terrorism/internet](http://www.un.org/terrorism/internet).
- [35] J. Qin, Y. Zhou, E. Reid, G. Lai, and H. Chen, “Analyzing terror campaigns on the internet: Technical sophistication, content richness, and Web interactivity,” *Int. J. Hum. Comput. Stud.*, vol. 65, no. 1, pp. 71–84, Jan. 2007, doi: 10.1016/j.ijhcs.2006.08.012.
- [36] K. Hausken, “The dynamics of terrorist organizations,” *Oper. Res. Perspect.*, vol. 6, Jan. 2019, doi: 10.1016/j.orp.2019.100120.
- [37] D. Harman, “U.S.-based ISIS cell fundraising on the dark web, new evidence suggests - Haaretz Com - Haaretz.com,” Apr. 10, 2015. <https://www.haaretz.com/.premium-isis-uses-bitcoin-for-fundraising-1.5366305> (accessed Jan. 24, 2021).
- [38] The Institute for National Security Studies (INSS), “Backdoor Plots: The Darknet as a Field for Terrorism,” 2013. <https://www.inss.org.il/index.aspx?id=4538&articleid=5574> (accessed Jan. 24, 2021).
- [39] Ş. Pektaş and J. Leman, “Militant Jihadism Today and Tomorrow,” 2019.
- [40] MEMRI, “The ‘Dark Web’ And Jihad: A Preliminary Review Of Jihadis’ Perspective On The Underside Of The World Wide Web ,” May 21, 2014. <https://www.memri.org/jttm/dark-web-and-jihad-preliminary-review-jihadis-perspective-underside-world-wide-web> (accessed Jan. 28, 2021).
- [41] A. Stenersen, *Al-Qaida’s Quest for Weapons of Mass Destruction: The History behind the Hype*. VDM Verlag Dr. Müller, 2008.
- [42] G. D. Koblenz, “Emerging Technologies and the Future of CBRN Terrorism,” *Wash. Q.*, vol. 43, no. 2, pp. 177–196, Apr. 2020, doi: 10.1080/0163660X.2020.1770969.
- [43] BBC, “Munich shooting: Manhunt after deadly attack at shopping centre - BBC News,” 2016. <https://www.bbc.com/news/world-europe-36870874> (accessed Jan. 22, 2021).
- [44] R. Bender and C. Alessi, “Munich Shooter Likely Bought Reactivated Pistol on Dark Net - WSJ,” Jul. 24, 2016. <https://www.wsj.com/articles/munich-shooter-bought-recommissioned-pistol-on-dark-net-1469366686> (accessed Jan. 22, 2021).
- [45] V. Vilic, “DARK WEB, CYBER TERRORISM AND CYBER WARFARE: DARK SIDE OF THE CYBERSPACE,” 2007. [https://www.researchgate.net/publication/324720749\\_DARK\\_WEB\\_CYBER\\_TERRORISM\\_AND\\_CYBER\\_WARFARE\\_DARK\\_SIDE\\_OF\\_THE\\_CYBERSPACE](https://www.researchgate.net/publication/324720749_DARK_WEB_CYBER_TERRORISM_AND_CYBER_WARFARE_DARK_SIDE_OF_THE_CYBERSPACE) (accessed May 10, 2021).
- [46] Clarivate, “Web of Science,” 2021. <https://clarivate.com/webofsciencegroup/solutions/web-of-science/> (accessed Sep. 22, 2021).

- [47] J. Qin, Y. Zhou, G. Lai, E. Reid, M. Sageman, and H. Chen, "The dark web portal project: Collecting and analyzing the presence of terrorist groups on the web," in *Lecture Notes in Computer Science*, 2005, vol. 3495, pp. 623–624, doi: 10.1007/11427995\_78.
- [48] J. Qin, Y. Zhou, E. Reid, G. Lai, and H. Chen, "Unraveling International Terrorist Groups' exploitation of the Web: Technical sophistication, media richness, and web interactivity," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 3917 LNCS, pp. 4–15, doi: 10.1007/11734628\_2.
- [49] J. Xu, H. Chen, Y. Zhou, and J. Qin, "On the topology of the dark web of terrorist groups," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 3975 LNCS, pp. 367–376, doi: 10.1007/11760146\_32.
- [50] J. R. Scanlon and M. S. Gerber, "Forecasting Violent Extremist Cyber Recruitment," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 11, pp. 2461–2470, Nov. 2015, doi: 10.1109/TIFS.2015.2464775.
- [51] T. Sabbah, A. Selamat, M. H. Selamat, R. Ibrahim, and H. Fujita, "Hybridized term-weighting method for Dark Web classification," *Neurocomputing*, vol. 173, pp. 1908–1926, 2016, doi: 10.1016/j.neucom.2015.09.063.
- [52] S. Celik, "Tertiary-level internet users' opinions and perceptions of cyberhate," *Inf. Technol. People*, vol. 31, no. 3, pp. 845–868, May 2018, doi: 10.1108/ITP-05-2017-0147.
- [53] J. Dalins, C. Wilson, and M. Carman, "Criminal motivation on the dark web: A categorisation model for law enforcement," *Digit. Investig.*, vol. 24, pp. 62–71, Mar. 2018, doi: 10.1016/j.diin.2017.12.003.
- [54] J. K. Saini and D. Bansal, "A Comparative Study and Automated Detection of Illegal Weapon Procurement over Dark Web," *Cybern. Syst.*, vol. 50, no. 5, pp. 405–416, Jul. 2019, doi: 10.1080/01969722.2018.1553591.
- [55] Y. Yang, A. R. Pah, and B. Uzzi, "Quantifying the future lethality of terror organizations," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 43, pp. 21463–21468, Oct. 2019, doi: 10.1073/pnas.1901975116.
- [56] A. Alharbi et al., "A Link Analysis Algorithm for Identification of Key Hidden Services," *Comput. Mater. Contin.*, vol. 68, no. 1, pp. 877–886, Mar. 2021, doi: 10.32604/cmc.2021.016887.
- [57] R. H. Kallet, "How to Write the Methods Section of a Research Paper," *Respir. Care* 49, pp. 1229–1232, 2004.
- [58] M. Edwards, A. Rashid, and P. Rayson, "A systematic survey of online data mining technology intended for law enforcement," *ACM Comput. Surv.*, vol. 48, no. 1, Sep. 2015, doi: 10.1145/2811403.

- [59] G. Kalpakis et al., "OSINT and the dark web," in *Advanced Sciences and Technologies for Security Applications*, Springer, 2016, pp. 111–132.
- [60] D. Jaclin, "Poached lives, traded forms: Engaging with animal trafficking around the globe," *Soc. Sci. Inf.*, vol. 55, no. 3, pp. 400–425, Sep. 2016, doi: 10.1177/0539018416648233.
- [61] A. AL-Imam et al., "Captagon: use and trade in the Middle East," *Hum. Psychopharmacol.*, vol. 32, no. 3, May 2017, doi: 10.1002/hup.2548.
- [62] K. Paul, "Ancient Artifacts vs. Digital Artifacts: New Tools for Unmasking the Sale of Illicit Antiquities on the Dark Web," *Arts*, vol. 7, no. 2, p. 12, Mar. 2018, doi: 10.3390/arts7020012.
- [63] A. R. Thomas and R. M. Schwartz, "At-risk populations to unintentional and intentional fentanyl and fentanyl+ exposure," *J. Transp. Secur.*, vol. 12, no. 3–4, pp. 73–82, Dec. 2019, doi: 10.1007/s12198-019-00202-1.
- [64] H. Chen, W. C. Chung, J. Qin, E. Reid, M. Sageman, and G. Weimann, "Uncovering the DarkWeb: A Case Study of Jihad on the Web," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 8, pp. 1347–1359, 2008, doi: 10.1002/asi.
- [65] H. Chen, "From Terrorism Informatics to Dark Web Research," 2011, pp. 317–341.
- [66] A. Baravalle, M. S. Lopez, and S. W. Lee, "Mining the Dark Web: Drugs and Fake Ids," *IEEE Int. Conf. Data Min. Work. ICDMW*, vol. 0, pp. 350–356, 2016, doi: 10.1109/ICDMW.2016.0056.
- [67] CIGI-IPSOS, "2019 CIGI-Ipsos Global Survey on Internet Security and Trust," 2019. <https://www.cigionline.org/internet-survey-2019> (accessed Mar. 20, 2021).
- [68] A. Silke, "Research on Terrorism," 2008, pp. 27–50.
- [69] Y. Yang, A. R. Pah, and B. Uzzi, "Quantifying the future lethality of terror organizations," vol. 116, no. 43, pp. 21463–21468, 2019, doi: 10.1073/pnas.1901975116.
- [70] A. Belasco, "The Cost of Iraq, Afghanistan, and Other Global War on Terror Operations Since 9/11," 2014. Accessed: Jun. 02, 2021. [Online]. Available: [www.crs.gov](http://www.crs.gov).
- [71] F. . Stokman, "Social Network Analysis," *Int. Encycl. Soc. Behav. Sci.*, 2001, Accessed: Apr. 27, 2021. [Online]. Available: <https://www.sciencedirect.com/topics/social-sciences/social-network-analysis>.
- [72] S. Nazah, S. Huda, J. Abawajy, and M. M. Hassan, "Evolution of dark web threat analysis and detection: A systematic approach," *IEEE Access*, vol. 8, pp. 171796–171819, 2020, doi: 10.1109/ACCESS.2020.3024198.

- [73] N. Memon and H. L. Larsen, "Investigative Data Mining Toolkit: A Software Prototype for Visualizing, Analyzing and Destabilizing Terrorist Networks," *Vis. Netw. Inf.*, pp. 1–24, 2006.
- [74] A. Luo, "What is content analysis and how can you use it in your research?," 2019. <https://www.scribbr.com/methodology/content-analysis/> (accessed Jun. 03, 2021).
- [75] A. S. Beshiri and A. Susuri, "Dark Web and Its Impact in Online Anonymity and Privacy: A Critical Analysis and Review," *J. Comput. Commun.*, vol. 07, no. 03, pp. 30–43, 2019, doi: 10.4236/jcc.2019.73004.
- [76] R. Zheng, Y. Qin, Z. Huang, and H. Chen, "Authorship analysis in cybercrime investigation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2665, pp. 59–73, 2003, doi: 10.1007/3-540-44853-5\_5.
- [77] M. Jordan, D. M. Blei, A. Y. Ng, and J. B. Edu, "Latent Dirichlet Allocation Sampling and Bayesian inference View project EM and optimization algorithms in statistical models View project Latent Dirichlet Allocation Michael I. Jordan," 2003. Accessed: Aug. 04, 2020. [Online]. Available: <https://www.researchgate.net/publication/221620547>.
- [78] Y. Zhou, J. Qin, E. Reid, G. Lai, and H. Chen, "Studying the presence of terrorism on the web," 2005, p. 402, doi: 10.1145/1065385.1065505.
- [79] H. Chen, "Dark Web Forum Portal," Springer, New York, NY, 2012, pp. 257–270.
- [80] A. Panchenko, A. Mitseva, M. Henze, F. Lanze, K. Wehrle, and T. Engel, "Analysis of Fingerprinting Techniques for Tor Hidden Services," 2017, doi: 10.1145/3139550.3139564.
- [81] A. A. AlQahtani and E. S. M. El-Alfy, "Anonymous connections based on onion routing: A review and a visualization tool," *Procedia Comput. Sci.*, vol. 52, no. 1, pp. 121–128, 2015, doi: 10.1016/j.procs.2015.05.040.
- [82] H. Prunckun, "Scientific Methods of Inquiry for Intelligence Analysis," Rowman & Littlefield, Sep. 05, 2014. <https://www.amazon.com.tr/Scientific-Methods-Inquiry-Intelligence-Analysis/dp/1442224320> (accessed Jun. 03, 2021).

# Augmented Artificial Neural Network Model for the COVID-19 Mortality Prediction: Preliminary Analysis of Vaccination in Turkey

 Sena Kır<sup>1</sup>,  Elif Elçin Günay<sup>2</sup>

<sup>1</sup> Department of Industrial Engineering, Sakarya University, Sakarya, 54050, Turkey; senas@sakarya.edu.tr; (090)264295-5965

<sup>2</sup> Department of Industrial Engineering, Sakarya University, Sakarya, 54050, Turkey; ekabeloglu@sakarya.edu.tr;

Received 22 September 2021; Accepted 26 January 2022; Published online 30 April 2022

## Abstract

The spread and severity of coronavirus disease 2019 (COVID-19) have a severe impact on our lives, such that over 5.8 million lives have been lost since it has been first emerged. Although prediction of the COVID-19 mortality may be inevitably accompanied by uncertainty, it is helpful for health politicians and public health decision-makers to take proper precautions to diminish the pandemic's severity. Therefore, this study proposed a mortality prediction model for the deaths that occur on-day, lag 1 day, lag 7 day, and lag 14 day in Turkey, considering 16 variables under four categories as follows: (i) severity of the disease, (ii) vaccination policy as a preventive strategy, (iii) exposure duration in society, (iv) time series impact. The developed Augmented-Artificial Neural Network (ANN) model took advantage of Auto-Regressive Integrated Moving Average (ARIMA) and ANN models to capture the linear and nonlinear components of the mortality. The proposed model was able to predict mortality with the lowest error compared to ARIMA and ANN models. A set of experiments was designed to reveal the impact of each responsible category on mortality. In the experimental study, it was observed that the impact of four categories from highest to the lowest importance on prediction performance were exposure duration in society, vaccination policy, severity of disease, and time series, respectively. According to these results, new virus-fighting policies can be developed, and the existing model can be used as a simulation tool with the new data to be obtained.

**Keywords:** ANN, ARIMA, coronavirus, vaccination, estimation

## 1. Introduction

Starting in December 2019, the coronavirus 2019 (COVID-19) pandemic has been a global threat all around the world. According to the World Health Organization (WHO), there have been over 400 million confirmed COVID-19 cases, including nearly 5.8 million deaths [1]. These numbers have been increasing, which continue to affect our lives, education, and economies severely day by day. To diminish the burden of the pandemic, researchers and scientists put a tremendous amount of effort into how to prevent and treating the pandemic [2]. From this aspect, it is evident that predicting the spread of COVID-19 and its impacts contributes to understanding the adversity of the pandemic and developing effective preventive public health emergency strategies for authorities to limit the disease spread promptly [3]. Also, the medical systems can be designed to be more robust to patient overflows, illness, and deaths considering the predictions of the spread of COVID-19 [4].

Estimating the future trend of the COVID-19 and its impact (e.g., hospitalization, mortality, and demand for ICU beds) has been the focus of recent studies given in Table 1. Due to the dependency of epidemic diseases on many different factors and uncertainties [5], the time series modeling approach is quite favorable. Time-series methods help establish a valid prediction model when there is insufficient data to explain the relationship between the dependent and independent variables [6]. Especially, ARIMA [5] and Artificial Neural Network (ANN) models [7] took overwhelming attention because of their capability to work with noisy, complex, and incomplete data.

Table 1 Approaches employed to predict the impact of COVID-19

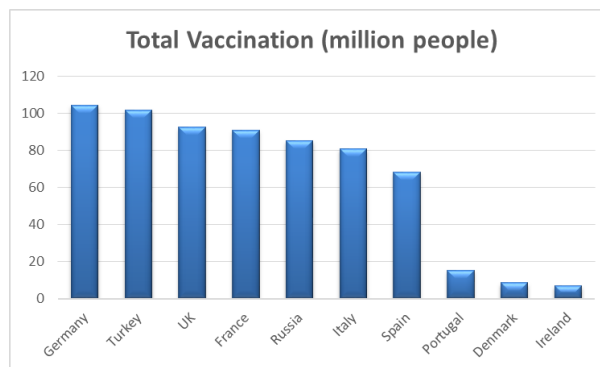
Source	Prediction method/approach	Countries	Dependent variables	Independent variable
[14]	SEIR	China	The inflection point, possible ending time, and total infected cases	Confirmed infected cases, recovered cases latent time, quarantine time, and effective reproduction number
[15]	ARIMA and NARNN	India	Infected cases	Confirmed cases
[16]	Several machine learning models	Italy, the US, France, and the UK	Infected cases, the spread of the virus (growth rate), and the mortality rate	Weather variables and census features
[17]	Logistic function, Hill function, Gompertz function, and LSTM	China and Netherlands	Infected cases	Population size, infected case, static traffic network, and dynamic traffic network
[18]	PNN, RBFNN, and GRNN	India	Deaths (mortality rate)	Deaths and confirmed cases
[19]	MLPICA-ANFIS	Hungary	Infected cases and the mortality rate	Odd days and even days' cases, and mortality rates
[20]	ARIMA, and ANN	India	Infected cases	Confirmed cases
[21]	Gompertz model, Logistic model, and ANN	Mexico	Infected cases	Confirmed cases
[22]	RNN-LSTM variants	India	Daily and weekly infected cases	Confirmed cases
[23]	Machine learning techniques (EN, PCR, PLSR, KNN, RT, RF, GBM, ANN)	US metropolitan counties	Daily and cumulative infected cases	Daily and cumulative confirmed cases, county-level demographic, environmental, mobility, and time-series data
[24]	ANN	Pakistan	Deaths, recovered, and infected cases	Date (day number)
[25]	ANFIS improved by FPA-SSA	China	Daily infected cases	Confirmed cases
[26]	ARIMA (with Hannan, Rissanen algorithm)	India, US, Brazil, Russia, and Spain	Total infected cases	Total confirmed cases
[27]	ARIMA	Italy, Spain, and Turkey	Deaths and infected cases	Confirmed cases
[28]	ARIMA, NARNN, and LSTM	Denmark, Belgium, Germany, France, United Kingdom, Finland, Switzerland, and Turkey	Cumulative infected cases	Cumulative confirmed case
[29]	Recursive based model, Boltzmann function-based model, and Beesham's model	Iran and Turkey	Cumulative deaths and infected cases	Cumulative deaths and confirmed cases
[30]	SEIR	Turkey	Infected cases, intensive care needs, hospitalizations, and deaths	Average and age-specific Infection Fatality Ratio (IFR), infection rate, census data, age-specific hospitalization, and intensive care ratios,
[31]	Gene expression programming	India	Deaths and infected cases	Deaths and confirmed cases
[32]	ARIMA, and NARNN	Egypt	Cumulative infected cases	Cumulative confirmed cases
[4]	Bayesian optimization-based LSTM	USA, Switzerland, India, Slovakia, Russia, Uruguay, Greenland, Malta, Denmark, Brazil, Zimbabwe, Japan, Mexico, Germany, and Norway	Deaths, recovered, infected cases, and country risk classification	Time data, country, confirmed cases, recovered cases, deaths, and weather data

(ANFIS: adaptive network-based fuzzy inference system, ANN: artificial neural network, ARIMA: autoregressive integrated moving average, EN: elastic net model, FPA: flower pollination algorithm, GBM: gradient boosted tree models, GRNN: generalized regression neural network, KNN: k-nearest neighbors regression model, LSTM: long-short term memory, MLPICA: multi-layered perceptron-imperialist competitive algorithm, NARNN: nonlinear autoregressive neural network, PCR: principal components regression model, PLSR: partial least squares regression model, PNN: probabilistic neural network, RBFNN: radial basis function neural network, RF: random forests model, RNN: recurrent neural network, RT: regression tree model, SEIR: susceptible exposed infectious recovered model, SSA: salp swarm algorithm,)

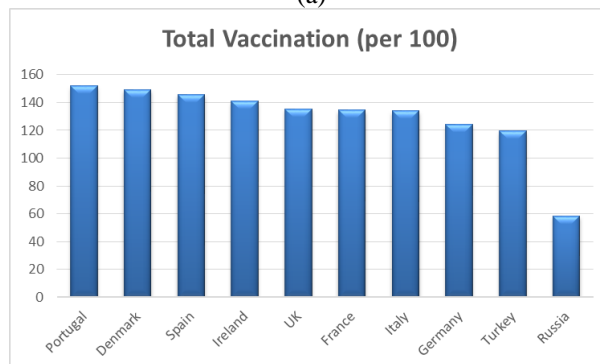
The literature review papers [8-13] are helpful to have more information about the approaches employed to detect and predict the spread of COVID-19. In one of the recent reviews, [33] criticizes state-of-the-art prediction approaches providing generic tools and neglecting major factors, such as social distancing, test rate, and vulnerability issues related to chronic diseases, which significantly affect the prediction accuracy. Ignoring the impact of those factors can deviate the predictions dramatically. In this manner, vaccination is a highly influential factor that affects the spread of the COVID-19 because it is considered a practical way to develop herd immunity [34]. Otherwise, establishing herd immunity via infections may cause a vast number of deaths, as experienced in Sweden [35]. As more people are vaccinated, fewer individuals are expected to be infected, and the severity of the disease is expected to diminish [34]. Predictions before vaccination started are insufficient to accurately predict the spread and impact of outbreaks. Therefore, there is a need to accommodate vaccination impact in the mortality prediction models.

Considering the impact of vaccination, we propose an augmented-ANN approach for COVID-19 mortality prediction for the deaths that occur on-day, lag 1 day, lag 7 day, and lag 14 day. The augmented-ANN utilizes the two most preferred techniques, ANN and ARIMA. Due to the complex and random structure of the epidemic disease, mortality data may include both linear and nonlinear components. Therefore, the augmented-ANN provides a better fit to capture the linear part by ARIMA and the nonlinear part by ANN. Taking into account the suggestion of the previous studies [33, 23], we accommodate the following significant factors that impact the prediction accuracy: (i) severity of the disease, (ii) vaccination policy as a preventive strategy, (iii) time series, (iv) exposure duration in society.

Since the middle of January 2021, Turkey has followed serious vaccination policies successfully, which lead the country to take second place (see Figure 1a) in terms of total vaccination and ninth place in total vaccination per hundred (see Figure 1b) in Europe. Besides, to the best of our knowledge, there is no study on Turkey's mortality prediction after the vaccination has been initiated; instead, mortality prediction research has focused on countries in Europe and China.



(a)



(b)

Figure 1 Vaccination rates and numbers of top 10 European countries [36]



Our study contributes to the literature in the following aspects: i) integration of ARIMA and ANN to capture both linear and nonlinear components of the data, ii) accommodation of the vaccination rates and numbers to better explain the mortality, iii) prediction of mortality in Turkey where the nationwide vaccination program against COVID-19 has been present. The remaining sections are organized as follows. Section 2 presents the precautions taken to fight the epidemic in Turkey and discusses the responsible factors for COVID-19 mortality. Section 3 presents the ARIMA, ANN, and augmented-ANN models. Section 4 demonstrates the results and the advantage of the proposed model and examines the impact of responsible factors on mortality. Section 5 concludes the study by discussing the limitations and suggesting possible future extensions.

## 2. Data Pretreatment: Turkey Case

In addition to various control measures, such as the general use of face mask and travel bans, stay-at-home policies has been quite strictly applied in Turkey to mitigate the spread of COVID-19. Especially when the rate of increase in COVID-19 cases is high, citizens who violate measures have been subjected to penalties or fines. Various precautions have been taken considering the spread of the virus since March 13, 2020, when the first case was seen in Turkey. These precautions include partial lockdowns (for weekends or specific dates and times), lockdown over 65 and under 18, transition to distance education, full lockdown, etc. By lockdowns, the social distance has been maintained; therefore, the increase in COVID-19 cases and mortality has been reduced.

Policies, such as weekend lockdowns and lockdowns for over the age of 65 and under 20, have restricted community movements. The reflection of these policies can be observed by the Google community mobility report. The report includes the trends on community movements over time by location across six categories: retail and recreation, grocery and pharmacy, parks, transit stations, workplaces, and residential. Google mobility data shows the visitors' relative change in the categorized locations compared to baseline days, representing a typical day. The baseline day is given as the median value over the five weeks from January 3 to February 6, 2020 [37]. The community mobility graph for Turkey between January 14, 2021, and July 03, 2021, as discussed in this study, is shown in Figure 2.

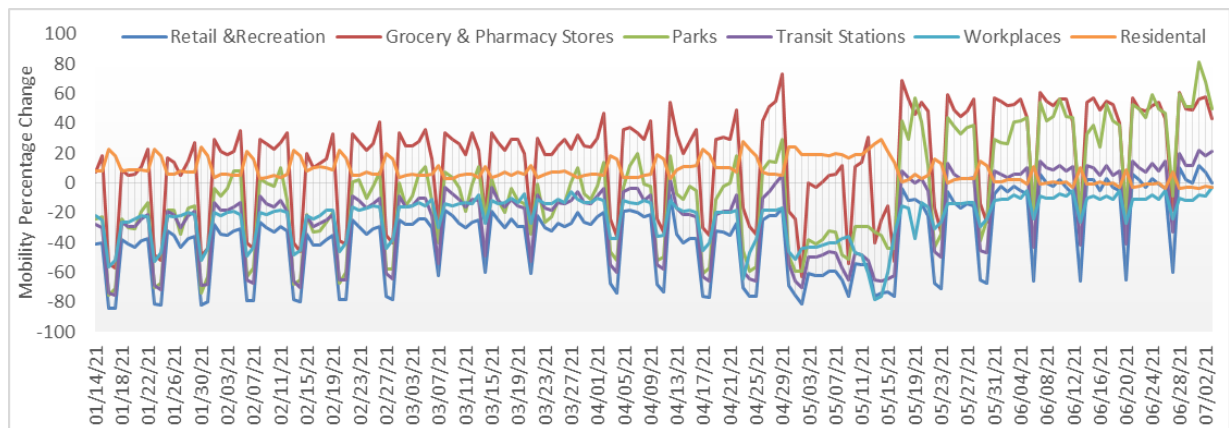


Figure 2 Mobility percentage change in Turkey between January 14 and February 19, 2021

In Figure 2, for example, the weekend lockdowns strictly encourage people to stay at home; thus, generate a decreasing pattern for the weekends followed by an increasing pattern for the weekdays in all the mobility categories except for residential, as expected. Similarly, the effect of full lockdown between April 29 and May 17 on mobility changes can also be seen from the graph in Figure 2.

Considering the COVID-19 data in Turkey and relevant studies from literature, factors affecting mortality can be summed up following four categories: (i) severity of the disease [14, 4], (ii) vaccination policy as a preventive strategy [34], (iii) exposure duration in society [23], (iv) time series [23]. First,

the severity of the disease is associated with the number of severe patients since the number of deaths will be affected by the critical health status of the individuals. The rate of COVID-19 pneumonia is another factor that represents the severity of the disease because pneumonia fills air cells in the lungs with fluid and is deadly [38]. Moreover, the number of patients recovering is incorporated; as more people recover, fewer deaths occur. Additionally, the severity of the disease is also related to late diagnosis. Therefore, the number of COVID-19 PCR tests daily is considered as a factor affecting mortality. We integrated vaccination data as a preventive strategy to the model as a second category. Starting January 14, 2021, to maximize the vaccination's efficiency, Turkey has set a vaccination plan that prioritizes individuals under high risks, such as healthcare workers, older people (above 65 years old), handicapped, and the staff of the nursing homes. Then, gradually the vaccination plan is extended to include all the individuals 18 years old and above. Sinovac and Biontech (since April 12) vaccines are used, and each individual receives two doses approximately 3-4 weeks apart. Three months after the second dose, the third (booster) dose is recommended. In terms of vaccination data, total vaccinations, people fully vaccinated (received two doses), the ratio of total vaccinations to population, and the ratio of people fully vaccinated to population are determined as input variables. Third, we also integrated exposure duration in society to the prediction model by community mobility report retrieved by Google. Finally, because of weekend lockdown policies, mobility has different structures during weekends and weekdays. Thus, the status of the day as weekday or weekend is incorporated as an indicator of the movement trends. All 15 variables considered affecting the mortality in Turkey, the sources, and the types of data were presented in Table 2.

Table 2 Input variables, data sources, periods, and types in ANNs

Sym.	Variable	Source	Data Period	Data Type
X <sub>1</sub>	Pneumonia rate in patients	TR Ministry of Health <sup>a</sup>		Continuous
X <sub>2</sub>	# of severe patients	TR Ministry of Health <sup>a</sup>		Discrete
X <sub>3</sub>	# of recoveries	TR Ministry of Health <sup>a</sup>		Discrete
X <sub>4</sub>	# of PCR test	TR Ministry of Health <sup>a</sup>		Discrete
X <sub>5</sub>	Total vaccinations	Our World in Data <sup>b</sup>		Discrete
X <sub>6</sub>	People fully vaccinated	Our World in Data <sup>b</sup>		Discrete
X <sub>7</sub>	The ratio of total vaccinations to population	Our World in Data <sup>b</sup>		Continuous
X <sub>8</sub>	The ratio of people fully vaccinated to population	Our World in Data <sup>b</sup>	01/14/2021- 07/03/2021	Continuous
X <sub>9</sub>	Retail and recreation percent change	Google <sup>c</sup>		Continuous
X <sub>10</sub>	Grocery and pharmacy percent change	Google <sup>c</sup>		Continuous
X <sub>11</sub>	Parks percent change	Google <sup>c</sup>		Continuous
X <sub>12</sub>	Transit stations percent change	Google <sup>c</sup>		Continuous
X <sub>13</sub>	Workplaces percent change	Google <sup>c</sup>		Continuous
X <sub>14</sub>	Residential percent change	Google <sup>c</sup>		Continuous
X <sub>15</sub>	Weekend or not	-		Categorical

<sup>a</sup> Turkish Republic Ministry of Health, 2021., <sup>b</sup> Our World in Data, 2021b., <sup>c</sup> Google, 2021. [39, 40, 37]

Taking into account 15 variables in Table 2, we attempt to predict mortality as an output variable. There is a lag between symptom severity and death [41], and an average of 7- day ICU stay is reported in the literature [42]. Hence, we predicted mortality as the number of deaths in four ways: on day, lag 1 day, lag 7 day, and lag 14 day in ANN models.

### 3. Methods

Three different prediction models were presented in this section considering the disease situation after vaccination started in Turkey. In the first model, time series analysis was performed with ARIMA in Section 3.1. Next, ANN models with different input variables related to mortality were created in Section 3.2. Last, to improve the prediction performance by handling complex data structures, we proposed an Augmented-ANN approach in Section 3.3.

### 3.1 ARIMA Model

Among many time series analysis models, ARIMA is preferred in different applications due to its advantages over other stochastic models, such as its superior forecasting capability and ability to provide greater information concerning the time-related change [43]. Particularly, when the explanatory variables describing the prediction variable are limited or unsatisfactory, the ARIMA modeling approach is instrumental since the only data input is the previous data records.

ARIMA models consist of three components: Autoregressive (*AR*), moving average (*MA*), and integrated (*I*). *AR* component indicates that the future values of the data vary over its past values, while *MA* component reflects the purely random regression errors. Also, the last component *I* refers to the differencing period used to forecast future values [6]. These three components are usually represented in the form of ARIMA( $p, d, q$ ) where  $p$  refers to the order of *AR* polynomials,  $d$  is the degree of difference, and  $q$  is the order of *MA* polynomials [44]. Given in a time series ( $Z_t$ ), ARIMA models can be in the form of AR( $p$ ), MA( $q$ ), ARMA( $p, q$ ), or ARIMA ( $p, d, q$ ) depending on the values of  $p$ ,  $d$ , and  $q$ . Equations (1-3) show the general formula of AR ( $p$ ), MA ( $q$ ), and the ARMA( $p, q$ ) respectively where  $\phi$  and  $\theta$  are autoregressive and moving average parameters,  $\alpha$  is a constant, and  $\varepsilon$  is the random errors [5].

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + \varepsilon_t \quad (1)$$

$$Z_t = \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2)$$

$$Z_t = \alpha + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (3)$$

In Box-Jenkins methodology [45], building an ARIMA model consists of three stages: (i) model identification, (ii) parameter estimation, and (iii) diagnostic testing. The identification process includes applying transformation algorithms to convert data to the stationary form. Stationary is a pre-requirement for ARIMA models, which refers to a constant statistical characteristic (mean, variance, autocorrelation structure) over time. In this stage, the data should be tested in terms of stationarity. Augmented Dickey-Fuller (ADF) test is a stationarity test used in addition to a time-series plot. When data is in a nonstationary form due to trend or heteroscedasticity, logarithmic transformation, power transformation, or differencing should be applied to stabilize the data. After a tentative model is defined, the following stage includes the estimation of the autoregressive and the moving average parameters presented in a way that the overall error is minimized. The parameters are decided according to the lowest Akaike Information Criterion (AIC), which is presented in Equation (4), where  $n$  is the data number of observations,  $r=p+q+1$  and  $\hat{\sigma}_a^2$  is a maximum likelihood prediction.

$$AIC_{p,q} = \frac{-2 \ln(\text{maximized likelihood}) + 2r}{n} \approx \ln(\hat{\sigma}_a^2) + r \frac{2}{n} + \text{constant} \quad (4)$$

The final stage includes the performance evaluation of the forecasting. Commonly used performance evaluation criteria in the literature are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) presented as follows in Equations (5-7).

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (z_t - \hat{z}_t)^2} \quad (5)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |z_t - \hat{z}_t| \quad (6)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{z_t - \hat{z}_t}{z_t} \right| \quad (7)$$

### 3.2 Artificial Neural Networks

The ANNs are nature-inspired computational modeling tools that simulate human learning to predict future data and make decisions. The ANNs are widely accepted in many disciplines for modeling complex real-world problems [46]. Because first, ANN models have flexible nonlinear function mapping capability that helps approximate the variable in interest with high accuracy [47, 48]. Second, as a data-driven approach, ANN can capture the uncertain hidden relationships between variables [49]. Third, ANNs have an adaptive structure that provides more robust generalizable models for nonstationary environments. However, in contrast to the advantages mentioned earlier, the performance of the ANN compared to the linear model is inconsistent. Some researchers reported that the linear structure of the data without much disruption might be the reason for outperforming linear time series. Therefore, instinctively developing ANN models for any type of data does not guarantee good performance.

In the literature, feed-forward ANNs were preferred mostly for a wide range of prediction applications [50], and also multi-layered perceptron feed-forward ANNs were found favorable for application in predicting deadly infectious disease outbreaks [46]. In this paper, a two-layer feed-forward backpropagation neural network architecture was preferred to predict mortality. The ANN model has 15 input variables as presented in Table 1, two hidden layers, and one output. The same ANN model is replicated for four outputs: as on day, lag 1 day, lag 7 day and lag 14 day. The data set was decomposed into three sets: 80% for training, 10% for testing, and 10% for validation. The tangent sigmoid function was preferred as a neural transfer function in the hidden layer, and the Purelin function is employed in the output layer. The neuron numbers of hidden layers were determined as 8 and 10. Levenberg-Marquardt (LM), Bayesian Regularization (BR), and Scaled Conjugate Gradient (SCG) backpropagation algorithms were deployed as training algorithms.

Preferred training algorithms have different superior features across each other. LM algorithm, which has high accuracy prediction recognition rate in supervised learning networks [51], is known to be a fast and stable convergence training algorithm for moderate-sized feed-forward neural networks (up to several hundred weights) [52]. Based on the success of conjugate gradient methods in solving large-scale unconstrained optimization problems, the SCG algorithm is usually efficient for large multilayer networks, which have more than a thousand or more weights and biases [53]. The BR algorithm can result in good generalization for small or noisy data sets. It is quite challenging to collect all COVID-19 data with complete accuracy. Therefore, assuming that the obtained data is noisy, the BR training algorithm is expected to give better results than other algorithms for the COVID-19 data set. Besides, the BR algorithm updates the squared weights  $E_w$  and squared errors  $E_D$  by regularization parameters  $\alpha$  and  $\beta$ . It limits the size of network parameters by regularization to create a network with better generalization ability. Regularization forces the network to keep the weight and bias values at smaller values; this helps the network reduce overfitting and capture noise [54]. For detailed information on the BR training algorithm, see Hagan et al. [54].

### 3.3 Augmented-ANN

Real-world problems are seldom pure linear or nonlinear and usually merge both structures, implying that neither a single technique is sufficient for modeling [55]. Besides, due to their complex structure, there has been agreed-upon prediction literature that no universal model acquires distinct patterns evenly. Both theoretical and empirical findings indicated that an effective way to improve their predictive performance is particularly combinations of quite different models [6]. Hence, to adequately capture both the linear and nonlinear components of the data, the ARIMA and ANN models for linear and nonlinear data structure can be combined in one model.

COVID-19 mortality data is complex by nature since neither all the responsible factors nor their impact can be truly captured yet. Various studies utilized different factors, such as age, gender, and chronic disease history, to understand the spread and consequences of the COVID-19 pandemic. However, there are still many unknowns in terms of underlying factors and their impact. Besides, a concern about the nonlinear structure of COVID-19 data was mentioned by [56]. To adequately acquire different patterns in the data, we propose an Augmented-ANN model. The integration of ANN and ARIMA allows us to provide a better fit for the complex mortality data. The ARIMA model captures the linear relationships between present and prior mortality values, while ANN captures the nonlinear relationships.

The framework of the proposed prediction methodology is presented in Figure 3. The methodology initiates by creating the best performing ARIMA ( $p,d,q$ ) model. The outcomes of the ARIMA model are considered as a baseline prediction that has been built upon the linear correlation structures of the current and past deaths. Then, in addition to other responsible factors in Table 2 ( $X_1-X_{15}$ ), the ARIMA predictions ( $X_{16}$ ) are included as input variables to the ANN model for predicting mortality.

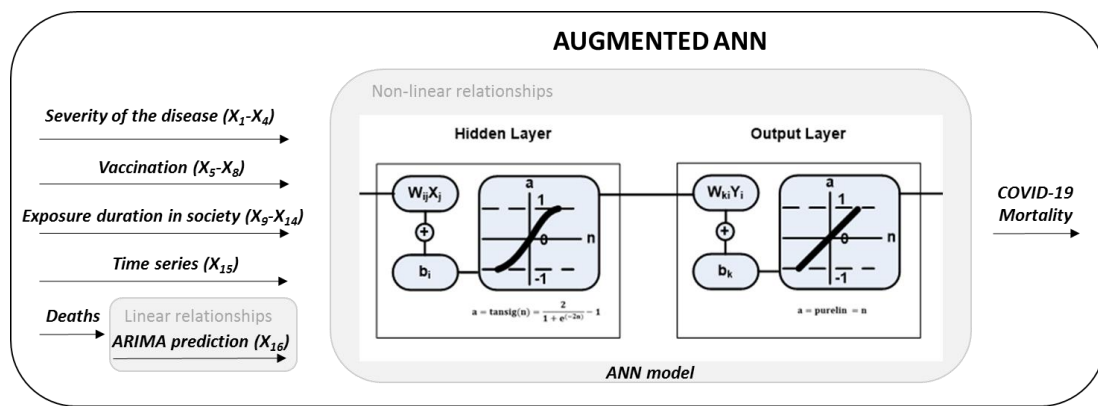


Figure 3 The framework of the proposed Augmented-ANN models

#### 4. Results and Discussion

This section discusses the results of the ARIMA, ANN, and Augmented-ANN models employed to predict the number of deaths under subsections Section 4.1 to 4.3, respectively. The modeling performances of the three models were evaluated in Section 4.4 to demonstrate the appropriateness and effectiveness of Augmented-ANN. Besides, a sensitivity analysis was conducted to identify the impact of each factor on mortality prediction.

##### 4.1 ARIMA Results

The original mortality time series data were tested by Augmented Dickey-Fuller (ADF) test for stationarity. After the stationary assumption was satisfied ( $p= 0.0269$ ), the parameters of the ARIMA model were determined based on the smallest AIC value that was chosen among the possible combinations of values of 0, 1, and 2 for  $p$  and  $q$ , respectively. Accordingly, AR (1) model with “1541.503” AIC value was the best model to represent the mortality data. It is worth to mention that AR (1) is a special version of ARIMA model with  $p=1, d=0, q=0$  values. The estimations of the AR(1) model were presented in Table 3.

Table 3 AR (1) model parameters

	Coefficient	Standard error	z-Statistic	p-Value	95% CI for the coefficient	
					Lower limit	Upper Limit
$\phi_1$	0.998	0.0056	178.32	0.000	0.9870	1.0089

## 4.2 ANN Results

The parameters such as type of training algorithms, the number of hidden neurons affect the performance of ANNs. Considering their impact, three training algorithms (LM, BR, and SCG) and two levels of hidden layers (8 and 10) were included in the experimental design. A total of 24 experiments (i.e., 3 training algorithms  $\times$  2 levels of hidden layers  $\times$  4 outputs) were run based on full factorial design. Performance comparisons of all ANN models in terms of MAE, RMSE, and MAPE were performed. However, due to the conflictive scores of performance parameters (MAE, MSE, and RMSE) across each experiment, the performance parameters were normalized to the range of 0-1 and the overall score is calculated by weighting all parameters equally ( $w_{MAE} = w_{MSE} = w_{RMSE} = 0.33$ ). The model with the lowest overall score shows the minimum error, thus the best performance. Table 4 presents the results of ANN experiments.

Table 4 ANN models in detail

Training Alg.	# of Hidden Neurons	Output	# of output data	Performance Measures			Overall Score	Correlation Coefficients			
				MAE	RMSE	MAPE		Training	Validation	Test	All
LM	10	On day	170	6.346	8.880	0.049	0.271	.997	.992	.994	.997
LM	8			3.369	6.584	0.027	0.118	1	.994	.985	.998
BR	10			2.362	4.563	0.022	0.043	1	-	.985	.998
BR	8			3.174	6.602	0.027	0.115	1	-	.994	.999
SCG	10			13.553	17.509	0.116	0.759	.983	.987	.981	.983
SCG	8			13.098	16.740	0.103	0.693	.988	.978	.953	.984
LM	10	Lag 1 day	169	4.360	9.486	0.037	0.228	1	.987	.981	.995
LM	8			4.802	8.441	0.042	0.222	.999	.997	.971	.996
BR	10			2.574	9.516	0.016	0.144	1	-	.968	.995
BR	8			3.182	5.109	0.029	0.086	.999	-	.992	.999
SCG	10			9.449	14.163	0.071	0.495	.992	.980	.983	.989
SCG	8			8.698	12.548	0.065	0.432	.993	.985	.991	.991
LM	10	Lag 7 day	163	4.357	10.712	0.035	0.252	1	.987	.959	.994
LM	8			4.680	8.178	0.036	0.196	.999	.989	.989	.996
<b>BR</b>	<b>10</b>			<b>2.121</b>	<b>3.707</b>	<b>0.020</b>	<b>0.013</b>	<b>1</b>	-	<b>.991</b>	<b>1</b>
BR	8			2.930	4.947	0.024	0.065	1	-	.992	.999
SCG	10			8.804	12.473	0.063	0.422	.992	.988	.989	.991
SCG	8			11.190	14.587	0.087	0.572	.989	.990	.982	.988
LM	10	Lag 14 day	156	6.579	9.584	0.051	0.294	.996	.991	.992	.995
LM	8			3.884	10.603	0.028	0.220	1	.975	.972	.994
BR	10			4.330	8.534	0.027	0.172	.999	-	.980	.996
BR	8			4.351	7.932	0.028	0.476	.990	-	.974	.996
SCG	10			10.171	14.049	0.076	0.514	.993	.952	.982	.989
SCG	8			9.710	12.866	0.072	0.471	.990	.992	.992	.991

In Table 4, the best performing ANN model was indicated as italic and bold. Among all outputs, the performance of the lag 7 day mortality prediction (MAE= 2.121, RMSE= 3.707, MAPE= 0.02, overall score= 0.013) was the highest. Our finding complies with the literature stating the presence of several days lag between severity of disease and the number of deaths [41, 42]. Additionally, it was also observed that the BR algorithm succeeded a better fit in comparison to other training algorithms for all the outcomes. In contrast, the performance of the SCG algorithm was the worst. This might be because of the BR algorithm's superiority for modeling the noisy structure of COVID-19 mortality data. Moreover, it was observed that a higher number of neurons were effective to improve the prediction accuracy, and 10 hidden neurons outperformed 8 hidden neurons for on-day, lag 7 day, and lag 14 day predicted death cases.

## 4.3 Augmented-ANN Results

This subsection reports the results of the Augmented-ANN in which the ARIMA predictions were included as an input to strengthen ANNs. Because the performance of the Augmented-ANN also

depends on the training algorithm and the number of hidden neurons, as mentioned in Section 4.2, we repeated the same set of experimental designs for the augmented-ANN models. The performance of each different Augmented-ANN model was presented in Table 5.

Table 5 Augmented- ANN models in detail

Training Alg.	# of Hidden Neurons	Output	# of output data	Performance Measures			Overall Score	Correlation Coefficients			
				MAE	RMSE	MAPE		Training	Validation	Test	All
LM	10	On day	170	3.862	7.622	0.032	0.448	1	.997	.983	.997
LM	8			3.775	7.070	0.033	0.433	.999	.996	.981	.997
BR	10			0.938	3.150	0.007	0.119	1	-	.995	.999
BR	8			2.638	5.259	0.023	0.304	1	-	.979	.999
SCG	10			7.062	10.264	0.053	0.719	.994	.999	.989	.994
SCG	8			7.989	11.148	0.060	0.806	.983	.993	.994	.993
LM	10	Lag 1 day	169	0.811	1.138	0.006	0.060	1	1	1	1
LM	8			0.488	0.800	0.004	0.030	1	1	1	1
<b>BR</b>	<b>10</b>			<b>0.171</b>	<b>0.542</b>	<b>0.001</b>	<b>0.000</b>	<b>1</b>	-	<b>1</b>	<b>1</b>
BR	8			0.271	0.598	0.002	0.008	1	-	1	1
SCG	10			5.323	6.810	0.043	0.528	.998	.998	.994	.997
SCG	8			2.711	3.600	0.024	0.269	.999	.999	.998	.999
LM	10	Lag 7 day	163	3.946	8.739	0.030	0.471	.999	.993	.962	.996
LM	8			3.974	7.974	0.030	0.452	1	.992	.981	.997
BR	10			3.073	10.982	0.025	0.472	1	-	.940	.993
BR	8			3.706	11.591	0.023	0.500	1	-	.954	.992
SCG	10			9.638	13.550	0.071	0.973	.993	.985	.973	.990
SCG	8			9.961	13.297	0.074	0.994	.989	.994	.992	.990
LM	10	Lag 14 day	156	5.563	12.402	0.045	0.685	.999	.949	.974	.991
LM	8			5.109	10.625	0.033	0.571	.999	.984	.970	.994
BR	10			1.367	5.548	0.007	0.196	1	-	.982	.998
BR	8			2.984	11.810	0.023	0.481	1	-	.935	.992
SCG	10			9.042	12.361	0.066	0.898	.993	.991	.976	.991
SCG	8			9.384	12.685	0.070	0.936	.992	.982	.994	.991

In Table 5, the highest prediction accuracy was accomplished for lag 1 day death prediction with 0.171 MAE, 0.542 RMSE, 0.001 MAPE, and 0.000 overall score. The best performing Augmented-ANN model was succeeded with BR training algorithm and 10 hidden neurons.

#### 4.4 Performance Comparison

The performance of the ARIMA, ANN, and Augmented-ANN are compared in Table 6. It is important to mention that in Table 6 ARIMA model – AR(1) – presents the results of time series and solely considers the linear relationship between current and previous COVID-19 deaths. The ANN row reports the performance of the ANN, considering the impact of different inputs on mortality prediction. Finally, the performance of the Augmented-ANN is presented in the last row. Based on the results, the lowest MAE, RMSE, and MAPE indicated that the Augmented-ANN predicts mortality with the lowest error. Confirmed deaths and predictions of on-day, lag 1 day, lag 7 day, and lag 14 day are presented in Figure 4 for Augmented-ANN.

Table 6 Comparison of the best performing models

Methods	MAE	RMSE	MAPE
AR(1)	8.670	13.545	0.064
ANN	2.121	3.707	0.020
Augmented-ANN	0.171	0.542	0.001



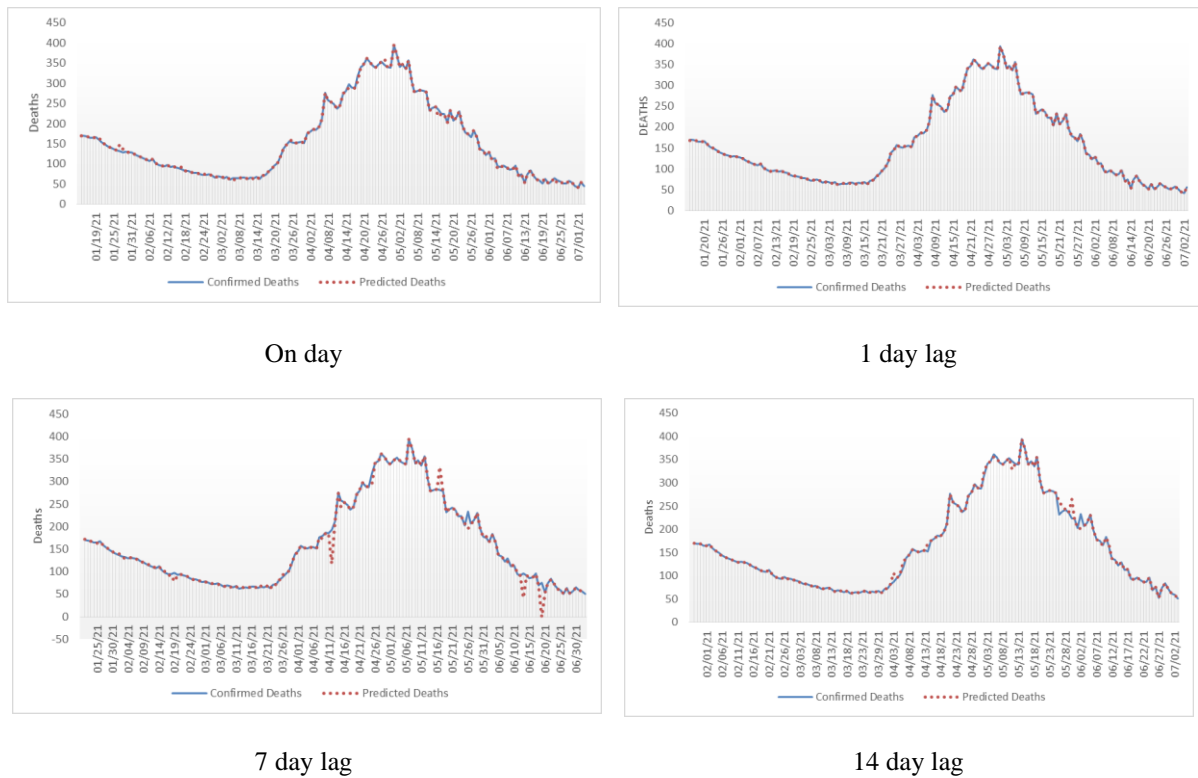


Figure 4 Confirmed and predicted deaths for best performing Augmented-ANN model

#### 4.5 Impact of model parameters on prediction performance

The prediction performance of the model intends to improve as the number of variables increase. However, the question is to identify which input variables have the strongest association with mortality. In this section, we performed a sensitivity analysis to discuss the impact of the input variables on the prediction performance of the Augmented-ANN model. Four new Augmented-ANN models were created considering the categories of the input variables, namely severity of the disease variables ( $X_1-X_4$ ), vaccination policy as a preventive strategy variables ( $X_5-X_8$ ), exposure duration in society variables ( $X_9-X_{14}$ ), and time series variable ( $X_{15}$ ). Each category's impact was explored by dropping the associated variables while remaining the other three categories in the prediction model. The procedure is repeated for all possible combinations of input variables (categorized in four subsets) to examine the weakest predictors for mortality. The results of the four models were presented in Table 7 in addition to the base model so that their influence on mortality could be discussed.

Table 7 Impact of each category on explaining the mortality for Augmented-ANN model

Impact of category	Variables added	Variables dropped	MAE	MAPE	RMSE	Overall score	Coefficient of determination ( $R^2$ )
Base model	$X_1-X_{16}$	-	0.171	0.001	0.542	0.236	0.999
Time series	$X_1-X_{14}, X_{16}$	$X_{15}$	0.287	0.002	0.472	0.529	1.000
Severity of disease	$X_5-X_{16}$	$X_1-X_4$	0.267	0.002	0.570	0.724	0.999
Vaccination policy	$X_1-X_4, X_9-X_{16}$	$X_5-X_8$	0.278	0.002	0.565	0.787	0.999
Exposure duration in society	$X_1-X_8, X_{15}-X_{16}$	$X_9-X_{14}$	0.338	0.003	0.560	0.963	0.999

According to the results in Table 7, the  $R^2$ - values of all models are above 0.999, which indicates that more than 99.9% of the variation can be explained in COVID-19 mortality with the variables considered. The base model in which all input variables were taken into account outperforms the other models (MAE=0.171, MAPE=0.001, RMSE=0.542, and overall score= 0.236). The second-best model was observed with a MAE of 0.287, a MAPE of 0.002, a RMSE of 0.472, and an overall score of 0.529 after dropping the time series variable. This remarked that the time series had the lowest impact on model

performance, i.e., prediction accuracy. Thus, dropping associated variables resulted in a minimal change in MAE, MAPE, RMSE, and  $R^2$ . The performance of the model was the worst when the variables of exposure duration in the society category were removed. Additionally, dropping vaccination policy variables accounted for higher errors. These two categories (exposure duration in society and vaccination policy) have higher impact than the severity of disease category in mortality prediction. Maintaining social distance, applying stay-at-home orders to limit community movement, and getting vaccinated appear to be a safe solution for community health.

## 5. Conclusion and Limitations

Despite the massive amount of studies available to predict impacts of COVID-19, the research for improving the effectiveness of the prediction has been the focus of the research because of the pandemic's enormous effect on society, the healthcare system, and the economy. Prediction models with increased accuracy assist authorities in taking the complete picture of adverse effects; thus, helping them prompt proactive strategies to fight the pandemic. In this regard, we presented ARIMA, ANN, and augmented-ANN models to predict the number of on-day, lag 1 day, lag 7 day, and lag 14 day COVID-19 deaths in Turkey, considering the data after vaccination has started. ARIMA model encapsulated the linear relationships between current and past data. ANN model accommodated four groups of responsible factors, namely (i) severity of the disease, (ii) vaccination policy, (iii) time series, and (iv) exposure duration in society in the prediction model. Finally, augmented-ANN integrated ARIMA and ANN models to incorporate the ARIMA predictions as a baseline to the established ANN model. Integrating two methods improved prediction accuracy; thus, augmented-ANN achieved the best performance, followed by ANN and ARIMA. The dominance of the ANN models (pure ANN and augmented-ANN) may be attributed to the nonlinear data structure of the COVID-19 mortality.

In order to examine the impact of the responsible factors on mortality, we set up an experimental design and compared the performance of the models by either adding or dropping the responsible categories. Our experiments revealed that the basic model that includes all four categories outperforms the other models. The mobility (integrated through exposure duration in society) has the highest impact on prediction performance so it should not be ignored in the model. Additionally, vaccination policy has the second-highest impact on mortality prediction. On the other hand, dropping time series variables accounted for a minimal change in the error, stating the lowest impact on prediction performance.

Since the accuracy of prediction models plays a crucial role in adopting preventive measures, it is vital to exploit the one with the desirable precision. The results of this research can assist authorities in effectively planning their resources such as ICU bed and staff planning, developing and implementing a shutdown policy that limits the public movement. The model can also be used as a simulation tool to investigate the impact of different regularizations, such as restricting the internal movement, closing nonessential shops, etc.

Even though our study's novelties in terms of incorporating vaccination into the prediction model and integrating ARIMA with ANN to boost accuracy, there are some limitations of the study. First, we could not employ the daily COVID-19 cases in the prediction model since it was not clear whether the daily numbers published by the health ministry of Turkey indicate the number of new "patients" or "cases". Second, the model is based on the current stay-at-home regularizations. The change on these orders will directly influence mobility data, which has the highest impact on mortality. Nevertheless, the prediction model can be updated according to the changes in mobility to provide more precise predictions. In future work, we aim to investigate the effect of various movement restrictions on COVID-19 mortality under the influence of vaccination.

### Conflict of Interest Statement

The authors declare no competing interests.

### Author's Contributions

All authors contributed equally to the study.

## References

- [1] WHO, “World Health Organization COVID-19 Dashboard,” 2021. [Online] Available: <https://covid19.who.int/> [Accessed: 02-Feb-2022]
- [2] A. Hernandez-Matamoros, H. Fujita, T. Hayashi, and H. Perez-Meana, “Forecasting of COVID-19 per regions using ARIMA models and polynomial functions,” *Appl. Soft Comp.*, vol. 96, 106610, 2020.
- [3] S. Zhang, M. Diao, W. Yu, L. Pei, Z. Lin, and D. Chen, “International Journal of Infectious Diseases Estimation of the reproductive number of novel coronavirus (COVID-19) and the probable outbreak size on the Diamond Princess cruise ship: a data-driven analysis,” *Int. J. Infect. Dis.*, vol. 93, pp. 201–204, 2020.
- [4] R. Pal, A. A. Sekh, S. Kar, and D. K. Prasad, “Neural network based country wise risk prediction of COVID-19,” *Appl. Sci.*, vol. 10, no. 18, 6448, 2020.
- [5] Z. Ceylan, “Estimation of COVID-19 prevalence in Italy, Spain, and France,” *Sci. Total Environ.*, vol. 729, 138817, 2020.
- [6] M. Khashei and M. Bijari, “A novel hybridization of artificial neural networks and ARIMA models for time series forecasting,” *Appl. Soft Comp.*, vol. 11, no. 2, pp. 2664-2675, 2011.
- [7] A. Mollalo, K. M. Rivera, and B. Vahedi, “Artificial neural network modeling of novel coronavirus (COVID-19) incidence rates across the continental United States,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 12, 4204, 2020.
- [8] I. E. Agbehadji, B. O. Awuzie, A. B. Ngowi, and R. C. Millham, “Review of big data analytics, artificial intelligence and nature-inspired computing models towards accurate detection of COVID-19 pandemic cases and contact tracing,” *Int. J. Environ. Res. and Public Health*, vol. 17, no. 5, 5330, 2020.
- [9] F. N Khan, A. A. Khanam, A. Ramlal, and S. Ahmad, A review on predictive systems and data models for COVID-19. In *Computational Intelligence Methods in COVID-19: Surveillance, Prevention, Prediction and Diagnosis*. Springer, Singapore, pp. 123-164, 2021.
- [10] S. Lalmuanawma, J. Hussain, and L. Chhakchhuak, “Applications of machine learning and artificial intelligence for COVID-19 (SARS-CoV-2) pandemic: A review,” *Chaos, Solitons & Fractals*, vol. 139, 110059, 2020.
- [11] Y. Mohamadou, A. Halidou, P. T. Kapen, “A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19,” *Appl. Intell.*, vol. 50, no. 1, pp. 3913-3925, 2020.
- [12] I. Rahimi, F. Chen, A. H. Gandomi, “A review on COVID-19 forecasting models,” *Neural Comput. and Applic.*, pp. 1-11, 2021.
- [13] H. Swapnarekha, H. S. Behera, J. Nayak, and B. Naik, “Role of intelligent computing in COVID-19 prognosis: A state-of-the-art review,” *Chaos, Solitons & Fractals*, vol. 138, 109947, 2020.
- [14] L. Peng, W. Yang, D. Zhang, C. Zhuge, and L. Hong, “Epidemic analysis of COVID-19 in China by dynamical modeling,” *arXiv preprint arXiv:2002.06563*, 2020.
- [15] F. M. Khan and R. Gupta, “ARIMA and NAR based prediction model for time series analysis of COVID-19 cases in India,” *J. Saf. Sci. Resilience*, vol. 1, no. 1, pp. 12-18, 2020.
- [16] Z. Malki, E. S. Atlam, A. E. Hassanien, G. Dagnew, M. A. Elhosseini, and I. Gad, “Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches,” *Chaos, Solitons and Fractals*, vol. 138, 110137, 2020.
- [17] M. A. Achterberg, B. Prasse, L. Ma, S. Trajanovski, M. Kitsak, and P. Van Mieghem, “Comparing the accuracy of several network-based COVID-19 prediction algorithms,” *Int. J. Forecasting*, In Press.
- [18] S. Dhamodharavadhani, R. Rathipriya, and J. M. Chatterjee, “COVID-19 mortality rate prediction for India using statistical neural network models,” *Front. Public Health*, vol. 8, 2020.
- [19] G. Pinter, I. Felde, A. Mosavi, P. Ghamisi, R. Gloaguen, “COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach,” *Mathematics*, vol. 8, no. 6, 890, 2020.
- [20] L. Moftakhar, S.E.I.F., Mozghan, and M. S. Safe, “Exponentially increasing trend of infected patients with COVID-19 in Iran: a comparison of neural network and ARIMA forecasting models,” *Iranian Journal of Public Health*, 49, 2020.
- [21] O. Torrealba-Rodriguez, R. A. Conde-Gutiérrez, and A. L. Hernández-Javier, “Modeling and

- prediction of COVID-19 in Mexico applying mathematical and computational models,” *Chaos, Solitons and Fractals*, 138, 109946, 2020.
- [22] P. Arora, H. Kumar, and B. K. Panigrahi, “Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India,” *Chaos, Solitons and Fractals*, vol. 139, 110017, 2020.
- [23] C. P. Kuo and J. S. Fu, “Evaluating the impact of mobility on COVID-19 pandemic with machine learning hybrid predictions,” *Sci. Total Environ.*, vol. 758, 144151, 2021.
- [24] I. Ahmad and S. M. Asad, “Predictions of coronavirus COVID-19 distinct cases in Pakistan through an artificial neural network,” *Epidemiology and Infection*, vol. 148, 2020.
- [25] M. A. Al-qaness, A.A. Ewees, H. Fan, and M. Abd El Aziz, “Optimization method for forecasting confirmed cases of COVID-19 in China,” *J. Clin. Med.*, vol. 9, no. 3, 674, 2020.
- [26] A. K. Sahai, N. Rath, V. Sood, and M. P. Singh, “ARIMA modelling & forecasting of COVID-19 in top five affected countries,” *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 14, no. 5, pp. 1419-1427, 2020.
- [27] L. Bayyurt and B. Bayyurt, “Forecasting of COVID-19 cases and deaths using ARIMA models,” *Medrxiv*, 2020.
- [28] I. Kırbas, A. Sözen, A. D. Tuncer, and F. Ş. Kazancıoğlu, “Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches,” *Chaos, Solitons and Fractals*, 138, 110015, 2020.
- [29] M. Niazkar, T. G. Eryılmaz, H. R. Niazkar, and Y. A. Türkkan, “Assessment of Three Mathematical Prediction Models for Forecasting the COVID-19 Outbreak in Iran and Turkey,” *Comp. Math. Methods in Med.*, 2020.
- [30] S. Arslan, M. Y. Ozdemir, and A. Ucar, “Nowcasting and Forecasting the Spread of COVID-19 and Healthcare Demand in Turkey, a Modeling Study,” *Front. Public Health*, vol. 8, 2020.
- [31] R. Salgotra, M. Gandomi, and A. H. Gandomi, “Time series analysis and forecast of the COVID-19 pandemic in India using genetic programming,” *Chaos, Solitons & Fractals*, vol. 138, 109945, 2020.
- [32] A. I. Saba and A. H. Elsheikh, “Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks,” *Process Saf. Environ. Protection*, vol. 141, pp. 1-8, 2020.
- [33] K. C. Santosh, “COVID-19 prediction models and unexploited data,” *J. Med. Sys.*, vol. 44, no. 170, 2020.
- [34] O. Sharma, A. A. Sultan, H. Ding, and C. R. Triggler, “A Review of the Progress and Challenges of Developing a Vaccine for COVID-19,” *Frontiers in Immunology*, vol. 11, 2413, 2020.
- [35] F. Jung, V. Krieger, F. T. Hufert, and J. H. Küpper, “Herd immunity or suppression strategy to combat COVID-19,” *Clinical Hemorheology and Microcirculation*, pp. 1-5, 2020.
- [36] Our World in Data, “Statistics and Research: Coronavirus (COVID-19) Vaccinations,” 2021. [Online] Available: <https://ourworldindata.org/covid-vaccinations> [Accessed: 09-Sep-2021]
- [37] Google, “Community Mobility Reports,” 2021. [Online] Available: <https://support.google.com/covid19-mobility/answer/9824897?hl=en> [Accessed: 20-Jul-2021]
- [38] F. Pan, et al., “Factors associated with death outcome in patients with severe coronavirus disease-19 (COVID-19): A case-control study,” *Int. J. Med. Sci.*, vol. 17, no. 9, 1281, 2020.
- [39] Turkish Republic Ministry of Health, “COVID-19 Information Platform,” 2021. [Online] Available: <https://covid19.saglik.gov.tr/TR-66935/genel-koronavirus-tablosu.html> [Accessed: 20-Jul-2021]
- [40] Our World in Data, “COVID-19 Vaccinations Public Data: Turkey,” 2021b. [Online] Available: [https://github.com/owid/covid-19-data/blob/master/public/data/vaccinations/country\\_data/Turkey.csv](https://github.com/owid/covid-19-data/blob/master/public/data/vaccinations/country_data/Turkey.csv) [Accessed: 20-Jul-2021]
- [41] P. Chrusciel and S. Szybka, “On the lag between deaths and infections in the first phase of the COVID-19 pandemic,” *medRxiv*, 2021.
- [42] G. Grasselli, et al., “Baseline characteristics and outcomes of 1591 patients infected with SARS-CoV-2 admitted to ICUs of the Lombardy Region, Italy,” *Jama*, vol. 323, no. 16, pp. 1574-1581, 2020.
- [43] S. Morid, and V. Smakhtin, “Drought forecasting using artificial neural networks and time series

- of drought indices,” *Int. J. Climatol.*, vol. 27, no. 15, pp. 2103-2111, 2007.
- [44] X. Li, C. Zhang, B. Zhang, and K. Liu, “A comparative time series analysis and modeling of aerosols in the contiguous United States and China,” *Sci. Total Environ.*, vol. 690, pp. 799-811, 2019.
- [45] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [46] M. D. Philemon, Z. Ismail, and J. Dare, “A review of epidemic forecasting using artificial neural networks,” *Int. J. Epidemiol. Res.*, vol. 6, no. 3, pp. 132-143, 2019.
- [47] G. Cybenko, “Approximations by superpositions of a sigmoidal function,” *Math. Control Signal Systems*, vol. 2, pp. 303–314, 1989.
- [48] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feed forward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [49] G. Zhang, B. E. Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks: the state of the art,” *Int. J. Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [50] R. M. Rizk-Allah, and A. E. Hassanien, “COVID-19 forecasting based on an improved interior search algorithm and multi-layer feed forward neural network,” *arXiv preprint arXiv:2004.05960*, 2020.
- [51] A. A. Suratgar, M. B. Tavakoli, A. Hoseinabadi, “Modified Levenberg–Marquardt method for neural networks training,” *World Acad. Sci. Eng. Technol.*, vol. 6, no. 1, pp. 46-48, 2005.
- [52] H. Yu, and B. M. Wilamowski, “Levenberg-Marquardt training,” *Industrial Electronics Handbook*, vol. 5, no. 12, 1, 2011.
- [53] N. Andrei, “Scaled conjugate gradient algorithms for unconstrained optimization,” *Comput. Optim. Appl.*, vol. 38, no. 3, pp. 401-416, 2007.
- [54] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design*. PWS Publishing Co., 1997.
- [55] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159-175, 2003.
- [56] S. I. Alzahrani, I. A. Aljamaan, and E. A. Al-Fakih, “Forecasting the spread of the COVID-19 pandemic in Saudi Arabia using ARIMA prediction model under current public health interventions,” *J. Inf. and Public Health*, vol. 13, no. 7, pp. 914-919, 2020.

# Performance Analysis of Chaotic Neural Network and Chaotic Cat Map Based Image Encryption

 Sefa Tunçer<sup>1</sup>,  Cihan Karakuzu<sup>2</sup>

<sup>1</sup>Sefa Tunçer; Bilecik Seyh Edebali University; sefa.tuncer@bilecik.edu.tr;  
Bilecik Seyh Edebali University; cihan.karakuzu@bilecik.edu.tr;

Received 30 September 2021; Revised 09 February 2022; Accepted 23 February 2022; Published online 30 April 2022

## Abstract

Nowadays, chaotic systems are used quite often, especially in image encryption applications. Hypersensitivity to the initial conditions, limited field-changing signs and irregular movements make chaotic systems one of the critical elements in scientific matters such as cryptography. Chaotic systems are divided in two parts as discrete time and continuous time in terms of their dimensions and properties. Gray level image encryption applications generally use one-dimensional and color image encryption applications generally use multi-dimensional chaotic systems. In this study, Tent Map, Cat Map, Lorenz, Chua, Lu chaotic systems were used for chaotic neural network based image coding application and Logistic Map and 3D Cat Map chaotic systems were used for 3D chaotic Cat Map based image encryption application. The encrypted image and the original image were examined by various analysis methods. As a result of the examinations, it is seen that both algorithms give very successful results in key size, key sensitivity, entropy analysis, histogram analysis and correlation coefficient analysis. According to the analysis, it has been shown that the chaotic neural network-based image encryption algorithm is more secure and successful.

**Keywords:** image encryption, encryption based on chaotic systems, chaotic neural network

## 1. Introduction

With the developing technology, various algorithms have been developed to ensure the security of digital images. In addition, analyzes are carried out to test the reliability of these algorithms. Use of chaotic systems in image encryption significantly increases security and efficiency in literature. In this study, performance analyzes are performed two different chaotic system based image encryption algorithms. Key size, key sensitivity, entropy analysis, histogram analysis and correlation coefficient analysis are performance criteria taken into consideration.

Prusty et al. [1] mixed the image using Arnold Cat Map chaotic system and generated key and random numbers using Henon Map and encrypted images. They have achieved successful results by encrypting images in different formats. Li et al. [2] implemented image cryptography using Tent Map and Lorenz chaotic systems. It has successfully passed key widths greater than 256 bits, randomness, histogram and correlation tests. For this reason, it can be said that there is a successful encryption. Liu and Wang [3] performed encryption and decryption by combining Red, Green and Blue (R, G, B) values in three different images of the same size, respectively. They used the SHA-256 hash function for key in encryption and Lorenz chaotic system to generate random numbers. Wang et al. [4] aimed to obtain a stronger algorithm by generating different control parameters in each iteration of image encryption. At this point, the image is encrypted with a different key at each step, and very good results are obtained in terms of speed and security. Wong et al. [5] used a chaotic Standard Map to implement an image encryption scheme based on simple addition and replacement operations. They have tried to develop this chaotic cryptographic algorithm based on speed, and as a result they have managed to encode a 512x512 size grayscale image below 100 milliseconds. Zeghid et al. [6] used the AES algorithm, which is used in text encryption in the literature, for image encryption. The AES algorithm was modified using a key generator to get rid of some deficiencies in the image cipher. Thus, they obtained a stronger algorithm in tests such as key width, histogram analysis, correlation coefficient and entropy analysis. Xiao et al. [7] have developed a grayscale image encryption algorithm using Arnold Cat Map and Chen chaotic systems. It has been seen to be successful based on the results obtained from the safety tests. In

this study, mathematical operations used in image encryption based on chaotic neural network (CNN) are similar to algorithm used by Xiao et al. Hongjun and Xingyuan [8] have developed a robust algorithm against noises caused by any reason using the Chebyshev Map chaotic system in the image. They obtained original image with minimum loss against the noise in the encrypted image. Randomness is increased by using MD5 in generation of encryption key to increase confidentiality. They also randomly determine the initial conditions of chaotic systems. In CNN, key is not randomly determined, but it is more secure in terms of key length.

Çavuşoğlu and Al-Sanabani encrypt 256x256 images using lightweight encryption algorithms. Small-sized images are encrypted by choosing lightweight encryption algorithms, which are generally preferred in the field of the Internet of Things. By looking at the performance analysis, it was tried to determine which encryption algorithm would be more efficient. S-AES and LBlock algorithm are suitable in terms of security and speed. Chaotic systems are more advantageous in terms of randomness and speed, but lightweight algorithms are more suitable for IoT systems [9]. Chaotic systems, which are used in many areas, are used especially in optimization algorithms to avoid local extrema and to search better in search space. Demirci and Yurtay [10] stated that chosen chaotic system can also be effective in finding global best solution in optimization algorithm. Similarly, different chaotic systems affect performance in image encryption as well. Süzen and Duman keep data on the blockchain by encrypting it with Advanced Encryption Standard (AES-256 bits) symmetric encryption algorithm. In this way, they try to ensure data integrity and confidentiality. AES, which is efficient in terms of key length and speed, is often used for text encryption [11]. Symmetric encryption algorithms do not always give desired results in histogram analysis, speed, key sensitivity and correlation coefficient analysis compared to chaotic systems in image encryption.

In the following sections, chaotic systems, chaotic neural network based color image encryption and 3-dimensions (3D) chaotic Cat Map based gray level image encryption algorithms have been investigated. The performances of these algorithms against the analysis methods are compared with each other. According to the results of the analysis, it was determined which algorithm was more effective in image encryption.

## 2. Chaotic Systems Used In Image Encryption

In this study, chaotic systems used for image encryption and their properties are given below. These chaotic systems differ in terms of their use and dimensions.

### 2.1 Tent Map

The equation of the Chaotic Tent Map system is shown in Equation 2.1 [12-13]. It is a one dimensional and discrete time system.

$$f(a, x) = \begin{cases} \left\lfloor \frac{M}{a} x \right\rfloor, & 0 \leq x \leq a \\ \left\lfloor \frac{M}{M-a} (M - x) \right\rfloor + 1, & a < x \leq M \end{cases} \quad (2.1)$$

where a value  $a \in [1, M]$  is an integer, and  $\lfloor x \rfloor$  and  $\lceil x \rceil$  represent the upper and lower limit values of  $x$ , respectively. The  $M$  value is usually chosen according to the plain text and  $M = 256$  for an 8 bit image.

### 2.2 2D Cat Map

For an  $N \times N$  gray level image, Cat Map is defined as in Equation 2.2. The  $p$  and  $q$  are the control parameters of the chaotic system and positive integers.  $(x, y)$  and  $(x', y')$  are the positions before and after coordinate values, respectively.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = Q \begin{pmatrix} x \\ y \end{pmatrix} \bmod(N) = \begin{bmatrix} 1 & p \\ q & pq + 1 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod(N) \quad (2.2)$$

### 2.3 3D Cat Map

This system is obtained by expanding the 2D Cat Map system. Chen et al. [14] designed to replace the pixel values of an image in a limited area without any loss (Equation 2.3).  $a_x, a_y, a_z, b_x, b_y, b_z$  are control parameters that are all positive integers.

$$C = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix} \quad (2.3)$$

### 2.4 Lorenz

The Lorenz chaotic system was developed in 1962-63 (Equation 2.4). The constant parameters are  $a = 10$ ,  $b = 8/3$ ,  $c = 28$  for the system to exhibit chaotic behavior [15-17].

$$\begin{cases} x'(t) = a(y(t) - x(t)) \\ y'(t) = -x(t)z(t) + cy(t) \\ z'(t) = x(t)y(t) - bz(t) \end{cases} \quad (2.4)$$

### 2.5 Chua

The equation of the chaotic chaotic system is shown in Equation 2.5. In the system of equations,  $a, b$  are constant parameters and  $f(x(t)) = 2x(t) - x(t)/7$ . The system should be selected as  $a = 10$ ,  $b = 100/7$  for chaotic behavior [18].

$$\begin{cases} x'(t) = a(y(t) - f(x(t))) \\ y'(t) = x(t) - y(t) + z(t) \\ z'(t) = -by(t) \end{cases} \quad (2.5)$$

### 2.6 Lü

The equation of the chaotic chaotic system is shown in Equation 2.6. The constant parameters should be selected as  $a = 36$ ,  $b = 3$ ,  $c = 20$  for chaotic behavior [19].

$$\begin{cases} x'(t) = a(y(t) - x(t)) \\ y'(t) = -x(t)z(t) + cx(t) \\ z'(t) = x(t)y(t) - bz(t) \end{cases} \quad (2.6)$$

### 2.7 Chen

The equation of the chaotic chaotic system is shown in Equation 2.6. The constant parameters should be selected as  $a = 35$ ,  $b = 3$  ve  $c \in [20,28.4]$  for chaotic behavior [7,20]. The change of parameter  $c$  clearly affects behavior of the chaotic system.

## 3. Chaotic Neural Network Based Image Encryption

A chaotic neural network (CNN) based image encryption algorithm developed by Bigdeli et al. [12] has been implemented. The encryption algorithm consists of three separate phrases consisting of a chaotic key generation block, a chaotic network layer (CNL) and a permutation network layer (PNL). The block structure of this encryption method is shown in Figure 1. The second and third layers come from 3 inputs 3 outputs and 3 neurons. The chaotic key generation block supports the network by producing appropriate weight and bias values for these layers.



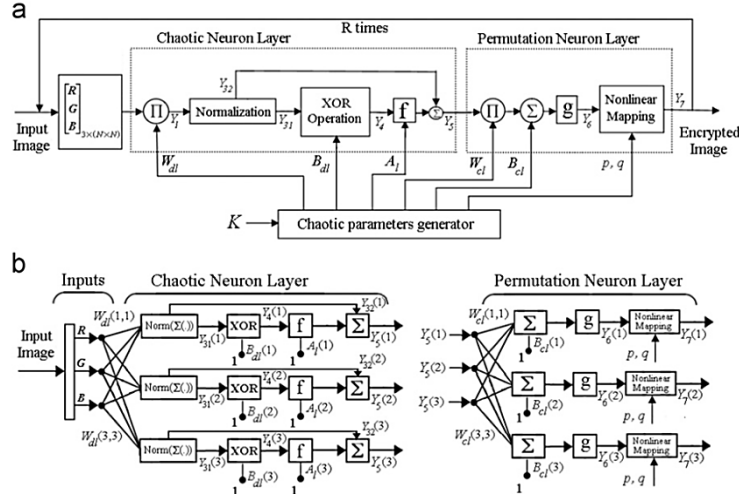


Figure 1 The encryption process (a) block scheme, (b) network scheme [12]

1. A 160-bit key is chosen for the algorithm. As shown in Fig. 2, the 160-bit key is divided into 5 groups, and  $x_1(0), y_1(0), z_1(0), x_2(0), y_2(0), z_2(0), x_3(0), y_3(0), z_3(0)$  input parameters are determined. The  $N_0$  value is chosen to avoid the problem of transiting chaotic systems.  $R$  is the number of repetitions.  $x_1(N_0), y_1(N_0), z_1(N_0), x_2(N_0), y_2(N_0), z_2(N_0), x_3(N_0), y_3(N_0), z_3(N_0)$  are obtained.  $k$  is number of steps.

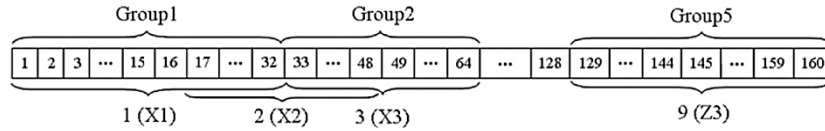


Figure 2 9 key generation from 160 bit verification code [12]

2. An  $F$  image is selected in  $N \times N$  pixel dimensions. Since chaotic systems are passed  $N_0$  times iteration, up to  $N_0$  will not be used again. The values of the three chaotic systems are iteratively calculated for  $N_0 + i$ , where  $i = (r - 1) \times (N \times N) + 1, \dots, r \times (N \times N)$ . Number of iterations of the chaotic system represents  $i = 1, 2, \dots, r \times (N \times N)$ . In each iteration, the values of  $W_{dl}, A_l$  and  $B_{dl}$  are calculated using Equations 3.1, 3.2, 3.3, 3.4 and 3.5, respectively.

$$W_{dl} = \begin{bmatrix} x_1(N_0 + i) & x_2(N_0 + i) & x_3(N_0 + i) \\ y_1(N_0 + i) & y_2(N_0 + i) & y_3(N_0 + i) \\ z_1(N_0 + i) & z_2(N_0 + i) & z_3(N_0 + i) \end{bmatrix} + \alpha I \quad (3.1)$$

$$a(j, i) = \text{mod} \left( \left( |x_j(N_0 + i)| - \text{floor} (x_j(N_0 + i)) \right) \times 10^{14}, 255 \right) + 1, \quad j = 1, 2, 3 \quad (3.2)$$

$$A_l(i) = [a(1, i), a(2, i), a(3, i)]^T \quad (3.3)$$

$$b(j, i) = \text{mod} \left( \left( |y_j(N_0 + i)| - \text{floor} (y_j(N_0 + i)) \right) \times 10^{14}, 255 \right) + 1, \quad j = 1, 2, 3 \quad (3.4)$$

$$B_{dl}(i) = [b(1, i), b(2, i), b(3, i)]^T \quad (3.5)$$

3.  $W_{dl}$  represents the weight matrix of the CNL,  $A_l$  and  $B_{dl}$  represent the bias matrices of the CNL,  $\text{mod}(x, y)$  represents modulo  $y$  of  $x$ ,  $\text{floor}(x)$  is less than or equal to  $x$  itself. Another matrix  $W_{cl}$ , which is the weight matrix of CNL, is used for linear mixing of the three color components obtained from the output of the chaotic neural network. It is used to change the positions of R, G, B components. Hence, it is a matrix of  $3 \times 3$  dimensions, and there is only one '1' in each row and column. Equations 3.6, 3.7, 3.8 and 3.9 are used to determine  $W_{cl}$ .

$$D_i = [x_1(N_0 + i), y_2(N_0 + i), z_3(N_0 + i)] \quad (3.6)$$

$$w_{1,i} = \arg(\max(D_i)) \quad (3.7)$$

$$w_{2,i} = \arg(\max(D_i)) \quad (3.8)$$

$$W_{cl,i}(1, w_{1,i}) = W_{cl,i}(2, w_{2,i}) = 1 \quad (3.9)$$

4. In Equations 3.7 and 3.8,  $\arg(\max(D_i))$  gives index of the maximum value in the vector  $D_i$ . Next, non-zero terms of first and second rows of the matrix  $W_{cl}$  are determined. After these operations, the non-zero term of the third line is determined such that there is only one '1' in each row and column of the matrix  $W_{cl}$ . The variation of the positions of the RGB values in an image can be exemplified as in Equation 3.10.

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} B \\ R \\ G \end{bmatrix} \quad (3.10)$$

5. The input image  $F$  is assumed to be  $N \times N$  pixels. The size of  $F$  image is  $N \times N \times 3$  with RGB. The value of  $k$ -th pixel of RGB components can be expressed as  $X_k = [R_k, G_k, B_k]^T$ ,  $k = 1, \dots, (N \times N)$ . All of the color information in the image is transformed into a matrix of three rows from  $3 \times (N \times N)$ . The matrix  $X$  is calculated as the input of the CNL.

Some operations must be applied to each column of the matrix to obtain the secret information  $F$ . The operations are as follows:  $X_1$  (Equation 3.11) is obtained based on the values  $F_k$ ,  $k = 1, \dots, (N \times N)$  and  $i = (r - 1) \times (N \times N) + 1, \dots, r \times (N \times N)$ . Then normalization is applied (Equation 3.12) in the range of 0-255 for  $X_2$ .  $X_3$  is obtained depending on  $X_2$  Equation 3.13. Next,  $X_{31}$  (Equation 3.14) and  $X_{32}$  (Equation 3.15) are determined based on  $X_3$  for use in subsequent operations. Value of  $X_4$  (Equation 3.16) is determined by special or (XOR) operation. In this step, XOR operation means that bit XOR process. Then the chaotic activation function (Equation 3.17) is applied. The function  $f$  represents Tent Map.

$$X_{1,k} = W_{dl}(i)F(k) \quad (3.11)$$

$$X_2(k) = \text{Normalization}(X_1(k)) \quad (3.12)$$

$$X_3(k) = \text{floor}(X_2(k)) + \text{mod}(X_2(k), \text{floor}(X_2(k))) \quad (3.13)$$

$$X_{31}(k) = \text{floor}(X_2(k)) \quad (3.14)$$

$$X_{32}(k) = \text{mod}(X_2(k), \text{floor}(X_2(k))) \quad (3.15)$$

$$X_4(k) = \text{XOR}(X_{31}(k), B_{dl}(i)) \quad (3.16)$$

$$X_5(k) = f(X_4(k), A_l(i)) + X_{32}(k) \quad (3.17)$$

6.  $X_5$  matrix of dimensions  $3 \times (N \times N)$ , which is the output of the CNL, is mixed in two stages in the PNL. In the first step, each column of  $X_5$  is mixed linearly with  $X_5(k)$ ,  $k = 1, \dots, (N \times N)$ .

$$X_6(k) = g(W_{cl}(i)X_5(k)) \quad (3.18)$$

The activation functions of the neurons which the parameters of the neural network layer structure, weight matrix and bias vector are arranged for determined purposes. The purpose of this study is to change the positions and pixel values of the R, G and B components using neural structure. Appropriate weight and bias values are selected for this. The result  $g(x) = x$  achieves a high performance ratio, but not sometimes. The weight matrix,  $W_{cl}$ , is calculated as described in step 3.

7. In this step, the outputs of the linear permutation step are mixed. Hence, each line of the matrix  $X_6$  is arranged in an  $N \times N$  matrix so that three output  $N \times N$  matrices are obtained. Then each matrix is mixed with 2B Cat Map permutation algorithm. Non-linearly mixed matrices are designated as R,

G and B values of the encrypted image, that is  $X_7$ . If the final iteration is not reached ( $r < R$ ),  $F = X_7$  is made.  $r$  is incremented by 1 and step 3 is returned. In this way, the encrypted image ( $F_{end}$ ) is obtained in the last iteration and the encryption process is completed.

#### 4. Image Encryption Based on 3d Chaotic Cat Map

3D chaotic Cat Map (CCM) based image encryption algorithm was developed by Chen et al. [14]. It comes from the following steps.

1. Key generation. 128 bit array are selected as a key and divided into a few of the parameters of the 3B Cat Map and the eight group mapped onto the Logistic Map  $a_x, b_x, a_y, b_y, a_z, b_z, L_g$  and  $T$  values. The encryption block diagram is shown in Fig. 3.
2. A two-dimensional image is transformed into three-dimensional matrices. The image consists of  $W$  pixel width and  $H$  pixel height. First, all pixels of image are partitioned into a number of clusters of  $N_1 \times N_1 \times N_1, N_2 \times N_2 \times N_2, \dots, N_k \times N_k \times N_k$  dimensions respectively. The following condition must be satisfied in order to divide an image into a matrix of several cubes.

$$W \times H = N_1^3 + N_2^3 + \dots + N_k^3 + R \quad (4.1)$$

where  $N_k \in \{2, 3, \dots, M\}$  is the length of one edge of each cube,  $M$  is the maximum number of cubes, and  $R \in \{0, 1, \dots, 7\}$  is the number of unused pixels after all cubes have been constructed.

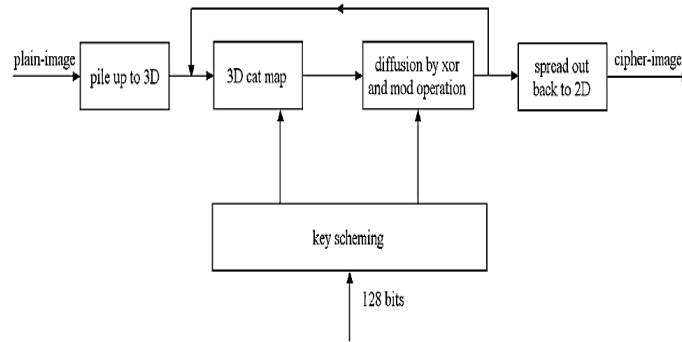


Figure 3 Image encryption block diagram of 3D Cat Map [14]

3. 3D Cat Map is performed. For each cubes mixture, a three dimensional discrete Cat Map is applied using control parameters  $a_x, b_x, a_y, b_y, a_z, b_z$ . In this process, the initial values  $x(0) = L_g$  and  $S(0) = T$  are selected and the diffusion process is performed according to the algorithm in Equations 4.2 and 4.3.  $L_g$  is the floating-point number in the range (0,1),  $T$  is an integer.  $L_g$  is the initial value of the chaotic logistic map. Eq. 5.3 expresses  $E(i)$  the encrypted pixel value,  $E(i - 1)$  the previous encrypted pixel value,  $I(i)$  the current pixel value, and  $N$  the gray level image color level. In gray level images, the color level is represented by  $N = 256$ .

$$x(i + 1) = 4x(i)[1 - x(i)] \quad (4.2)$$

$$E(i) = \varphi(i) \oplus \{[I(i) + \varphi(i)] \bmod N\} \oplus E(i - 1) \quad (4.3)$$

4. The cubes mixed in three dimensions are transformed into two dimensional images again.
5. The operations are repeated regularly until security reaches a appropriate level in steps 3 and 4. As the number of iterations increases, a more secure encryption takes place, but disadvantages arise, such as computational cost and time delays.

## 5. Analyzes and Conclusions

The security levels of CNN and CCM based image encryption algorithms have been tested. Various analyzes have been made for this. Analyzes have shown that the CNN is safer.

### 5.1 Key size security

The key size ensures that the encryption algorithm is robust against brute force attacks. The CNN has 224 bits and the 3B CCM has a 128 bit long key. From this point of view, CNN is more secure against brute force attacks.

### 5.2 Key sensitivity

Chaotic systems considerably increase key sensitivity. When the input parameters of the chaotic system obtained by the key change  $10^{-14}$ , it becomes impossible to obtain the original image from the encrypted image. This sensitivity can be up to  $10^{-16}$  at some points. From this point of view, CNN system, which uses more chaotic systems, is more secure.

### 5.3 Information entropy analysis

The degree of uncertainty in a system is entropy. The information entropy  $E(m)$  of an  $m$  message is calculated as in Equation 5.1 [12].

$$E(m) = -\sum_{j=0}^{2^n-1} p(m_j) \log_2 \frac{1}{p(m_j)} \quad (5.1)$$

$p(m_j)$  is the probability of occurrence of  $m_j$ . Each symbol has equal probability  $p(m_j) = 2^{-8}$ , if  $n = 8$ . In this case, the distribution term in the entropy is  $E(m) = 8$ . The entropy values of the RGB in the encrypted image are given in Table 1. Both CNN and CCM have successful results.

Table 1 Information entropy analysis values

Method	CNN			CCM
	R	G	B	Gray Level
Lena	7.9928	7.9937	7.9928	7.9967
Baboon	7.9931	7.9938	7.9931	7.9970
Peppers	7.9930	7.9934	7.9929	7.9971

### 5.4 Histogram analysis

The histograms of the images encoded with CNN and CCM are shown in Fig. 4 and Fig. 5, respectively. It shows an almost homogeneous distribution, when the histogram of encrypted images is examined. At this point, no information about the pixel in the original image can be obtained from the encrypted image. Therefore, there is no statistical attack on the encrypted image, the original image and the proposed encryption process. Both CNN and CCM showed successful results in histogram analysis.

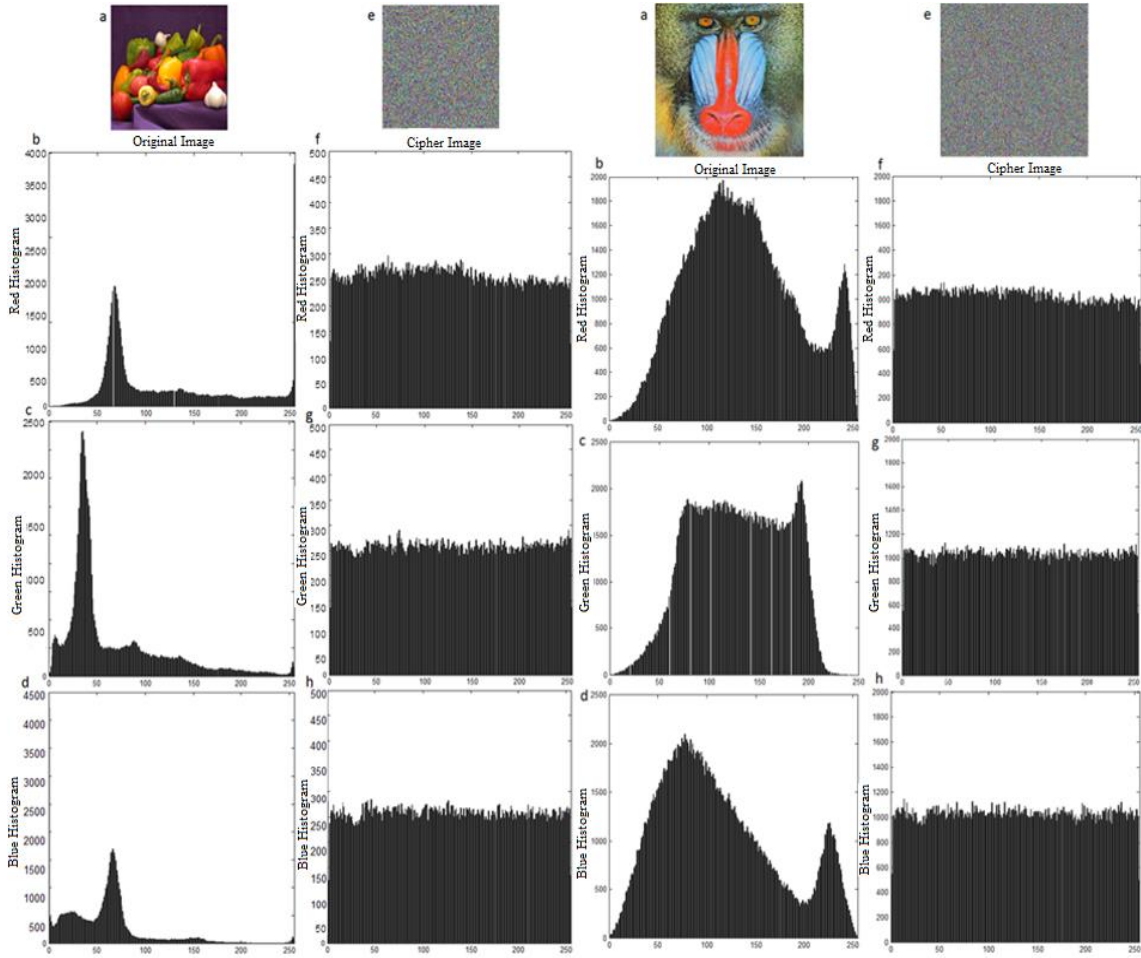


Figure 4 Red (b, f), green (c, g) and blue (d, h) color histograms of the original (a) and encrypted (e) states of peppers and baboon images for CNN

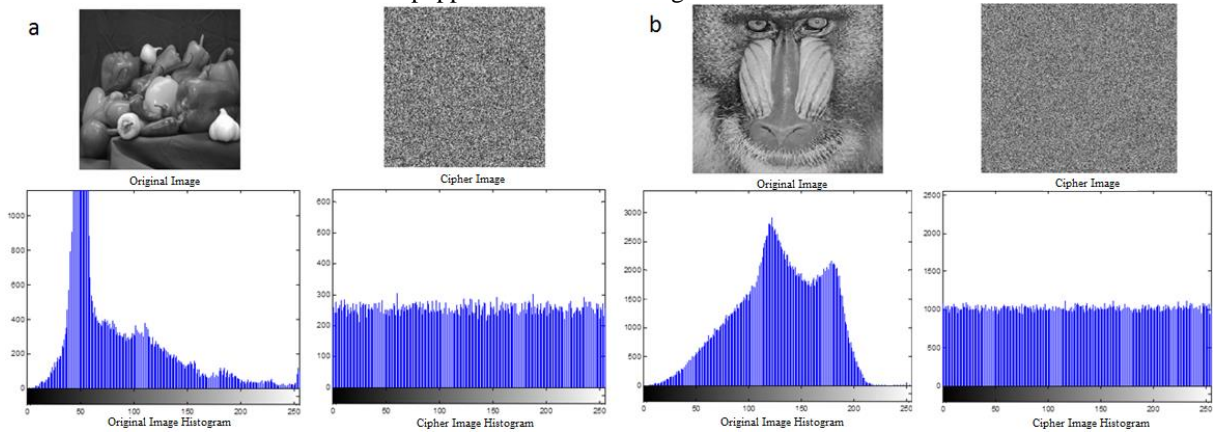


Figure 1 Gray level histograms of original and scrambled states of peppers (a) and baboon (b) images for CCM

### 5.4 Correlation coefficient analysis

Correlation explores the relationship between two or more values in terms of decrease and increase. Correlation value in an original image will be high. An effective image encryption algorithm should reduce the correlation between adjacent pixels. The average correlation values of the vertical, horizontal and diagonal pixels of 3 different images encrypted with CNN and CCM are shown in Table 2. According to these results, it can be said that CNN is more successful.

Table 2 Vertical, horizontal and diagonal correlation values

	Original Image			Cipher Image (CNN)			Cipher Image (CCM)		
	Vertical	Horizontal	Cross	Vertical	Horizontal	Cross	Vertical	Horizontal	Cross
Lena	0.94	0.9227	0.8983	-0.0432	-0.0145	-0.0011	-0.2035	0.0008	-0.007
Baboon	0.7585	0.83	0.7727	0.0079	-0.0038	-0.0273	-0.3078	-0.0031	0.003
Peppers	0.9772	0.9793	0.9630	0.0098	-0.0141	0.0143	-0.0841	-0.0100	-0.011

5000 adjacent pixel pairs were randomly selected from the baboon image for CNN. The vertically, horizontally and diagonally adjacent pixels in the original and ciphered images are compared (Figure 6). It is seen that the correlations of the adjacent pixels of the original image are highly close to each other and the adjacent pixels of the encrypted image is close to zero.

Similar to CNN, 5000 adjacent pixel pairs were randomly selected from the baboon image for CCM. The vertically, diagonally adjacent pixels in the original and ciphered images are compared (Figure 7). It is seen that the correlations of the adjacent pixels of the original image are highly close to each other and the adjacent pixels of the encrypted image is close to zero.

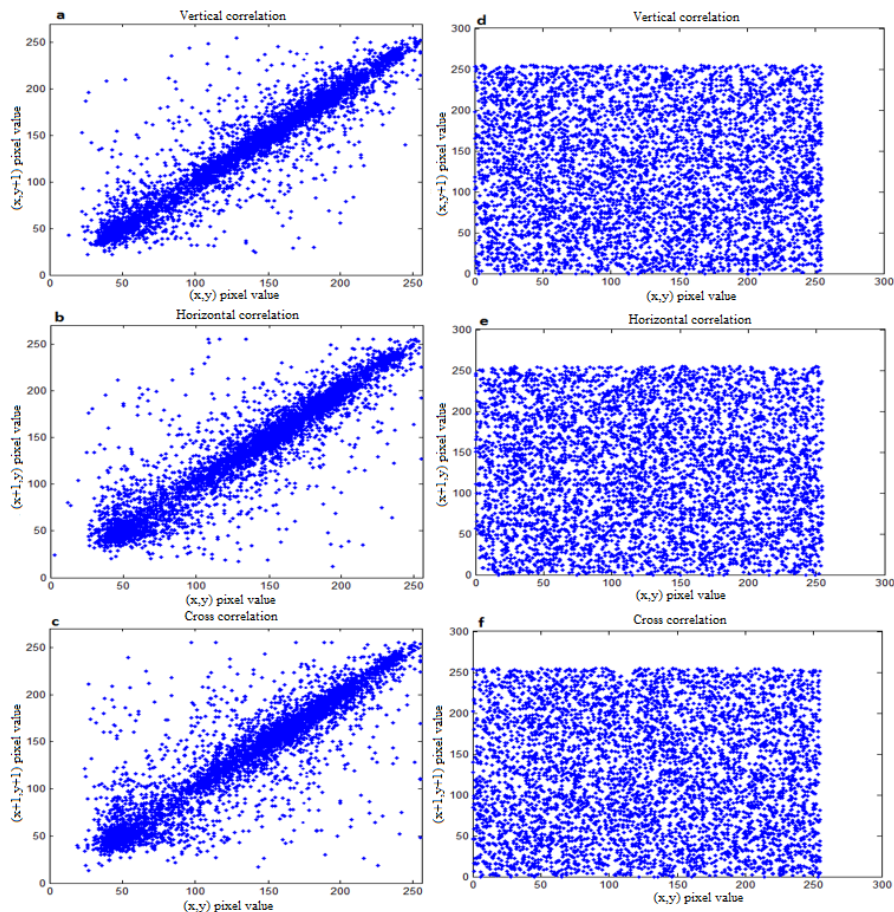


Figure 2 Vertical, diagonal correlation graphs of original (a, b, c) and encrypted (d, e, f) baboon images for CNN



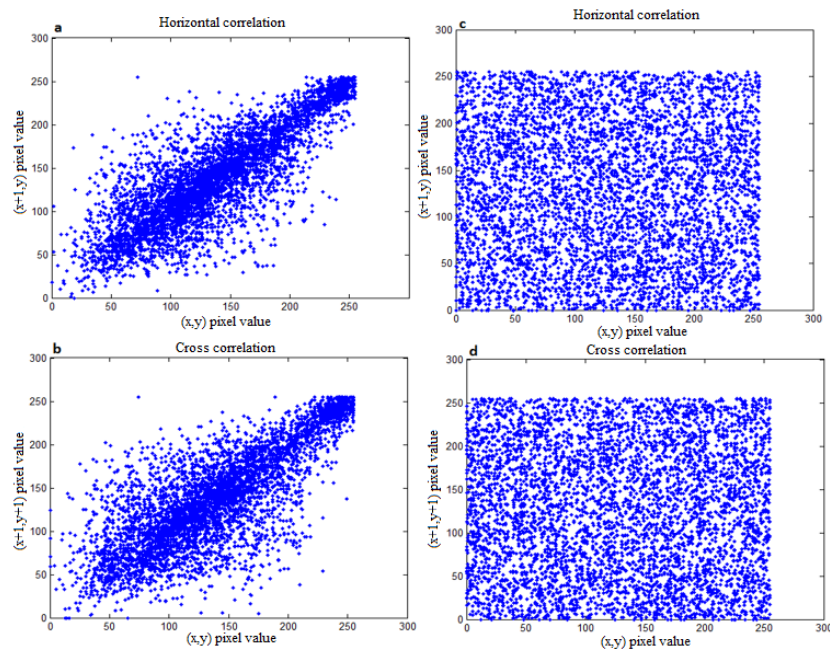


Figure 3 Vertical, horizontal, diagonal correlation graphs of original (a, b) and encrypted (c, d) baboon images for CCM.

Based on all analyzes, CNN and CCM algorithms can be simply compared as in Table 3. Key sensitivity line looks same for both algorithms, it has a more precise and effective structure due to the chaotic systems used by the CNN. While the correlation coefficients are at the maximum level of 0.01 in CNN, this ratio can be at level of 0.3 in CCM depending on selected picture. Therefore, it can be said that CNN is better in terms of correlation analysis. As a result, both algorithms gave desired results, but CNN was found to be a safer and more successful algorithm than CCM. In addition, both gray level and color image encryption can be done with CNN.

Table 3 Analysis results of CNN and CCM based algorithms

Parameter / Method		CNN	CCM
Key size security		224 bit	128 bit
Key sensitivity		$10^{-14}$ degrees	$10^{-14}$ degrees
Histogram analysis		Secure	Secure
Correlation coefficient analysis	Vertical	-0.0432	-0.2035
	Horizontal	-0.0145	-0.0008
	Cross	-0.0011	-0.0070
Information entropy analysis		R: 7.9930 G: 7.9938 B: 7.9929	Gray: 7.9921

## References

- [1] A.K. Prusty, A. Pattanaik, S. Mishra, "An Image Encryption & Decryption Approach Based on Pixel Shuffling Using Arnold Cat Map & Henon Map", International Conference on Advanced Computing and Communication Systems, 1-6, 2013.
- [2] J. Li, Y. Xing, C. Qu, J. Zhang, "An Image Encryption Method Based on Tent and Lorenz Chaotic Systems", Software Engineering and Service Science (ICSESS), 582-586, 2015.
- [3] H. Liu and X. Wang, "Triple-image encryption scheme based on one-time key stream generated by chaos and plain images", The Journal of Systems and Software, 86:826-834, 2013.
- [4] Y. Wang, K. Wong, X. Liao, T. Xiang, G. Chen, "A chaos-based image encryption algorithm with variable control parameters", Chaos, Solitons and Fractals, 41:1773-1783, 2009.
- [5] K. Wong, B.S. Kwok, W.S. Law, "A fast image encryption scheme based on chaotic standard map", Elsevier Physics Letter A, 372(15):2645-2652, 2008.

- [6] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, R. Tourki, “A Modified AES Based Algorithm for Image Encryption”, *International Journal of Computer Science and Engineering*, 1(1):70-75, 2007.
- [7] D. Xiao, X. Liao, P. Wei, “Analysis and improvement of a chaos-based image encryption Algorithm”, *Chaos, Solitons and Fractals*, 40:2191–2199, 2009.
- [8] L. Hongjun and W. Xingyuan, “Color image encryption based on one-time keys and robust chaotic maps”, *Computers and Mathematics with Applications*, 59:3320-3327, 2010.
- [9] Ü. Çavuşoğlu and H. Al-Sanabani, “The Performance Comparison of Lightweight Encryption Algorithms”, *Sakarya University Journal of Computer And Information Sciences*, 2(3):158-169, December 2019.
- [10] H. Demirci and N. Yurtay, “Effect of the Chaotic Crossover Operator on Breeding Swarms Algorithm”, *Sakarya University Journal of Computer And Information Sciences*, 4(1):120-130, April 2021.
- [11] A. A. Süzen and B. Duman, “Blockchain-Based Secure Credit Card Storage System for E-Commerce”, *Sakarya University Journal of Computer And Information Sciences*, 4(2):204-215, August 2021.
- [12] N. Bigdeli, Y. Farid, K. Afshar, “A novel image encryption/decryption scheme based on chaotic neural networks”, *Engineering Applications of Artificial Intelligence* 25:753–765, 2012.
- [13] N. Masuda and K. Aihara, “Cryptosystems With Discretized Chaotic Maps”, *IEEE Transactions On Circuits And Systems: Fundamental Theory And Applications*, 49(1):28-40, 2002.
- [14] G. Chen, Y. Mao, C.K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps”, *Elsevier Chaos, Solitons and Fractals*, (21):749–761, 2004.
- [15] E.N. Lorenz, “Deterministic Nonperiodic Flow”, *Journal of the Atmospheric Sciences*, 20:130-141, 1963.
- [16] O.A. González, G. Han, J.P. de Gyvez, and Edgar, “CMOS Cryptosystem Using a Lorenz Chaotic Oscillator”, *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '99*, 5:442-445, 1999.
- [17] D. Li, Z. Yin, “Connecting the Lorenz and Chen systems via nonlinear control”, *Commun. Nonlinear Sci. Numerical Simulation*, 14(3):655–667, 2009.
- [18] T. Botmart and P. Niamsup, “Adaptive control and synchronization of the perturbed Chua’s system”, *Math. Comput. Simulation*, 75(1–2):37–55, 2007.
- [19] J. Lü, G. Chen, S. Zhang, “The compound structure of a new chaotic attractor” *Chaos Solitons Fractals*, 14(5):669–672, 2002.
- [20] G. Chen and T. Ueta, “Yet another chaotic attractor”, *Int J Bifurcat Chaos*, 9(7):1465–6, 1999.



# Using a Convolutional Neural Network as Feature Extractor for Different Machine Learning Classifiers to Diagnose Pneumonia

 Enes AYAN<sup>1</sup>

<sup>1</sup>Corresponding Author; Kirikkale University, Department of Computer Engineering; 75450, Yahsihan/Kirikakale, Turkey; enesayan@kku.edu.tr

Received 04 November 2021; Revised 14 February 2022; Accepted 01 March 2022; Published online 30 April 2022

## Abstract

Pneumonia is a general public health problem. It is an important risk factor, especially for children under 5 years old and people aged 65 and older. Fortunately, it is a treatable disease when diagnosed in the early phase. The most common diagnostic method known for the disease is chest X-Rays. However, the disease can be confused with different disorders in the lungs or its own variants by experts. Therefore, computer-aided diagnostic systems are necessary to provide a second opinion to experts. Convolutional neural networks are a subfield of deep learning and they have demonstrated success in solving many medical problems. In this paper, Xception which is a convolutional neural network was trained with the transfer learning method to detect viral pneumonia, normal cases, and bacterial pneumonia in chest X-Rays. Then, five different machine learning classification algorithms were trained with the features obtained by the trained convolutional neural network. The classification performances of the algorithms were compared. According to the experimental results, Xception achieved the best classification performance with an accuracy of 89.74%. On the other hand, SVM achieved the closest classification performance to the convolutional neural network model with 89.58% accuracy.

**Keywords:** Artificial Intelligence, Deep Learning, Pneumonia, Convolutional Neural Networks, Machine Learning

## 1. Introduction

Pneumonia is an infection that appears in the lungs caused by viruses or bacteria. It presents a great threat, especially for children. Annually, 1.4 million children die because of pneumonia. 18% of these children are under 5 years old [1]. Pneumonia also threatens people aged 65 and older and those with chronic lung diseases [2]. Every year an average of 450 million people contracted pneumonia worldwide [3]. Fortunately, through early diagnosis and suitable medication pneumonia can be treated before it becomes deadly [4]. The most common method utilized for the diagnosis of pneumonia is chest X-Ray images [5]. But pneumonia could be confused with other diseases or harmless abnormalities in chest X-Rays. Because of diagnostic errors, the patient's status gets may worse, it may even result in death [6]. The diagnostic accuracy is dependent on the experience and attention of the radiologist. As a result, diagnostic decisions could be subjective and error-prone. On the other hand, training a radiologist with enough experience is required a long time and costs [7]. Also, it is very difficult to come across a radiologist in rural areas of low-income countries. Therefore, computer-aided diagnosis (CAD) systems are necessary to assist radiologists to diagnose pneumonia in chest X-Rays [8]. Artificial intelligence is a popular research area in terms of solving medical problems. Recently, Convolutional Neural Networks (CNNs) is a subfield of deep learning that has become pretty popular in computer vision tasks. CNNs are special deep artificial neural networks that have been designed inspiring by the mammalian visual cortex [9]. Some applications of CNNs in different computer vision problems are object classification, instance segmentation, and object localization [10]. CNNs also have been achieved promising results in solving medical problems using medical image data (e.g., breast cancer detection [11], brain tumor segmentation [12], skin cancer detection [13]).

The diagnosis of pneumonia in chest X-Rays using CNNs is a popular research area. Most of the studies in the literature focus on re-training of pre-trained CNN models with appropriate transfer learning and fine-tuning strategies. Recent studies have generally focused on only distinguishing between normal and

pneumonia cases. Although bacterial and viral types of pneumonia are similar, their treatments are different. There are few studies that classify cases into viral pneumonia bacterial pneumonia and normal cases. The motivation of this study is utilizing a CNN model to be a feature extractor for different machine learning algorithms and comparing their pneumonia classification performance in chest X-Ray images. For this purpose, the Xception CNN model was trained using convenient transfer learning and fine-tuning strategies to classify chest X-Rays into three different classes (normal, viral pneumonia, and bacterial pneumonia). After the training process, the convolutional part of the CNN model was separated from fully connected layers and was used as a feature extractor for various machine learning classifiers. The provided features by the convolutional layers were used to train K-Nearest Neighbor (KNN), Decision Trees (DT), Support Vector Machines (SVM), Logistic Regression (LR), Naive Bayes (NB), and Fully Connected Neural Networks (FC). The trained algorithms were evaluated on test data and their classification performances were compared.

The rest of the paper is organized as follows: Section 2 presents related works. Section 3 introduces the dataset, materials and methods. Section 4 includes experimental results. Section 5 provides a brief discussion. The last section contains the conclusions of the study.

## 2. Related Works

Thanks to open access datasets, many studies have been carried out to diagnose pneumonia in chest X-Rays using CNNs. Some studies which are using the same dataset as in this study are following. Rajpurkar et al. [14] proposed a 121-layer CNN (CheXNet) model for pneumonia diagnosis. The proposed CNN model in the study was trained with 100000 chest X-Rays, to classify 14 different disorders in the lungs, including pneumonia. According to the test results, the proposed model achieved better classification performance than the radiologists in terms of detecting pneumonia. Kermany et al. [15] trained a pre-trained CNN model utilizing the transfer learning strategy to diagnose pneumonia in chest X-Rays. Liang et al. [16] designed a CNN model via residual connections and dilated convolutions to diagnose pneumonia. They also focused on the effect of transfer learning in pneumonia classification from chest X-Rays. Chounhan et al. [17] trained five different pre-trained CNN via suitable transfer learning strategies for diagnosing pneumonia. Then, they proposed an ensemble method, combining the estimates of the CNN models. Gu et al. [18] diagnosed bacterial and viral pneumonia using a method that comprises two stages. In the first stage, they segmented the lungs with a Fully Convolutional Network (FCN) model. In the second stage, they used a Deep Convolutional Neural Network (DCNN) to detect pneumonia from segmented lungs.

Mittal et al. [19] accomplished the diagnosis of pneumonia in chest X-Rays with the Capsule Network (CapsNet) which is a CNN model including multilayer capsules. Prayog et al. [20] utilized a siamese convolutional network (SCN) to classify chest X-Ray images into 3 classes (normal, viral pneumonia, bacterial pneumonia). Rahman et al. [21] have trained four different pre-trained CNN models (AlexNet, ResNet18, DenseNet201, and SqueezeNet) by transfer learning method to classify pneumonia in chest X-Ray images. Hasimi et al. [22] proposed a weighted CNN classifier using various models: ResNet18, Xception, Inception V3, DenseNet 121, and MobileNet V3 to diagnose pneumonia. Mahmud et al. [23] designed a CNN model named as CovXNet. They utilized depthwise convolution in varying dilation rates in the proposed architecture to differentiate Covid19 from pneumonia. Asnaoui [24] compared single and ensemble learning CNN models (InceptionResNet V2, ResNet 50, MobileNet V2) classification performances of pneumonia and Covid19. Darici et al. [25] used a CNN voting ensemble methodology to classify chest X-Rays into three classes.

Most of the studies in the literature focus on training a pre-trained CNN model for diagnosing pneumonia in chest X-rays. But determining fully connected layer size and neuron number in the classification part of CNN models is complicated and requires experience. The main advantage of our work is to show a CNN model can be used as a feature extractor to simple machine learning algorithms. Instead of using hand-crafted features, a pre-trained CNN model can be used easily as a feature extractor to machine learning algorithms and can be achieved successful results.

### 3. Materials and methods

In this study, our main idea is to compare the classification performance of a CNN model's fully connected layers with different machine learning classifiers. The proposed methodology is figured out in Figure 1. First of all, we trained a pre-trained Xception model using transfer learning and fine-tuning methods to classify pneumonia in chest X-Rays. Then, the convolutional part of the CNN model was separated from fully connected layers and was used as a feature extractor for various machine learning classifiers. The provided features by the convolutional layers of the model were used to train DT, SVM, LR, NB, KNN. In the final step, the trained machine learning algorithms were evaluated on test data and their individual classification performances were compared.

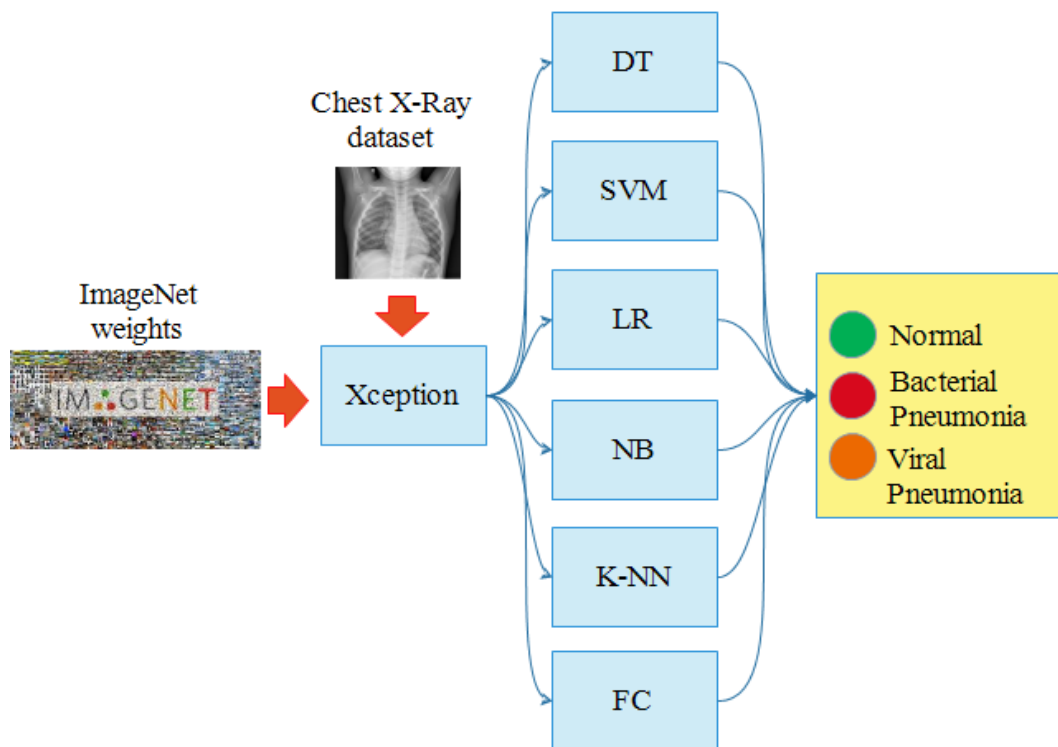


Figure 1 A semantic representation of our methodology

#### 3.1 Dataset

The dataset used in the study was collected from Guangzhou Women and Children's Medical Center and contains chest X-Rays of children under the age of five [26]. The dataset includes a total of 5856 RGB chest X-Rays including bacterial pneumonia, normal, and viral pneumonia cases. The dataset is split into the train (5232) and test (624) sets by its creators. We used a hold-out validation strategy, our test and validation sets are the same. But, in the validation process, we applied online data augmentation to the validation set. The class distribution of the dataset is given in Table 1. In addition, some sample images in different classes from the dataset are given in Figure 2. We resized all images to (299x299x3) to match the default input size of Xception and normalized in the range of [0-1].

Table 1 Distribution of dataset in terms of classes, train validation, and test

Class	Train	Validation	Test
Normal	1349	234	234
Viral Pneumonia	1345	148	148
Bacterial Pneumonia	2538	242	242
Total	5232	624	624

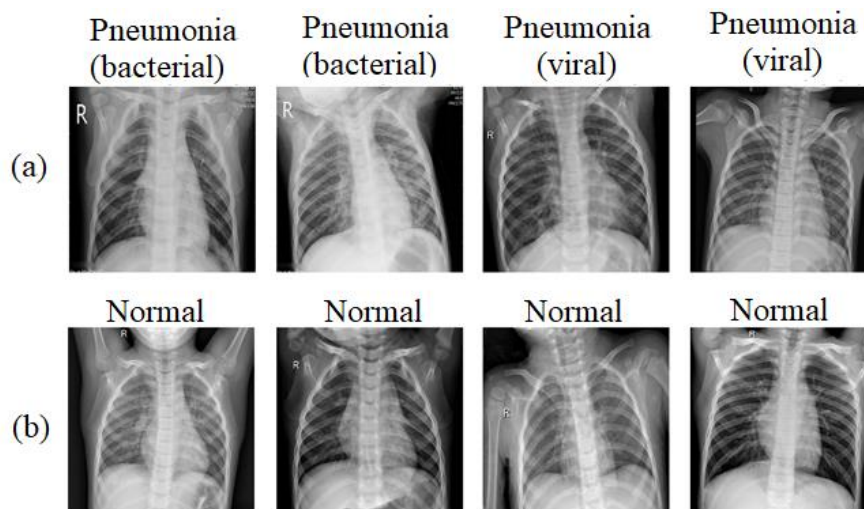


Figure 2 Case samples from the dataset

### 3.2 Convolutional neural networks

CNNs use a backpropagation algorithm to learn features automatically from data [27]. A CNN architecture consists of four basic components. These are convolutional layers, activation function, pooling layers, and fully connected layers. In the convolution layer, filters of a specified given number and size (3x3, 5x5, 7x7), scan the image in order to reveal meaningful features. After the convolutional layer, an activation function is applied to the obtained features. Activation functions help the model to learn more complex patterns. The Rectified Linear Unit (ReLU) is the most preferred activation function in CNN models. The pooling layers are used to reduce the size of feature maps by retaining the most important information. They also help to reduce calculation costs. The obtained feature maps after stacks of convolution and pooling layers are used in fully connected layers (FC) as the final classification output of a CNN model. In this study, we utilized Xception CNN architecture which was proposed by F. Chollet in 2017 [28]. Xception model was inspired by Inception V3 architecture. The model consists of a linear stack of reverse order depthwise separable convolutions and residual connections. Depthwise separable convolution operation consists of a depthwise convolution followed by a pointwise convolution. Recently, it is mostly preferred convolution type instead of the conventional convolution operation. Depthwise separable convolution is more efficient in terms of computations cost according to conventional convolution operation. It also has fewer parameters to adjust as compared to standard convolution, which helps to reduce overfitting. In Xception architecture depthwise separable convolution operation was modified. In the modified version, pointwise convolution was followed by depthwise convolution. This modification has a similarity with the inception module in Inception V3 architecture. The Xception architecture is composed of 14 modules and 36 convolutional layers in total. All modules have residual connections, except for the first and last modules. The last module is terminated by global average pooling instead of a flatten layer to preserve spatial features of feature

maps. In this study, two FC layers with 512 neurons were added after the global average pooling layer to Xception architecture. The ReLU activation function was used in these FC layers. We were determined the FC layer number and their neuron size trying different configurations based on the performance of validation and train data. The output of the fine-tuned Xception model was arranged as 3 neurons to classify bacterial pneumonia, normal and viral pneumonia. In order to avoid overfitting and covariate shift problems, dropout, L2 regularization, and batch normalization methods were used after every FC layer (see Figure 3).

### 3.3 Transfer learning and data augmentation

Transfer learning is the use of knowledge and experience obtained to solve a problem in solving another similar problem [29]. Few researchers have been training a CNN model from scratch in recent years. They use weights of pre-trained models that were trained on a large dataset such as ImageNet [30]. Pre-trained CNN models can be used directly or integrated into a new model to solve new computer vision problems. In this way, the training process is reduced. There are some tricks by using pre-trained model weights. The most important of criteria is the similarity between the new dataset and the large dataset in which the model was trained previously. This similarity plays a critical role in determining the number of layers that will not participate in training. The initial convolutional layers learn data-independent general features such as edges, corners, simple textures. Through further layers, more complex features emerge such as more complex patterns and textures, objects, and parts of objects. For this reason, while training a CNN model with a new dataset, initial layers may not participate in training according to the data similarity between the new dataset and the large dataset which the model was trained previously. Consequently, a predetermined number of layer weights doesn't update. This procedure is called fine-tuning. Fine-tuning has a positive effect on reducing training time and improving classification performance [31]. In this study, we fine-tuned Xception architecture. We trained the fine-tuned model 10 times over 25 epochs. In the first training procedure, 10% percent of the model was frozen then we increased the freezing rate 10% in the next trainings. The best results were achieved at a 60% freeze rate. The first 60 layers of the model were frozen and didn't train. This number corresponds to 60% of the whole CNN model (see Figure 3). The total parameters size of fine-tuned Xception are 22,178,859 and we trained 14,021,627 of the total. The classification performance of CNNs is directly proportional to the amount of training data [32]. However, it is not possible to obtain sufficient data for every medical problem. Data augmentation is a commonly used method for increasing the number of samples in a dataset. It improves the model's generalization capability, prevents overfitting, and increases the model accuracy [33]. We used online (training time) data augmentation in this study. The train and validation datasets were augmented using various image processing methods such as zooming, rotation, shearing, width shift, horizontal flipping, height shift, and rescaling. Detailed parameters of the image augmentation methods are given in Table 2.

Table 2 Image augmentation parameters

Hyperparameter	Value
Zoom range	0.2
Shear range,	0.2
Rotation range	40
Rescale	1./255- [0,1]
Width shift	0.2
Height shift	0.2
Fill mode	Nearest
Horizontal flip	True

### 3.4 Experimental setup

The fine-tuned Xception model was retrained over 25 epochs with 32 batch sizes. We used the softmax activation function in the output layer. The categorical cross-entropy loss function was used to calculate model error and Adam optimization algorithm was preferred to minimize the loss function. Python programming language and Keras deep learning framework was used to implement the proposed comparison. All trials have been carried out on a computer with a 1080 Ti graphics card and Ubuntu operating system. The fine-tuned CNN model's accuracy and loss graphs are shown in Figure. 4. The gap between the training accuracy, validation accuracy, and training loss, validation loss graphs indicate an overfitting problem. It is clearly seen in Figure 4 the gap is an acceptable extent and there is not any overfitting sign. The fluctuation in the validation accuracy and validation loss graphs is due to the dropout layers. We determined the model hyperparameters batch size, epochs size, learning rate, regularization coefficient by trying various values based on the performance of validation and train data. The hyperparameters of the fine-tuned model are given in Table 3. The training and testing times of all algorithms are given in Table 4.

Table 3 Fine-tuned Xception model parameters

Hyperparameter	Value
Dropout	0.5
Learning rate	0.001
L2 regularization	0.001
Adam beta 1	0.9
Adam beta 2	0.999
Epochs	25

Table 4. The training and testing times of algorithms in seconds.

Algorithm	Training Time	Testing Time
Xception	1325	11
KNN	6.46	4.59
SVM	23	1.43
NB	0.22	0.022
DT	5.95	0.014
LR	1.35	0.025

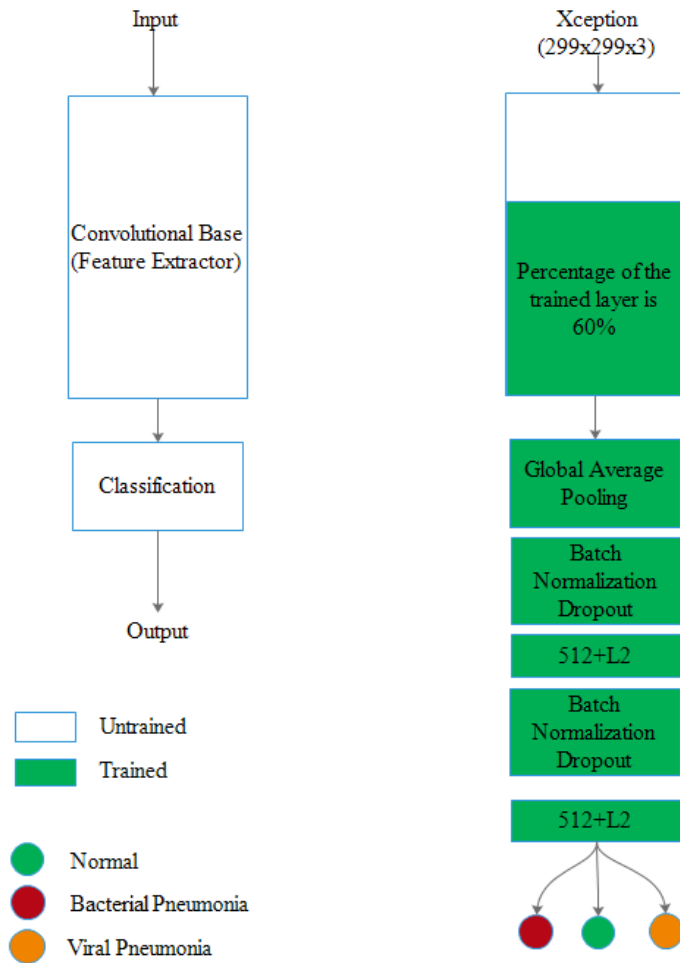


Figure 3 Fine-tuned Xception model

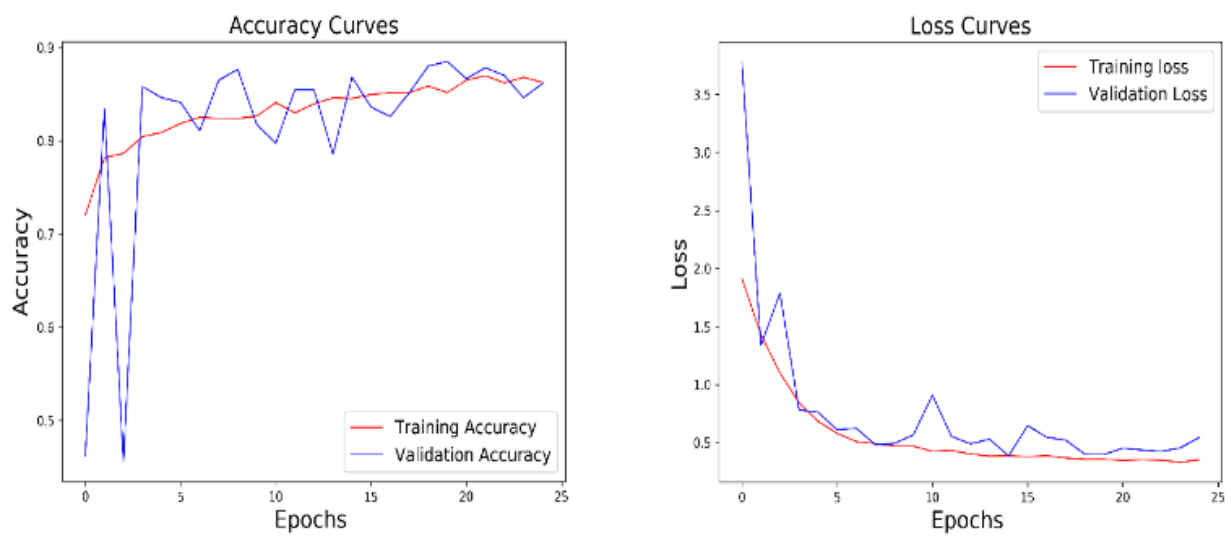


Figure 4 Accuracy and loss graphs of fine-tuned Xception model

### 3.5 Machine learning classifiers

Hand-crafted features have been used commonly in computer vision problems such as image classification. They are determined by human experts for the related classification problem and the determined features are extracted from the raw image data by using various image processing methods. Then the classical machine learning algorithms use hand-crafted features for solving related classification problems. However, there are some drawbacks of hand-crafted features which affect the performance of the classifier such as inexperienced experts, the unsuccess of image processing methods in feature extraction. Therefore, the generalization ability of machine learning algorithms decreases in different datasets of the same problem. Hand-crafted features are started to be less preferred after deep learning models (CNNs), which self-learn models the necessary features to solve the classification problem from raw image data. So, in this study, we used convolutional layers of the fine-tuned Xception model as a feature extractor for machine learning classifiers instead of hand-crafted features. After the global average pooling layer, the obtained 2048 features were used to train classical machine learning methods. We compared the classification performance of fine-tuned CNN model with SVM [34], KNN [34], DT [34], NB [34], LR [34] algorithms. We utilized the popular machine learning framework scikit-learn. The detailed information about used algorithms in the study is following.

**Support Vector Machines:** The SVM is a popular supervised machine learning algorithm used in classification and regression problems. It can be used to solve linear or nonlinear problems. The main idea of SVM is to find an optimal line or hyperplane which separates the data into classes. The algorithm utilizes support vectors and margins to find optimal line or hyperplane between the classes. It is not fast compared to other machine learning algorithms but it has more accurate and is less prone to overfitting [34].

We tried SVM with different kernels such as linear, radial basis, sigmoid, and polynomial functions. But best result was achieved by the sigmoid function. The C and gamma parameters were determined as 1.0 and 0.0053 respectively.

**K-Nearest Neighbor:** The KNN is one of the simplest machine learning algorithms which is based on the similarity between the new data points and their nearest neighbors. It is a non-parametric algorithm and can be used for classification and regression problems. The K value represents the nearest neighbor number of the new data point. It is also known as a lazy algorithm because it doesn't learn from data at the training stage, it stores the train data and uses them at the test stage [34]. In this study, we determined the K value (72) by taking the square root of the training data.

**Naive Bayes:** NB is a classification algorithm based on Bayes' Theorem. So, its predictions are based on the conditional probabilities of data. The algorithm assumes every feature in the dataset is independent of other features and makes predictions on this assumption. In real-world problems, the features generally depend on each other so this is one of the drawbacks of the NB algorithm [34]. There are different types of NB algorithm. We utilized the Gaussian Naïve Bayes algorithm in this study.

**Decision Tree:** DT is a tree-based supervised algorithm that can be used for classification or regression problems. In a decision tree, internal nodes namely decision nodes represent features, branches represent decision rules and leaf nodes represent final decisions. There are different types of DTs such as CART, ID3, C4.5 according to the used homogeneity criterion (Gini index, Entropy) [34]. We preferred CART DT algorithm in this study.

**Logistic Regression:** LR is a supervised classification algorithm. It is a transformed version of linear regression by using the sigmoid function and cross-entropy loss function. But it uses for classification problems and gives probabilistic outcomes between 0-1. There are different types of logistic regression such as binomial, multinomial, and ordinal [34]. In this study, we used a multinomial version of LR. We used default parameters NB, DT, and LR in scikit learn framework.

### 3.6 Performance Metrics

The performances of all algorithms were validated by accuracy (Acc.), recall (sensitivity), specificity (Spe.), precision (Pre.), F1 score, average (Avg.) recall(sensitivity), average specificity, average precision, and average F1 score metrics. The formulas of all metrics are detailed below. In binary



classification problems, TP indicates the number of positive cases that are identified correctly by the classifier. TN indicates the number of negative cases that are identified correctly by the classifier. FP indicates the number of negative cases that are identified incorrectly positive by the classifier. FN indicates the number of positive cases that are identified incorrectly negative by the classifier. However, in multi-class problems, TP, TN, FP, and FN are calculated considering each class to other classes. Therefore, all metrics are calculated by class-based.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Recall - (Sensitivity) = \frac{TP}{TP + FN} \quad (2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$F1 - Score = 2x \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (5)$$

#### 4. Results

We evaluated all algorithms with test data (624). Table 5 indicates the classification results of algorithms. We also calculated the confusion matrices of each algorithm (see Figure 5).

Table 5 Classification results (%) of different machine learning algorithms

Method	Acc.	Avg. Pre.	Avg. Spe.	Avg. Recall	Avg. F1 score
DT	71.63	74.64	86.20	70.93	69.94
SVM	89.58	<b>89.59</b>	94.69	87.71	88.35
LR	77.24	79.48	89.28	77.25	75.51
NB	81.09	82.34	91.25	81.16	79.64
KNN	82.05	82.65	91.41	81.84	80.62
Xception	<b>89.74</b>	85.95	<b>94.92</b>	<b>89.74</b>	<b>89.72</b>

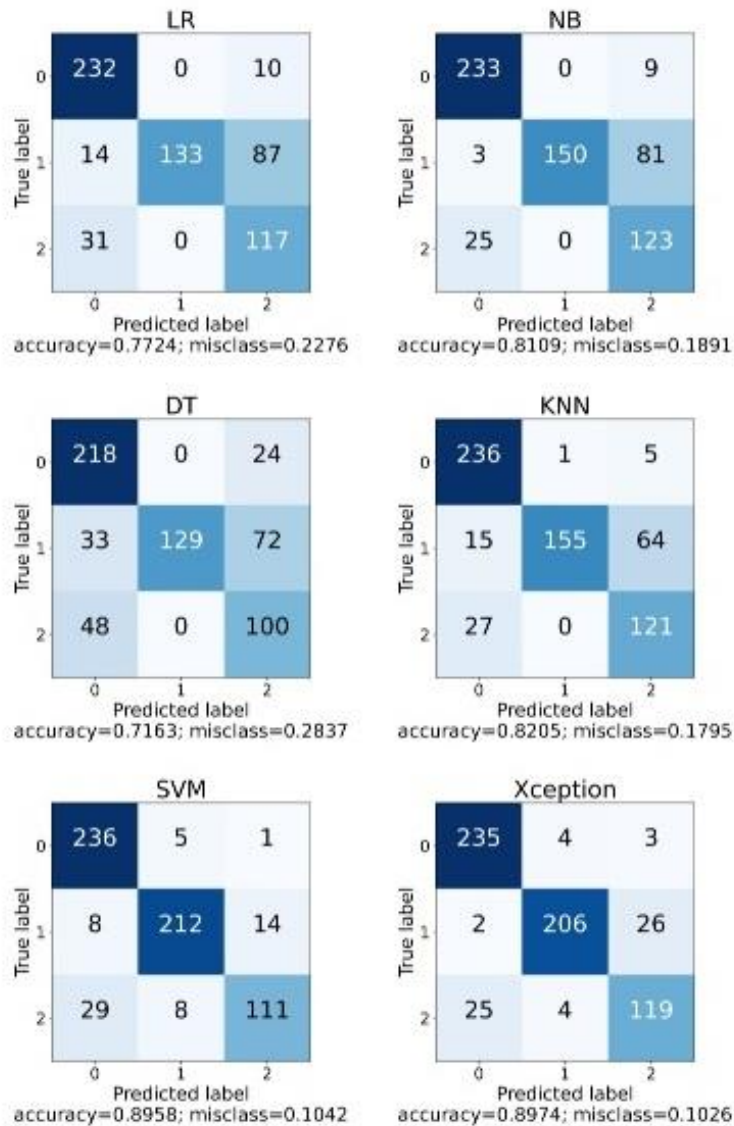


Figure 5 Confusion matrices of algorithms

## 5. Discussion

In this work, we compared the classification performances of classical machine learning algorithms with a fine-tuned CNN model. The five machine learning algorithms have been trained with the 2048 features extracted by the fine-tuned CNN model. The chest X-Ray images were classified as bacterial pneumonia (0), normal (1) and, viral pneumonia (2). According to Table 5, the fine-tuned CNN model is more successful than other classifiers in terms of all metrics. On the other hand, when the confusion matrices given in Figure 5 are examined, it is seen that the SVM and KNN algorithm is more successful than the others in detecting bacterial pneumonia. While the SVM was more successful in the classification of normal images, NB was outperformed other algorithms in the detection of viral pneumonia. When the assumption of independence of features, NB classifier performs better compared to other models, it needs less training data [34]. There are fewer viral pneumonia cases in the training data. Among the five machine learning algorithms, SVM is the most successful algorithm with an accuracy rate of 89.58%. SVM also achieved very close results in terms of average specificity (94.69), average recall (87.71), and average f1 score (88.35) considering the performance of fine-tuned Xception model. In addition, it outperformed fine-tuned Xception model at average precision metric (89.59) (see Table 5). According to confusion matrices in Figure 5 SVM is more successful than fine-tuned Xception model in detecting

bacterial pneumonia (1 more case) and normal cases (6 more cases). In contrast, the fine-tuned Xception model was better than SVM in detecting viral pneumonia cases (8 more cases). Table 6 and Table 7 shows class-based classification results of SVM and fine-tuned Xception model respectively. Recall (Sensitivity) is regarded as the most important metric in medical studies. SVM outperformed fine-tuned Xception model in terms of sensitivity metric in bacterial pneumonia and normal cases But, fine-tuned Xception achieved better sensitivity value at detecting viral pneumonia cases. As a result, fine-tuned Xception achieved better classification performance than other machine learning algorithms. In addition, it is clearly seen from Table 6 and Table 7 that the detection of bacterial pneumonia cases is easier, while viral pneumonia cases are more difficult to detect. The main reason for this situation is that the number of images of bacterial pneumonia during the training is more than the other two classes, and the model has learned this type of pneumonia better than others.

Table 6 Class based classification results (%) of SVM algorithm

Class	Pre.	Recall	Spe.	F1-score	Sample
0 (Bacterial Pneumonia)	86.45	97.52	90.51	91.65	242
1 (Normal)	94.22	90.60	96.73	92.37	234
2 (Viral Pneumonia)	88.10	75.00	96.84	81.02	148

Table 7 Class based classification results (%) of fine-tuned Xception model

Class	Pre.	Recall	Spe.	F1-score	Sample
0(Bacterial Pneumonia)	89.69	97.11	92.93	93.25	242
1(Normal)	96.26	88.03	97.94	91.96	234
2(Viral Pneumonia)	80.41	80.41	93.90	80.41	148

After the outbreak of Covid 19, most of the studies have been focused on differentiae pneumonia from Covid 19. There are few studies for detecting types of pneumonia and normal cases in chest X-Ray images. Prayogo et al used SCN models to classify chest X-Rays into 3 three classes normal, bacterial, and viral pneumonia. They achieved 80.03 accuracy and 79.59 F1-score values; our results are better than theirs in terms of all metrics. They used the same dataset used in our study [26]. Darici et al used a CNN ensemble voting methodology to diagnose viral pneumonia, bacterial pneumonia, and normal cases. They used three CNN models, two of them were designed by them and the last one was the pre-trained Inception V3 model. The used dataset in their study is the same as in our study [26]. However, they utilized offline data augmentation to balance the dataset. The classification performance of their CNN ensemble voting method is given in Table 8. We used an imbalanced dataset in our experiments and but our results are better than theirs. Mahmud et al designed a CNN model (CovXNet) utilizing depthwise separable convolutions to classify chest X-Rays into four classes (normal cases, viral pneumonia, bacterial pneumonia, and Covid-19). The used dataset in their study is the same within used in our study [26]. They used a two-step transfer learning methodology. First, they trained the model with chest X-Ray dataset which are including normal cases, viral pneumonia, and bacterial pneumonia images. Then, they trained again pre-trained model with chest X-Ray dataset which are including normal cases and Covid-19 cases. Their proposed CNN model achieved better results in terms of all metrics. However, designing and training a CNN model from scratch needs too much experience and effort. In addition, their inception model classification performance is less successful than our method (see Table 8).

On the other hand, retraining a pre-trained CNN model for a specific problem has some challenges. There are hyperparameters in the retraining process such as the number of freezing layers, fully connected layer size, fully connected neuron number, epoch size. Suitable hyperparameters are obtained after different experiments and this may take some time. This is the limitation of our study.

Table 8 The performance comparison of the proposed method with literature

References	Classes	Acc	Recall	Pre.	F1-score
Prayogo et al. [20]	3	80.03	80.03	79.23	79.59
Darici et al. [25]	3	78.00	75.00	77.0	75.00
Mahmud et al. [23]	3	91.70	92.10	92.90	92.60
Mahmud et al. (Inception) [23]	3	81.10	84.90	75.40	78.90
Proposed SVM	3	89.58	87.71	89.59	88.35
Proposed Xception	3	89.74	89.74	85.95	89.72

## 6. Conclusion

In this study, we trained a fine-tuned Xception model to classify chest X-Rays into three classes bacterial pneumonia, normal and viral pneumonia. Then, the fine-tuned Xception model was used as a feature extractor for various machine learning algorithms. Five machine learning algorithms (SVM, KNN, LR, NB, DT) were trained using the 2048 features extracted by the fine-tuned Xception model's convolutional layers. Then, we compared the classification performance of algorithms. According to the test results, the fine-tuned Xception model achieved better classification results with 89.74% accuracy than other classifiers. Among the machine learning classifiers, SVM algorithm achieved the best score with 89.58% accuracy. Our findings show that a pre-trained CNN model can be used successfully as a feature extractor instead of handcrafted features for different machine learning algorithms.

In future work, we will use multiple CNN models as feature extractors. Then, we will concatenate extracted features and will use them to train an ensemble of ML classifiers.

## Acknowledgments

Author 1: performed the design and implementation of the research, analysis of the results and writing the article. There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person / institution in the article prepared.

## References

- [1] D. You, G. Jones, and T. Wardlaw, "Levels & Trends in Child Mortality: Report 2011. Estimates Developed by the UN Inter-Agency Group for Child Mortality Estimation.," New York: United Nations Children's Fund 2011.
- [2] WHO, "Priority diseases and reasons for inclusion," in *Chapter 6.22-Pneumonia*, 2014.
- [3] O. Ruuskanen, E. Lahti, L. C. Jennings, and D. R. Murdoch, "Viral pneumonia," *The Lancet*, vol. 377, no. 9773, pp. 1264-1275, 2011.
- [4] D. E. Drake, A. Cohen, and J. Cohn, "National hospital antibiotic timing measures for pneumonia and antibiotic overuse," *Quality Management in Healthcare*, vol. 16, no. 2, pp. 113-122, 2007.
- [5] WHO, "Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children," Geneva: World Health Organization 2001.
- [6] M. I. Neuman *et al.*, "Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children," *Journal of Hospital Medicine*, vol. 7, no. 4, pp. 294-298, 2012.

- [7] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375-9389, 2017.
- [8] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221-248, 2017.
- [9] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," *Journal of Magnetic Resonance Imaging*, vol. 49, no. 4, pp. 939-954, 2019.
- [10] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*, 2010, pp. 253-256: IEEE.
- [11] M. A. Al-Antari, M. A. Al-Masni, and T.-S. Kim, "Deep learning computer-aided diagnosis for breast lesion in digital mammogram," *Deep Learning in Medical Image Analysis*, pp. 59-72, 2020.
- [12] H. Li, A. Li, and M. Wang, "A novel end-to-end brain tumor segmentation method using improved fully convolutional networks," *Computers in biology medicine*, vol. 108, pp. 150-160, 2019.
- [13] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.
- [14] P. Rajpurkar *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.
- [15] D. S. Kermany *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122-1131. e9, 2018.
- [16] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Computer Methods and Programs in Biomedicine*, p. 104964, 2019.
- [17] V. Chouhan *et al.*, "A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.
- [18] X. Gu, L. Pan, H. Liang, and R. Yang, "Classification of bacterial and viral childhood pneumonia using deep learning in chest radiography," in *Proceedings of the 3rd International Conference on Multimedia and Image Processing*, 2018, pp. 88-93.
- [19] A. Mittal *et al.*, "Detecting Pneumonia using Convolutions and Dynamic Capsule Routing for Chest X-ray Images," *Sensors*, vol. 20, no. 4, p. 1068, 2020.
- [20] K. A. Prayogo, A. Suryadibrata, and J. C. Young, "Classification of pneumonia from X-ray images using siamese convolutional network," *Telkomnika*, vol. 18, no. 3, pp. 1302-1309, 2020.
- [21] T. Rahman *et al.*, "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray," *Applied Sciences*, vol. 10, no. 9, p. 3233, 2020.
- [22] M. F. Hashmi, S. Katiyar, A. G. Keskar, N. D. Bokde, and Z. W. Geem, "Efficient pneumonia detection in chest xray images using deep transfer learning," *Diagnostics*, vol. 10, no. 6, p. 417, 2020.
- [23] T. Mahmud, M. A. Rahman, and S. A. Fattah, "CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization," *Computers in biology medicine*, vol. 122, p. 103869, 2020.
- [24] K. El Asnaoui, "Design ensemble deep learning model for pneumonia disease classification," *International Journal of Multimedia Information Retrieval*, vol. 10, no. 1, pp. 55-68, 2021.
- [25] M. B. Darici, Z. Dokur, and T. Olmez, "Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images," *International Journal of Intelligent Systems Applications in Engineering*, vol. 8, no. 4, pp. 177-183, 2020.
- [26] D. Kermany and M. Goldbaum, "Labeled optical coherence tomography (OCT) and Chest X-Ray images for classification," *Mendeley Data*, vol. 2, 2018.
- [27] J. Koushik, "Understanding convolutional neural networks," *arXiv preprint arXiv:1605.09081*, 2016.
- [28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251-1258.
- [29] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

- [30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255: IEEE.
- [31] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352-2449, 2017.
- [32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.
- [33] E. Ayan and H. M. Ünver, "Data augmentation importance for classification of skin lesions via deep learning," in *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1-4: IEEE.
- [34] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd, 2017.

# Calculation of Driving Parameters for GOA4 Signaling System using Machine Learning Methods

 Mehmet Taciddin AKÇAY<sup>1</sup>,  Abdurrahim AKGUNDOĞDU<sup>2</sup>

<sup>1</sup>Corresponding Author; Department of Electrical-Electronics Engineering, Faculty of Engineering, Halic University, Istanbul, Turkey; mehmettaciddinakcay@halic.edu.tr

<sup>2</sup>Department of Electrical-Electronics Engineering, Faculty of Engineering, Istanbul University Cerrahpasa, Istanbul, Turkey; akgundogdu@iuc.edu.tr

Received 5 May 2021; Revised 4 March 2022; Accepted 10 March 2022; Published online 30 April 2022

## Abstract

Among the electromechanical components of the rail system, the rail system vehicle is one of the most important units that carrying the passenger load. In terms of the efficiency of the signalization system, it is very critical to create the optimum vehicle driving profile. While many parameters of the vehicle are important during the design of the driving profile, determining the acceleration and braking accelerations directly affects this characteristic. The use of programmable devices and software instead of human factors is becoming more and more widespread day by day with the developed technology in rail transportation systems. Among the software used, artificial intelligence and machine learning applications constitute a large share in the general distribution. Especially, in the driverless (GOA4) signaling systems, these softwares become more important. In this study, the estimation of Vehicle Acceleration and Braking Acceleration with travel time has been done by using Machine Learning Methods. The ideal results obtained were given comparatively and interpreted on the graphics.

**Keywords:** Acceleration, braking, driverless, machine learning, vehicle.

## 1. Introduction

Today, the rail system technology is constantly renewed and updated in the best way to respond to the needs of the age at an adequate level. Especially the high development in the field of electrical-electronics is of interest to the systems used in this field, and new systems are replaced by old systems. With the high performance achieved in systems where communication and data systems used in rail system technology in our age are integrated, the systems leave themselves to automatic software and control methods. Especially with the efficient management of data with big data technology, great advantages have been provided to the rail system operation in controlling and controlling the systems. High success is achieved in machine learning applications, artificial intelligence algorithms, and applications with the wide data networks obtained from the systems. These studies also help companies in optimization of operational traffic by increasing their management skills. It is the most important subsystem signaling system that closely concerns electromechanical issues, operating performance, and RAMS competencies in rail systems. While designing the signalization system, compliance with EN 50126, EN 50128, and EN 50129 standards is the most basic principle. The rating of the signaling system according to the automation level (GOA) is specified in the IEC 62290 standard. Among these ratings, GOA 4 has the highest performance in terms of the highest automation level and features it provides. At this level, the vehicle is operated completely without a driver, while the system is managed with a full automation level. It is important to minimize the problems experienced for the signalization system, obtain the targeted RAMS values, and increase the performance. Figure 1 gives the graphic expressing the situations related to the malfunctions that occur.

Actions are developed against the problems experienced in the business according to the decrease, the same progress, and increase of the malfunctions. Therefore, possible errors in the signaling system are prevented with the highest automation level following SIL 4 criteria. At the GO4 signaling level, the acceleration and braking (driving characteristic) of the vehicle are automatically activated.

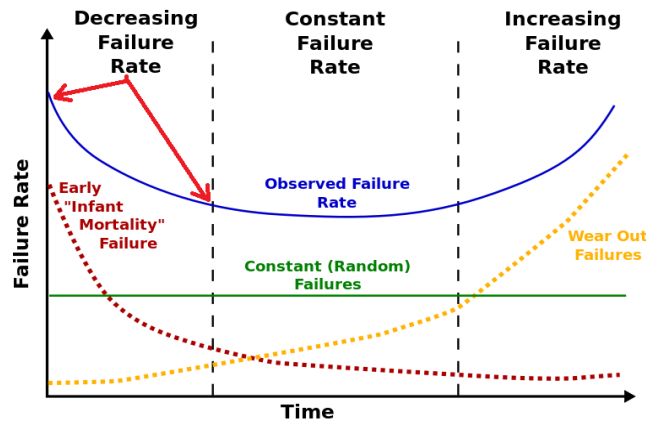


Figure 1 Failure Rate Situation

In this study, the estimation of Vehicle Acceleration and Braking Acceleration has been carried out using Machine Learning Methods. In the second part of this article, the model proposed for the estimation of vehicle acceleration and braking acceleration, and the input-outputs used to create the structure of the model and the proposed method are explained. By presenting the simulation results obtained in the third section, a comparison has been made with other similar methods. In the last part, the evaluation of the article is given in detail. Various studies have been carried out in the literature on this subject. While estimating the arrival times of vehicles with vehicle speed measurement done in the study [1], the method for calculating vehicle motion resistance with the help of a vehicle data collection device is explained with [2]. In [3], the prediction of critical speed for the vehicle on soft ground is studied, and in [4] a low-speed estimation has been made with the time signal-based warping algorithm. In [5], the simulation of the vehicle movement has been done by the element increment method. While traffic speed estimation is made with the deep learning method in [6], a study has been done on speed prediction models with [7] and hybrid systems have been investigated. In [8], the estimation of vehicle motion resistance with the help of a vehicle onboard system was examined. While the estimation of the cruise speed is made using learning methods in [9], in [10] the situation of estimating the vehicle speed according to the energy consumption has been investigated. While estimating the average vehicle speed according to weather conditions and traffic characteristics with [11], in [12] the vehicle travel speed was estimated by using machine learning methods. While real-time arrival estimates were made in light rail systems with [13] and [14], vehicle speed was calculated with ground vibrations in [15]. In this study, the performance outputs used by the signaling system and effectiveness in the vehicle speed profile were estimated by machine learning methods.

## 2. Material and Method

### 2.1 Experimental Study and Simulation

In this study, the distance between two stations with the setup set up, the acceleration, braking acceleration, and travel time parameters of the signaling system was used for the structure of the design. Each row from the data sets represents the data sets obtained separately from each other for all data sections. While the system was in operation, records were taken and the characteristics of the speed profiles were reached. Figure 2 shows the experimental setup.

After the driving algorithm is applied, speed position profiles are formed and the data for the specific acceleration curve for the vehicle with the following query code sequence below is taken from the recorded part. Figure 2 shows how the driving profile is created with the characteristic driving parameters of the vehicle. This curve is limited according to the operating speed limits, and starting and braking situations are created according to the conditions determined in the operation with the control blocks.



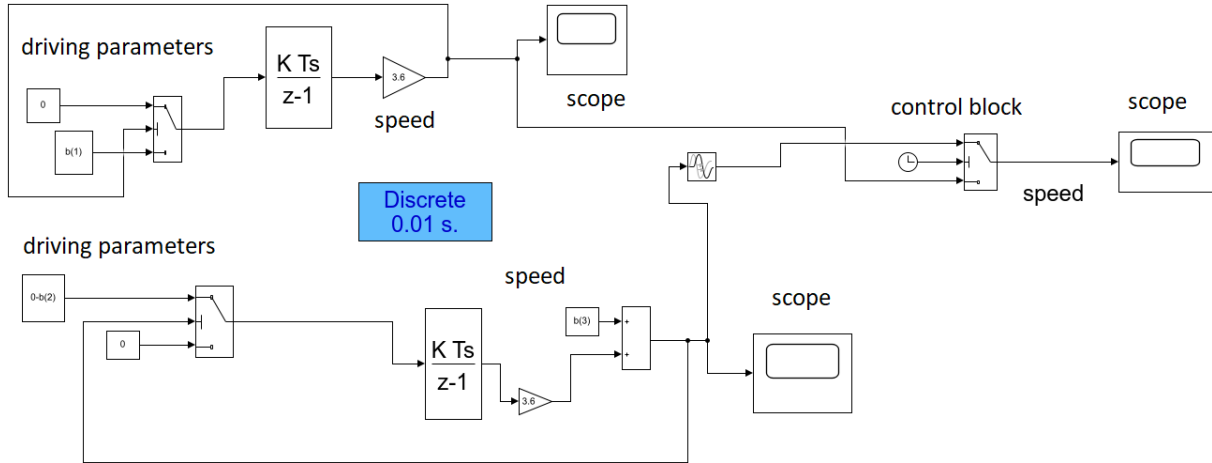


Figure 2 Experimental setup

Code 1 Query

```

1  for datas=1:length(speed.signals.values)
2  if speed.signals.values(datas)>=max operation speed
3  acceleration(n)=speed.time(datas);
4  n=n+1;
5  end
6  if speed.signals.values(datas)<0
7  total(m)=zaman.time(datas);
8  m=m+1;
9  end
10 end
    
```

## 2.2 Data Set

Within the scope of the study, 300 data arrays were used, while the distance between stations and maximum operating speed were used as input parameters, vehicle acceleration, deceleration, and travel speed were used as output parameters. The distribution of some of the data is given in Figure 3.

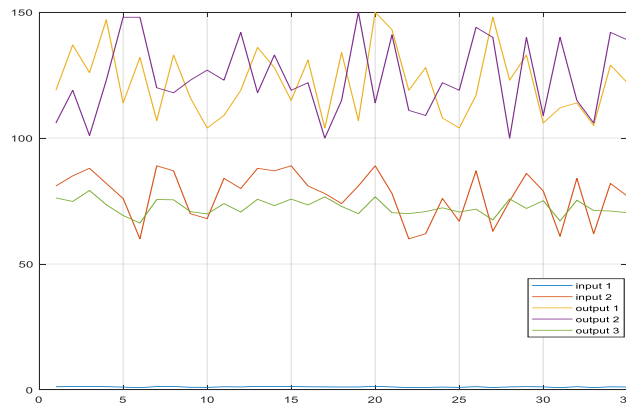


Figure 3 Graph of a section of data

The number of data was chosen according to the success of the methods, and the data realized under operating conditions were preferred. Values ranging from 60 to 90 km/h were used for the maximum operating speed. As the output, the acceleration values of the vehicle in the starting and braking states were used, and this data varies between 1 and 1.5 m/s<sup>2</sup>. The distance between the stations was taken between 1 and 2 km since this design was made for the metro stations.

## 2.3 Machine Learning Methods

Machine learning is the computer modeling of systems that make predictions by making inferences from operations on data using mathematics and statistics. The model is created with the current data set and the algorithm used. Machine learning is used to get the maximum performance from models. In this study, Linear Regression, Random Forest, Multi-Layer Perceptron, and K-Neighbors Regressor models, which are among the supervised learning methods, were applied to the data.

### 2.3.1 Linear Regression

Linear regression is a method used to model the relationship between one or more independent variables and another dependent variable. The purpose of linear regression; is to find the values of  $\beta_i$  using given  $x$  and  $y$ . Once the  $\beta_i$  values are found,  $y$ 's of unknown value can be predicted from the  $x$  values [16,17]. Linear regression can be formulated as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon_i \quad (1)$$

In this equation;

$y_i$ : dependent variable

$x_i$ : explanatory variables

$\beta_0$ : y-intercept (constant term)

$\beta_i$ : is the regression coefficient

$\beta_p$ : slope coefficients for each explanatory variable

$\epsilon$  = the model's error term.

### 2.3.2 Random Forest

Random Forest is a supervised learning algorithm. The Random Forest creates multiple decision trees and combines them with the bagging method to obtain a more accurate and stable forecast. Figure 4 shows a simple Random Forest model.

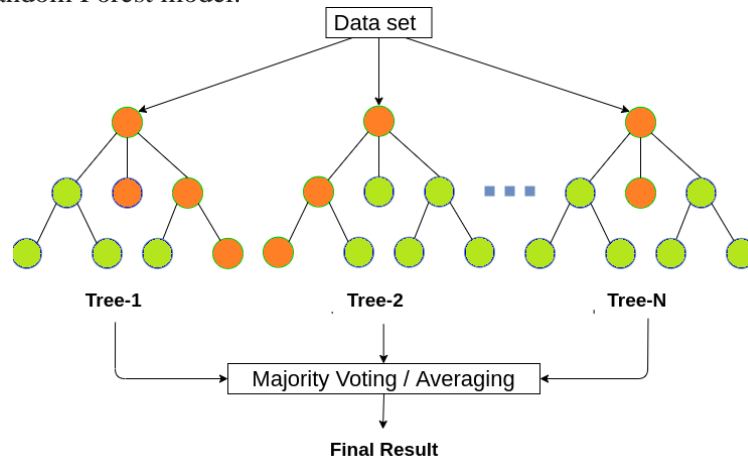


Figure 4 An example of the random forest with N decision trees [18]

The main advantage of the random forest is that it can be used for both classification and regression problems that make up most of the current machine learning systems [19, 20]

### 2.3.3 Multi-Layer Perceptron (MLP)

MLP is the simplest in artificial neural networks. It consists of three layers: input, hidden, and output. The input layer is entered into the system and is the layer where it is processed towards the hidden layer. In these networks, error learning is performed by distributing backward in each transaction cycle [21]. A nonlinear activation function transmits the sum of the weighted input signals. The actual observation

results are compared with the results of the network and the error of the network is calculated. Then the calculated network error is propagated back by the system and the weights of the coefficients are updated [22].

MLP generally contains one or more hidden neuronal layers. After these neurons are the output layer of the neurons. The network learns the linear and nonlinear relationships between input and output vectors through transfer functional neuron layers. An example MLP model is shown in Figure 5.

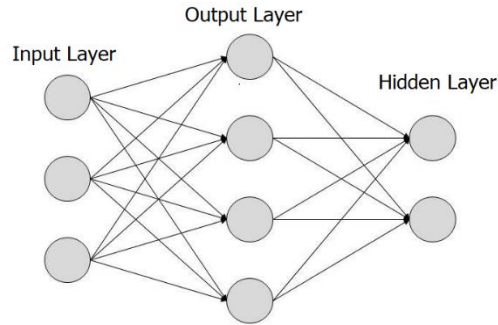


Figure 5 Example of an MLP model with three inputs and two outputs

### 2.3.4 K Neighbors Regressor

K-NN, which is sensitive to the distance function, is a non-parametric learning algorithm. The main idea of the algorithm is that if the most similar training data in the feature space belongs to a cluster, it includes the training data in this cluster. To determine to which cluster the training data in the feature space belongs to, the distance between training data is determined by distance functions such as Euclidean, Manhattan, Minkowski, and Kullback-Leibler [23, 24].

## 2.4 Performance Calculations

### 2.4.1 Statistical Performance Validation

In this study, Root Mean Square Error (RMSE) and coefficient of determination ( $R^2$ ) values were calculated. These performance values can be formulated as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (T_i - Y_i)^2}{N}} \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (T_i - Y_i)^2}{\sum_{i=1}^N (T_i - \bar{T})^2} \quad (3)$$

where  $T_i$  is the targets,  $\bar{T}$  is the mean of all target values,  $Y_i$  is the neural network outputs and  $N$  is the number of samples. These performance results were used as a reference for the comparisons about the all methods [25].

## 3. Results and Discussion

In this study, acceleration braking acceleration and travel times are estimated simultaneously with three different machine learning methods. While performing regression, the data were subjected to training and testing processes in 10 groups with the cross-validation method. The regression curves obtained after each method are shown in Figure 6, Figure 7, Figure 8, and Figure 9 respectively. RMSE values

and  $R^2$  values obtained after the estimates are shown in Table I. When Table I is examined, it is seen that the most successful method is the MLP.

Table 1 Performance evaluations of different models

Model	RMSE	$R^2$
Random Forest	9.79	0.30
kNN	8.88	0.42
Linear Regression	8.04	0.52
MLP	7.69	0.56

Figure 6 shows the regression distribution related to the estimation results of the kNN method, and a partial success has been achieved in this case.

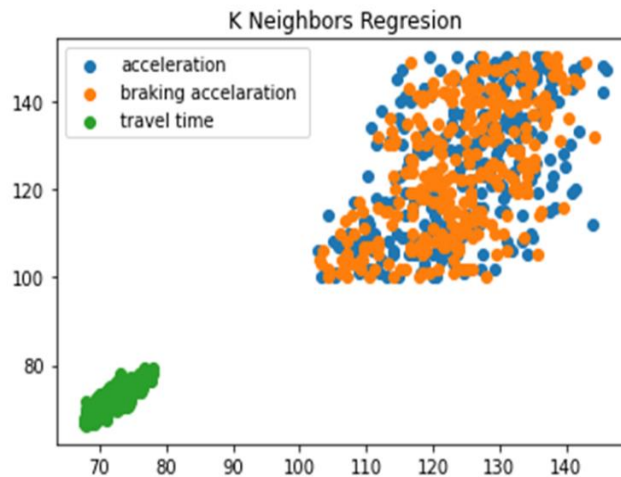


Figure 6 Regression curves of models (kNN)

The regression distribution of the Random Forest method is given in Figure 7, and in this case, it can be understood from the rate of deviation at the 45 degree slope line as it can be seen that less successful results are obtained compared to the kNN method.

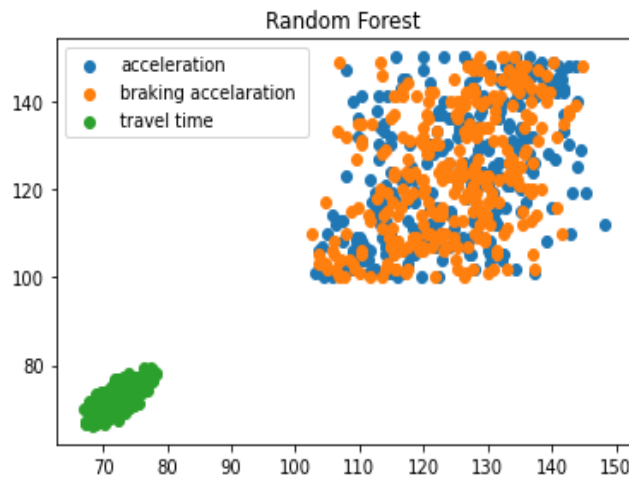


Figure 7 Regression curves of models (Random Forest)

Figure 8 shows the regression distribution related to the estimation results of the Linear Regression method, and in this case, the most successful results were obtained compared to the previous two methods.

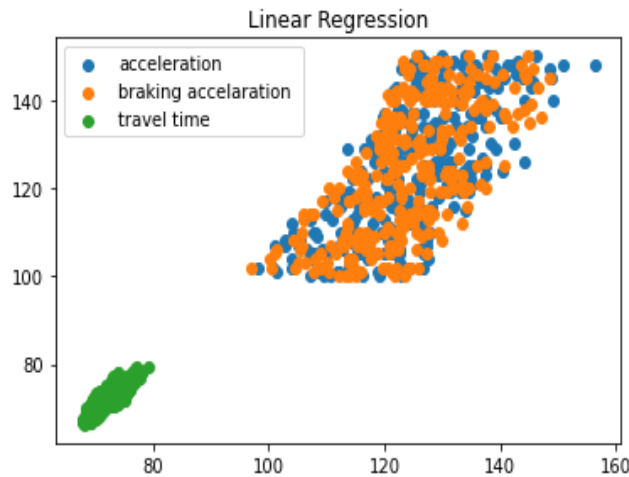


Figure 8 Regression curves of models (Linear Regression)

While Figure 9 shows the regression distribution related to the estimation results of the MLP method, in this case, the most successful estimation results compared to the previous methods were obtained. In this case, while the  $R^2$  value is obtained as 0.56, it is understood that the results achieved under these conditions are close to ideal and success, and at the same time, the results seem to be at a usable level.

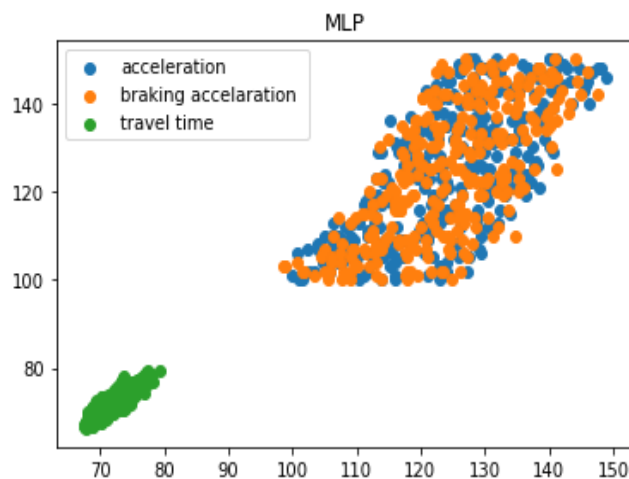


Figure 9 Regression curves of models (MLP)

The results obtained for different methods are given in Table 2. The SGD method proposed in this study and the most successful results are obtained, and the performance values of AdaBoost, Random Forest, Neural Network, kNN, Decision Trees and SVM methods, which are frequently used in the literature, can be seen in these tables, respectively.

#### 4. Conclusion

In this study, the driving parameters of the driverless signaling system with GOA4 technology used in subway lines were estimated using machine learning methods. Speed profiles for the vehicle were supported by calculating vehicle acceleration, braking acceleration, and travel times. The performance values of the results obtained with the study using Random Forest, kNN, Linear Regression, and MLP methods are given on the table and interpreted. RMSE and  $R^2$  values were obtained in calculations as performance criteria. The study aims to integrate new technologies used in the design of automatic driverless systems with machine learning and artificial intelligence applications and to increase operational performance. Analyzes can be diversified with the help of new machine learning methods, artificial intelligence applications, and algorithms by using different data series and types for the estimation of signaling system parameters with future studies.

## Acknowledgments

We would like to thank Istanbul Metropolitan Municipality, Rail System Department, for its support during the realization of this study.

## References

- [1] Y. Chen and L. R. Rilett, "A train speed measurement and arrival time prediction system for highway-rail grade crossings", *Transportation Research Record Journal of the Transportation Research Board*, vol. 2608, no.1., pp. 96-104, 2017.
- [2] T. Ogawa, S. Manabe, G. Yoshikawa, Y. Imamura, and M. Kageyama, "Method of Calculating Running Resistance by the Use of the Train Data Collection Device", *Quarterly Report of RTRI*, no. 58, pp. 21-27, 2017.
- [3] K. N. Cosgriff, E. G. Berggren, A. M. Kaynia, N. N. Dam, and N. A. Mortensen, "new method for estimation of critical speed for railway tracks on soft ground", *International Journal of Rail Transportation*, vol. 6, no. 4, pp. 203-217, 2018.
- [4] S. Hensel, M. Marinov, "Time Signal Based Warping Algorithms for Low Speed Velocity Estimation of Rail Vehicles", *Annual Journal of Electronics*, no. 8., pp. 177-180, 2014.
- [5] G. Xu, F. Li, J. Long, and D. Han, "Train movement simulation by element increment method", *Journal of advanced transportation*, no. 50, pp. 2060–2076, 2017.
- [6] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A Deep Learning Model for Traffic Speed Prediction", in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018, pp. 3470-3476.
- [7] A. Dhamaniya and S. Chandra, "Speed Prediction Models for Urban Arterials under Mixed Traffic Conditions", *Procedia-Social and Behavioral Sciences*, no. 104, pp. 342 – 351, 2013.
- [8] S. Aradi, T. Becsi, and P. Gaspar, "Estimation of running resistance of electric trains based on on-board telematics system", *International Journal of Heavy Vehicle Systems*, no. 22, pp. 277-291, 2015.
- [9] M. Gmira, M. Gendreau, A. Lodi, and J. Potvin, "Travel Speed Prediction Based on Learning Methods For Home Delivery", *Interuniversity Research Center On Business Networks, logistics and transport*, pp. 1-34, 2018.
- [10] M. Bysveen, *Vehicle speed prediction models for consideration of energy demand within road design*, Norwegian University of Science and Technology, Civil and Environmental Engineering, Master's Thesis, 2017.
- [11] B. Mirbaha, M. Saffarzadeh, S. A. Beheshty, M. Aniran, M. Yazdani, and B. Shirini, "Predicting Average Vehicle Speed in Two Lane Highways Considering Weather Condition and Traffic Characteristics", *IOP Conference Series: Materials Science and Engineering*, pp. 1-7, 2017.
- [12] M. Gmira, M. Gendreau, A. Lodi, and J. Potvin, "Travel speed prediction using machine learning techniques", *ITS World Congress*, pp. 1-10, 2017.
- [13] E. Naye, *Real-time arrival prediction models for light rail train systems*, Royal Institute of Technology, Department of Engineering, Master's Thesis, 2014.
- [14] O. Cats, "Real-Time Predictions for Light Rail Train Systems", *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-10, 2014.
- [15] G. Kouroussis, D. P. Connolly, M. Forde, and O. Verlinden, "Train speed calculation using ground vibrations", *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 229, no. 5, pp. 466-483, 2015.
- [16] D. Freedman, *Statistical Models: Theory and Practice*, Cambridge: Cambridge University Press, pp.25, 2005.
- [17] A. C. Rencher, and W. F. Christensen, "Chapter 10, Multivariate regression – Section 10.1, Introduction", *Methods of Multivariate Analysis, Wiley Series in Probability and Statistics, 709 (3rd ed.)*, John Wiley & Sons, pp. 19, 2012.
- [18] Available: <https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/>

- [19] T. K. Ho, “Random Decision Forest”, *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, 14-16 August, pp. 278-282, 1995.
- [20] L. Breiman, “Random Forests”, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [21] J. A. Freeman and D. M. Skapura, “Neural Networks Algorithms”, *Applications and Programming Techniques*, New York, USA: Addison-Wesley Publishing Company, 1991.
- [22] G. J. Garret and C. Wu, “Knowledge-based modeling of material behavior with neural networks”, *Journal of Engineering Mechanics*, vol. 117, no.1, pp. 132-153, 1991.
- [23] T. M. Cover, and P. E. Hart, “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [24] J. Walters-Williams, Y. Li, “Comparative study of distance functions for nearest neighbors”, *In Advanced Techniques in Computing Sciences and Software Engineering*, Springer, Dordrecht, pp. 79-84, 2010.
- [25] M. Akçay, “Estimation of Constant Speed Time for Railway Vehicles by Stochastic Gradient Descent Algorithm”, *Sakarya University Journal of Computer and Information Sciences*, vol. 3, no. 3, pp. 355-365, 2020.

# A Comparative Study On COVID-19 Prediction Using Deep Learning And Machine Learning Algorithms: A Case Study On Performance Analysis

 Hilal ARSLAN<sup>1</sup>,  Orhan ER<sup>2</sup>

<sup>1</sup>Corresponding Author; Ankara Yıldırım Beyazıt University, [hilalarslanceng@gmail.com](mailto:hilalarslanceng@gmail.com)

<sup>2</sup>Izmir Bakırçay University; [orhan.er@bakircay.edu.tr](mailto:orhan.er@bakircay.edu.tr)

Received 04 May 2021; Revised 17 August 2021; Accepted 29 March 2022; Published online 30 April 2022

## Abstract

COVID-19 disease has been the most important disease recently and has affected serious number of people in the world. There is not proven treatment method yet and early diagnosis of COVID-19 is crucial to prevent spread of the disease. Laboratory data can be easily accessed in about 15 minutes, and cheaper than the cost of other COVID-19 detection methods such as CT imaging and RT-PCR test. In this study, we perform a comparative study for COVID-19 prediction using machine learning and deep learning algorithms from laboratory findings. For this purpose, nine different machine learning algorithms including different structures as well as deep neural network classifier are evaluated and compared. Experimental results conduct that cosine k-nearest neighbor classifier achieves better accuracy with 89% among other machine learning algorithms. Furthermore, deep neural network classifier achieves an accuracy of 90.3% when one hidden layer including 60 neurons is used to detect COVID-19 disease from laboratory findings data.

**Keywords:** COVID-19 disease, SARS-CoV-2, laboratory data, machine learning, deep learning

## 1. Introduction

Severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2) belongs to a betacoronavirus family and has caused a disease known as COVID-19. COVID-19 has been announced as a worldwide pandemic and has caused many deaths worldwide. According to the World Health Organization (WHO) report published on March 9, 2021, the total number of patients who were positive in the diagnosis of COVID-19 is 119,450,269 and the number of patients who died from COVID-19 is 2,647,662. The common symptoms of the disease are tiredness, cough, fever, sore throat as well as problems in breathing and there is no proven treatment for COVID-19. Zoabi et al. [1] investigated importance of these symptoms and they reported that headache and sore throat were identified as the most important symptoms. Several studies have shown that people who have a chronic respiratory disease, older and male are more affected by COVID-19 disease. Furthermore, people with crucial medical diseases like cancer and cardiovascular illness are affected seriously from COVID-19.

This pandemic has led to an exponential increase in hospitalization demands. Furthermore, it continues to challenge the healthcare system due to shortages in medical equipment. It is important to forecast people who will be more likely to develop severe illness including death. The most common technique used in the diagnosis of COVID-19 is Reverse transcriptase polymerase chain reaction (RT-PCR); however, some countries have not enough RT-PCR tests, which causes infection rates to increase sharply. Another common technique to diagnose COVID-19 is to use computer tomography (CT) scans. However, they are unable to discriminate between COVID-19 and other illnesses like flu. Moreover, they are not effective in screening for SARS-CoV-2 in the general population. To overcome all these limitations and effectively use clinical decision-making equipment as well as healthcare resources, machine learning and deep learning aided systems have been developed, recently.

Machine leaning techniques are actively used in COVID-19 detection from genome sequences [2,3] and common symptoms of COVID-19 [4]. Furthermore, they are used to estimate the severity of COVID-19 infected into the patient [1]. Besides, deep learning techniques are efficiently used to predict COVID-



19 from medical imaging [5,6]. They are also efficiently used to interpret clinical findings from various types of cancers [7] and biomedical studies [8].

In this study, we provide a COVID-19 detection method by performing machine and deep learning techniques using laboratory findings of the patients. The rest of the paper is organized as follows. In Section 2, Explanatory information about the machine learning and deep learning techniques performed in this study is given. In Section 3, material and methods are given. In Section 4, the experimental results obtained as a result of using machine learning and deep learning methods are presented and interpreted. The last section presents important results that can be shown as a guide for future studies.

## 2. Related Work

In this section, we give machine and deep learning algorithms detecting COVID-19 positive cases from laboratory findings and symptoms rather than RT-PCR or CT imaging. Although these studies are limited in the literature, many state-of-the-art machine [9, 10] and deep learning algorithms [6] recently published to detect COVID-19 from medical images or genome sequences can be found in [11,12, 13, 14].

Zoabi et al. [1] developed a machine learning algorithm to diagnose COVID-19. They designed their methods based on basic information and symptoms without using any medical equipment. The features of their dataset include information about cough, fever, sore throat, shortness of breath, headache as well as sex and age information. Their method achieves an auROC of 0.86. Cabitza et al. [15] evaluated five machine learning techniques on the data including blood tests. They performed logistic regression, naive bayes, random forest, SVM, and KNN methods. They achieve satisfactory results and concluded that machine learning techniques based on blood tests can detect COVID-19 cases fast compared to the RT-PCR tests. Unal and Dudak [16] implemented naive bayes, SVM, KNN, and decision tree methods on the dataset including 19 features which are sex, age, the state of pneumonia as well as the state of various types of diseases such as asthma, diabetes, kidney failure, and hypertension. They showed that SVM achieved an accuracy of 100%. Alakus and Turkoglu [5] compared deep learning approaches to diagnose COVID-19 using laboratory findings. Their methods achieve an accuracy of 68.6%. Jiang et al. [17] developed an artificial intelligence tool to predict patients at risk for COVID-19. They used the data containing 11 features which are blood count, hemoglobin, temperature, Na<sup>+</sup>, creatinine, K<sup>+</sup>, a liver enzyme, myalgias, gender, lymphocyte count, and age. They performed logistic regression, KNN, decision tree, and SVM classifiers. They concluded that myalgias, hemoglobin, and the liver enzyme are important features and the most predictive. Batista et al. [18] used SVM, random forest, neural network, gradient boosted trees and logistic regression to diagnose COVID-19 using laboratory findings. Their method achieved the best AUC score with 0.84 when SVM and random forest methods are used. Schwab et al. [19] evaluated predictive models using logistic regression, neural network, random forest, different SVM methods and gradient boosting using demographic, clinical and blood analysis data containing 111 features. They obtained the best performance with 66% AUC score when gradient boosting method is used. Göreke et al. [20] proposed a new architecture using deep neural networks for diagnosing COVID-19 from laboratory findings. Shaban [21] proposed a novel hybrid diagnosing system based on deep neural network and fuzzy inference engine.

## 3. Material and Methods

In this section, the data sets and methods used in the training of artificial intelligence algorithms proposed for the solution of the COVID-19 prediction problem will be explained.

### 3.1 Data Description

The main data set used for this study was obtained from routine blood test results performed on 600 patients on admission to the emergency room at San Raffaele Hospital (OSR) between 19 February 2020 and 31 May 2020 [22]. COVID-19 positivity for each case was determined according to the result of the SARS-CoV-2 molecular test performed with RT-PCR on nasopharyngeal swabs. All samples have eighteen features and given in Table 1.

Table 1 The features that are used.

Name	Abbreviation
Hematocrit	HTC
Hemoglobin	HGB
Platelets	Thrombocytes
Red blood Cells	RBC
Lymphocytes	L
Leukocytes	WBC
Basophils	
Eosinophils	EBM
Monocytes	-
Serum Glucose	-
Neutrophils	N
Urea	-
Proteina Creativa	PCR
Creatinine	CR
Potassium	K
Sodium	NA
Alanine transaminase	ALT
Aspartate transaminase	AST

### 3.2 COVID-19 Detection Using Machine Learning Algorithms

Machine learning teaches computers the natural learning ability of humans. It also achieves this through experiences. Machine learning algorithms learn data without depending on a predetermined equation. Machine learning uses three types of methods as shown in Figure 1:

1. Supervised learning that trains known data to predict future outputs,
2. Unsupervised learning detecting hidden patterns in input data,
3. Reinforcement learning helping you to take your decisions sequentially based on interacting with the environment.

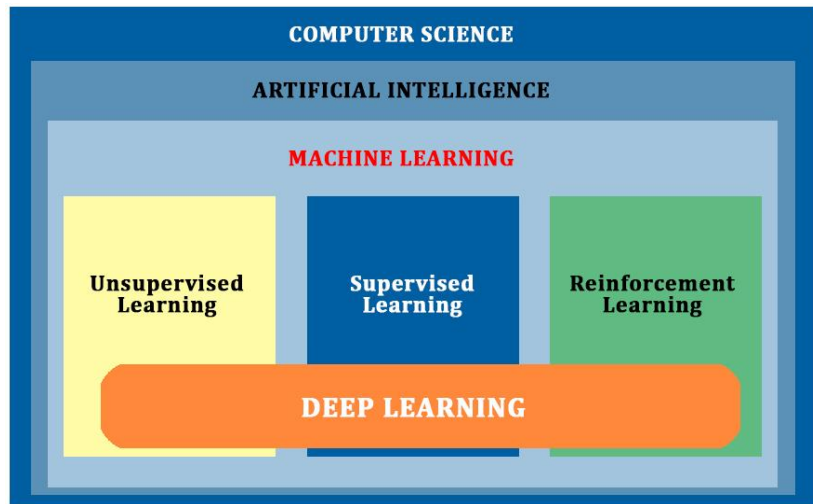


Figure 1 The Relationship between machine learning and deep learning in computer science [23]

The most well-known machine learning models in the field of artificial intelligence are given in Figure 2 [23].

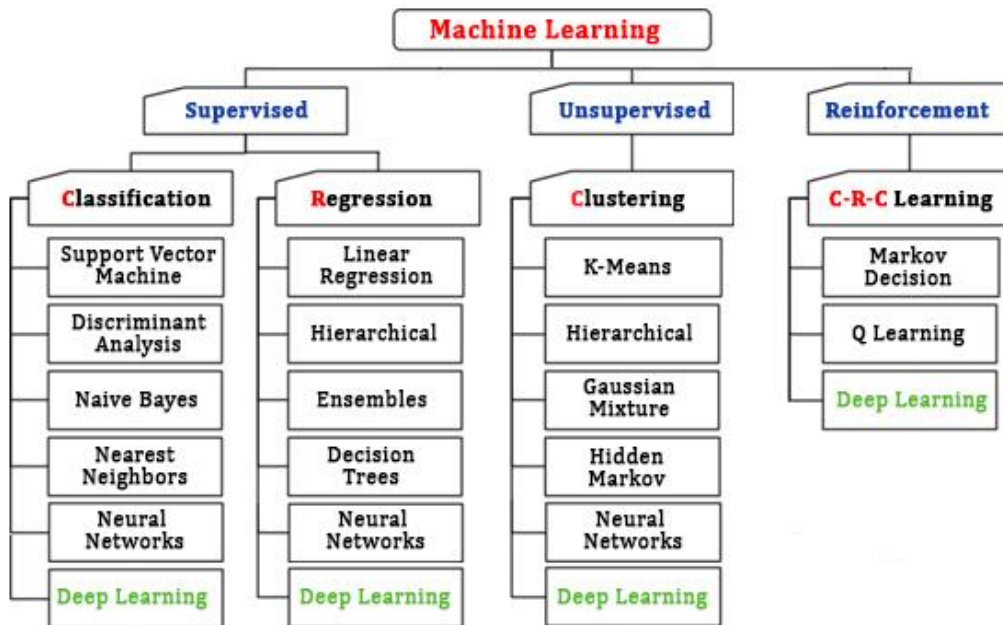


Figure 2 Machine learning models

Supervised machine learning enables the network to learn by making evidence-based predictions from the data. A supervised learning algorithm performs the training of the network by taking a known set of input data for examples and known output data as shown in Figure 3. It creates a mathematical model to generate reasonable estimates that fit the output data. Supervised learning using regression and classification techniques is used to develop predictive models.

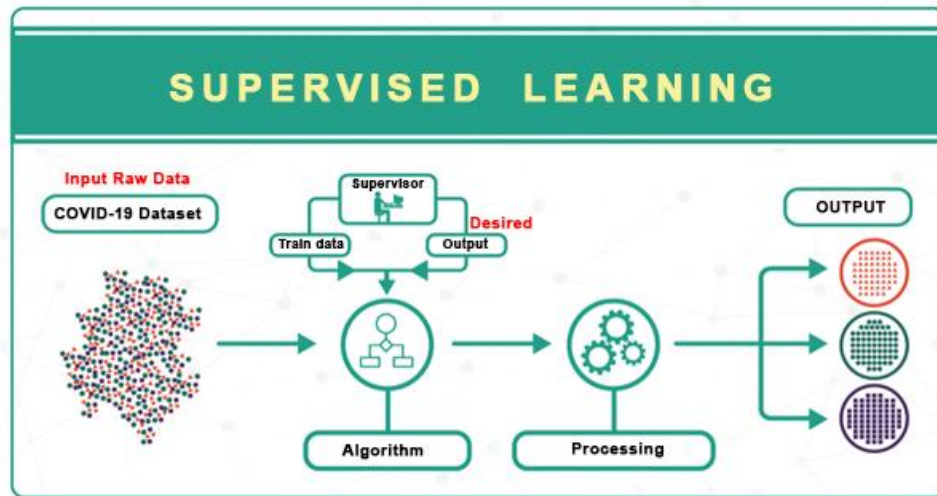


Figure 3 Supervised learning [23]

Unsupervised learning finds hidden patterns or intrinsic structures in data as shown in Figure 4. It performs the learning process by making inferences from data sets consisting of input data without labeled outputs. Clustering is the most common unsupervised learning technique. It is used to find meaningful relationships or groups hidden in data in exploratory data analysis processes. Market research, object recognition and gene sequence analysis are the most used research areas of clustering [23].

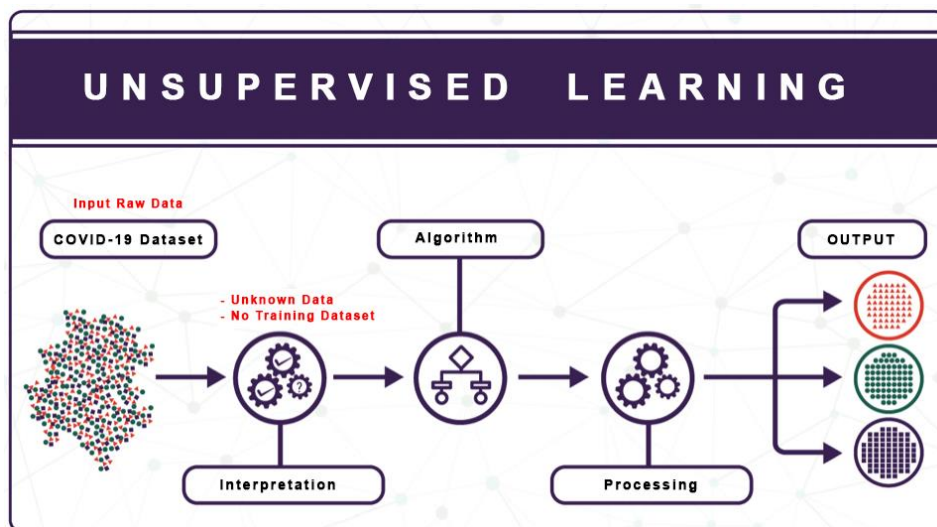


Figure 4 Unsupervised learning [23]

Reinforcement learning, unlike unsupervised learning, provides a high control area in the learning process as shown in Figure 5. Reinforcement learning is based on the concept of unsupervised learning and requires software agents to determine its ideal behavior in data. This structure has been created in a way that helps the performance of the machine to grow. In order to help the train the network, an operator is informed about the progress of the network with simple feedback [23].

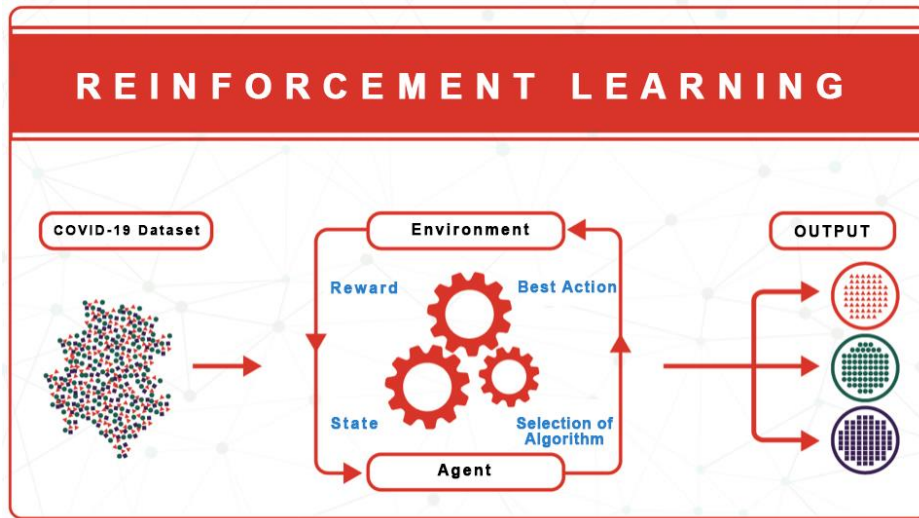


Figure 5 Reinforcement learning [23]

In this study, since the detection of COVID-19 is a classification problem, only classification methods were used among these 3 models. The most preferred deep learning techniques, machine learning and its derivatives were used for the detection process on Covid-19 data. These techniques are given in Table 2.

Table 2 Different algorithms to detection of COVID-19.

Neural Network		SVM		Trees	
1	Narrow Neural Network	15	Linear SVM	29	Boosted Trees
2	Medium Neural Network	16	Quadratic SVM	30	Bagged Trees
3	Wide Neural Network	17	Cubic SVM	31	Subspace Discriminant
4	Bilayered Neural Network (FF)	18	Fine Gaussian SVM	32	RUSBoosted Trees
5	Trilayered Neural Network (FF)	19	Medium Gaussian SVM	33	Fine Tree
6	Probabilistic Neural Network	20	Coarse Gaussian SVM	34	Medium Tree
7	Learning Vector Quantization	<b>KNN</b>		35	Coarse Tree
8	Multilayer NN (1 Hidden Layer)	21	Fine KNN	<b>Deep Neural Network</b>	
9	Multilayer NN (2 Hidden Layer)	22	Medium KNN	36	Deep NN with CNN layer pre-layer (1 Hidden Layer)
10	Multilayer NN (3 Hidden Layer)	23	Coarse KNN		
<b>Naive Bayes</b>		24	Cosine KNN	37	Deep NN with CNN layer pre-layer (2 Hidden Layer)
11	Gaussian naive Bayes	25	Cubic KNN		
12	Kernel Naive Bayes	26	Weighted KNN	38	Deep NN with CNN layer pre-layer (3 Hidden Layer)
<b>Discriminant</b>		27	Subspace KNN		
13	Linear Discriminant	<b>Regression</b>		Used to detection of COVID-19	
14	Quadratic Discriminant	28	Logistic Regression		

The following systematic machine learning workflow shown in Figure 6 is used in this study. In this way, it helped to overcome machine learning challenges for detection of COVID-19.



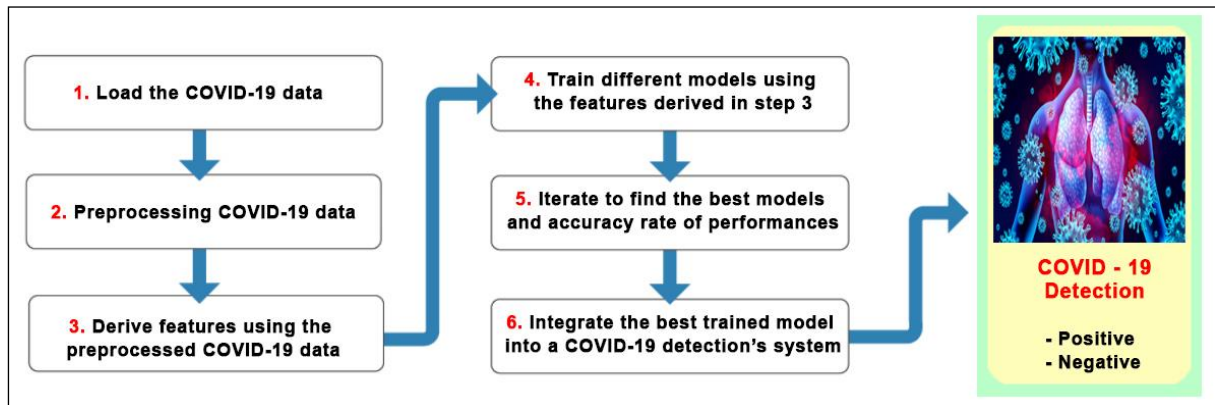


Figure 6 Operation steps of the COVID-19 model

### 3.3 COVID-19 Detection Using Deep Neural Network Algorithm

Deep learning is a class of machine learning that teaches computers the skills that they naturally learn through experiences. The most well-known deep learning method is convolutional neural network (CNN).

The convolutional neural network is a type of feed-forward artificial neural network whose connection between neurons is inspired by the visual cortex of animals. CNN is among the most popular deep learning algorithms and it learns to perform the classification process directly from image, video, text, or audio files. CNN is quite similar to ordinary ANNs, and it consists of neurons with learnable weight and bias values just like ordinary ANNs [24-27]. The biggest difference of CNN from ordinary ANNs is that by nature it assumes its inputs as two- or three-dimensional images. This situation causes a significant reduction in the number of network parameters, and at the same time, it prevents overfitting in problems with a high number of features and a low amount of data, thereby increasing efficiency. The main reason for using the CNN architecture in this study is that it automatically extracts features while learning as shown Figure 7. With this aspect, it provides more advantageous and successful results than ordinary machine learning algorithms.

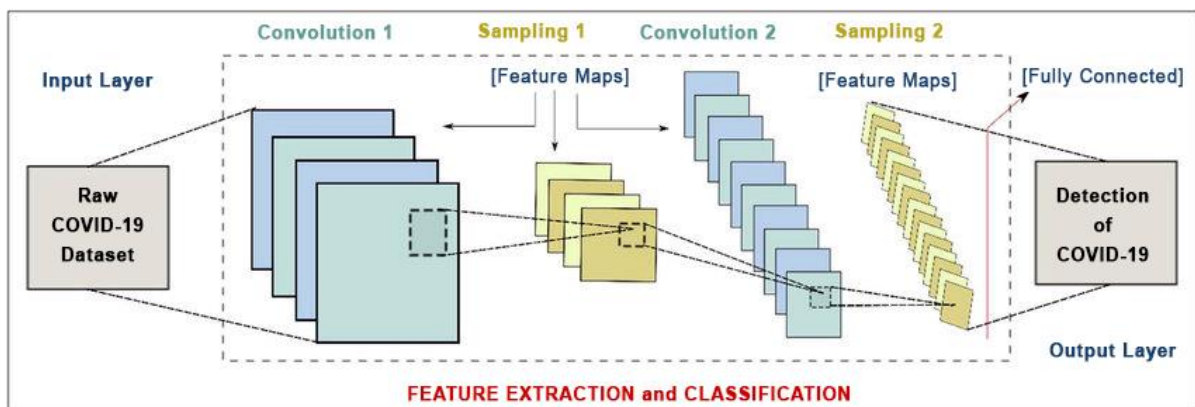


Figure 7 Classical deep learning with CNN structure

The details of the proposed deep NN model are given in Table 3.

Table 3 The structure of deep NN

Layers	Numerical Value
1. Input Layer (Raw Data)	18x600
2. Convolution Layer	4x4x600 (filter)
3. Sampling Layer	2X2X600 (filter)
4. Fully-Connected Layer	1X600

In our study, a hybrid CNN pre-layer deep artificial neural network is used. For this purpose, one convolution layer and two classifiers are used for the CNN structure to detect COVID-19.

A typical convolutional neural network has three types of layers that are repeated in different numbers and combinations between the input layer and the fully connected layer. These layers are the convolution layer, the ReLU layer, and the Pooling layer, respectively. While creating a CNN, these three types of layers are repeated over and over to adjust the depth of the network.

A CNN with appropriate parameter values (weights and bias) will correctly assign a given input to one of the classes it has in its output. The purpose of CNN training is to find the most appropriate CNN parameters to perform the correct classification process for a classification problem. At the start of the CNN training, the values of weights and biases are assigned random values or the process may be started using the parameter values of a different neural network performing similar tasks, which is a method that we call transfer learning.

After the initial values of parameter values are determined, each sample in the training set is fed as input to the CNN to calculate class scores generated for each class in its output. By applying these generated class scores to a cost function, whether the CNN produces appropriate results with the existing parameters or not is determined. If our parameter values are not suitable in terms of intuitiveness, our cost function will generate relatively high values. In essence, training a CNN is the process of finding CNN parameters that minimize the cost function [28].

For a single class NN classifier with a single neuron at its output layer, the cost function calculates the cost of each sample in the training set at the NN output. For a multi-class NN classifier with multiple neurons at its output layer, the cost function calculates the cost for all classes at the NN output for each sample in the training set. Cost values for each sample are generated based on the output class they belong.

NN training is the finding of network parameters that minimize the cost function for the training set. The backpropagation algorithm used in the training of NNs is an algorithm that minimizes the cost function. The partial derivatives obtained in the backpropagation algorithm are used together with the gradient descent [29] or a more advanced optimization algorithm to find network parameters that minimize the cost function. In this study, the gradient descent algorithm was preferred as the cost function.

#### 4. Results

In this section, we evaluate and compare machine learning and deep learning results based on the classification accuracy to detection of COVID-19. For this purpose, dataset containing the laboratory findings of the patients retrieved from the Israelita Albert Einstein Hospital (Sao Paulo, Brazil) is used [22]. The performance of machine learning algorithms was evaluated with k-fold cross validation method ( $k = 5$ ), one of the data enhancement methods. We applied nine different kinds of classification algorithms and the classification accuracies are presented in Table 4. All numerical results are obtained by using MATLAB application on a core intel processor under Windows 10 operating system.

First we evaluate results of decision tree classifier. Three different kinds of decision tree algorithms (fine tree, medium tree, and coarse tree) are applied for performance comparison, and coarse tree achieves the best accuracy with 86% among them. Second, we applied two different discriminant analysis algorithm (linear and quadratic discriminant) and the better result is obtained when linear discriminant method achieving an accuracy of 87.5% is used. Next, we look at the results of logistic regression classifier, it achieves 88.2% accuracy. We performed 2 different types of Naive-Bayes (NB) classifier (Kernel NB and Gaussian NB). The results of Naive Bayes classifier are close to the results of decision tree classifier and Kernel Naive Bayes method achieves better accuracy with 86.3%. We applied 6 different types of SVM methods (linear, quadratic, cubic, coarse gaussian, medium gaussian, and fine gaussian SVM), and the better result is obtained when medium gaussian achieving an accuracy of 88.8% is used. Next we evaluate results of the KNN classifier. We applied 6 different kinds of the KNN classifier (cubic, coarse, fine, medium, weighted and cosine KNN). Cosine KNN method achieves the best accuracy of 89.0%. We applied five types of ensemble classifiers (bagged trees, boosted trees, subspace discriminant, RUSBoosted tree, and subspace KNN). Bagged trees method achieves better results with an accuracy of 87.8%. Next we look at results of classical neural network classifiers (narrow, medium, wide, bilayered, trilayered neural networks). All types of neural networks give close results and better accuracy (85.5%) is achieved when wide neural network method is used. Finally, we look at the results of multilayer perceptron.

In this study, 36 different artificial intelligence methods were classified on COVID-19 data to detection. When all methods are compared, the highest performance has been obtained by synthesizing the deep neural network model CNN proposed as a hybrid model. Table 4 shows that the other models perform approximately the same. This is because the features given to the models are automatically selected with the CNN structure. It is seen that the feature extraction layer in this CNN structure improves accuracy performance on COVID-19 dataset.

Table 4 Accuracy results of machine and deep learning methods

Method		Accuracy (%)
Decision Trees	Coarse Structure	86.0
	Fine Structure	85.3
	Medium Structure	85.3
Discriminant Analysis	Linear Disc. Structure	87.5
	Quadratic Disc. Structure	85.3
Logistic Regression Classifier	Logistic Regression Structure	88.2
Naïve Bayes	Kernel Structure	86.3
	Gaussian Structure	85.0
SVM	Medium Gaussian Structure	88.8
	Linear Structure	87.0
	Fine Gaussian Structure	86.7
	Coarse Gaussian Structure	86.7
	Quadratic Structure	84.2
	Cubic Structure	83.0



KNN	Cosine Structure	<b>89.0</b>
	Cubic Structure	88.2
	Weighted Structure	87.7
	Medium Structure	87.7
	Fine Structure	84.8
	Coarse Structure	86.7
Ensemble Classifiers	Bagged Trees Structure	87.8
	Subspace Discriminant Structure	86.5
	Subspace KNN Structure	86.3
	Boosted Trees Structure	84.8
	RUSBoosted Trees Structure	80.7
Neural Network Classifiers	Wide NN Structure	85.5
	Medium NN Structure	85.2
	Narrow NN Structure	84.7
	Bilayered NN Structure	84.5
	Trilayered NN Structure	84.2
Multilayer Perceptron	Two Hidden-Layers Structure	88.7
	Three Hidden-Layers Structure	87.4
	One Hidden-Layer Structure	86.3
Our recommendation: Deep Neural Network with CNN Structure	One Hidden-Layer Structure	<b>90.3</b>
	Two Hidden-Layers Structure	89.2
	Three Hidden-Layers Structure	86.4

Now we evaluate accuracy results of deep neural network classifier. In the proposed model, it was observed how different hidden layer neuron numbers affect the performance and is given in Figure 8. As seen in Figure 8, it is understood that the excessive increase in the number of hidden layers negatively affects the performance of the network. It is understood that this is due to the increase in the mathematical computational burden of the model.

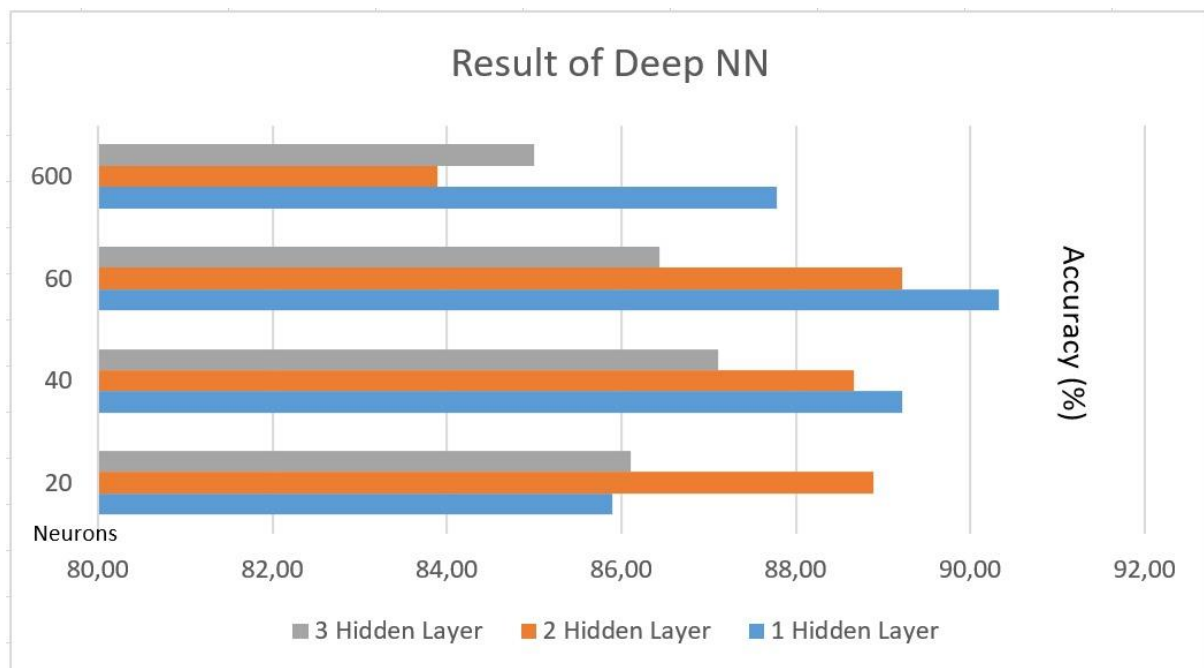


Figure 8. Result of Deep NN on COVID-19's detection

As a result, when the results obtained are examined, it is understood that the artificial intelligence methods mentioned above have an acceptable success for the detection of COVID-19. Accuracy performance can be expected to increase by increasing the number of samples in the dataset. In addition, performance can be increased with different hybrid models. However, it is seen that working with this current situation is successful.

## 5. Conclusion

COVID-19 disease has caused severe and deadly complications and fast and accurate detection of COVID-19 is important to prevent spreading from one person to another. This paper presents a comparative study for COVID-19 detection using various types of machine and deep learning methods from laboratory finding data. As the conclusion, the following results can be summarized:

- Nine different types of machine learning methods with different variants are performed and classification accuracy changes from 80.7 % to 89 %.
- Classification accuracies of decision tree, naive bayes, and neural network classifiers are close to each other.
- Cosine KNN method achieves the best accuracy with 89.0 % among other machine learning methods.
- Ensemble classifier achieves an accuracy of 87.8 %.
- In the deep neural network classifier, the number of neurons at a hidden layer is changed from 20 neurons to 600 neurons and the number of the hidden layer is changed from 1 to 3. The best classification accuracy is obtained when one hidden layer including 60 neurons is obtained.
- Accuracies of deep neural network classifier change from 83.9 % to 90.3 %.
- Performance of machine learning and deep learning method is close to each other and deep neural network method achieves the best result.

In future studies, we will combine supervised and unsupervised techniques to increase overall accuracy.

## References

- [1] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms" *Digital Medicine*, 4(1):3, December 2021.
- [2] H. Arslan and H. Arslan, "A new covid-19 detection method from human genome sequences using CpG island features and knn classifier", *Engineering Science and Technology, an International Journal*, 2021.
- [3] H. Arslan, "Machine learning methods for covid-19 prediction using human genomic data", *Proceedings*, 74(1), 2021.
- [4] W. Shang, J. Dong, Y. Ren, M. Tian, W. Li, J. Hu, and Y. Li, "The value of clinical parameters in predicting the severity of COVID-19", *Journal of Medical Virology*, 92(10), pp. 2188-2192, June 2020.
- [5] T. B. Alakus and I. Turkoglu, "Comparison of deep learning approaches to predict covid-19 infection", *Chaos, Solitons Fractals*, 140:110120, 2020.
- [6] M. Alazab, A. Awajan, A. Mesleh, A. Abraham, V. Jatana, and S. Alhyari, "Covid-19 prediction and detection using deep learning", *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp.168-181, 2020.
- [7] V. Andriasyan, A. Yakimovich, F. Georgi, A. Petkidis, R. Witte, D. Puntener, and U. F. Greber, "Deep learning of virus infections reveals mechanics of lytic cells", *Cell Biology*, October 2019.
- [8] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Zidek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. T. Jones, D.

- Silver, K. Kavukcuoglu, and D. Hassabis, "Improved protein structure prediction using potentials from deep learning", *Nature*, 577(7792), pp. 706-710, January 2020.
- [9] Y. Zoabi, S. Deri-Rozov, and N. Shomron, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms", *Digital Medicine*, 4(1):3, December 2021.
- [10] L. J. Muhammad, E. A. Algehyne, S. S. Usman, A. Ahmad, C. Chakraborty, and I. A. Mohammed, "Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset", *SN Computer Science*, 2(1):11, February 2021.
- [11] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. Varkonyi-Koczy, U. Reuter, T. Rabczuk, and P. M. Atkinson, "COVID-19 Outbreak Prediction with Machine Learning", *Algorithms*, 13(10):249, October 2020.
- [12] M. H. Tayarani, "Applications of artificial intelligence in battling against covid-19: A literature review", *Chaos, Solitons & Fractals*, 142:110338, January 2021.
- [13] S. Kushwaha, S. Bahl, A. K. Bagha, K. S. Parmar, M. Javaid, A. Haleem, and R. P. Singh, "Significant applications of machine learning for covid-19 pandemic", *Journal of Industrial Integration and Management*, 5(4), December 2020.
- [14] F. D. Felice and A. Polimeni, "Coronavirus Disease (COVID-19): A Machine Learning Bibliometric Analysis", *In Vivo*, 34(3 suppl):1613-1617, 2020.
- [15] F. Cabitza, A. Campagner, D. Ferrari, C. D. Resta, D. Ceriotti, E. Sabetta, A. Colombini, E. D. Vecchi, G. Banfl, M. Locatelli, and A. Carobene, "Development, evaluation, and validation of machine learning models for covid-19 detection based on routine blood tests", *Clinical Chemistry and Laboratory Medicine (CCLM)*, 59(2), pp. 421-431, 2021.
- [16] Y. Unal and M. N. Dudak, "Classification of covid-19 dataset with some machine learning methods", *Journal of Amasya University the Institute of Sciences and Technology*, 1:30 - 37, 2020.
- [17] X. Jiang, M. Coffee, A. Bari, J. Wang, X. Jiang, J. Huang, J. Shi, J. Dai, J. Cai, T. Zhang, Z. Wu, G. He, and Y. Huang, "Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity", *Computers, Materials & Continua*, 62(3):537-551, 2020.
- [18] A. F. M. Batista, J. L. Miraglia, J. L. Rizzi, T. H. Donato, A. D. Porti Chivegatto, "COVID-19 diagnosis prediction in emergency care patients: a machine learning approach", *Medrxiv*, 2021-06-30.
- [19] P. Schwab, A. D. Schutte, B. Dietz, and S. Bauer, "Clinical predictive models for COVID-19: Systematic study", *Journal of Medical Internet Research*, 22(10):21439, October 2020.
- [20] V. Göreke, V. Sari, S. Kockanat, "A novel classifier architecture based on deep neural network for COVID-19 detection using laboratory findings", In *Applied Soft Computing*, vol. 106, 2021.
- [21] W. M. Shaban, A. H. Rabie, A. I. Saleh, M.A. Abo-Elsoud, "Detecting COVID-19 patients based on fuzzy inference engine and Deep Neural Network", In *Applied Soft Computing*, vol. 99, 2021. 106906, T. B. Alakus and I. Turkoglu, "Comparison of deep learning approaches to predict COVID-19 infection", In *Chaos, Solitons & Fractals*, vol. 140, 2020.
- [22] F. Cabitza et al. "Development, evaluation, and validation of machine learning models for covid-19 detection based on routine blood tests", In *Clinical Chemistry and Laboratory Medicine (CCLM)*, 59(2):421-431, 2021.
- [23] Mathworks, "Introducing Machine Learning", *e-book about MATLAB and Simulink, the MathWorks Inc.*, 2016.
- [24] O. Er, A. C. Tanrikulu, A. Abakay, and F. Temurtas, "An approach based on probabilistic neural network for diagnosis of Mesothelioma's disease", *Computers & Electrical Engineering*, 38(1), 75-81, 2012.
- [25] L. LE, Y. Zheng, G. Carneiro, L. Yang, "Deep learning and convolutional neural networks for medical image computing", *Advances in Computer Vision and Pattern Recognition, Springer*, 2017.
- [26] H. A. Aghdam, E. J. Heravi, "Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification", *Springer*, 1st edn., 2017.
- [27] E. Olmez, V. Akdogan, M. Korkmaz, and O. Er, "Automatic Segmentation of Meniscus in Multispectral MRI Using Regions with Convolutional Neural Network (R-CNN)", *Journal of Digital Imaging*, 33, 916-929, 2020.

[28] D. B. Aydın and O. Er, "A New Proposal For Early Stage Diagnosis of Urinary Tract Infection Using Computers Aid Systems", *Sakarya University Journal of Computer and Information Sciences*, vol. 1, 2018.

[29] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent", *In Proceedings of the 22nd international conference on Machine learning*, pp. 89-96, 2005.

# Comparison of Different Machine Learning Algorithms to Predict the Diagnostic Accuracy Parameters of Celiac Serological Tests

 Özgül ÖZER<sup>1</sup>,  Nazlı ARDA<sup>2</sup>

<sup>1</sup>Corresponding Author; Institute of Graduate Studies in Science, Istanbul University Turkey;  
ozgul.ozer@ogr.iu.edu.tr

<sup>2</sup> Department of Molecular Biology and Genetics, Faculty of Science, Istanbul University, Turkey;  
narda@istanbul.edu.tr

Received 27 March 2022; Revised ; Accepted 4 April 2022; Published online 30 April 2022

## Abstract

Celiac disease; is an autoimmune digestive system disease characterized by chronic intestinal inflammation and villus atrophy affecting genetically predisposed individuals. Diagnosis is based on serological tests and small bowel biopsy. Because of the diversity in the clinical features of the disease, various patient profile and the non-standardized serological tests, it is difficult to diagnose the celiac disease. Sensitivity, specificity, positive and negative predictive values are important parameters for the accuracy of the tests and they are missing in some clinical studies. It is difficult to standardize the tests with these missing values for clinicians. The aim of this study is to train different machine learning algorithms and to test their performance in prediction of the diagnostic accuracy parameters of celiac serological tests. Decision trees are effective machine learning algorithms for predicting potential covariates with %88,7 accuracy.

**Keywords:** machine learning, diagnostic test accuracy, CAD diagnosis of celiac disease, celiac serological tests

## 1. Introduction

Celiac disease (CD) is the inflammation of the small intestine caused by dietary gluten in genetically predisposed individuals. The incidence of the disease is %1 in most countries. Patients are required to follow a gluten-free diet life-long. Nutritional can not be absorbed sufficiently as the result of villus atrophy [1]. While the symptoms of the disease are similar to many other diseases and these symptoms are different in each individual, it is very difficult to diagnose. %80-90 of the patients are still undiagnosed while only %10 of the patients know that they have celiac disease [2]. In a serologic screening research, involving more than 17,000 Italian schoolchildren, the ratio of individuals who know their disease to those who do not know is 1/7 [3,4].

Although the patient profile with celiac disease may be variable, serological tests are a cheap and non-invasive method for clinicians to identify the disease. The usage of serological tests has also been suggested for the follow up of patient dietary compliance. Antibodies against to gluten proteins in the foods and to structural proteins in intestinal mucosa (endomysium, reticulin, transglutaminase) are the targets of the tests. In 1960s, it is found that the gliadin compounds in wheat are involved in the pathogenesis of the disease. Anti-gliadin antibodies (AGA) are the first autoantibodies used in the diagnosis of celiac disease and then anti-endomysium (EMA) antibodies began to be used in the diagnosis at 1980s. Endomysium is a structural protein of intestinal tissue.

It is not recommended to use Anti-endomysium antibodies in patients with mild bowel lesions (Marsh 3A) and children under 2 years of age. In 1990s, the role of the tissue transglutaminase (tTG) enzyme in celiac pathogenesis is well understood and tTG antibody tests are became very popular at diagnosis. [5]. We can use Anti-gliadin antibodies (AGA) for screening aims while anti-tissue transglutaminase (dTG) and anti-endomysium (EMA) autoantibodies are giving better results at diagnosis and patient follow-up [6].

Diagnostic test accuracy determines if the test identify the target situation accurately. There are some parameters like sensitivity, specificity, likelihood ratios, Youden's index which tells us the diagnostic

accuracy of tests. These parameters can be calculated from  $2 \times 2$  contingency table that includes the number of true-positive, true-negative, false-positive, and true-positive test results. Sensitivity is the ratio of individuals correctly identified with target situation. A test with 100% sensitivity means all diseased individuals are correctly identified. There are no false negatives. These parameters differ between analysis. Specificity and sensitivity of some assays are lower than expected in some clinical applications [7,8,9].

Machine learning is extensively applied in the field of medical informatics, including gene and protein structure prediction, genome analysis, drug discovery, text mining and image processing. There are limited number of studies about the prediction of diagnostic test accuracy parameters using machine learning algorithms [10,11].

Machine learning workflows are complex and difficult to understand since the accuracy of the algorithms is distinct from each other. Decision trees provide high classification accuracy and can be used in different areas of medical decision making. Simple decisions are used for prediction consecutively in decision tree algorithms. Bayesian classifier is also one of the most useful and effective predictive data mining method. Naive Bayes models uses the method of maximum likelihood for parameter estimation in practical applications. A family of algorithms based on a common principle are used for training instead of a single algorithm [12,13]. Random forests have been successfully used in classification, regression and clustering tasks. Boosting is also a flexible nonlinear regression procedure that helps improving the accuracy of trees [14,15].

KNIME Platform is a very useful tool for applying machine learning algorithms for beginners without coding background. Procedures like clicks, drags, and drops can be followed easily. This paper describes the overall process of applying different machine learning algorithms via the KNIME analytics platform in a simple way [16].

## **2. Material and Method**

### **2.1. Dataset and Data Preprocessing**

The Pubmed database was searched (January 2000- January 2022) for clinical studies assessing the accuracy of celiac serological tests. 80 Studies including sensitivity, specificity, positive and negative predictive values were included. We processed and analyzed the data using the Konstanz Information Miner (KNIME) analytics platform. The procedures to install KNIME extensions were followed. After installing Knime extensions, we created the Knime workflow.

Datasets are transferred to Knime workflow with CSV reader node. The input table is split into two partitions (%70 train dataset, %30 test dataset) with partitioning node as shown in Figure 1-2.

Sensitivity was designed as target value since there is a correlation between sensitivity and the other values.



Row ID	Sensitivity	Specificity	PPV	NPV
Row0	87,1	94,1	95,2	84,4
Row1	91	97	91	97
Row2	93	98,2	93,9	98,5
Row3	95,7	94,3	95,7	94,4
Row4	100	97,7	100	98
Row5	90,1	75	7,7	99,7
Row6	90,1	87,1	27,3	99,4
Row7	89,5	87,5	21,4	99,5
Row8	95,2	93	93,7	96,8
Row9	90,9	90,9	28,6	99,6
Row10	97,9	92,5	89,4	99
Row11	86,8	42,9	3,57	99,3
Row12	98,82	100	95,35	100
Row13	98	38	57	96
Row14	90	54	98,7	12,5
Row15	85	59	98,1	14,2
Row16	100	61	100	98,7
Row17	83	30	95,2	9,5
Row18	92	94	89	96
Row19	76	95	84	92
Row20	76	85	83	79
Row21	76	68	65	79
Row22	95	66	71	94
Row23	91	96,8	91,2	96,8
Row24	98,4	100	95,5	100
Row25	100	95	100	98
Row26	100	81	100	93
Row27	89,3	87,1	90,4	93,4
Row28	90	90	61	98
Row29	96	82	79	97
Row30	99	74	89	96
Row31	99	68	92	95
Row32	99	62	93	94
Row33	100	51	95	92

Row ID	Sensitivity	Specificity	PPV	NPV
Row36	100	19	98	88
Row37	100	21	98	88
Row38	97,4	93,3	90,3	98,2
Row39	97,4	93,8	83,3	100
Row40	95	93	93	95
Row41	95,3	74,2	91	85,2
Row42	100	94	100	94
Row43	99,5	98,3	99,6	98,1
Row44	100	95,7	100	95,9
Row45	99,5	42,9	42,9	99,5
Row46	99,5	73,3	84,6	98,9
Row47	99,1	82,6	73,1	99,5
Row48	98,2	87	66,3	99,5
Row49	100	83,5	82	96,9
Row50	99	79	78,9	99,1
Row51	100	100	100	100
Row52	100	100	100	100
Row53	100	93,7	100	94,4
Row54	100	100	100	100
Row55	94	95,8	93,1	91,6
Row56	64,5	95,3	83,3	88
Row57	87	73	84	77
Row58	96,9	91	94,5	97,2
Row59	97,04	90,24	88,1	97,62
Row60	80,7	96,9	66	98,5
Row61	81,9	90,6	65,9	95,8
Row62	89,2	90,6	76,3	96,1
Row63	95,2	75,2	85,7	90,8
Row64	87	66	58	95
Row65	98,5	84,4	98,2	86,8

Figure 1 Accuracy parameters of serological tests

Row ID	Sensitivity	Specificity	PPV	NPV
Row66	94,2	76,9	92,9	70,7
Row67	93	29	42	89
Row68	93	13	25	86
Row69	99,1	60	94,7	90,5
Row70	99,1	25	80	90,5
Row71	90	93,3	70	98,1
Row72	90	50	40	92,8
Row73	96	89	97	87
Row74	98	91	92	98
Row75	93	53	73	86
Row76	98	79	79	98
Row77	94	52	87	74
Row78	92	80	67	96
Row79	95	93	50	99,5
Row80	87	82	86	83

Figure 2 Accuracy parameters of serological tests

## 2.2. Applying Machine Learning Algorithms

4 different machine learning algorithms are used after partitioning. Decision tree learner, naives bayes learner, random forest learner, gradient boosted trees learner nodes are trained with training datasets while predictors nodes made predictions with test datasets. Scorer nodes calculated and represent the accuracy statistics as shown in Figure 3.

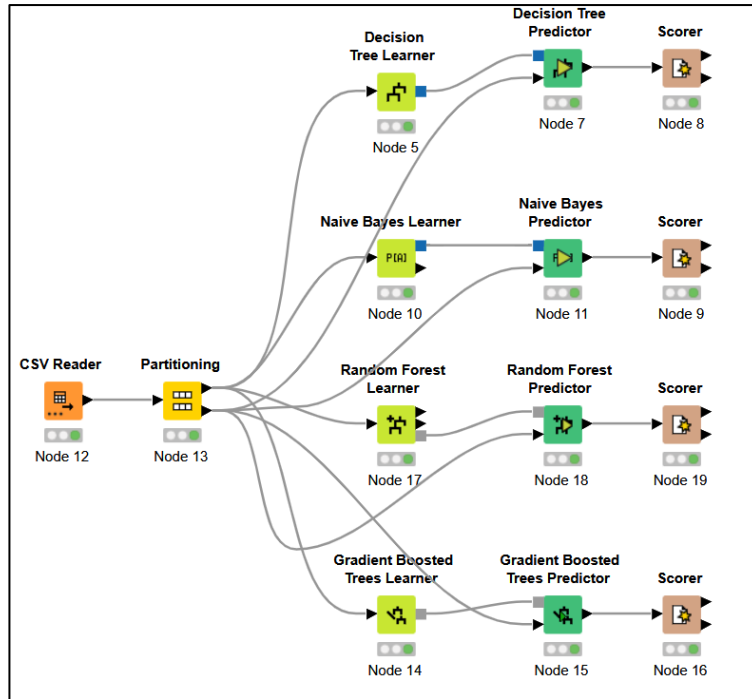


Figure 3 Knime workflow

### 3. Results

Accuracy values of sensitivity predictions are; %88 for decision tree predictor, %70 for naive bayes predictor, %100 for random forest predictor and %71 for gradient boosted trees predictor as shown in Figure 4-7.

Decision tree predictor node provided highest Cohen’s kappa value with 0,87 while naive bayes predictor node the lowest value with 0,67. Decision tree predictor provided the lowest error rate with 0,1 while naive bayes predictor calculated the highest error value.

Name	Value
d Cohen's kappa	0.8790525785318326
i #False	9
i #Correct	71
d Error	0.1125
d Accuracy	0.8875
s knime.workspace	C:\Users\ozgul\knime-workspace

Figure 4 Accuracy statistics for decision tree predictor node

Name	Value
d Cohen's kappa	0.6703862660944205
i #False	24
i #Correct	56
d Error	0.3
d Accuracy	0.7
s knime.workspace	C:\Users\ozgul\knime-workspace

Figure 5 Accuracy statistics for naive bayes predictor node



Name	Value
d Cohen's kappa	1.0
i #False	0
i #Correct	80
d Error	0.0
d Accuracy	1.0
s knime.workspace	C:\Users\ozgul\knime-workspace

Figure 6 Accuracy statistics for random forest predictor node

Name	Value
d Cohen's kappa	0.6871280394490733
i #False	23
i #Correct	57
d Error	0.2875
d Accuracy	0.7125
s knime.workspace	C:\Users\ozgul\knime-workspace

Figure 7 Accuracy statistics for gradient boosted trees predictor

#### 4. Conclusion

Data mining approaches have been successfully applied to different practical problems not only in clinical medicine but also in epidemiological studies and meta-analysis. These approaches can offer predictions for missing parameters which are in fact not ignorable in meta-analyses and systemic reviews. Machine Learning algorithms can highlight the gaps in the evidence based medicine by predicting potential covariates [17,18].

%100 accuracy of random forest predictor in this study, can be explained with overfittig and the small number of sample size. Decision tree predictor which provides %88,7 accuracy can be used as a effective machine learning algorithm for predicting potential covariates for missing values in meta analyses.

## References

- [1] D. Schuppan, “Current concepts of celiac disease pathogenesis,” *Gastroenterology*, vol. 119, pp. 234–242, 2000.
- [2] S. Lohi et al, “Increasing prevalence of coeliac disease over time,” *Alimentary Pharmacology & Therapeutics*, vol. 26, no. 9, pp. 1217-25, 2005.
- [3] M. Parizade, Y. Bujanover, B. Weiss V., Nachmias and B. Shainberg, “Performance of Serology Assays for Diagnosing Celiac Disease in a Clinical Setting,” *Clinical and Vaccine Immunology*, vol. 16, pp. 1576–1582, 2009.
- [4] A. Fasano and C. Catassi, “Current approaches to diagnosis and treatment of celiac disease: An evolving spectrum”, *Gastroenterology*, vol. 120, no. 3, pp. 636-51, 2001.
- [5] A. Marlou and A.D. Leffler, “Serum Markers in the Clinical Management of Celiac Disease,” *Digestive Disease*, vol. 33, pp. 236–243, 2015.
- [6] D. Basso et al. “A new indirect chemiluminescent immunoassay to measure Anti-Tissue Transglutaminase antibodies,” *J Pediatr Gastroenterol Nutr.*, vol. 43, pp. 613-8, 2006.
- [7] P. Eusebi, “Diagnostic Accuracy Measures,” *Cerebrovascular Diseases*, vol.36, pp. 267–272, 2013.
- [8] A. Hoyer and A. Zapf, “Studies for the Evaluation of Diagnostic Tests,” *Deutsches Ärzteblatt International*, vol. 18, pp. 555–60, 2021.
- [9] O. Rozenberg, A. Lerner, A. Pacht, M. Grinberg, D. Reginashvili, C. Henig and M. Barak, “A new algorithm for the diagnosis of celiac disease,” *Cellular & Molecular Immunology*, vol. 8, pp. 146–149, 2011.
- [10] Z. Obermeyer and J. E. Emanuel, “Predicting the Future-Big Data, Machine Learning, and Clinical Medicine,” *N Engl J Med*, vol. 375, no.13, pp. 1216–1219, 2016.
- [11] M. Saken, M. Y. Banzragch and N. Yumusak, “Impact of image segmentation techniques on celiac disease classification using scale invariant texture descriptors for standard flexible endoscopic systems,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 29, pp. 598 – 615, 2021.
- [12] R. Bellazzi and B. Zupan, “Predictive data mining in clinical medicine: Current issues and guidelines,” *International Journal of Medical Informatics*, vol.77, pp.81–97, 2008.
- [13] Y. Long, L. Wang and M. Sun, “Structure Extension of Tree-Augmented Naive Bayes,” *Entropy*, vol. 21, pp.721, 2019.
- [14] Z. Zhang, Y. Zhao, A. Canes, D. Steinberg and O. Lyashevskaya, “Predictive analytics with gradient boosting in clinical medicine,” *Annals of Translational Medicine*, vol. 7, no.7, pp.152, 2019.
- [15] M. Song, H. Jung, S. Lee, D. Kim, and M. Ahn, “Diagnostic Classification and Biomarker Identification of Alzheimer’s Disease with Random Forest Algorithm,” *Brain Sciences*, vol. 11, no. 4, pp. 453, 2021.
- [16] A. W. Warr, “Scientific workflow systems: Pipeline Pilot and KNIME,” *J Comput Aided Mol Des.*, vol. 26, no.4, pp. 801–804, 2012.
- [17] Y. Yuan and R. Little, “Meta-Analysis of Studies with Missing Data,” *Biometrics*, vol.65, pp. 487-496, 2009.
- [18] J. M. Schauer, K. Diaz, T.D. Pigott and J. Lee, “Exploratory Analyses for Missing Data in Meta-Analyses and Meta-Regression: A Tutorial,” *Alcohol and Alcoholism*, vol. 57, pp. 35-36, 2022.

## A Novel Hybrid Binary Farmland Fertility Algorithm with Naïve Bayes for Diagnosis of Heart Disease

 Vafa Radpour<sup>1</sup>,  Farhad Soleimanian Gharehchopogh<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

<sup>2</sup>Corresponding Author; Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran; [bonab.farhad@gmail.com](mailto:bonab.farhad@gmail.com)

Received 203 August 2021; Revised 08 April 2022; Accepted 14 April 2022; Published online 30 April 2022

### Abstract

One of the essential aims of intelligent algorithms concerning the diagnosis of heart disease is to achieve accurate results and discover valuable patterns. This paper proposes a new hybrid model based on Binary Farmland Fertility Algorithm (BFFA) and Naïve Bayes (NB) to diagnose heart disease. The BFFA is used for Feature Selection (FS) and the NB for data classification. FS can be employed to discover the most beneficial features. Four valid and universal UCI datasets (Hearts, Cleveland, Hungary, and Switzerland) were used to diagnose heart disease. Each dataset included 13 main features. The evaluation of the proposed model is simulated in MATLAB 2017b. The number of features in four datasets of Heart, Cleveland, Hungary, and Switzerland is equal to 13, which was reduced to six for each dataset through the BFFA to better the efficiency of the proposed model. For evaluation, the accuracy criterion, the criterion of accuracy in the proposed model for all features in the four datasets, Heart, Cleveland, Hungary, and Switzerland, is equal to 82.25%, 86.91%, 89.32%, and 89.24%, respectively. Results of the proposed model showed appropriateness in comparison to some other methods. In this paper, the proposed model was compared with other methods, and it was found that the proposed model possessed a better accuracy percentage.

**Keywords:** Diagnosis of Heart Disease, Binary Farmland Fertility Algorithm, Naïve Bayes Algorithm, Feature Selection, Classification, Accuracy Percentage

### 1. Introduction

Heart diseases are usually asymptomatic in the early stages, and heart attack and brain stroke are the first warning signs [1, 2]. Heart disease is one of the deadly silent diseases that cause more deaths than cancer in some countries. High blood pressure, high blood fats, obesity, and diabetes are risk factors for heart disease [3]. These factors gradually cause damage to the heart. Therefore, a physician must regularly monitor such risk factors for regular tests and examinations [4]. Heart disease affects the heart and blood supply to the peripheral areas of the body. Heart disease generally refers to a condition that causes the coronary arteries to narrow or become blocked, leading to heart failure, pain in the chest, or stroke [5].

Since the disease diagnosis is not easy in most cases, the physician must review and analyze the patient's tests and past decisions made for patients with similar conditions to make the right decision. In other words, the physician will need knowledge and experience. However, due to many patients and multiple tests per patient, an automated tool is required to explore patients with heart problems. One of the essential methods used for data inference is intelligent disease-diagnosis systems [6]. Collecting and recording data on heart diseases by intelligent systems is significant. Therefore, machine learning methods obtain valuable relationships between pathogens [7].

The entailing necessity in designing intelligent disease-diagnosis systems is to assist physicians in diagnosing the due disease and accelerating the testing process. Computer-based technology has significantly increased medical diagnosis, disease treatment, and patient follow-up processes [8, 9]. There is a need to diagnose it early to reduce the mortality rate related to a deficiency in coronary heart disease. As such, the existence of multiple features concerning the analysis and diagnosis of the disease proves the work of the physician brutal [10]. As a result, experts need an accurate tool that comprehensively considers these risk factors and reveals the exact impact of these ambiguous features.

In the present study, encouraged to create such a valuable and accurate tool, the researchers designed an intelligent system that identifies heart disease [11].

Concerning various interfering factors, the diagnosis of heart disease has always been a pivotal issue for physicians and teams of specialists. Recently, intelligent computational algorithms are often used in medical science to diagnose and determine the severity of various diseases. Intelligent algorithms help physicians as assisting tools in diagnosis with an incident of fewer errors. Up to now, different algorithms have been used to diagnose various diseases. In the present study, the researchers proposed a new model for the diagnosis of heart disease based on the Farmland Fertility Algorithm (FFA) [12] and NB [13]. It used the BFFA for FS and the NB algorithm for samples classification.

The FFA was presented by *Shayanfar* and *F.S.Garahchapagh* in 2018 [12]. The algorithm is a nature-inspired meta-heuristic algorithm that mimics fertility practices of agricultural lands. The FFA is under the inspiration of soil strategy [14, 15]. Also, the algorithm has undergone 20 mathematical tests through optimization equations. NB is among the machine learning methods for categorization [16]. As a probability hypothesis, different classes are considered separately in the technique—each new training data augmentations or diminutions the probability of correctness of the primary views. Then finally, the ideas with the highest likelihood are considered a particular class and labeled.

FS, an optimization problem, becomes a critical pre-process tool in machine learning, simultaneously minimizing feature size and maximizing model generalization. FS possesses a particular place and ranks in most research because there are various features in numerous data sets. Many are unused or do not have a positive informational aspect [17]. Non-elimination of such features does not impose a problem concerning the nature of the information. Still, they increase the computational load of the system and make it harder for accurate identification. With the increased number of features, feature space rises too. As such, data analysis and classification become significantly more complicated. Data are widely dispersed in related areas, creating significant problems concerning the supervised and unsupervised algorithms. This phenomenon is known as the size/bulk problem and is based on the fact that working with high-dimensional data is often tricky and time-consuming. A large number of features is apt to increase the noise in the data. Thus classification algorithm error rises, especially if the number of samples is smaller than the number of related features.

The significant contributions of this paper are:

- A hybrid BFFA with NB is proposed for the diagnosis of heart disease.
- A fast hybrid dimensionality reduction method for classification is proposed.
- BFFA is applied to remove the weakest feature and choose the best features.
- It is evaluated on four valid and universal UCI datasets (Heart, Cleveland, Hungary, and Switzerland).
- The proposed model shows excellent efficiency and competitive classification performance.

This paper is organized as follows: In Section 2, the related works are explained briefly. Section 3 provides a summary of the proposed model procedures. Section 4 provides experimental results and discussion. Finally, the conclusion and future work directions are summarised in Section 5.

## 2. Related Works

Several conducted studies in heart disease diagnosis algorithms are presented in this section. In [18], the scientists created a fuzzy algorithm for detecting heart diseases using medical knowledge and intelligence principles. Their model was based on the data set obtained from 1000 patients, including patients with heart disease and healthy individuals at *Tohid Hospital in Sanandaj* via resorting to MATLAB, featuring a fuzzy toolbox. The results of the performed experiments on the collected dataset demonstrated that the proposed model was able to 98% for accurate diagnosis of heart diseases.

In [19], the author proposed an Artificial Neural Networks (ANN) model for diagnosing heart disease. They used the data obtained from 270 patients extracted from the UCI site's valid data set, including 14 features (one feature related to class). According to the results, it was shown that the model of ANN

with multilayer perceptron structure and an accuracy of 83.33% performed the classification operation for the set of test observations. The results also indicated that the model's accuracy in classifying the heart disease response variable samples was 87.75% and 83.33% for the training and test samples.

In [20], the authors presented a model based on a hybrid Whale Optimization Algorithm (WOA) and Simulated Annealing (SA) to detect the influential factors in diagnosing heart disease. The Support Vector Machine (SVM) algorithm was used for effective disease classification. It was evaluated with the Cleveland Heart Disease Database in the UCI Valid Database. Generally, there were 13 features with 270 samples of sick and healthy individuals. The hybrid algorithm identified the number of compelling disease features based on the best solution. The proposed method identified ten valuable features with an accuracy of 87.78%.

In [21], the authors proposed a "Designing a Cardiovascular Disease Prediction System using a Support-vector Machine" model to diagnose heart disease. The dataset of 270 individuals, including 13 features, was used. The evaluation criteria in the system were classification rate and sensitivity. The system's performance was 85% and 85.8%, respectively, based on the mentioned indexes.

According to regression tree and classification, a model based on an artificial neural network (ANN) and variable selection to predict coronary artery disease was proposed [23]. The dataset included nine information variables of 13228 individuals who experienced angiography in Tehran Heart Center (4059 people without coronary artery disease and 9169 people with the disease). According to the regression tree and classification, the coronary artery disease prediction model was created based on an ANN multilayer perceptron (ANN-MLP) and variable selection method. After seven times of modeling and comparing the developed models, the final model, including all available variables occupying an area under the rocking curve of 0.754, the sensitivity of 92.41, and the accuracy of 74.19, was obtained.

[22] proposed a fuzzy-differential hybrid model based on a fuzzy expert system (FES) to predict heart disease risk. For evaluation and validation of the proposed model, a data set including 380 samples acquired from *Parsyan Hospital* was used. The results showed that the FES had a functional accuracy of 85.52%, which was enhanced to 97.93% after applying the fuzzy genetic evolutionary model fuzzy-differential hybrid method to 97.67%. The results indicated that the proposed hybrid fuzzy genetic evolutionary model significantly improved the performance of the FES.

Sabbagh Gol [23] used the C4.5 algorithm to diagnose heart disease. The study was applied and descriptive by nature. Standard data from the credible UCI site with the Cleveland dataset was used. The database contains 303 samples (6 samples were missing). According to the utilized model, the variables of high cholesterol, gender, old age, high maximum heartbeat, and thallium scan above three and abnormal ECG had the most significant impact on coronary heart disease.

In [24] used, various ANN such as Multilayer Perceptrons (MLP), Learning Vector Quantization (LVQ), and BR to predict heart disease. The study was analytical, and its database contained 200 records of non-attributive types. The most important criteria for a disease diagnosis system were two indexes of specificity and sensitivity. These two indexes were calculated in the test and experiment phases of the study. According to the law of Back-Propagation of error, the best accuracy of the model was related to the ANN-MLP, equal to 88%. It was also observed that the elimination of discrete parameters had a positive effect on the convergence rate of the neural network and could improve the prediction accuracy by 85%. Table (1) illustrates a comparison of the proposed models for diagnosing heart disease

Table 1 Comparison of Proposed Models for Diagnosis of Heart Disease

Refs	Models	Datasets	Number of data samples	Number of features	FS	Percentage of accuracy
[18]	Fuzzy system for predicting heart disease	Collected	1000	8	8	98
[19]	Artificial Neural Networks	reference and standard	270	13	13	83.33
[20]	Hybrid WOA with SA	reference and standard	270	13	10	87.78
[21]	SVM	reference and standard	270	13	13	85
[25]	Decision tree and ANN	Collected	350	9	9	93.4
[26]	ANNs and variable selection based on regression tree	Collected	13228	9	5	74.19
[22]	Fuzzy-differential hybrid model based on a fuzzy expert system	Collected	380	5	5	97.67
[27]	C4.5	reference and standard	303	14	10	90.1
[23]	C4.5	reference and standard	297	14	5	80.2
[24]	Multi-layer artificial neural network	Collected	200	14	10	88

In [18], eight features (age, smoking, blood pressure, harmful fats, history, family, diabetes, gender) were used for evaluation. In [19], 13 features (large vessels (Nbr-ves), stress reduction (ST-dep), defect, chest pain, stress peak (Peak-ST), heart rate, angina, gender, age, static ECG (Res-elec), blood pressure (Blood-press), blood sugar, and serum cholesterol (Serum-chol)) were used for evaluation. In [20], ten features were selected as the main features for diagnosing heart disease. In [21], 13 features were chosen as the main features for diagnosing heart disease. In [25], out of 9 features, all features were added to the system for better detection. The [26] dataset includes nine risk features: age, sex, obesity, abdominal obesity, family history, smoking, high blood fats, diabetes, and high blood pressure. In [22], according to the results, the accuracy of the fuzzy expert model was 85.52%, which has increased to 97.93% after applying the hybrid Fuzzy-GA model and has grown to 97.67% after applying the hybrid Fuzzy-DE model. In [27], 14 features were used for evaluation, of which ten features had the more accurate diagnosis.

### 3. Proposed Model

A model based on BFFA with NB for diagnosing heart disease was proposed in the present study. MATLAB 2017b was used to simulate the proposed model, and 80% of the samples were used for the training phase and the remaining 20% for the test phase. Figure (1) shows the flowchart of the proposed model.

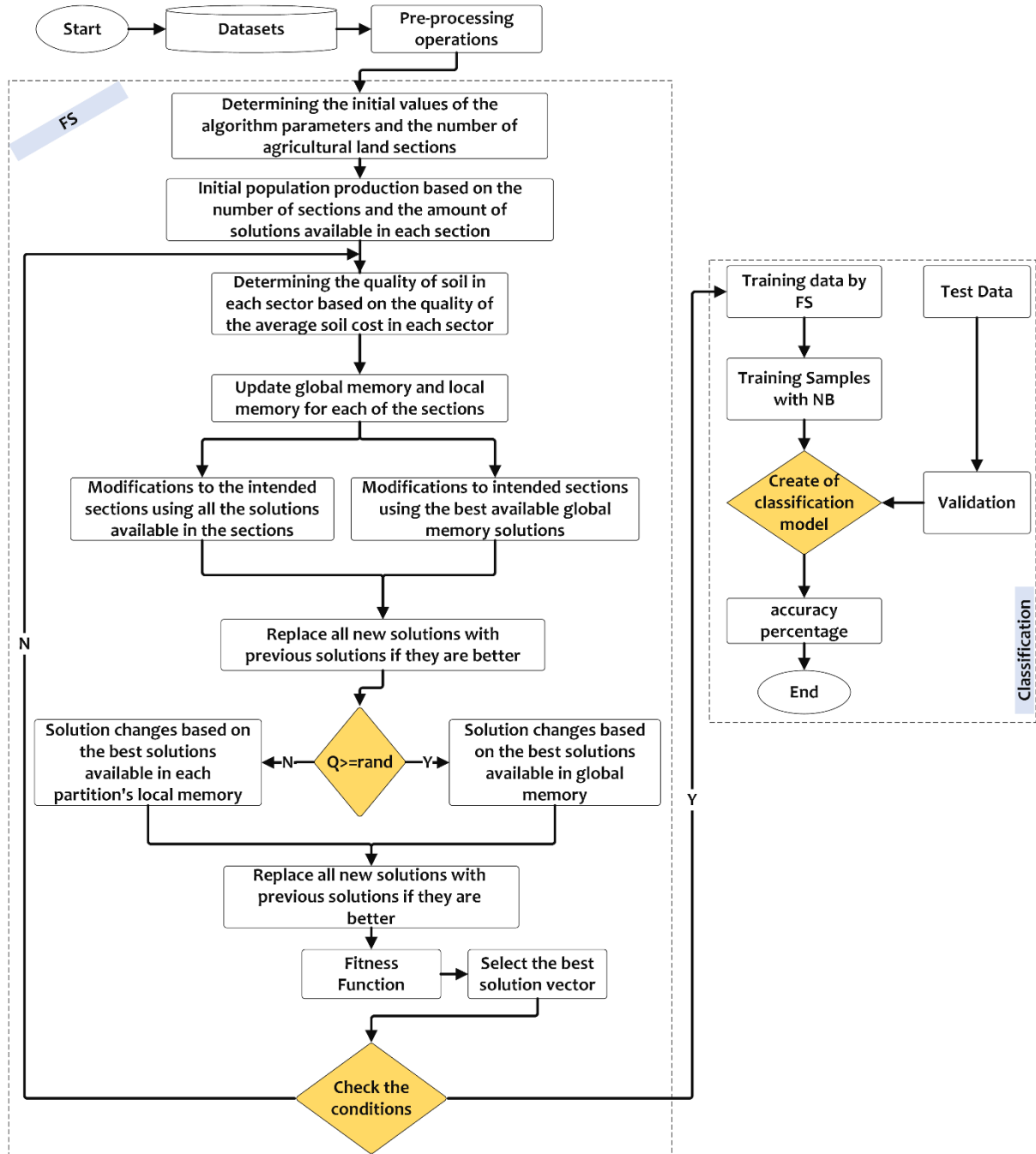


Figure 1 Proposed Model Flowcharts

### 3.1 Pre-processing

In this stage, the data were normalized, and the missing values were replaced. Normalization unifies and approximates the specific range of different features to some extent to achieve more accurate results. The average method was used concerning the issue of missing values. One of the most common normalization methods, according to Eq. (1), is the Min-Max method. Each data set is mapped to arbitrary intervals in this method to know the minimum and maximum values in advance. The minimum data ( $x_{min}$ ) and the maximum data ( $x_{max}$ ) are the minimum and maximum values in each variable ( $x_i$ ) respectively.

$$N_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

### 3.2 Feature Selection

In the present study, the BFFA was used for FS. Easy implementation, low complexity, and reasonable convergence rate are the main reasons for selecting the FFA regarding the FS issue. The V-Shaped function was used to convert the FFA to the binary mode. The procedure was first used by *Mirjalili* to convert a continuous state to binary mode [28]. In constant optimization issues, the solution vectors contain valid values, and in binary optimization issues, they have values of zero or one. The steps of the BFFA are as follows.

*Initial Values:* The production of the initial population is defined in this step, which refers to the number of sections in the farmland and the number of solutions accessible in each part. The number of initial populations is defined according to Eq. (2).

$$N = n \times k \quad (2)$$

In Eq. (2), The  $k$  parameter determines the number of sections connected to the optimization problem,  $n$  shows the number of solutions available in each land area, and parameter  $N$  indicates the total number of populations in the search space. Thus, the search space is divided into ( $k$ ) parts, including several solutions.

Eq. (3) and Eq. (4) are used to determine the quality of every single section of the farmland. The quality of each section of the farmland is acquired by averaging the available solutions in each section of the farmland. Eq. (3) separates solutions in each section. As such, the average of each key is calculated separately.

$$\begin{aligned} \text{Section}_s &= x(a_j), a = n * (s - 1) : n * s \\ s &= \{1, 2, \dots, k\}, j = \{1, 2, \dots, D\} \end{aligned} \quad (3)$$

In Eq. (3),  $s$  shows the number of segments,  $x$  is equal to all solutions in the search space, and  $j = [1, \dots, D]$  indicates the dimension of the variable  $x$ .

$$\begin{aligned} \text{Fit\_Section}_s &= \text{Mean}(\text{all Fit}(x_{ij}) \text{ in Section}_s); s = \{1, 2, \dots, k\}, i \\ &= \{1, 2, \dots, n\} \end{aligned} \quad (4)$$

In Eq. (4), *Fit\_Section* defines the level of the quality of section's solutions of the farmland of which each part has a specific rate and indicates the average fit or suitability of all the solutions in each section in the search space. Thus, the average number of accessible solutions in each part is obtained and stored in *Fit\_Sections* for each area of land.

*Updating Memory:* The local memory of each section and the global memory are updated after the solutions, and the average of each section of the farmland is determined. Some of the best states of each section and the best states of all sections are stored locally and globally.

After determining the circumstance of each part by Eq. (4), the part that has the worst condition will experience the most significant changes. According to Eq. (5) and Eq. (6), all the solutions in the worst part of the farmland would be hybrid with one in the global memory.

$$h = \alpha * \text{rand}(-1, 1) \quad (5)$$

$$X_{new} = h * (X_{ij} - X_{MGlobal}) + X_{ij} \quad (6)$$

In Eq. (6),  $X_{MGlobal}$  is a random solution among the available solutions in the global memory, and  $\alpha$  is a number between zero and one that must be assigned at the beginning of land fertility evaluation. The parameter  $X_{ij}$  is a solution in the worst section of the farmland picked to perform the changes, and  $h$  is a decimal number computed according to Eq. (5). As a result,  $X_{new}$  delivers a new solution featuring the adapted changes. After making changes in the worst section of the farmland, the other parts must be hybrids with the available solutions in the entire search space.



*Soil Composition:* Some available solutions in all locations are combined with the best existing solution ( $Best_{Global}$ ) to improve the quality of the available solutions in each section. The hybridization of the supposed solution with  $Best_{Global}$  or  $Best_{local}$  is determined by Eq. (7).

$$H = \begin{cases} X_{new} = X_{ij} + \omega_1 * (X_{ij} - Best_{Global}(b)), Q > rand \\ X_{new} = X_{ij} + rand(0, 1) * (X_{ij} - Best_{local}(b)), else \end{cases} \quad (7)$$

In Eq. (7), a new solution is generated based on two methods. In Eq. (7),  $Q$  is a parameter in [0-1] that determines the hybridization of solutions with the best global ( $Best_{Global}$ ).  $\omega_1$  is an integer and should be specified at the beginning of the algorithm. The value gradually declines due to the iteration of the algorithm.  $X_{ij}$  parameter is the solution selected to apply the changes from all sections. As a result,  $X_{new}$  is a new solution and is generated according to the user changes.

In the proposed model, the V-shaped function was applied to place the processes of the BFFA. Therefore, the V-shaped function continuously changes the position of the solutions in the FFA towards the binary state according to Eq. (8) [28].

$$V(X_i^d(t)) = \left| \left( \frac{\sqrt{\pi}}{2} \int_0^{\left(\frac{\sqrt{\pi}}{2} X_i^d(t)\right)} e^{-t^2} dt \right) \right| \quad (8)$$

In Eq. (8)  $X_i^d$  is the continuous value of the  $i$  solution in the  $d$ th dimension in iteration  $t$ . The output of the V-shaped transfer function is still in a constant state between 0 and 1, so a threshold must be set to convert it to a binary value, the random point of which is determined by Eq. (9). The V-shaped function converts the solutions to binary values to FS.

$$X_i^d(t+1) = \begin{cases} 0 & \text{if } rand < V(X_i^d(t)) \\ 1 & \text{if } rand \geq V(X_i^d(t)) \end{cases} \quad (9)$$

In Eq. (9)  $X_i^d$  shows the position of the  $i$ th solution in the population of the FFA in iteration  $t$  in the  $d$  dimension. Also,  $rand$  is a number between zero and one of the uniform distribution types. Therefore, the solutions of the proposed model move in a binary search space using Eq. (8).

*Fit Function:* The fit function is used to measure solutions. Eq. (10) is used as the fitness function for the proposed model. In Eq. (10)  $\gamma_R(d)$  is classification error,  $R$  is the number of selected attributes, and the parameters  $\alpha$  and  $\beta$  are in the range of zero and one,  $N$  is the total number of attributes.

$$Fitness = \alpha \times \gamma_R(d) + \beta \times \frac{|N| - |R|}{|N|} \quad (10)$$

*Ultimate Conditions:* Finally, the ultimate conditions of the algorithm are examined according to iteration. If the final condition is approved, the algorithm stops. Otherwise, the algorithm continues to work to create the ultimate conditions.

### 3.3 Classification

NB algorithm is a probabilistic learning algorithm derived from the Bayesian theory; it is a type of classification that creates classes based on conditional probabilities. To detect the data, the NB algorithm balances the decisions of different classes and then selects the best one for them. The NB algorithm explicitly operates on various hypothetical probabilities. The NB probability theory is defined according to Eq. (11).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (11)$$

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c) \quad (12)$$

If  $D$  is an instruction set of instances and contains class labels, then each sample is represented by an  $n$ -dimension feature vector  $X = \{x_1, x_2, \dots, x_n\}$ , where  $x_1, x_2, \dots, x_n$  are the attributed featured values of  $A_1, A_2, \dots, A_n$  in sample  $X$ . Assuming the existence of  $m$  class  $C_1, C_2, \dots, C_n$  and the new data sample (unlabeled), the NB classification will predict if the unknown sample  $X$  belongs to the class with the highest secondary probability.

### 3.4 Evaluation Criteria

Five criteria were used to evaluate the proposed model according to Table (2). True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) parameters are the main parameters concerning the evaluation criteria. TP entails the number of genuinely positive samples that have been correctly tagged. TN parameters involve the number of genuinely negative samples that have been correctly classified. FP parameters indicate the number of models without the disease but mistakenly diagnosed positive because of test results. An FN parameter shows the number of cases whose actual class is positive, and the classification algorithm has wrongly detected their category as a hostile class.

Table 2 Evaluation Criteria

Equations	Descriptions
$R = \frac{TP}{TP+FN}$	Recall (Sensitivity)
$P = \frac{TP}{TP+FP}$	Precision
$\frac{TN}{TN+FP}$	Specificity
$\frac{TP+TN}{TP+TN+FP+FN}$	Accuracy
$F - \text{Measure} = \frac{2 * P * R}{P + R}$	F-Measure

Precision indicates the number of actual samples regarding the total number of pieces. Recall means the number of positive examples that have been correctly detected on all truly positive models. It measures how well a test has been performed in diagnosing the disease in the absence of the disease. The accuracy percentage of a classification method on the training data set is the percentage of observations of the training set correctly classified by the method used. The F-Measure criterion is based on a hybridization of accuracy and callback.

## 4. Evaluation and Results

The statistical population in the present study consisted of a collection of data on heart disease derived from the credible and global UCI site. The dataset had a class attribute (diagnosis) that indicated the presence or absence of heart disease according to the values of the features. The value of one distinguished affliction by the disease, and the value of two indicated non-affliction. Table (3) shows the characteristics of the datasets in detail.

Table 3 Characteristics of the Heart Disease Data set

Datasets	Main Samples	FS	Number of missing samples	Number of samples used
Heart Disease [29]	270	13	No	0
Cleveland [30]	303	13	Somewhat	297
Hungary [31]	294	13	For all samples	amendment
Switzerland [32]	123	13	For all samples	amendment

The results in Table (4) illustrate why the proposed model in this study was selected to diagnose heart disease compared to other algorithms. The number of iterations is equal to 100, and the number of initial populations is equal to 30 in all algorithms. According to the results, it is evident that the proposed model has more accuracy percentage than other algorithms. The accuracy of the proposed model on the Heart, Cleveland, Hungary, and Switzerland datasets is 88.25, 86.91, 89.32, and 89.24.

Table 4 The Results of the Models based on Different Criteria

Datasets	criteria	ABC-NB	PSO-NB	FA-NB	<b>Proposed Model</b>
Heart Disease	Accuracy	<b>86.13</b>	<b>87.46</b>	<b>86.92</b>	<b>88.25</b>
	Precision	85.73	86.52	85.76	87.42
	Recall	85.95	86.80	86.13	88.05
	F-Measure	85.84	86.66	85.94	87.73
Cleveland	Accuracy	<b>84.67</b>	<b>83.45</b>	<b>85.72</b>	<b>86.91</b>
	Precision	84.16	82.83	85.48	85.81
	Recall	84.55	83.18	85.92	86.69
	F-Measure	84.35	83.00	85.70	86.25
Hungary	Accuracy	<b>86.35</b>	<b>87.14</b>	<b>85.78</b>	<b>89.32</b>
	Precision	85.96	86.73	84.90	88.67
	Recall	86.21	87.06	85.35	89.11
	F-Measure	86.06	86.89	85.12	88.89
Switzerland	Accuracy	<b>87.13</b>	<b>86.49</b>	<b>85.56</b>	<b>89.24</b>
	Precision	86.28	85.42	84.76	88.76
	Recall	86.94	85.97	85.23	89.09
	F-Measure	86.61	85.69	84.99	88.92

Table (4) shows that in all the datasets, the proposed model carry-outs better than the other methods, demonstrating the predictive performance strength of the proposed model. Figure (2) indicates that the proposed model has a global search ability and convergence potency that outperforms other running time methods. It is evident from Table (4) and Figure (2) that the proposed model has shown competitive performance compared to FA-NB, ABC-NB, and PSO-NB.

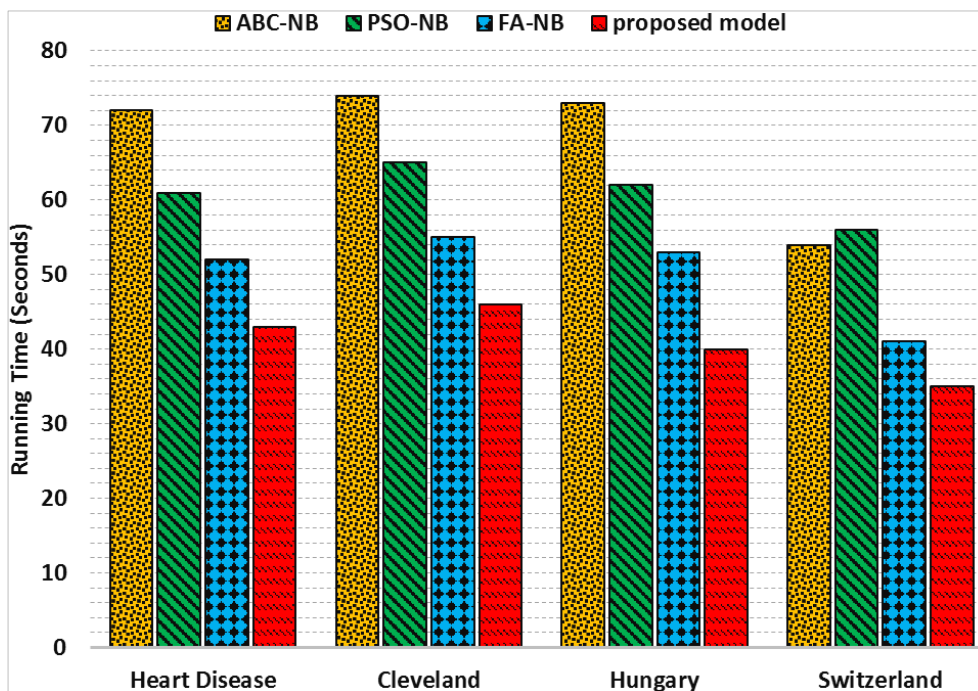


Figure 2 The running time in seconds for the proposed model and other algorithms

Figure (3) indicates convergences of ABC-NB, PSO-NB, FA-NB, and the proposed model. Additionally, Figure (3) confirms that the proposed model found the best possible optimum in iterations

71, 21, 11, and 61 for Heart Disease, Cleveland, Hungary, and Switzerland, respectively. The proposed model found the best optimum in fewer iterations than ABC-NB, PSO-NB, and FA-NB. The results revealed that the proposed model had a better convergence than ABC-NB, PSO-NB, and FA-NB.

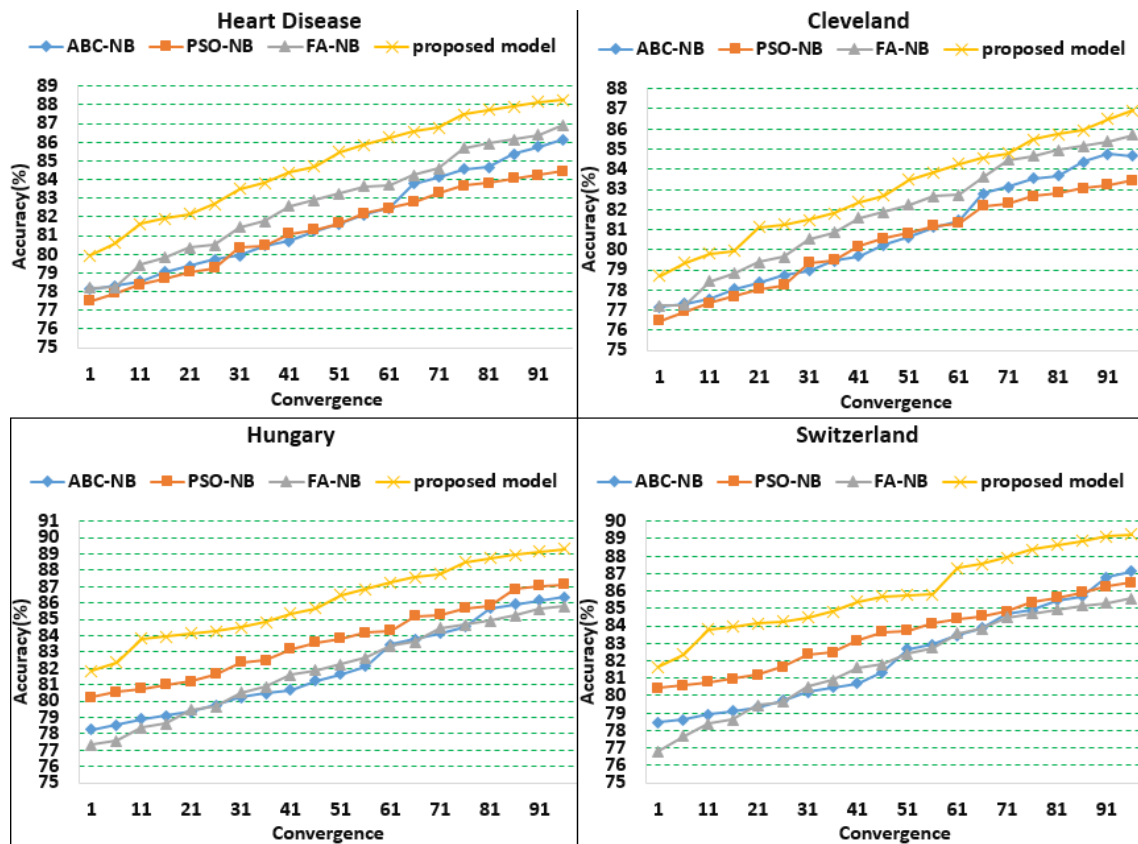


Figure 3 Convergence analysis between ABC-NB, PSO-NB, FA-NB, and proposed model

Table (5) shows the accuracy percentage of the different models based on FS. The accuracy percentage of the proposed model on the Heart data set with six features is equal to 92.23. The accuracy rate on the Cleveland dataset with six features in the proposed model is equivalent to 90.67%. The accuracy percentage of the proposed model on the Hungarian dataset with six features is equal to 92.68. The accuracy percentage of the proposed model on the Swiss dataset with six features is similar to 90.15. The outcomes indicate that the proposed model has a better accuracy percentage than other models.

Table 5 The Accuracy Percentage of the Models based on Feature Selection

Datasets	FS	ABC-NB	PSO-NB	FA-NB	Proposed Model
Heart Disease	6	91.35	91.59	91.42	92.23
	7	91.17	91.12	91.03	91.95
	9	90.07	90.25	89.79	90.56
	10	89.91	89.85	89.23	90.14
Cleveland	6	89.68	89.95	90.19	90.67
	7	89.24	89.63	89.96	90.21
	9	88.97	89.12	89.31	89.82
	10	88.65	88.84	88.91	89.28
Hungary	6	90.23	90.56	91.15	92.68
	7	89.75	90.32	90.89	92.32
	9	89.42	90.15	90.63	91.67
	10	89.12	89.92	90.25	91.38
Switzerland	6	90.72	90.25	89.91	91.15
	7	90.33	89.78	89.51	90.68
	9	89.86	89.16	88.36	90.17
	10	88.36	87.69	87.45	89.75

#### 4.1 Comparison and Evaluation

Table (6) compares the proposed model with other models based on the accuracy percentage. The accuracy percentage of ANN, a hybridization of WOA-SA, SVM, Gravitational Search Algorithm-KNN (GSA-KNN), and Particle Swarm Optimization-KNN (PSO-KNN) on the Heart dataset was equal to 83.33, 87.78, 85, 82.96 and 83.7, respectively. The accuracy percentage of the proposed model is 88.25, which is higher than other models.

Table 6 Accuracy Percentage of Models based on Feature Selection

Datasets	Models	Refs	Accuracy
Heart Disease	ANNs	[19]	83.33
	WOA-SA	[20]	87.78
	SVM	[21]	85.00
	GSA-KNN	[33]	82.96
	PSO-KNN		83.7
	<b>Proposed Model</b>	-	88.25
Cleveland	C4.5	[23]	80.2
	SVM	[34]	72.36
	NB		76.19
	Deep Belief Network (DBF)		78.69
	PSO-DBF		86.44
	<b>Proposed Model</b>	-	86.91
Hungary	SVM	[34]	75.76
	NB		80.95
	Deep Belief Network		87.10
	PSO-DBF		87.10
	<b>Proposed Model</b>	-	89.32
Switzerland	SVM	[34]	75.51
	NB		76.47
	Deep Belief Network		77.78
	PSO-DBF		84.00
	<b>Proposed Model</b>	-	89.24

The accuracy percentage of the C4.5 [23], SVM, NB, DBF, and PSO-DBF on the Cleveland dataset were equal to 80.2, 72.36, 76.19, 78.69, and 86.44, respectively. The accuracy percentage of the proposed model was higher than other models and similar to 86.91. The SVM, NB, DBF, and PSO-DBF accuracy percentages on the Hungary dataset were 75.76, 80.95, 87.10, and 87.10. The accuracy percentage of the proposed model is 89.32, which is higher than other models.

#### 4.2 Discussion

Analysis and results have been undertaken in (Cleveland, Hungary, and Switzerland) by using the study [34] for comparison. Accordingly, SVM, NB, Deep Belief Network, and PSO-DBF have been the existing studies to compare the proposed model. The analysis has been done concerning the accuracy obtained results in Table 6. Though the previous works applied various methods for classifying the diseases' dataset, the proposed model showed a high accuracy rate, confirming its efficiency in predicting conditions. NB showed high efficiency and excellent capability to solve complex pattern classification problems [34]. Generally, NB is a valuable and rapid method. Moreover, the selection of the most suitable neighbors by NB accelerates the diagnosis process, as it does not consider all neighbors of the evaluated item. Therefore, the proposed model is a fast and accurate decision-making system for detecting diseases.

The results indicated that the accuracy of the proposed model was 88.25% in heart, 86.91% in Cleveland, 89.32% in Hungary, and 89.24% in Switzerland. However, the proposed model indicated high accuracy of 89% than other methods. In addition, the proposed model showed high recall, precision, and f-measure rate than the other models. The proposed model showed superior performance and provided a

balance between the number of features and classification accuracy. The proposed model used the binary operation to enhance the searching process to find essential features.

## 5. Conclusion and Future Works

The paper processed various datasets derived from the UCI standard database. The present study predicted heart disease by applying patients with heart disease characteristics via BFFA and NB. The proposed model consisted of feature normalization, replacement of lost values, and FS. The most important features were identified in the study, and a better performance concerning precision, sensitivity, and accuracy was achieved by selecting the features based on the BFFA. The Hungary dataset's SVM, NB, DBF, and PSO-DBF accuracy were 75.51%, 76.47%, 77.87%, and 84.00%, respectively. The accuracy of the proposed model was equal to 89.24%. With a higher of 89%, this model has better performance than SVM, NB, DBF, and PSO-DBF in diagnosing heart disease. Conducted experiments and simulations showed that the medical system introduced in this study approved better performance on the heart patient database and had different accuracy percentages on different datasets. In future studies, the sensitivity of the BFFA parameters may be discovered. The BFFA may also be compared to other FS algorithms, using various large datasets and different classifiers to take better results.

## References



- [1] M. Langarizadeh, S. M. A. Sadr-ameli, and M. Soleymani, "Development of Vital Signs Monitoring Decision Support System for Coronary Care Unit Inpatients," *Journal of Health Administration*, vol. 20, no. 67, pp. 75-88, 2017.
- [2] L. B. Sorkhabi, F. S. Gharehchopogh, and J. Shahamfar, "A systematic approach for pre-processing electronic health records for mining: case study of heart disease," *International Journal of Data Mining and Bioinformatics*, vol. 24, no. 2, pp. 97-120, 2020.
- [3] M. Hassanzadeh, I. Zabbah, and K. Layeghi, "Diagnosis of Coronary Heart Disease using Mixture of Experts Method," *Journal of Health and Biomedical Informatics*, vol. 5, no. 2, pp. 274-285, 2015.
- [4] S. M. S. Shah, F. A. Shah, S. A. Hussain, S. Batool, "Support Vector Machines-based Heart Disease Diagnosis using Feature Subset, Wrapping Selection and Extraction Methods", *Computers & Electrical Engineering*, vol. 84, no. 1, pp. 106628, 2020.
- [5] T. Vivekanandan, and N. C. S. N. Iyengar, "Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," *Computers in Biology and Medicine*, vol. 90, no. 1, pp. 125-136, 2017.
- [6] S. M. S. Shaha, S. Batoolb, I. Khana, M. U. Ashrafac, S. H. Abbasa, S. A. Hussaina, "Feature extraction through parallel Probabilistic Principal Component Analysis for heart disease diagnosis," *Physica A: Statistical Mechanics and its Applications*, vol. 482, no. 1, pp. 796-807, 2017.
- [7] S. Nazari, M. Fallah, H. Kazemipoor, A. Salehipour, "A fuzzy inference- fuzzy analytic hierarchy process-based clinical decision support system for diagnosis of heart diseases," *Expert Systems with Applications*, vol. 95, no.1, pp. 261-271, 2018.
- [8] A.M. Alqudah, "Fuzzy expert system for coronary heart disease diagnosis in Jordan," *Health and Technology*, vol. 7, no. 2, pp. 215-222, 2017.
- [9] S. Javadzadeh, H. Shayanfar, and F. S. Gharehchopogh, "A Hybrid Model based on Ant Lion Optimization Algorithm and K-Nearest Neighbors Algorithm to Diagnose Liver Disease," *Ilam-University-of-Medical-Sciences*, vol. 28, no. 5, pp. 76-89, 2020.
- [10] M. H. F. Zarandi, A. Seifi, M. M. Ershadi, and H. Esmaeeli, "An Expert System Based on Fuzzy Bayesian Network for Heart Disease Diagnosis," *North American Fuzzy Information Processing Society Annual Conference, NAFIPS 2017: Fuzzy Logic in Intelligent System Design*, vol. 648, pp. 191-201, 2017.

- [11] S. Safdar, S. Zafar, N. Zafar, and N. F. Khan, "learning based decision support systems (DSS) for heart disease diagnosis: a review," *Artificial Intelligence Review*, vol. 50, no. 4, pp. 597-623, 2018.
- [12] H. Shayanfar, and F. S. Gharehchopogh, "Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 71, pp. 728-746, 2018.
- [13] Y. Jiang, H. Lin, X. Wang, and D. Lu, "A Technique for Improving the Performance of Naive Bayes Text Classification," *International Conference on Web Information Systems and Mining, WISM 2011: Web Information Systems and Mining*, vol. 6988, pp. 196-203, 2011.
- [14] A. Benyamin, F. S. Gharehchopogh, and S. Barshandeh, "Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems," *International Journal of Intelligent Systems*, vol. 36, no. 3, pp. 1270-1303, 2021
- [15] A. Hosseinalipour, F. S. Gharehchopogh, M. Masdari, and A. Khademi, "A novel binary farmland fertility algorithm for feature selection in analysis of the text psychology," *Applied Intelligence*, vol. 51, pp. 4824-4859, 2021.
- [16] S. Khalandi, and F. S. Gharehchopogh, "A New Approach for Text Documents Classification with Invasive Weed Optimization and Naive Bayes Classifier," *Journal of Advances in Computer Engineering and Technology*, vol. 4, no. 3, pp. 167-184, 2018.
- [17] S. Ardam, and F. S. Gharehchopogh, "Diagnosing Liver Disease using Firefly Algorithm based on Adaboost," *Journal of Health Administration*, vol. 22, no. 1, pp. 61-77, 2019.
- [18] V. Maihami, A. Khormehr, and E. Rahimi, "Designing an expert system for prediction of heart attack using fuzzy systems," *HBI Journals*, vol. 21, no. 4, pp. 118-131, 2016.
- [19] M. Kazemi, H. Mehdizadeh, and A. Shiri, "Heart disease forecast using neural network data mining technique," *Ilam-University-of-Medical-Sciences*, vol. 25, no. 1, pp. 20-32, 2017.
- [20] Z. Hassani, and M. Khosravi, "Diagnosis of Coronary Heart Disease by Using Hybrid Intelligent Systems Based on the Whale Optimization Algorithm Simulated Annealing and Support Vector Machine," *Engineering Management and Soft Computing*, vol. 4, no. 2, pp. 79-93, 2019.
- [21] M. S. Mahmoodi, "Designing a Heart Disease prediction System using Support Vector Machine," *Journal of Health and Biomedical Informatics*, vol. 4, no. 1, pp. 1-10, 2017.
- [22] R. Akhoondi, and R. Hosseini, "A Novel Fuzzy-Genetic Differential Evolutionary Algorithm for Optimization of A Fuzzy Expert Systems Applied to Heart Disease Prediction," *Soft Computing Journal (SCJ)*, vol. 6, no. 2, pp. 32-47, 2017.
- [23] H. Sabbagh Gol, "Detection of Coronary Artery Disease Using C4.5 Decision Tree," *Journal of Health and Biomedical Informatics*, vol. 3, no. 4, pp. 287-299, 2017.
- [24] Zabbah, M. Hassanzadeh, and Z. Koohjani, "The Effect of Continuous Parameters on The Diagnosis of Coronary Artery Disease Using Artificial Neural Networks," *Journal of Torbat Heydariyeh University of Medical Sciences (Journal of Health Chimes)*, vol. 4, no. 4, pp. 29-39, 2017.
- [25] R. Safdari, M. Ghazi Saeedi, M. Gharooni, M. Nasiri, and G. Argi, "Comparing performance of decision tree and neural network in predicting myocardial infarction," *Journal of Paramedical Sciences & Rehabilitation*, vol. 3, no. 2, pp. 26-35, 2014.
- [26] Mahmoudi, R. A. Moghadam, M. H. Moazzam, S. Sadeghian, "Prediction model for coronary artery disease using neural networks and feature selection based on classification and regression tree," *Shahrekord-University-of-Medical-Sciences*, vol. 15, no. 5, pp. 47-56, 2013.
- [27] H. Tahmasbi, M. Jalali, and H. Shakeri, "An Expert System for Heart Disease Diagnosis Based on Evidence Combination in Data Mining," *Journal of Health and Biomedical Informatics*, vol. 3, no. 4, pp. 251-258, 2017.
- [28] S. Mirjalili, and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013.
- [29] Statlog, "statlog+(heart)," 1997.[Online]. Available: [https://archive.ics.uci.edu/ml/datasets/statlog+\(heart\)](https://archive.ics.uci.edu/ml/datasets/statlog+(heart)). [Accessed: 25-May-2021].
- [30] cleveland, "cleveland," 2005, [Online]. Available: <https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/cleveland.data>. [Accessed: 25-May-2021].
- [31] hungarian, "hungarian," 1998, [Online]. Available: <https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/processed.hungarian.data>. [Accessed: 25-May-2021].

- [32] switzerland ,” switzerland ,“ 2002, [Online]. Available:<https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/processed.switzerland.data>. [Accessed: 25-May-2021].
- [33] S. Nagpal, S. Arora, S. Dey, A. Shreya, “Feature Selection using Gravitational Search Algorithm for Biomedical Data. *Procedia Computer Science*,” vol. 115, no. 1, pp. 258-265, 2017.
- [34] A. M. Alhassan, and W. M. W. Zainon, “Taylor Bird Swarm Algorithm Based on Deep Belief Network for Heart Disease Diagnosis,” *Applied Sciences*, vol. 10, no. 18, pp. 1-20, 2020.



# Detection of Heart Rate Variability from Photoplethysmography (PPG) Signals Obtained by Raspberry Pi Microcomputer

 Ziyne Pamuk<sup>1</sup>,  Ceren Kaya<sup>2</sup>

<sup>1</sup>Corresponding Author; Zonguldak Bulent Ecevit University, Department of Biomedical Engineering, Zonguldak, Turkey; ziyne.pamuk@beun.edu.tr; +90 372 291 26 23

<sup>2</sup> Zonguldak Bulent Ecevit University, Department of Biomedical Engineering, Zonguldak, Turkey; ceren.kaya@beun.edu.tr, crnkaya@hotmail.com;

Received 16 November 2021; Revised 30 March 2022; Accepted 26 April 2022; Published online 30 April 2022

## Abstract

Photoplethysmography (PPG) signals are signals obtained as a result of optically measuring volumetric changes in capillaries. Volumetric changes in capillaries also depend on the work of heart. According to recent researches, it has been seen that PPG signals contain a lot of information about the physiological and biological state of related person. Most of these studies are based on the analysis of characteristics and waveforms of PPG signals obtained with a single wavelength in time and frequency domains. In this study, 10 minutes of data was taken from the left index finger of a 24-year-old male, which was positioned horizontally using a MAX30100 sensor and Raspberry Pi 4 microcomputer. Experiments are carried out in the fully resting state of a male volunteer in outdoors and stressful environments. While the MAX30100 sensor shows the heartbeat on the screen, it also gives PPG signal data, which is a single wavelength, into a .csv file as received data. In these cases, five different time domain parameters of received PPG signals are extracted. When the results are interpreted, it is seen that all results are meaningful and consistent.

**Keywords:** photoplethysmography, heart rate variability, time domain parameters, raspberry pi

## 1. Introduction

Heart rate variability (HRV), which is a non-invasive measurement, is an important parameter in the prognosis of some diseases, especially cardiovascular diseases. The parameter called HRV, which is revealed by making use of the effect of the sympathetic branch of the autonomic nervous system (ANS) accelerating heart and parasympathetic branch (nervus vagus) slowing the heart, is also used as an important stress measurement parameter in terms of providing information about the sympathovagal balance. Changes in ANS activation are different in patients and healthy individuals.

ANS responds to internal and external factors by regulating visceral functions such as cardiovascular function, respiration, thermoregulation, neuroendocrine secretion, gastrointestinal and genitourinary functions, and plays a role in the control of vital functions [1]. It is a system that transmits impulses from the central nervous system to peripheral organs and is more effective. Thus, it controls the heart rate, the contractile force of heart, the contraction and dilatation of vessels, the contraction and relaxation of smooth muscles in various organs, and the secretion from the endocrine and exocrine glands.

ANS is anatomically and functionally divided into two as sympathetic and parasympathetic. The preganglionic fibers of parasympathetic nervous system (PSS) arise from the brainstem and are known as craniosacral fibers [2]. The vagus carries fibers to the heart, lungs, and other organs and forms the principal parasympathetic innervation of these organs. PSS causes a decrease in heart rate and blood pressure and is more concerned with the restoration and conservation of energy by increasing the digestion, absorption and excretion of nutrients [2].

The nuclei of sympathetic preganglionic nerve fibers are located in the sympathetic ganglion chain located in the lateral horns between T1-L2 of the spinal cord. Since the adrenal medulla is stimulated by sympathetic preganglionic fibers, adrenaline is released as a result of stimulation of nicotinic acetylcholine receptors. The neurotransmitter noradrenaline is found in the majority of postganglionic sympathetic nerve endings, in the adrenal medulla, and in the presynaptic terminal. Synthesis of

adrenaline and noradrenaline in sympathetic postganglionic nerve endings and adrenal medulla is similar, but noradrenaline is converted to adrenaline more in the adrenal medulla [2].

The heart is under the control of sympathetic and parasympathetic nerves. A strong sympathetic stimulus can increase the normal heart rate from 70 beats per minute (bpm) in young individuals to approximately 250 bpm. Sympathetic stimuli can increase the volume of blood pumped and ejection pressure, increasing the force of heart contraction up to twice. The amount of blood pumped by the heart per minute (cardiac output) can increase by more than 100% with sympathetic stimulation.

Parasympathetic impulses, strong impulses of parasympathetic nerve fibers reaching the heart with the vagus nerves, can stop the heartbeat for a few seconds, and severe vagus impulses can reduce the force of contraction of the heart by 20-30% [3]. The vagus fibers radiate mainly to the atria rather than the ventricles, where strong contraction of the heart occurs. Therefore, vagus impulses reduce heart rate without significant reduction in cardiac contractility [3].

The rhythm produced due to spontaneous depolarization of the sinus node in the heart is called intrinsic heart rhythm [4]. Heart rate varies with age and gender. Hypoxia, exercise, and temperature are also among other factors that can affect intrinsic heart rate [5, 6]. Sympathetic and parasympathetic systems, which are the components of the autonomic system, are the main variables that affect the intrinsic heart rate.

## 2. Related Works

The first use of HRV information was made in 1965 by Hon. In his study, they showed that a difference was observed between each heartbeat before the visible changes in heart rate in babies in the womb [7]. HRV analyzes have an increasing importance in diagnostic use, especially in cardiology. Huikiri and Stein, and Zuern et al. demonstrated HRV knowledge as a risk assessment tool in patients with myocardial infarction recovery. In a large number of studies, it has been observed that the risk of mortality increases within a few years after myocardial infarction in patients with reduced or abnormal HRV [8, 9].

Studies have shown that patients with reduced HRV values are at risk of sudden cardiac death. When the researchers examined the records of healthy-looking people whose Electrocardiography (ECG) holter recordings were taken before their sudden death, they found that these people had low HRV values. Likewise, low HRV values were found in holter recordings taken just before the sudden death of angina patients [10, 11].

In many studies, it has been shown that the low SDNN value, which is one of the HRV time domain parameters, can be used as a tool to predict the mortality rate. Kearney et al. found that low SDNN value, low serum sodium content and high sodium creatinine amount indicate an increased risk of death. In the study of Nolan et al. on 433 patients, it was shown that the decreased SDNN value could predict death, not sudden death, in developing heart failure [12-14]. When the HRV values of the control group and the patient group with chronic heart failure were compared, it was observed that there was a significant difference between them. In addition, the low frequency spectral component of HRV was found to be associated with increased mortality in patients with chronic heart failure [15].

Today, in another study on HRV, 24-hour recordings were taken from participants between the ages of 20-70 to show the effect of age and gender on HRV, and when time domain analyzes were made, a decrease was observed in all parameters, especially in parasympathetic activity, due to aging. In other more detailed studies, it has been observed that parasympathetic activity decreases rapidly until the age of 80 and then increases again. In the same study, studies that showed decreased HRV in patients at risk of sudden cardiac death and low HRV values in healthy patients whose ECG holter records were examined before their sudden death were mentioned [16].

In a classification-based study in the form of HRV discrimination obtained from long-term measurements, records with normal sinus rhythm and records from PhysioBank of the patient with heart failure were used. Using long-term (24-hour) recordings of 54 patients with normal sinus rhythm (30 males, 24 females) and 29 heart failure patients (8 females, 2 males, 14 genders of the remaining 21

subjects unknown), aged 34-79 years, 9 different HRVs were used. The time domain parameter was calculated. As a result of the study, it was observed that the standard deviation (SDNN) parameter in 9 different long-term HRV measurements using linear discriminant analysis had the highest class discriminating power in the discrimination of diseased and healthy individuals [17].

Autonomic activity during sleep is assessed using spectral analysis of HRV. In the study, polygraphic recordings were taken from 11 healthy people during the night, and very low frequency, low frequency and high frequency components were evaluated. It was observed that the total spectrum power and very low frequency component were quite high in REM sleep, and the low frequency/high frequency ratio reflecting the sympathovagal balance reached its maximum value in REM sleep [18].

ECG recordings of 8 minutes under controlled breathing (12-15 breaths/minute) were taken from 202 patients aged  $52 \pm 9$  years with moderate and severe chronic heart failure. Decreased short-term LF power during controlled breathing when HRV analyzes are performed is a strong predictor of sudden death in patients with chronic heart failure and is independent of other variables [19].

In the studies, HRV was calculated based on the ECG signals of the person in stressful and non-stressful situations using different combinations of stress situations. In the study of Brunelli and Poggio, the mensa test was used to stress the person and ECG signals were taken throughout the experiment. As a result of time and frequency axis analysis, when mental activity state and resting state were compared, it was determined that the mean RR interval was lower and the pNN50 value was higher in the resting state. Although there was no significant difference in frequency axis analysis, there was an increase in the LF/HF ratio in the mental activity state [20]. The relationship between the stress caused by visual stimuli and HRV was examined and it was determined that significant changes occur in HRV in case of visual stress [21, 22].

In the study by Biel et al., the effects of external noise sounds on stress were investigated. Five different noise sounds, including car horn, baby crying, drill (drilling sound) and sounds from the construction site, were listened to 17 participants in the experiment. A total of 10 minutes of ECG recording was taken. In the first 1-minute part of the experiment procedure, the person was listened to a relaxing music and this part was accepted as the basic level. Then, after listening to each noise sound, there is a 1-minute rest period. As a result of the analysis, it was concluded that these noise sounds increase the stress level [23].

In the study of Mayya et al., which used PPG signals instead of ECG, PPG signals were obtained from 49 volunteers under two conditions. In the first case, the subject was asked to relax for 10 minutes and the signal was received for 4 minutes. In the second case, signals were recorded under five different situations that would put the person in a state of stress. These five situations are; Stroop color/word test, mental arithmetic test, memory test, public speaking test and counting backwards. Each stage took approximately 2 minutes. In addition, after each stage was completed, the subjects were asked to rate their stress levels between 1 and 5. The baseline was accepted as the lowest stress level and the scores from the subjects were averaged for each stage. As a result of the analyzes, it was examined whether there was a statistically significant difference between the baseline and each task, and it was seen that the RMSSD, pNN50, HF parameters, which are time and frequency domain analyzes, were statistically different in the baseline and in each stress condition [24].

In the study by Lu et al., which was conducted to compare HRV information obtained from ECG and PPG signals, 7 minutes of simultaneous ECG and PPG signals were recorded from 42 participants. Ag/AgCl electrodes were placed in lead-I arrangement for ECG recordings. For the PPG recording, the recording was taken from the left ear lobe. The 5-minute artifact-free portions of the recordings were used for analysis. The time and frequency domain parameters obtained from the analyzes were compared with each other with the help of statistical analysis methods. A correlation of over  $r = 0.95$  was found between the parameters of both measurements. This shows that the measurement results are highly correlated with each other [25].

ECG and PPG signals were obtained from 19 healthy individuals and these signals were sampled at 250 and 500 Hz sampling rates, respectively in Selveraj et al. study. While the detection of R waves in the ECG signal is made with the Pan and Tompkins algorithm, the peaks of the PPG signal; signal scaling,

thresholding, local peak detection, removal of detected very close peaks. When the values obtained from ECG and PPG signals were compared, it was seen that the lowest error occurred in SDNN with 2.46% and SD2 with 2% (reflecting long-term HRV in Poincare measurements), while the highest error occurred in pNN50 with 29.89%. In addition, pulses detected during PPG measurement do not contain high-frequency components associated with heartbeats, while ECG signals contain these features [26].

In the study of Taelman et al., the relationship between heart rate, HRV and mental stress was tried to be examined. Measurements taken from 28 participants were carried out in two stages, with and without mental testing for each subject. Mensa test was used for mental stress and ECG signals were taken throughout the experiment. As a result of time and frequency domain analysis, it was determined that the mean RR interval was lower when the mental activity state was compared with the resting state, and it was observed that the pNN50 value was higher in the resting state. Although there was no significant difference in frequency domain analysis, there was an increase in the LF / HF (low frequency / high frequency) ratio in the mental activity state [27].

In Wu and Lee, and Wu et al. studies, in which the relationship between stress caused by visual stimuli and HRV was examined, ECG signals were obtained from 50 participants. Signal acquisition phase was carried out in 2 parts. The first 5-minute period was the state of rest and no warning was applied to the subject. In the second 5 minutes, firstly, landscape pictures were shown for positive effect, and then black and white pictures were shown for visual stress. When the results were analyzed, it was seen that the RR intervals and the mean RR interval change in each different stimulus condition began to be wider when the subject was at rest, and narrower in the case of visual stress. Considering the frequency domain analysis, the average LF and HF power increased at the time of visual stress. Correspondingly, LF/HF was also increased, and significant changes in HRV were found under visual stress [28, 29].

In McDuff et al.'s study, data was collected by digital camera recording using a slightly different method in the signal collection stage for the measurement of stress. With a camera placed 3 m away from the person, a camera that can capture the effect of stress on the person's breathing and HRV was used. As a reference, blood volume pulses and electrodermal activity were obtained with the help of a finger sensor, and respiratory information was obtained by wearing a chest strap. 7 female 3 male 10 participants participated in the experiment. During the first two minutes, the signal was received while the participants were relaxed, then they were asked to count backwards from 4000 by subtracting 7 each time, and the signal was taken in this way for a while. These physiological signals and camera recordings were taken synchronously and then the results were compared. When the camera recordings and the physiological records used as reference are compared after the experiment, the relationship between them is  $r=1$  for heart rate,  $r= 0.93$  for respiration,  $r= 0.93$  for HR LF power,  $r= 0.93$  for HF, for LF/HF  $r=0.93$  and  $p<0.01$  were found [30].

In the study of Wang and Wang, which aimed to evaluate stress and emotion over HRV, ECG signals were obtained from 26 participants. In order to induce emotion in people, 3 types of movies were prepared for fear, relaxation and happiness, each of which is between 3-10 minutes. The picture matching game was used to complete the task in the appropriate time for the stress situation. In addition, to validate the induction of appropriate emotional information, subjects rated their emotional state using a Likert scale. When the results were analyzed, there was a decrease in the RR intervals in the 4 different mood states compared to the calm state. Heart rate in fear and stress increased more than in relaxation and happiness. SDNN index short interval changes of HRV under negative emotion (fear, stress) were higher than positive mood. In addition, the pNN50 value, which indicates vagus nerve activity, was suppressed under stress [31].

One study examined changes in sleep after supramaximal activity. It was observed that the duration of HRV at night of activity was shortened, and HRV values returned to normal only after two days [32]. In another study, recovery in HRV after single and multiple (4 times) WanT administration was examined after 20 minutes, 1 hour and 2 hours; It was observed that HRV values decreased more after multiple WanT applications, and resting HRV results could not be achieved in both applications after two hours [33].

In a study, the effect of short-term submaximal cycling ergometer loading on HRV recovery in different body positions (sitting, lying down, lying with feet up) was examined and HRV was greater in supine positions during a 15-minute rest period, but HRV decreased to resting values in all three positions. It was understood that it could not be reached completely [34]. There are also studies on other factors that affect HRV recovery. In a study examining the effects of pre-exercise energy drink and alcohol intake on HRV, lower HRV values were achieved after exercise in groups receiving alcohol and energy drinks [35].

Ten male swimmers in the 13-14 age group, who were athletes in a special swimming team in Ankara, voluntarily participated in the research. The mean age of the participants was  $13.40 \pm 0.52$  years, their height was  $168.70 \pm 8.35$  cm, their body weight was  $59.56 \pm 11.86$  kg, and their body fat ratio was  $14.51 \pm 5.69\%$ . The aim of this study was to examine the change in HRV parameters after 50 m sprint swimming activity and to understand the effect of ANS on the heart. As a result of the research, it was observed that high-intensity and short-term 50 m sprint swimming activity affected the sympathovagal balance by increasing the sympathetic activity on the heart and decreasing the vagal effect [36].

In a cross-sectional study by Ulu et al., 28 patients under dialysis treatment and 30 healthy controls were compared in terms of HRV. HRV was evaluated with 24-hour Holter electrocardiography. HRV was found to be significantly lower in patients with chronic renal failure under dialysis treatment compared to the control group (SDNN and pNN50 values were  $95.4 \pm 31.5$ ,  $127.5 \pm 38.8$ ,  $p=0.001$  and  $8.3 \pm 6.1$ , respectively).  $16 \pm 9$ ;  $p=0.71$ ) [37].

The contributions of the paper are as follows:

- None of the studies in literature are real-time. In all studies, data were first taken and then analyzed in the computer environment.
- In our study, reading and analysis of the data was done with Raspberry Pi 4 microcomputer. No normal computer was used anywhere. Python codes work easily on Raspberry Pi 4 microcomputer. We aim to produce a prototype with this study. For this purpose, a very small, portable, battery-powered hardware was preferred by disabling the computer.

### 3. Material and Method

#### 3.1 PPG Signal

PPG signal measurement is a noninvasive, electro-optical method that provides information about the volume of blood flowing at a test site of the body close to the skin. A PPG signal is obtained by light that illuminates the relevant area of body and then reflects or passes through that area [38]. The PPG signal is produced by the periodic beating of the heart. The waveform and characteristic parameters of a typical PPG signal are given in Figure 1.

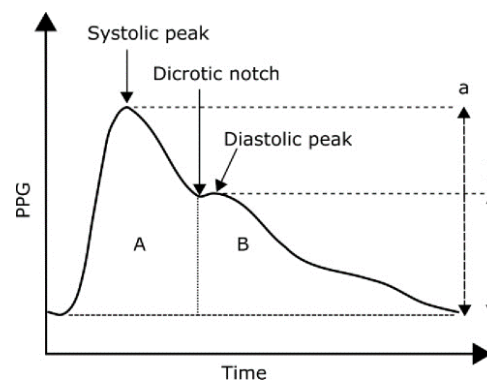


Figure 1 PPG Signal Waveform [39]

PPG signals are received optically from the body with the help of sensors with a light source and photodiode. PPG signals can be obtained by two different finger-type probes including transmissive and reflection modes, respectively as shown in Figure 2.

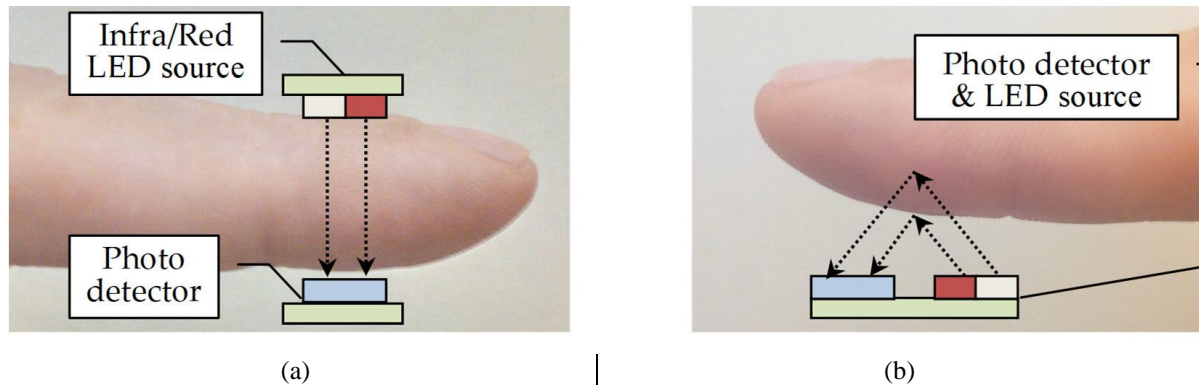


Figure 2 Finger-type PPG probes in (a) transmission (b) reflection modes [40]

In transmission mode sensor shown in Figure 2(a), the emitting LED and receiving photodiode are positioned opposite each other. This type of sensor can be used in areas of the body where light transmittance is high, such as fingertips and earlobes. Each time the heart pumps blood to the body, the blood density at the fingertip changes accordingly. Depending on the change in blood density, the intensity of the light sent from one surface of finger, reaching the opposite, also changes. As the blood density increases, the light intensity reaching the photodiode on the opposite side decreases. When the blood is drawn from the capillaries, the light intensity reaching the opposite side increases. Depending on this change, the light intensity on the photodiode changes and the PPG signal occurs [41].

In reflection mode sensor in Figure 2(b), the receiver and transmitter are positioned in the same direction. This type of sensor is used in areas of the body with low light transmission. In this sensor, unlike the transmission mode sensor, when the blood density increases, the amount of reflection will be higher, and the light intensity falling on the photodiode will increase. When the blood is withdrawn from the vessels, the light intensity falling on the photodiode will be less since the amount of reflection will decrease. As a result of this change in blood density, the PPG signal is obtained [41].

### 3.2 Heart Rate Variability

Heart Rate Variability (HRV) defines the change between heart beats based on the acceleration of heart rhythm with the activation of sympathetic nervous system and the slowing of heart rhythm with the activation of parasympathetic nervous system. Multiple methods are available to evaluate autonomic nervous system activity. These methods are; measurement of adrenaline and noradrenaline in the urine [42], measurement of muscle sympathetic activity [43] or HRV. Among these methods, the most preferred and reliable measurement is HRV.

HRV indicates synchronization between heartbeats. HRV increases when heartbeats are irregular, and HRV decreases if heartbeats are fairly regular. Analysis of HRV is a reliable and non-invasive method for assessing autonomic regulatory responses. This measurement technique is not only used to reveal autonomous function changes. It can also provide information about the prognosis of some diseases or the probability of a healthy person becoming sick in the future.

In this context, HRV provides information about the autonomic regulation of heart via the parasympathetic and sympathetic nervous systems [44]. In healthy individuals, HRV has a circadian rhythm. It increases at night and decreases during the day. A decrease in HRV is accepted as an indicator of a decrease in parasympathetic activity and an increase in sympathetic activity [45]. In clinical studies

conducted in this context, showing psychosomatic symptoms; HRV has been reported to be reduced in individuals with chronic fatigue syndrome [46] anxiety and depression [47] and work stress [48]. PPG and RR intervals should be recorded for a certain period of time to determine HRV. Various algorithms can be used to calculate HRV, from simple statistical methods to complex non-linear mathematical methods [2].

### 3.3 HRV Time Domain Parameters

Measurement of time-domain variables is evaluated by statistical calculation of variables in RR intervals [45]. By looking at the intervals between consecutive QRS complexes on a 24-hour Electrocardiography (ECG), heart rate or the distance between consecutive normal complexes (normal-normal (NN) intervals) is determined. From these records, various parameters such as mean NN interval, difference between longest and shortest NN interval, average heart rate can be calculated. In time-domain measurements, necessity of taking long sections of 24 hours and providing standard conditions causes a decrease in patient compliance [49].

In this study, HRV vector was created depending on the time between these points, along with the peaks detected from the filtered PPG signal. The mean and standard deviation values of created vector were determined and five different time domain parameters (SDNN, SDD, RMSSD, PNN20 and PNN50) were extracted.

RMSSD is the square root of mean of the square of the RR interval differences. This measurement estimates the high frequency components of heart rate in short-term normal-to-normal recordings that reflect a parasympathetic regulation of the heart. A decrease in RMSSD (below 10) associated with a low SDNN (below 20) is associated with the risk of developing cardiac disease [49]. SDD is the standard deviation of the adjacent RR interval differences [36]. pNN50 is the proportionality coefficient obtained by dividing the total number of RR intervals by NN50 [36]. pNN20 is the ratio obtained by dividing pNN20 by the total number of NNs [36].

SDNN is the standard deviation of the time (NN) between successive QRS complexes on ECG signal and reflects all cycle components responsible for variability throughout the recording. SDNN is affected by the recording time and values decrease as the recording time decreases and increase as it increases. Short-term (5min) recordings reflect high-frequency changes, while long-term (24-hour) recordings reflect low-frequency changes. Therefore, it would not be correct to compare SDNN values in long and short-term records. SDNN covers the time from the beginning to the end of the ECG signal recording and the times should be standardized [49]. The formulas of HRV time domain parameters used in the study are given in Table 1.

Table 1 HRV Time Domain Parameters

HRV Time Domain Parameters (Abbreviations)	HRV Time Domain Parameters (Full Forms / Description)	Formulas
SDNN	Standard deviation of NN intervals	$\sqrt{\frac{1}{N-1} \sum_{j=1}^N (RR_j - RR)^2}$
SDD	Standard deviation of the differences between successive NN intervals	$\sqrt{\frac{1}{N-1} \sum_{j=1}^{N-1} ( RR_j - RR_{j+1}  - RR_{dif})^2}$
RMSSD	Root mean square of successive RR interval differences	$\sqrt{\frac{1}{N-1} \sum_{j=1}^{N-1} (RR_{j+1} - RR_j)^2}$
PNN20	Percentage of successive RR intervals that differ by more than 20 ms	$\frac{NN20}{N-1} \times 100\%$
PNN50	Percentage of successive RR intervals that differ by more than 50 ms	$\frac{NN50}{N-1} \times 100\%$



### 3.4 MAX30100 Pulse Oximeter and Heart Rate Sensor Module

MAX30100 pulse oximeter and heart rate sensor is an optical sensor that takes readings from emitting two wavelengths of light from two LEDs (Red (650 nm) and Infrared (IR) (950 nm)) and then measures its absorption in the bloodstream against a photodiode. The MAX30100 sensor works on the principle of reflection mode sensor (See in Figure 2(b)). In the reflection sensor, as the amount of reflection increases when blood density increases, the light intensity falling on photodiode will increase. When the blood is withdrawn from the vessels, the light intensity falling on photodiode will be less since the amount of reflection will decrease. As a result of this change in blood density, the PPG signal is obtained [41]. This particular LED color combination is optimized for reading data from the fingertip. The signal is processed by a low noise analog signal processing unit and transmitted to the target MCU via the mikroBUS I<sup>2</sup>C interface. The MAX30100 sensor and system block diagram of sensor are shown in Figures 3 and 4, respectively.



Figure 3 MAX30100 Pulse Oximeter and Heart Rate Sensor Module [50]

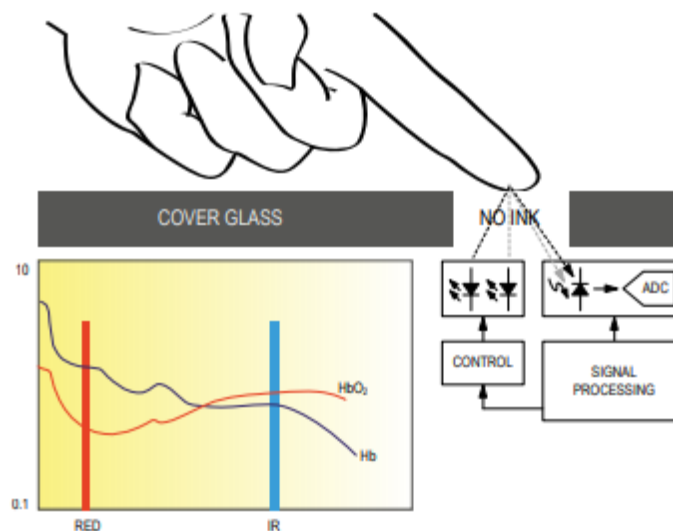


Figure 4 MAX30100 System Block Diagram [51]

In the working principle of MAX30100 sensor, DC and AC components belonging to the PPG signal are obtained through the sensor. In the MAX30100 sensor, a heart rate data sample consists of only one IR data point, so the received PPG signal for heart rate is obtained as a single wavelength signal.



### 3.5 Raspberry Pi 4 Complete Starter Kit

Raspberry Pi is a single board computer. This means that all units required for a computer such as processor, RAM memory, inputs/outputs are gathered on a single circuit board. Thanks to its small design and compact structure, it is possible to use these computers such as in robotic projects, smart home systems, embedded systems, kiosks, and even as desktop computers by connecting peripherals such as keyboard/mouse and screen. In addition to Linux operating systems, Raspberry Pi has the ability to run many ready-made systems specially developed to perform functions such as game machine, media center, network device. Raspberry Pi 4 model used in the study available in Complete Starter Kit is illustrated in Figure 5.



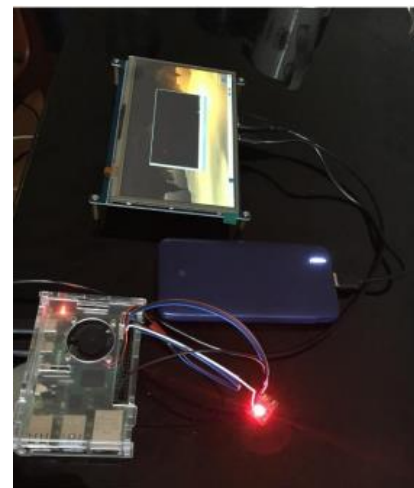
Figure 5. Raspberry Pi 4 Microcomputer [52]

## 4. Results and Discussion

Raspberry Pi 4, WaveShare LCD display, MAX30100 sensor and powerbank connections were established and the system was started. System setup connections are shown in Figure 6(a) and Figure 6(b).



(a)



(b)

Figure 6 System setup connections (a) combining system elements (b) adding MAX30100 sensor to the system Python software version 3.7.3 was installed on Raspberry Pi 4 microcomputer. The necessary libraries were downloaded for the study.

#### 4.1 Data Collection and Analysis

PPG signals were obtained from left index finger by placing the finger horizontally on the sensor from a 24-year-old male volunteer while he was at resting state in stressful and outdoor environments. In a study for HRV analysis, PPG signals were obtained from the subject over a 10-minute period [53]. In this study, 11 minutes of PPG signals were recorded by taking this information as a reference. While the PPG signal is being received, the lack of contact of the finger for 1 minute out of 11, when the finger touches the sensor, motion artifacts can affect the accuracy and sensitivity of the received data. For this reason, HRV detection was performed by processing the signals outside the first 1 minute portion during signal processing. In all cases where the signal is received, the volunteer is in a sitting position.

Two Python code files were written to read and analyze the data. The signals were first read with the `readdata.py` file and saved in separate `outdoordata.csv` and `stressdata.csv` files. Then, this .csv files was analyzed in the analysis code (`analysis.py`) and the HRV time domain attributes were extracted for each case.

#### 4.2 Data Preprocessing

In this study, a finger type heart rate sensor (MAX30100) was used to calculate heart rate. Finger heart rate sensor has one IR transmitter and one IR receiver. It works on the basis of detecting the amount of light sent by the transmitter as it passes through the finger, depending on the pulse value. After detecting the IR beam passing through fingertip, the resulting signal is amplified by two operational amplifiers. A pulse is then generated by a comparator. The pulse size is calculated by reading the produced pulse with microcomputer. While finding the impact length, empty and full parts of the impact are calculated separately and the impact length is found from the sum of the two. There is a transition time from the peak value to the trough value. Then, the number of times this pulse length occurs in a minute is found and the pulse is calculated [54].

Communication between MAX30100 sensor and Raspberry Pi 4 microcomputer is provided by I<sup>2</sup>C communication protocol. According to this communication protocol, the ground line between master and slave must be common, while the SDA (serial data, data line) and SCL (serial clock, clock pulse line used for data synchronization) pins must match. While data synchronization is provided on the SCL line, bidirectional data flow takes place on the SDA line. Digital data is stored in a 16-bit FIFO (first in first out). The I<sup>2</sup>C communication protocol allowed this data to be shared between the two systems [55]. The MAX30100 uses 1.8V and 3.3V power supplies, and the negligible standby current can be turned off via software, ensuring the power supply stays connected at all times. It has programmable high sampling rate and LED current feature [51]. The MAX30100 configuration settings set in the firmware of the Raspberry Pi 4 microcontroller are given in Table 2.

Table 2 MAX30100 Sensor Configuration Settings

Settings	Value
Mode Control	Heart Rate Mode (IR)
Heart Rate sampling rate control	100 sample per second
LED Pulse Width Control	400 $\mu$ s (14 bit)

Filtering processes were performed to strengthen the weak signal data contained in the raw PPG signal, and to remove the noise from the external environment. Bandpass filter codes were used while filtering the signal. The frequency range applied for this filter was chosen as 0.8 Hz - 2.5 Hz. While selecting

these frequency ranges, the highest (150 bpm) and lowest (48 bpm) pulse values were selected and the process was performed. These transactions are;

The lowest frequency value that the filter will pass =  $48(\text{bpm})/60(\text{second}) = 0.8 \text{ Hz}$

The highest frequency value passed by the filter =  $150(\text{bpm})/60(\text{second}) = 2.5 \text{ Hz}$

In stressful environment, PPG signals were obtained from a 24-year-old male volunteer through the MAX30100 sensor. The received PPG signal data is saved in the stressdata.csv file. In Figure 7, the raw PPG signal formed from the recorded data is given.

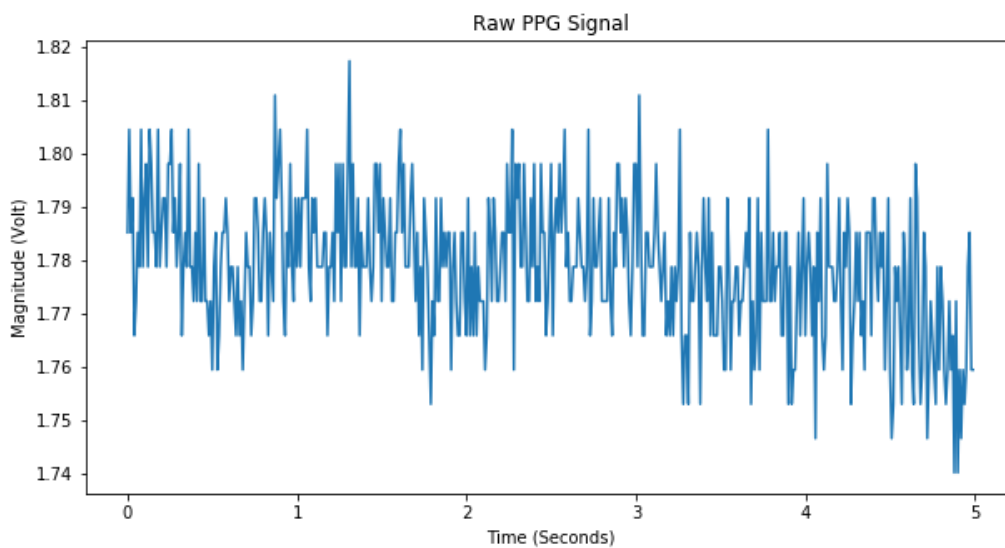


Figure 7 Raw PPG signal from stressdata.csv file

The signal obtained as a result of filtering the received PPG signals to remove noise and motion artifacts is shown in Figure 8.

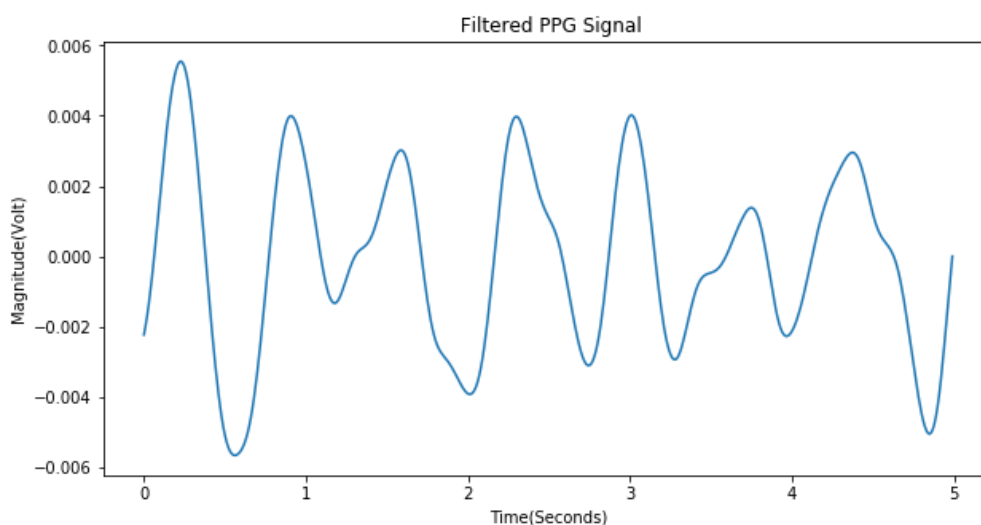


Figure 8 Filtered PPG signal from stressdata.csv file

The peaks and outlier removal of the filtered single-wavelength PPG signal received in the stress state of a 24-year-old male individual are given in Figure 9.

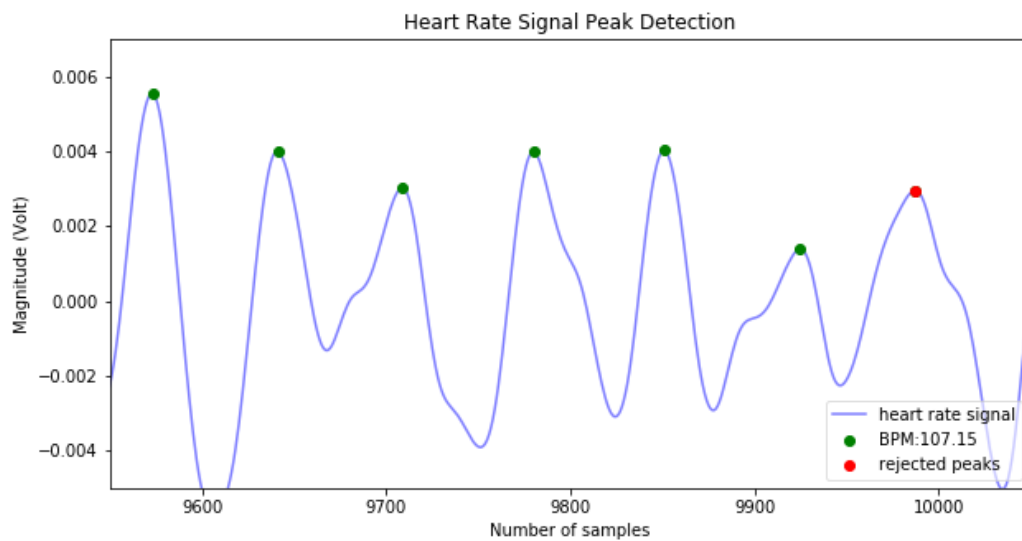


Figure 9 PPG signal peak detection from stressdata.csv file

In outdoor environment, PPG signals were obtained from a 24-year-old male volunteer through the MAX30100 sensor. The received PPG signal data is saved in the outdoordata.csv file. In Figure 10, the raw PPG signal formed from the recorded data is given.

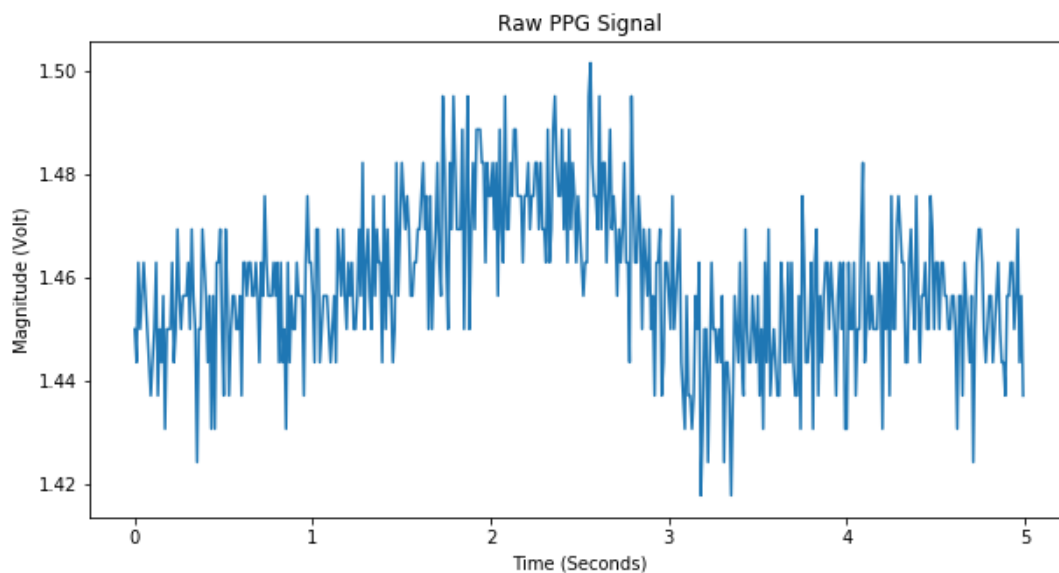


Figure 10 Raw PPG signal from outdoordata.csv file

The signal obtained as a result of filtering the received PPG signals to remove noise and motion artifacts is given in Figure 11.

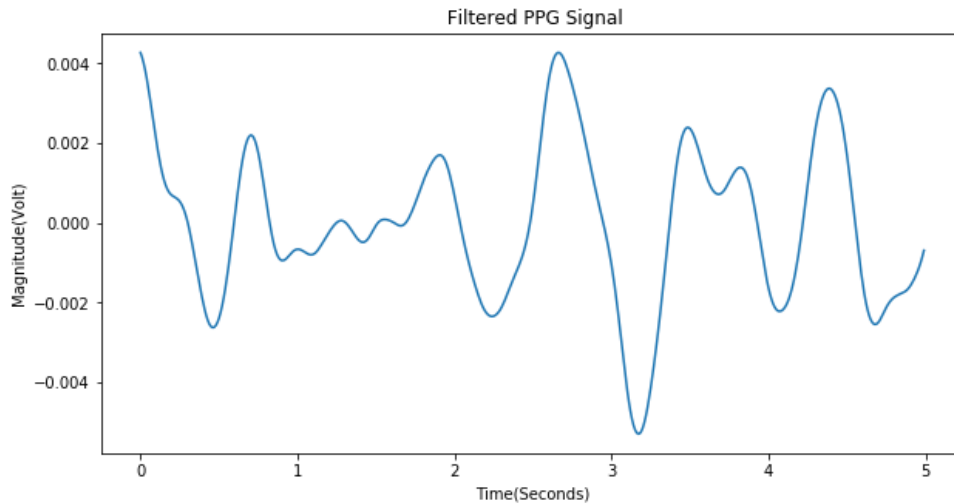


Figure 11 Filtered PPG signal from outdoordata.csv file

The peaks and outlier removal of the filtered single-wavelength PPG signal received in the outdoor state of a 24-year-old male individual are shown in Figure 12.

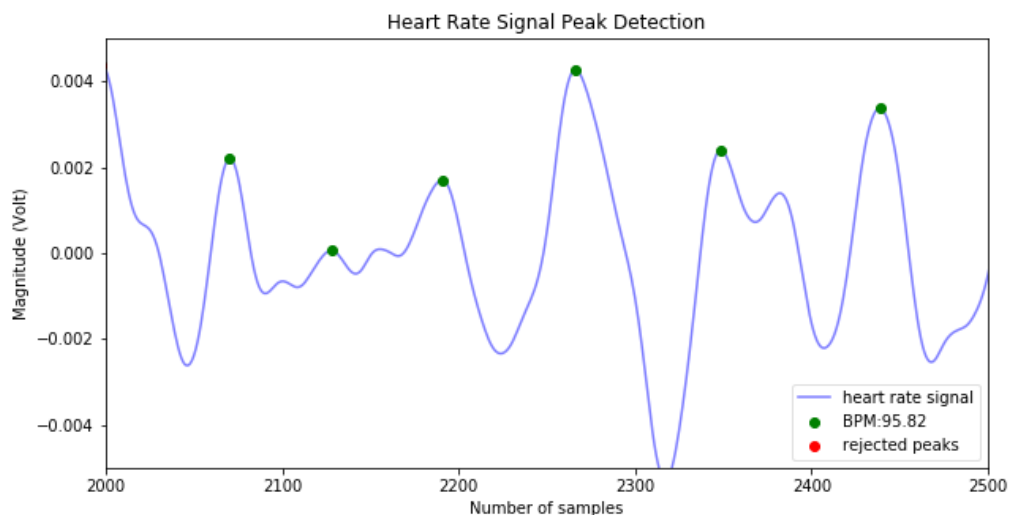


Figure 12 PPG signal peak detection from outdoordata.csv file

### 4.3 Feature Extraction

The heartpy library from Python libraries was used to obtain HRV time domain features (SDNN, SDSD, RMSSD, pNN20 and pNN50) after signal processing. In this study, PPG method was used to determine HRV. This method allowed us to interpret the heart HRV information as a result of filtering the 10-minute portion of the received PPG signals for 11 minutes under different conditions from MAX30100, a non-invasive photodiode heart rate sensor, finding the peaks and obtaining the time domain features.

A decrease in HRV is accepted as an indicator of a decrease in parasympathetic activity and an increase in sympathetic activity [45]. In clinical studies conducted in this context, showing psychosomatic symptoms; HRV has been reported to decrease in individuals with chronic fatigue syndrome, anxiety and depression, and work stress [46-48].

Analysis results of 2 different cases (stress status and outdoor environment) are given in Table 3. The HRV is basically based on SDNN.

Table 3 Time domain features of PPG signal received from a 24-year-old male in two cases

Cases	Pulse (bpm)	SDNN (ms)	SDSD (ms)	RMSSD (ms)	PNN20 (%)	PNN50 (%)
Stress Status (Mental Activity Intense, Worry, Anxiety)	107.15	126.42	37.28	66.20	0.76	0.47
Outdoor (Calm, Relaxed, No Anxiety)	95.82	159.85	44.49	84.41	0.79	0.63

As a result of the study, it was observed that stress and anxiety decreased SDNN, while outdoor environment increased SDNN. These results are consistent with the results in the literature.

## 5. Conclusion

Today, it is obvious that the immune system should be strong when fighting epidemics. In this study, the detection of HRV, which is an indicator of the immune system, was performed non-invasively using the PPG signal. The important point here is to determine which situations reduce HRV actually one of the time domain parameters, SDNN. It is a fact that we should stay away from environments that decrease SDNN, and engage in environments, situations, emotions and mental activities that increase SDNN. In fact, we have scientifically proven a fact known by the society. SDNN decreases in cases of anxiety, depression, being in closed environments, psychosomatic symptoms, chronic fatigue syndrome, and mental thoughts, but being in a relaxed mind in nature has an effect that increases SDNN.

In future studies, it is possible to use MAX30100 and MAX30102 sensors to study more mental activity, emotional states and receiving signals in different environments, and how the environments, emotions and mental activities in human life affect our health and immune system.

## Acknowledgments

This research has been supported by Scientific Research Projects Commission of Zonguldak Bulent Ecevit University, Turkey, under Project Number: 2020-39971044-02. We would also like to thank Abdelgader Obeida, Dilan Pişkin and İhsan Acuz for valuable contribution to the study.

## References

- [1] E. E. Benarroch, "The central autonomic network: functional organization, dysfunction, and perspective," In Mayo Clinic Proceedings, vol. 68, no. 10, pp. 988-1001, 1993.
- [2] J. V. Freeman, F. E. Dewey, D. M. Hadley, J. Myers and V. F. Froelicher, "Autonomic nervous system interaction with the cardiovascular system during exercise," Progress in Cardiovascular Diseases, vol. 48, no. 5, pp. 342-362, 2006.
- [3] J. E. Hall, Guyton ve Hall Tıbbi Fizyoloji. İstanbul, Güneş Tıp Kitabevi, 2017.
- [4] M. K. Lahiri, P. J. Kannankeril and J. J. Goldberger, "Assessment of autonomic function in cardiovascular disease: physiological basis and prognostic implications," Journal of the American college of Cardiology, vol. 51, no. 18, pp. 1725-1733, 2008.
- [5] A. D. Jose and F. Stitt, "Effects of hypoxia and metabolic inhibitors on the intrinsic heart rate and myocardial contractility in dogs," Circulation research, vol. 25, no. 1, pp. 53-66, 1969.
- [6] A. D. Jose, F. Stitt and D. Collison, "The effects of exercise and changes in body temperature on the intrinsic heart rate in man," American heart journal, vol. 79, no. 4, pp. 488-498, 1970.
- [7] E. H. Hon, "Electronic evaluations of the fetal heart rate patterns preceding fetal death, further

- observations," *Am J Obstet Gynecol*, vol. 87, no. 1, pp. 814-826, 1965.
- [8] H. V. Huikuri and P. K. Stein, "Clinical application of heart rate variability after acute myocardial infarction," *Frontiers in physiology*, vol. 3, no. 41, pp. 1-5, 2012.
- [9] C. S. Zuern, P. Barthel and A. Bauer, "Heart rate turbulence as risk-predictor after myocardial infarction," *Frontiers in physiology*, vol. 2, no. 99, pp. 1-8, 2011.
- [10] H. Mølgaard, K. E. Sørensen and P. Bjerregaard, "Attenuated 24-h heart rate variability in apparently healthy subjects, subsequently suffering sudden cardiac death," *Clinical Autonomic Research*, vol. 1, no. 3, pp. 233-237, 1991.
- [11] A. Pozzati, L. G. Pancaldi, G. Di Pasquale, G. Pinelli and R. Bugiardini, "Transient sympathovagal imbalance triggers "ischemic" sudden death in patients undergoing electrocardiographic Holter monitoring," *Journal of the American College of Cardiology*, vol. 27, no. 4, pp. 847-852, 1996.
- [12] M. T. Kearney et al., "Predicting death due to progressive heart failure in patients with mild-to-moderate chronic heart failure," *Journal of the American College of Cardiology*, vol. 40, no. 10, pp. 1801-1808, 2002.
- [13] M. Efeoglu, "Acil Tıp Eğitimi İçin EKG, EKG Kütüphanesi," 2014. [Online]. Available: <https://acilci.net/kategori/tibbi-kategoriler/kardiyoloji/ekg/litfl-ekg-kutuphanesi/>. [Accessed: 14-Oct-2021].
- [14] J. Nolan et al., "Prospective study of heart rate variability and mortality in chronic heart failure: results of the United Kingdom heart failure evaluation and assessment of risk trial (UK-heart)," *Circulation*, vol. 98, no. 15, pp. 1510-1516, 1998.
- [15] S. Guzzetti, R. Magatelli, E. Borroni and S. Mezzetti, "Heart rate variability in chronic heart failure," *Autonomic neuroscience*, vol. 90, no. 1-2, pp. 102-105, 2001.
- [16] B. Xhyheri, O. Manfrini, M. Mazzolini, C. Pizzi and R. Bugiardini, "Heart rate variability today," *Progress in cardiovascular diseases*, vol. 55, no. 3, pp. 321-331, 2012.
- [17] M. H. Asyali, "Discrimination power of long-term heart rate variability measures," *Proc. - 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 200-203, 2003.
- [18] P. Bušek, J. Vaňková, J. Opavský, J. Salinger and S. Nevšimalová, "Spectral analysis of heart rate variability in sleep," *Physiol res*, vol. 54, no. 4, pp. 369-376, 2005.
- [19] M. T. La Rovere et al., "Short-term heart rate variability strongly predicts sudden cardiac death in chronic heart failure patients," *Circulation*, vol. 107, no. 4, pp. 565-570, 2003.
- [20] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993.
- [21] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," *Pattern recognition*, vol. 25, no. 1, pp. 65-77, 1992.
- [22] D. Dumn, "Using a multi-layer perceptron neural for human voice identification," *Proc. - 4th Int. Conf. Signal Process. Applicat. Technol.*, 1993.
- [23] L. Biel, O. Pettersson, L. Philipson and P. Wide, "ECG analysis: a new approach in human identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 3, pp. 808-812, 2001.
- [24] S. Mayya, V. Jilla, V. N. Tiwari, M. M. Nayak and R. Narayanan, "Continuous monitoring of stress on smartphone using heart rate variability," *Proc. - 15th IEEE International Conference on Bioinformatics and Bioengineering*, pp. 1-5, 2015.
- [25] G. Lu, F. Yang, J. A. Taylor and J. F. Stein, "A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects," *Journal of medical engineering & technology*, vol. 33, no. 8, pp. 634-641, 2009.
- [26] N. Selvaraj, A. Jaryal, J. Santhosh, K. K. Deepak and S. Anand, "Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography," *Journal of medical engineering & technology*, vol. 32, no. 6, pp. 479-484, 2008.
- [27] J. Taelman, S. Vandeput, A. Spaepen and S. Van Huffel, "Influence of mental stress on heart rate and heart rate variability," *Proc. - 4th European conference of the international federation for medical and biological engineering*, pp. 1366-1369, 2009.
- [28] W. Wu and J. Lee, "Development of full-featured ECG system for visual stress induced heart rate variability (HRV) assessment," *Proc. - 10th IEEE International Symposium on Signal Processing*

- and Information Technology, pp. 144-149, 2010.
- [29] W. Wu, J. Lee and H. Chen, " Estimation of heart rate variability changes during different visual stimulations using non-invasive continuous ecg monitoring system," Proc. - International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, pp. 344-347, 2009.
- [30] D. McDuff, S. Gontarek and R. Picard, "Remote measurement of cognitive stress via heart rate variability," Proc. - 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2957-2960, 2014.
- [31] C. Wang and F. Wang, "An emotional analysis method based on heart rate variability," Proc. - IEEE-EMBS International Conference on Biomedical and Health Informatics, pp. 104-107, 2012.
- [32] P. B. Laursen, A. Said and B. Martin, "Nocturnal heart rate variability following supramaximal intermittent exercise," International Journal of Sports Physiology and Performance, vol. 4, no. 4, pp. 435-447, 2009.
- [33] P. J. Millar, M. Rakobowchuk, N. McCartney and M. J. Macdonald, " Heart rate variability and nonlinear analysis of heart rate dynamics following single and multiple Wingate bouts," Applied Physiology, Nutrition & Metabolism, vol. 34, no. 5, pp. 875-883, 2009.
- [34] O. F. Barak, D. G. Jakovljevic, J. Z. P. Gacesa, Z. B. Ovcin, D. A. Brodie and N. G. Grujic, " Heart rate variability before and after cycle exercise in relation to different body positions," Journal of sports science & medicine, vol. 9, no. 2, pp. 176-182, 2010.
- [35] U. Wiklund, M. Karlsson, M. Oöström and T. Messner, " Influence of energy drinks and alcohol on post-exercise heart rate recovery and heart rate variability," Clinical physiology and functional imaging, vol. 29, no. 1, pp. 74-80, 2009.
- [36] D. Aras, F. Akça and C. Akalan, "50 metre sprint yüzmenin 13-14 yaşlarındaki erkek yüzücülerde kalp hızı değişkenliğine etkisi," Spormetre Beden Eğitimi ve Spor Bilimleri Dergisi, vol. 11, no. 1, pp. 13-18, 2013.
- [37] R. Ulu, N. Gözel, İ. P. Yiğit, Z. Kemeç, K. A. Uğur, O. Doğdu and A. Doğukan, " Diyaliz Modalitelerinin Kalp Hızı Değişkenliği Üzerine Olan Etkisi," Türk Nefroloji Diyaliz ve Transplantasyon Dergisi, vol. 26, no. 1, pp. 93-97, 2017.
- [38] A. F. Aleixo, E. G. Lima, É. C. Leite, A. V. Inocêncio, L. T. Lins and M. A. Rodrigues, "Wearable Device for Acquisition of SpO<sub>2</sub> and Heart Rate," Proc. - XXVI Brazilian Congress on Biomedical Engineering, pp. 577-582, 2019.
- [39] M. Paul, A. F. Mota, C. H. Antink, V. Blazek and S. Leonhardt, " Modeling photoplethysmographic signals in camera-based perfusion measurements: optoelectronic skin phantom," Biomedical optics express, vol. 10, no. 9, pp. 4353-4368, 2019.
- [40] J. Přibíl, A. Přibílová and I. Frollo, "Comparative Measurement of the PPG Signal on Different Human Body Positions by Sensors Working in Reflection and Transmission Modes," In Engineering Proceedings, vol. 2, no. 69, pp. 1-7, 2020.
- [41] Ö. Yıldırım, Kalp aritmisinin çift dalgalı PPG sinyalleri kullanılarak belirlenmesi, Master Thesis, Dept. of Electrical and Electronics Engineering, Graduate Institute of Natural and Applied Sciences, Karadeniz Teknik University, Trabzon, Turkey, 2017.
- [42] D. S. Goldstein, R. McCarty, R. J. Polinsky and I. J. Kopin, "Relationship between plasma norepinephrine and sympathetic neural activity," Hypertension, vol. 5, no. 4, pp. 552-559, 1983.
- [43] B. G. Wallin and N. Charkoudian, " Sympathetic neural control of integrated cardiovascular function: insights from measurement of human sympathetic nerve activity," Muscle & nerve, vol. 36, no. 5, pp. 595-614, 2007.
- [44] T. B. Kuo and C. C. Yang, "Altered frequency characteristic of central vasomotor control in SHR," American Journal of Physiology-Heart and Circulatory Physiology, vol. 278, no. 1, pp. H201-H207, 2000.
- [45] A. H. Marques, M. N. Silverman and E. M. Sternberg, "Evaluation of stress systems by applying noninvasive methodologies: measurements of neuroimmune biomarkers in the sweat, heart rate variability and salivary cortisol," Neuroimmunomodulation, vol. 17, no. 3, pp. 205-208, 2010.
- [46] R. Freeman and A. L. Komaroff, "Does the chronic fatigue syndrome involve the autonomic nervous system?" The American journal of medicine, vol. 102, no. 4, pp. 357-364, 1997.
- [47] J. W. Hughes and C. M. Stoney, " Depressed mood is related to high-frequency heart rate variability during stressors," Psychosomatic medicine, vol. 62, no. 6, pp. 796- 803, 2000.



- [48] T. Chandola, A. Heraclides and M. Kumari, " Psychophysiological biomarkers of workplace stressors," *Neuroscience & Biobehavioral Reviews*, vol. 35, no. 1, pp. 51-57, 2010.
- [49] S. Açıkgöz and E. Diker, "Kalp hızı değişkenliği," *MN Kardiyoloji*, vol. 3, no. 1, pp. 275-278, 1996.
- [50] "MAX30100 Chip Heart Rate Sensor Module," 2021. [Online]. Available: [https://www.alibaba.com/product-detail/MAX30100-chip-heart-rate-sensor-module\\_1600142413057.html](https://www.alibaba.com/product-detail/MAX30100-chip-heart-rate-sensor-module_1600142413057.html). [Accessed: 24-June-2021].
- [51] "Maxim Integrated MAX30100 Datasheet," 2021. [Online]. Available: <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>. [Accessed: 24-June-2021].
- [52] "Raspberry Pi 4 Model B 4GB RAM Completely Upgraded," 2021. [Online]. Available: [https://www.alibaba.com/product-detail/Raspberry-Pi-4-Model-B-4GB\\_1600081890613.html](https://www.alibaba.com/product-detail/Raspberry-Pi-4-Model-B-4GB_1600081890613.html) [Accessed: 24-June-2021].
- [53] D. Yılmaz and B. Çiğdem B, "Epilepsi Hastalarında Levetirasetam Tedavisinin Otonom Sinir Sistemi Fonksiyonları Üzerine Etkileri," *Epilepsi*, vol. 26, no. 2, pp. 81-87, 2020.
- [54] K. L. Chong, D. Holden and T. Olin, "Heart Rate Monitor," 2010. [Online]. Available: [http://www.academia.edu/7884606/Heart\\_Rate\\_Monitor](http://www.academia.edu/7884606/Heart_Rate_Monitor) [Accessed: 24-June-2021].
- [55] G. Aydın and O. Özhan, "Pulse Oksimetre Tasarım Ve Analizinin Yapılması", *Proc. - 2. Ulusal Biyomedikal Cihaz Tasarımı ve Üretimi Sempozyumu*, pp. 46-48, 2017.

# Classification of Imbalanced Offensive Dataset – Sentence Generation for Minority Class with LSTM

 Ekin Ekinci<sup>1</sup>

<sup>1</sup>Sakarya University of Applied Sciences, Faculty of Technology, Computer Engineering Department; ekinekinci@subu.edu.tr; ORCID: 0000-0003-0658-592X

Received 10 February 2022; Revised 16 April 2022; Accepted 18 April 2022; Published online 30 April 2022

## Abstract

The classification of documents is one of the problems studied since ancient times and still continues to be studied. With social media becoming a part of daily life and its misuse, the importance of text classification has started to increase. This paper investigates the effect of data augmentation with sentence generation on classification performance in an imbalanced dataset. We propose an LSTM based sentence generation method, Term Frequency-Inverse Document Frequency (TF-IDF) and Word2vec and apply Logistic Regression (LR), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Multilayer Perceptron (MLP), Extremely Randomized Trees (Extra tree), Random Forest, eXtreme Gradient Boosting (Xgboost), Adaptive Boosting (AdaBoost) and Bagging. Our experiment results on an imbalanced Offensive Language Identification Dataset (OLID) that machine learning with sentence generation significantly outperforms.

**Keywords:** sentence generation, imbalance classification, offensive language, deep learning, machine learning

## 1. Introduction

Social media is an indispensable part of daily life and the way people communicate with each other is now shaped by this environment. Communication via social media is not only limited to people we know but is also frequently used in communication with people we do not know. This can be accomplished in a variety of ways, such as writing a comment under a photo, answering a question, or commenting on a topic. Although the free environment that social media offers to its users creates a situation where we can express all kinds of ideas in any way as if there are no restrictions, it is absolutely necessary to stay within certain limits. However, nowadays, it is seen that social media is used in the wrong ways that infringe on people's rights. Violation of human rights over social media is usually carried out with the use of offensive language. Offensive language use is a big problem in social media, leading to a large amount of research in detecting content against cyberbullying and hate speech [1]. Therefore, it is very critical to discover offensive language. Over the last ten years, a lot of progress has been done in the field of automatic detection and classification of offensive language and other related phenomena. [2-5]. The determination of whether a language is offensive or not is carried out by supervised methods. When dealing with problems involving imbalanced classification, however, supervised approaches encounter challenges. Imbalanced classification is one of the most important research topics to be handled in machine learning [6, 7]. Because the performance criteria for classification algorithms are designed to reduce classification error, while classifiers perform well on the majority class, they perform poorly on the minority class, which is the focus of attention [8]. To overcome class imbalance, oversampling techniques have an important place in the literature. Oversampling is defined as adding samples to a minority class and carried out in many different ways such as Random Oversampling (ROS), synthetic minority over-sampling technique (SMOTE), BorderLine SMOTE (B-SMOTE), K-Means SMOTE, Safe Level SMOTE (SL-SMOTE), SMOTE-Nominal and Continuous (SMOTE-NC), Adaptive Synthetic Sampling Approach (ADASYN) [9, 10]. ROS equalizes class membership by selecting and multiplying samples from minority classes until their balance is reached [11]. SMOTE is one of the most widely used oversampling techniques. SMOTE realizes oversampling by taking samples of each minority class and creates new synthetic examples by

using the sample and some of its KNN from the minority class [12]. B-SMOTE creates new samples by using minority borderline and other minority samples to determine the new samples [13]. K-Means SMOTE is the hybridization of the k-means clustering and SMOTE [14]. SL-SMOTE generates synthetic examples along line segments as in SMOTE but locates them near the largest safe level [15]. SMOTE-NC is a variant of SMOTE that supports discrete values [16]. ADASYN proposes sample distribution learning efficiently [17]. As a consequence, the aim of this article is to provide a new solution to imbalanced data using a deep learning algorithm to provide an effective solution for classifying offensive language. In the experiments, Offensive Language Identification Dataset (OLID)-sub-task A, which consists of English tweets annotated for offensive language, is used. The goal of this sub-task is to classify offensive language as offensive (OFF) and not-offensive (NOT). The data augmentation is realized by using LSTM, for feature representation two different methods namely TF-IDF and Word2vec are used. Then for both representation classification is realized by using base classifiers which are LR, SVM, KNN, MLP, and ensemble classifiers which are Extra-tree, Random Forest, Xgboost, AdaBoost and Bagging. Experiments are conducted on original and augmented datasets and the best results are achieved with TF-IDF represented augmented dataset by LR and SVM with an average 82% F-score.

The rest of the paper is organized as follows. Section 2 reviews studies in the literature on offensive language classification. In Section 3 pre-processing, feature representation, LSTM and the classification algorithms are explained. In Section 4 proposed data augmentation model is explained. In Section 5 experiments and results are given in detail. In the last section, the paper is concluded.

## 2. Related Works

Here, in chronological order, we provide a brief summary of the studies from literature in which sub-task A of OLID from the shared task SemEval-2019 Task 6 on Identifying and Categorizing Offensive Language in Social Media (OffensEval) was used [18].

Rozental et al. [19] proposed a Multiple Choice Convolutional Neural Network (MC-CNN) which was fed into contextual embedding generated from Twitter and achieved a 0.7868 F-score. In [20] Temporal Convolutional Neural Network with an attention layer was applied and the authors received a 0.4682 F-score. Also, they claimed that preserving as many even numbers of samples for each class as possible could increase classification accuracy when the dataset had a class imbalance ratio. Kumar et al. extracted uni-gram and bi-gram features and then used Linear SVM for classification [21]. The F-score obtained was 0.5282. Wu et al. applied an uncased BERT model pre-trained on model files with a 0.8057 F-score [22]. In [23], multi-layer RNN extracted features -benefited from ELMo embeddings, self-attention, character n-grams and node2vec- were given to gradient-boosted decision trees (GBDT) for classification and a 78.79% F1-score was obtained. Zang et al. proposed a 2-layer residual connected BiDirectional LSTM (BiLSTM) with double attention and their F1-score was 0.768 [24]. This architecture was designed as BiLSTM for contextual feature extraction, residual connected layer for deep feature synthesis and one attention mechanism for the extraction of semantic information of emojis and the other for output. Pavlopoulos et al. compared Perspective which is an API and BERT for offensive language classification and with a 0.7933 F1-Score Perspective had the upper hand over BERT [25]. To cope with class imbalance Modha et al. used pre-trained word vectors and applied LSTM, BiLSTM, CNN, and Stacked CNN [26]. However, they concluded that the proposed solution was not successful in classification according to the 0.7833 F-score achieved. In another study, an ensemble of CNN and RNN was used and low performance with a 0.5925 F-score was realized [27]. Pedersen trained logistic regression, data augmentation and logistic regression and a rule-based black-list approach [28]. Among these three approaches, the rule-based black-list was the best with a 0.73 F-score. Kebriaei et al. proposed machine learning algorithms, deep learning algorithms and the ensemble of these algorithms, and also, realized data augmentation from different sources [29]. The most successful F-score (0.76) was obtained with the application of SVM to expanded data. Pelicon et al. achieved a 0.8078 F-score with BERT [30]. In [31] with machine learning algorithms that were fed into multiple sentence embeddings 64.40% F-score was obtained. Results showed that the deep model dominated the SVM models with a 0.7793 F-score. LSTM classifiers with SVM predictions were used in three different ways

in Bansal et al.'s study [32]. In the first experiment LSTM classifier with SVM, predictions were used directly, in the second and third ones, classification was applied with the help of lookup list and hashtag parsing, respectively. The best F-score (0.7327) was achieved with the hashtag parsing approach. Oberstrass et al. proposed 6 different LSTM models and among these models LSTM trained with chars, words, stems and LSTM outputs gave the best F-score (0.767) [33]. Pătraș et al. aimed at three different models one of them was rule-based, the second one was lexicon-based and the last and best one with a 0.6446 F-score was external sources and offensive words in the training data [34]. Graff et al. applied B4MSA, FastText, and EvoMSA for classification and obtained a 0.774 F-score with EvoMSA [35]. In [36], 6 different architectures were trained and the most successful among them was RNN with an F-score of 0.74. In this RNN architecture, uni-gram and bi-gram features were used and in a fully connected (FC) layer, seven different machine learning models were performed. Pal et al. applied different machine learning and deep learning algorithms [37]. While the best among machine learning algorithms was LR-tri-gram with 0.7231 F-score, among deep learning algorithms the best was CNN-Glove with 0.7844 F-score. Rani and Ojha aimed to learn the impact of n-grams on offensive language detection [38]. It was observed that successful results were obtained (79.76%) with 1 gram from 1-4 grams. In [39], BiLSTM-Att -BiLSTM with attention layer- was proposed to extract semantic features and a 0.7682 F-score was obtained. In another study, BERT achieved state-of-the-art performance with a 0.798 F-score [40]. In Doostmohammadi et al.'s study comparison was made between SVM which took the word and character n-grams and BERT representations as input separately and the character-based deep model [41]. To cope with imbalanced class distribution, Ekinci et al. used oversampling, concept and word-embedding vectors based on expansion [4]. Then, applied weak and ensemble classifiers to the dataset and achieved an 85.66% F-score. Oswal made several experiments based on classification algorithms, feature representation techniques and sampling methods [42]. The best results were obtained with the original dataset by using LSTM-glove and achieved a 0.72 F-score. In [43] the authors devised fBert which was trained on 1.4 million offensive data from the SOLID dataset to deal with the imbalanced class problem. Compared to benchmarks, fBert outperformed with a 0.813 F-score. Muslim et al. improved fine-tuned BERT using a cost-sensitive and ensemble model and obtained an F1 score of 0.8207 [44].

### 3. Methodology and Applied Techniques

#### 3.1 Pre-processing

Data pre-processing is a must, necessitating the use of a technique known as data cleaning, which transforms raw data into a machine-readable format while separating noise from data [45]. In particular, social media data, such as Twitter data, must be subjected to a data cleaning step due to the special and non-ASCII characters, punctuation, numbers, misspellings, URLs, links, and @-mentions, retweets (RT), hashtags and emoticons it contains. To correct misspelled words autoencoder library of Python is used. So, all of these noises are removed from the dataset at hand. Then, other noises such as stopwords namely articles, prepositions and conjunctions are removed. For stopwords removing the stopword list in Python is used. All tweets have been converted to lowercase due to case insensitivity. As a final step, stemming is used to present each word on a stem or root base. Then, to give dataset as input to model some model specific pre-processing is realized. At first, a vocabulary is created which contains distinct words in the dataset. Every distinct word in the vocabulary is tokenized and every token is associated with a specific number. These token-numbers are used to represent sentences. For each sentence, number of token lengths ( $T$ ) minus one word n-grams (2 n-grams, 3 n-grams, ...,  $T$  n-grams) are created and these created n-grams are padded to be of equal size. The last element of this obtained sequence is the label and the remaining elements are the feature. Finally, the labels are encoded with one-hot encoding. To achieve these steps Python's Natural Language Toolkit (NLTK) is used.

### 3.2 Text Representation Models

#### 3.2.1 TF-IDF

The less a term appears in a document, the less informative that term provides and the less weight it is given. TF-IDF algorithm is widely used feature weighting techniques in Natural Language Processing (NLP) tasks. The reason why it is preferred so much is that it is simple and yet it is a very powerful model. The statistical method of multiplying TF and IDF to determine the significance of a word is known as TF-IDF. The formula for TF-IDF is given with Equation 1 below.

$$d_{i,j} = tf_{i,j} \times \log \frac{N}{n_i} \tag{1}$$

The weight of the term  $i$  in the document  $j$  is represented by the equation above  $d_{i,j}$  represents. The frequency of the term  $i$  in document  $j$  is  $tf_{i,j}$  and IDF of the term  $i$  is  $\log N/n_i$ . In the IDF, while  $N$  represents the number of documents,  $n_i$  shows how many documents the term  $i$  occurs in.

#### 3.2.2 Word2vec

Word2vec is one of the word embedding algorithms that maps each word to vectors that represent the semantic meaning of the word [46]. Word2vec, built on artificial neural network (ANN) architecture, is an unsupervised learning algorithm. Word2vec takes a big corpus of texts as input and generates a real-vector space with hundreds of dimensions by exploiting word context information. A vector in space represents each word in the corpus. In this vector space, distances between words can be calculated and semantically similar ones are also close to each other. Thanks to this method, some semantic conclusions can also be drawn for realizing NLP tasks.

Continuous Bag of Words (CBOW) and Skip-gram are two learning models in Word2vec. These two models were built over the n-gram model. While the CBOW model predicts a target word by using context words, the Skip-gram model predicts the context words of a particular word [47]. The input, projection, and output layers are the three layers of the CBOW and skip-gram models. The neural network (NN) architectures of CBOW and Skip-gram models are given in Figure 1.

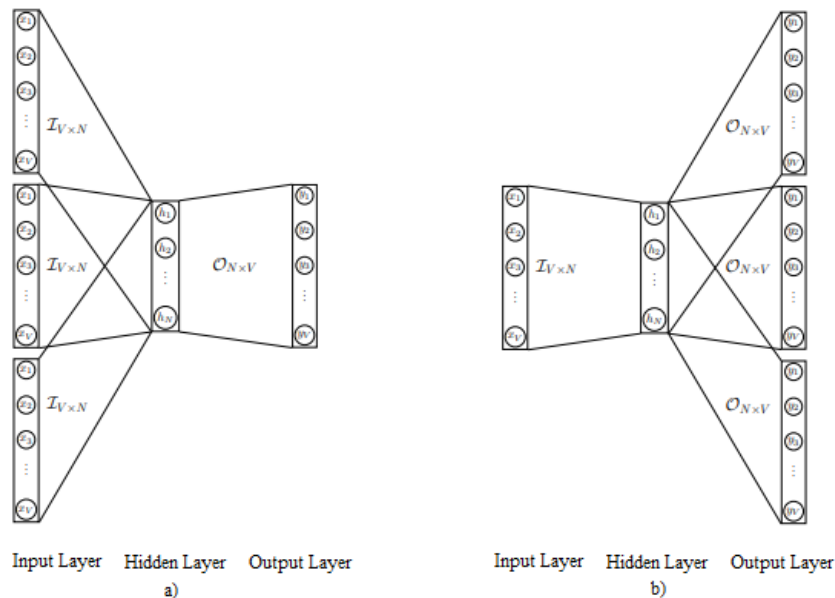


Figure 1 a) NN Architecture of CBOW b) NN Architecture of Skip-gram [48]

In the CBOW model, context words are represented with word vectors of one-hot encoded and each the size of the vocabulary ( $V$ ). The input layer consists of as many neurons as one-hot encoded context

words. The input layer is fully connected with the hidden layer. So connections must be stored in a weight matrix ( $I_{V \times N}$ ).  $N$  is the number of neurons in the hidden layer. The hidden layer and the output layer are also fully connected. And weight matrix between them is represented with  $O_{N \times V}$ . The output layer consists of  $V$  neurons and the output of this layer is the one-hot encoded target word.

In the Skip-gram model, the input layer consists  $V$  neurons which represent the one-hot-encoded word. As in the CBOW model, the input layer and hidden layer and hidden layer and output layer are fully connected in this model.  $I_{V \times N}$  represents the weight matrix between the input and hidden layer. There are  $N$  neurons in the hidden layer. The weight matrix between the hidden layer and output layer is represented with  $O_{N \times V}$ . The output layer consists of as many neurons as one-hot encoded context words each the size of the vocabulary.

Both CBOW and Skip-gram models find the output by using working logic of ANN.

### 3.3 LSTM

RNN is the name given to specialized neural networks for processing sequential data. In traditional ANN, the output is directly obtained from inputs. RNN is devised to overcome this drawback by adding cyclic connections to hidden neurons. In this way, between input and output a sequence-to-sequence mapping is realized, that is, the output is obtained based on the previous computation. However, the insufficient memory capacity of the RNN -single layer neural network with feedback loop- for long time steps requires a new architecture that has strong memorization ability.

LSTM with the repetitive module which has four operations (3 sigmoid and 1 tanh) provides memorization of information for these long time steps [49]. LSTM is a special RNN architecture with its hidden neurons composed of a memory cell and three gates namely the forget gate, input gate and output gate [50, 51].

The forget gate selects which information is taken off from the cell state (memory) based on the previous hidden state ( $h_{t-1}$ ) and the current input ( $x_t$ ). If the forget gate's output is 0, all information is forgotten, if it is 1, all information is taken into account. The mathematical expression of the forget gate is given with Equation 2:

$$f_t = \sigma(w^f x_t + w^f h_{t-1} + b^f) \quad (2)$$

where  $f_t$  is the output of forget gate,  $\sigma$  represents the sigmoid function.  $w^f$  indicates weight and  $b^f$  indicates bias of the forget gate.

The input gate determines which information is updated and added to the cell state. The output of input gate  $i_t$  is given with Equation 3:

$$i_t = \sigma(w^i x_t + w^i h_{t-1} + b^i) \quad (3)$$

where  $w^i$  indicates weight and  $b^i$  indicates bias.

The cell state of the network, and is updated by using Equations 4 and 5:

$$\tilde{C}_t = \tanh(w^c x_t + w^c h_{t-1} + b^c) \quad (4)$$

$$C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \quad (5)$$

where  $\tilde{C}_t$  represents the candidate values for cell state  $C_t$ .  $w^c$  indicates weight and  $b^c$  indicates bias of the cell state. Element-wise multiplication is represented with  $\odot$ .

What information in the cell state will be used is determined with the output gate. The output of this gate is given with Equation 6 and hidden state ( $h_t$ ) for time t Equation 7:

$$o_t = \sigma(w^o x_t + w^o h_{t-1} + b^o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

where  $w^o$  and  $b^o$  are the weight and bias of the output gate, respectively.

### 3.4 Classification Algorithms

#### 3.4.1 Logistic Regression

LR is a statistical binary classification method that uses regression analysis to predict a categorical variable based on one or more predictive variables [52]. LR is built on the principle that the value of the dependent variable is estimated using independent variables. In the model, while the class label  $Y$  is the dependent variable,  $X (x_1, x_2, \dots, x_n)$  is the set of independent variables which corresponds to attributes and used to predict  $Y$ .

#### 3.4.2 Support Vector Machines

SVM is a powerful classification algorithm grounded on statistical learning theory and has the capacity to classify the data that is both linear and nonlinear. The core idea behind this algorithm is to separate the two classes by finding the optimal separating hyperplane, that is, the decision boundary [53]. The SVM uses support vectors and margins to find this hyperplane. If the data is linearly separable, a separation hyperplane can easily separate data in the two-dimensional space. However, to separate nonlinear data converting this data to a higher dimension is necessary. At this stage, SVM uses non-linear mapping functions for this purpose. Training can be slow, but classification accuracy is high with its success in modelling complex and nonlinear decision boundaries.

#### 3.4.3 K Nearest Neighbor

The most important disadvantage of parametric methods is that they need prior knowledge about the distribution of the data. KNN is an easy-to-implement and non-parametric algorithm used when there is fewer or no prior knowledge about data distribution. There is no need to iterate this algorithm for tuning the parameters [54]. To assign a class label to a test instance, at first, the algorithm finds KNNs to that instance in training samples. Then, it assigns the class label that is most common in the  $k$  closest training instances to the test instance. In general, to calculate the distance between instances KNN uses Euclidean distance.

#### 3.4.4 Multilayer Perceptron

MLP is a layered feed-forward ANN trained with a backpropagation algorithm used for separating nonlinear data [55]. The input layer, hidden layer(s), and output layer form the layered structure. In the input layer, the number of neurons is equal to the number of features. The number of hidden layers and neurons in each layer are determined through trial and error. In the output layer, there are as many neurons as there are classes. Nodes in each layer are fully interconnected with nodes in the next layer. All connections have weights. Initially, these weights are randomly determined. After every iteration, the resulting error propagates backwards until the error falls below a certain value or a certain iteration occurs to adjust the weights.

#### 3.4.5 Extremely Randomized Trees

Extra tree is an ensemble learning algorithm that makes classification by combining results of independent decision trees into a forest [56]. Although it is a type of random forest, decision trees are created differently in this method. Each decision tree in extra trees is built using the whole dataset with feature subsets that are assembled at random.

#### 3.4.6 Random Forest

Random forest is an ensemble classifier that employs a large number of unpruned decision trees. Each decision tree that composes the forest is created with selected samples from the training dataset with the bootstrapping technique [57]. A randomly selected subset of features is used to separate the data according to the heterogeneity measure.

### 3.4.7 eXtreme Gradient Boosting

Xgboost is a faster, scalable and effective version of the gradient boosting decision tree (GBDT). The scalability of the Xgboost is so named because it can handle missing values without preprocessing [58], can process weights of instances and can work in parallel and distributed [59]. In Xgboost Classification and Regression Tree (CART) is used.

### 3.4.8 Adaptive Boosting

AdaBoost is widely used ensemble learning algorithms. Adaboost follows the logic of combining weak classifiers to create a strong classifier [60]. In this algorithm, at first, all training samples have the same weight and in the training process reweighting is realized. While correctly classified samples have a lower weight, erroneously classified samples have a higher weight. Thus, misclassified samples will become more important in the next iteration. This process will continue for all iterations and all samples will be reclassified at each iteration. It determines the classes of data samples by majority voting of decisions from weak classifiers applied in series.

### 3.4.9 Bagging

Bagging, which suggests a resampling mechanism is the oldest ensemble classifier [61]. Instead of making decisions according to the classification results obtained on a single dataset, a weak classifier is applied in parallel with the resampled training datasets in Bagging. Then, it determines the classes of data samples by majority voting of decisions from each training model.

## 4. Proposed Data Augmentation Method

In the data augmentation process, LSTM architecture is used. The input layer is designed as an embedding layer that takes the tokens. In this layer input dimension is the number of total words in OFF labelled tweets. The output dimension, which defines the size of the output embedded vectors for each token, is 10. Input dimension is equal to maximum sequence length minus one. We define only one hidden LSTM layer with 100 memory units. To prevent overfitting, the proposed network uses dropout with a probability of 10. The output layer is designed as a Dense layer with a softmax activation function and as an output, determines the probability of the best probable next word. And the number of iterations is equal to 100. The architecture of the model is After the model has been trained, the tokenized sequence given as input to the model. The architecture is given in Figure 2.

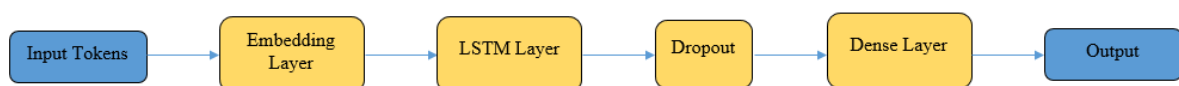


Figure 2 The Proposed Architecture

After each iteration, a new word is predicted as output of the model depending on the context. This output is added to previous input and combined input is given to model as a next input. If the input of the architecture is like “what the hell be”, one of the outputs of the architecture is “what the hell be like one see moron get drop”. For data augmentation, the Keras library of Python is used. In the experiments, the used model parameters are selected by using a grid search over multiple initialization seeds. From input to output, the generation is represented in Figure 3.



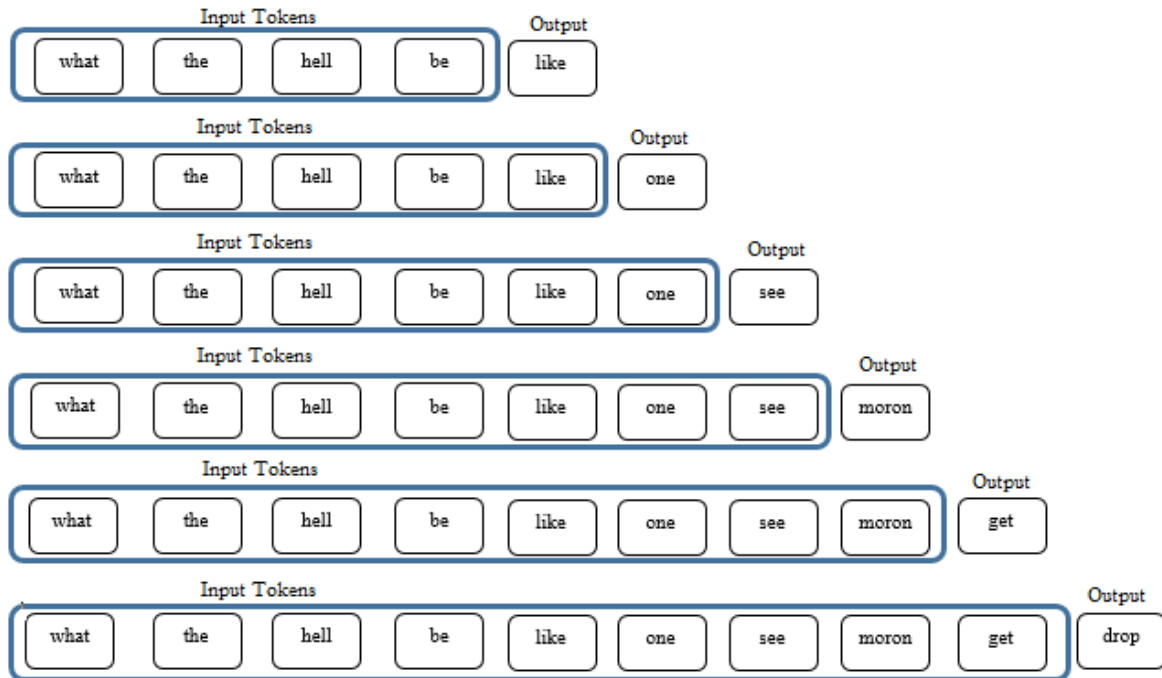


Figure 3 Sentence Generation

## 5. Experimental Study

### 5.1 Dataset

In the experiments Offensive Language Identification Dataset (OLID)- sub-task A is used. There are 13240 tweets in the dataset, with 8840 being labelled as not-offensive (NOT) and 4400 as offensive (OFF). The statistics of the imbalanced dataset are given in Table 1. As it is understood, the available dataset is an imbalanced dataset and an LSTM-based oversampling is applied to OFF labelled data within the scope of the study to increase the classification accuracy. With the application of LSTM based oversampling, the number of OFF tweets is 8840 and the total data is 17680.

Table 1 Statistics of the Imbalanced Dataset

		# of Term	# of Average Terms Per Tweet	Average Term Length
Label	NOT	160.719	18.18	4.26
	OFF	91.504	20.8	4.24

### 5.2 Performance Metrics

In order to assess the proposed models, we preferred F-score as the main evaluation metric. F-score is the harmonic mean of precision and recall. Precision is (p) the ratio of correctly classified not offensive tweets (tp) to all not offensive tweets (tp+fp). Recall (r) is the ratio of correctly classified not offensive tweets (tp) to all tweets that are classified as not offensive (tp+fn). The confusion matrix to calculate performance metrics is given in Table 1. The formula of the F-score is given with Equation 9.

Table 2 Confusion Matrix

		Actual	
		NOT	OFF
Predicted	NOT	tp	fp
	OFF	fn	tn

$$F - Score = \frac{2 \times p \times r}{p + r} \quad (9)$$

### 5.3 Realization of Experiments and Performance Results

In this study, we apply basic and ensemble classifiers to classify offensive tweets. For the implementation of classification algorithms scikit-learn library of Python is used. We divide the original and augmented datasets into train and test sets. As a train set, we take 80% of the dataset, as a test set, we take 20% of the dataset. While, in the training and test sets of the original dataset there are 10592 and 2648 samples, respectively, in the training and test sets of the augmented dataset there are 14144 and 3536 samples, respectively.

In the experiments, the l2-regularized LR algorithm with settings inverse regularization parameter  $C = 10$  and the liblinear optimization algorithm are used. The reason why we set the 10 to  $C$  parameter is that the smaller  $C$  provides better regularization. In the SVM, regularization parameter  $c = 1$  with a squared L2 penalty and RBF kernel are used. For the KNN algorithm, the selection of the  $k$  is important. So, to make a decision on  $k$  we use the Elbow method and see that the best value is 2 for both datasets. All points in each neighbourhood have equal weight and the Euclidean metric is preferred as a distance measure. While designing MLP 100 hidden layers are used, the activation function is relu and the solver is adam. 0.001 is assigned to the learning rate and 300 is assigned to the maximum iteration count.

The number of base classifiers is set to 100 when creating an extra tree classifier. As a split criteria Gini index is used. Minimum impurity decrease is defined as  $10^{-3}$ . The number of base trees in the Random forest is assigned to 100. As in Extra-tree, Gini is used as the split criterion for this algorithm. The tree's maximum depth has been set to 2. The number of base classifiers is set to 100 in Xgboost, as in other extra tree and random forest. The maximum depth of the tree is selected as 6 and the learning rate is selected as 0.1. In AdaBoost, 50 base estimators are used. In Bagging, the number of base estimators is 10 and bags are composed by replacement of samples.

The results for the original dataset are given in Table 3. When the results are examined for the original dataset it is clearly seen that TF-IDF representation is better than Word2vec representation in most cases in terms of performance achieved. Among the algorithms, while LR is the best for TF-IDF representation, KNN, Extra tree and Bagging are the best for Word2vec representation.

Table 3 Classification Results for Original Dataset (F-score)

Classifiers	Text Representations	
	TF-IDF	Word2vec
LR	0.76	0.53
SVM	0.73	0.52
KNN	0.56	0.58
MLP	0.71	0.52
Extra tree	0.76	0.58
Random Forest	0.52	0.52
Xgboost	0.72	0.53
AdaBoost	0.75	0.53
Bagging	0.74	0.58

The results for the augmented dataset are given in Table 4. As seen in the original dataset, TF-IDF representation performed better than Word2vec representation in most cases in augmented data. In addition, it is clearly seen that success has increased in the results obtained with both representations with data augmentation. Among the algorithms, while LR and SVM are the best for TF-IDF representation, MLP and Xgboost are the best for Word2vec representation.

Table 4 Classification Results for Augmented Dataset (F-score)

Classifiers	Text Representations	
	TF-IDF	Word2vec
LR	0.82	0.75
SVM	0.82	0.75
KNN	0.75	0.75
MLP	0.76	0.77
Extra tree	0.81	0.76
Random Forest	0.71	0.74
Xgboost	0.80	0.77
AdaBoost	0.80	0.75
Bagging	0.81	0.75

## 6. Conclusions

In this paper, we investigate the problem of how to improve classification performance on the imbalanced dataset. To achieve this, we conduct a comparative study of imbalanced data classification using LSTM based sentence generation method. For text representation, TF-IDF and Word2vec are used and LR, SVM, KNN, MLP, and ensemble classifiers which are Extra-tree, Random Forest, Xgboost, AdaBoost and Bagging algorithms are used to train the classifiers. Experimental results on the augmented dataset reveal that the TF-IDF based LR and SVM increase the F-score by 6% and 9%, respectively. We also investigate that classification with TF-IDF-based text representation often performs well on both original and augmented datasets compared to classification with Word2vec.

In the future, we will apply different sentence generation methods with different text representation methods to improve performance.

## References

- [1] S. Rosenthal, P. Atanasova, G. Karadzhov, M. Zampieri, and P. Nakov, "OLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification," *arXiv preprint arXiv:2004.14454*, 2020.
- [2] G. Wiedemann, E. Ruppert, R. Jindal and C. Biemann, "Transfer Learning from LDA to BiLSTM-CNN for Offensive Language Detection in Twitter," *arXiv preprint arXiv:1811.02906v1*, 2018.
- [3] H. Mubarak and K. Darwish K., "Arabic Offensive Language Classification on Twitter," *Lecture Notes in Computer Science*. Springer, Cham, 2019.
- [4] E. Ekinci, S. İlhan Omurca and S. Sevim, "Improve Offensive Language Detection with Ensemble Classifiers," *IJISAE*, vol. 8, no. 2, pp. 109–115, 2020.
- [5] M. Djandji, F. Baly, W. Antoun and H. Hajj, "Multi-Task Learning using AraBert for Offensive Language Detection," *Proc. - 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 97–101, 2020.
- [6] Y. Tung and Y. Q. Zhang, "Granular SVM with Repetitive Undersampling for Highly Imbalanced Protein Homology Prediction," *Proc. - 2006 IEEE International Conference on Granular Computing*, pp. 457–460, 2006.
- [7] J. Brownlee, *Imbalanced Classification with Python*. Machine Learning Mastery, 2020.
- [8] Q. Zou, S. Xie, Z. Lin, M. Wu and Y. Ju, "Imbalanced classification is one of most popular topics in the field of machine learning," *Big Data Res.*, vol. 5, pp. 2–8, 2016.
- [9] L. Wang, H. Cheng, Z. Zheng, A. Yang and X. Zhu, "Ponzi scheme detection via oversampling-based Long Short-Term Memory for smart contracts," *Knowl Based Syst.*, vol. 228, pp.1–12, 2021.
- [10] A. Gosain and S. Sardana, "Handling Class Imbalance Problem using Oversampling Techniques: A Review," *Proc. - 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 79–85, 2017.
- [11] E. L. Iglesias, A. S. Vieira and L. Borrajo, "An HMM-based over-sampling technique to improve text classification," *Expert Syst. Appl.*, 465, pp. 1–20, 2013.
- [12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-

- sampling Technique," *J Artif Intell Res.*, vol. 16, pp. 321–357, 2002.
- [13] H. A. Majzoub, I. Elgedawy, Ö. Akaydin and M. Köse Ulukök, "HCAB-SMOTE: A Hybrid Clustered Afnitive Borderline SMOTE Approach for Imbalanced Data Binary Classification," *Arab. J. Sci. Eng.*, vol. 45, pp. 3205–3222, 2020.
- [14] G. Douzas, F. Bacao and F. Last, "Improving imbalanced learning through a heuristic over-sampling method based on k-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, 2018.
- [15] C. Bunkhumpornpat, K. Sinapiromsaran and C. Lursinsap, "Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem," *Proc. - Pacific-Asia conference on knowledge discovery and data mining*, pp. 475–482, 2009.
- [16] S. Darabi and Y. Elor, "AE-SMOTE: A Multi-Modal Minority Oversampling Framework," pp. 1–19, 2020.
- [17] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah and A. Hussain, "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.
- [18] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra and R. Kumar, "Predicting the Type and Target of Offensive Posts in Social Media," *Proc. - NAACL-HLT*, pp. 1415–1420, 2019.
- [19] A. Rozental and D. Biton, "Amobee at SemEval-2019 Tasks 5 and 6: Multiple choice over contextual embedding," *arXiv preprint arXiv:1904.08292.*, 2019.
- [20] M. Sridharan and T. R. Swapna, "Amrita School of Engineering-CSE at SemEval-2019 Task 6: Manipulating attention with temporal convolutional neural network for offense identification and classification," *Proc. - 13th International Workshop on SemEval*, pp. 540–546, 2019.
- [21] R. Kumar, G. Bhanodai, R. Pamula, and M. R. Chennuru, "bhanodaig at SemEval-2019 Task 6: Categorizing offensive language in social media," *Proc. - 13th International Workshop on SemEval*, pp. 547–550, 2019.
- [22] Z. Wu, H. Zheng, J. Wang, W. Su and J. Fong, "Bnu-hkbu uic nlp team 2 at semeval-2019 task 6: Detecting offensive language using bert model," *Proc. - 13th International Workshop on SemEval*, pp. 551–555, 2019.
- [23] G. Aglionby, C. Davis, P. Mishra, A. Caines, H. Yannakoudakis, M. Rei, E. Shutova and P. Buttery, "CAMsterdam at SemEval-2019 Task 6: Neural and graph-based feature extraction for the identification of offensive tweets," *Proc. - 13th International Workshop on SemEval*, pp. 556–563, 2019.
- [24] Y. Zhang, B. Xu and T. Zhao, "CN-HIT-MI. T at SemEval-2019 Task 6: Offensive Language Identification Based on BiLSTM with Double Attention," *Proc. - 13th International Workshop on SemEval*, pp. 564–570, 2019.
- [25] J. Pavlopoulos, N. Thain, L. Dixon and I. Androutsopoulos, "Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert," *Proc. - 13th International Workshop on SemEval*, pp. 571–576, 2019.
- [26] S. Modha, P. Majumder, D. Patel, "DA-LD-Hildesheim at SemEval-2019 task 6: tracking offensive content with deep learning using shallow representation," *Proc. - 13th International Workshop on SemEval*, pp. 577–581, 2019.
- [27] G. L. De la Peña and P. Rosso, "DeepAnalyzer at SemEval-2019 Task 6: A deep learning-based ensemble method for identifying offensive tweets," *Proc. - 13th International Workshop on SemEval*, pp. 582–586, 2019.
- [28] T. Pedersen, "Duluth at SemEval-2019 task 6: Lexical approaches to identify and categorize offensive tweets," *arXiv preprint arXiv:2007.12949*, 2019.
- [29] E. Kebriaei, S. Karimi, N. Sabri and A. Shakery, "Emad at SemEval-2019 task 6: offensive language identification using traditional machine learning and deep learning approaches," *Proc. - 13th International Workshop on SemEval*, pp. 600–603, 2019.
- [30] A. Pelicon, M. Martinc and P. K. Novak, "Embeddia at semeval-2019 task 6: Detecting hate with neural network and transfer learning approaches," *Proc. - 13th International Workshop on SemEval*, pp. 604–610, 2019.
- [31] V. Indurthi, B. Syed, M. Shrivastava, M. Gupta and V. Varma, "Fermi at SemEval-2019 Task 6: Identifying and categorizing offensive language in social media using sentence embeddings," *Proc. - 13th International Workshop on SemEval*, pp. 611–616, 2019.

- [32] H. Bansal, D. Nagel and A. Soloveva, "HAD-Tübingen at SemEval-2019 Task 6: Deep learning analysis of offensive language on Twitter: Identification and categorization," *Proc. - 13th International Workshop on SemEval*, pp. 622–627, 2019.
- [33] A. Oberstrass, J. Romberg, A. Stoll and S. Conrad, "HHU at SemEval-2019 Task 6: Context does matter-tackling offensive language identification and categorization with ELMo," *Proc. - 13th International Workshop on SemEval*, pp. 628–634, 2019.
- [34] G. F. Patras, D. F. Lungu, D. Gifu and D. Trandabat, "Hope at SemEval-2019 Task 6: Mining social media language to discover offensive language," *Proc. - 13th International Workshop on SemEval*, pp. 635–638, 2019.
- [35] M. Graff, S. Miranda-Jiménez, E. Tellez and D. A. Ochoa, "INGEOTEC at SemEval-2019 task 5 and task 6: A genetic programming approach for text classification," *Proc. - 13th International Workshop on SemEval*, pp. 639–644, 2019.
- [36] Y. HaCohen-Kerner, Z. Ben-David, G. Didi, E. Cahn, S. Rochman and E. Shayovitz, "JCTICOL at SemEval-2019 Task 6: Classifying offensive language in social media using deep learning methods, word/character n-gram features, and preprocessing methods," *Proc. - 13th International Workshop on SemEval*, pp. 645–651, 2019.
- [37] P. Mukherjee, M. Pal, S. Banerjee and S. K. Naskar, "JU\_ETCE\_17\_21 at SemEval-2019 Task 6: Efficient Machine Learning and Neural Network Approaches for Identifying and Categorizing Offensive Language in Tweets," *Proc. - 13th International Workshop on SemEval*, pp. 662–667, 2019.
- [38] P. Rani and A. K. Ojha, "KMI-coling at SemEval-2019 task 6: exploring N-grams for offensive language detection," *Proc. - 13th International Workshop on SemEval*, pp. 668–671, 2019.
- [39] L. S. M. Altın, A. B. Serrano and H. Saggion, "Lastus/taln at semeval-2019 task 6: Identification and categorization of offensive language in social media with attention-based bi-lstm model," *Proc. - 13th International Workshop on SemEval*, pp. 672–677, 2019.
- [40] P. Aggarwal, T. Horsmann, M. Wojatzki and T. Zesch, "LTL-UDE at SemEval-2019 Task 6: BERT and two-vote classification for categorizing offensiveness," *Proc. - 13th International Workshop on SemEval*, pp. 678–682, 2019.
- [41] E. Doostmohammadi, H. Sameti and A. Saffar, "Ghmerti at SemEval-2019 task 6: a deep word- and character-based approach to offensive language identification," *arXiv preprint arXiv:2009.10792*, 2020.
- [42] N. Oswal, "SemEval-2019 (OffensEval): Identifying and Categorizing Offensive Language in Social Media," *arXiv preprint arXiv: 2104.04871v1*, 2021.
- [43] D. Sarkar, M. Zampieri, T. Ranasinghe and A. Orarbia, "fBERT: A Neural Transformer for Identifying Offensive Content," *arXiv preprint arXiv: 2109.05074v1*, 2021.
- [44] F. Muslim, A. Purwarianti and F. Z. Ruskanda, "Cost-Sensitive Learning and Ensemble BERT for Identifying and Categorizing Offensive Language in Social Media," *Proc. - ICAICTA*, pp. 1–6, 2021.
- [45] A. S. Neogi, K. A. Garg, R. K. Mishra and Y. K. Dwivedi, "Sentiment analysis and classification of Indian farmers' protest using twitter data," *Int. J. Inf. Manage.*, vol. 1, no. 2, pp. 100019, 2021.
- [46] E. M. Dharma, F. L. Gaol, H. L. H. S. Warnars and B. Soewito, "The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (CNN) Text Classification," *J. Theor. Appl. Inf.*, vol. 100, no. 2, pp. 349–359, 2022.
- [47] M. S. Başarslan and F. Kayaalp, "Sentiment Analysis on Social Media Reviews Datasets with Deep Learning Approach," *SAUCIS*, vol. 4, no. 1, pp. 35–49, 2021.
- [48] J. V. Lochter, P. R. Pires, C. Bossolani, A. Yamakami and T. A. Almeida, "Evaluating the impact of corpora used to train distributed text representation models for noisy and short texts," *Proc. - 2018 International Joint Conference on Neural Networks*, pp. 1–8, 2018.
- [49] A. Zhao, L. Qi, J. Dong and H. Yu, "Dual channel LSTM based multi-feature extraction in gait for diagnosis of Neurodegenerative diseases," *Knowl. Based Syst.*, vol. 145, pp. 91–97, 2018.
- [50] B. Kaya and A. Günay, "Twitter Sentiment Analysis Based on Daily Covid-19 Table in Turkey," *SAUCIS*, vol. 4, no. 3, pp. 302–311, 2021.
- [51] Y. Canbay, A. İsmetoğlu and P. Canbay, "Deep Learning and Data Privacy in Diagnosis of Covid-19," *J. Eng. Sci. Technol.*, vol. 9, no. 2, pp. 701–715, 2021.

- [52] E. Ekinci, S. İlhan Omurca and N. Acun, "A Comparative Study on Machine Learning Techniques using Titanic Dataset," *Proc. - 7th International Conference on Advanced Technologies*, pp. 411–416, 2018.
- [53] D. Chen, H. Bourlard and J. P. Thiran, "Text identification in complex background using SVM," *Proc. - 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 621–626, 2001.
- [54] M. Jogin, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," *Proc. - 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, pp. 2319–2323, 2018.
- [55] S. Lallahem, J. Mania, A. Hani and Y. Najjar, "On the use of neural networks to evaluate groundwater levels in fractured media," *J. Hydrol.*, vol. 307, no. 1-4, pp. 92–111, 2005.
- [56] K. Kaur and S. K. Mittal, "Classification of mammography image with CNN-RNN based semantic features and extra tree classifier approach using LSTM," *Mater. Today.*, pp. 1–7, 2020.
- [57] S. Sevim, E. Ekinci and S. İlhan Omurca, "Multi-view Document Classification with Co-training," *Proc. - 28th IEEE Conference on Signal Processing and Communications Applications*, pp. 1–4, 2020.
- [58] D. A. Rusdah and H. Murfi, "XGBoost in handling missing values for life insurance risk prediction," *SN Appl. Sci.*, vol. 2, no. 8, pp. 1–10, 2020.
- [59] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *arXiv preprint arXiv: 1603.02754v3*.
- [60] E. Ekinci and H. Takçı, "Comparing ensemble classifiers: Forensic analysis of electronic mails," *Global Journal on Technology*, vol. 4, no. 2, pp. 167–173, 2013.
- [61] G. Liang, X. Zhu, and C. Zhang, "An empirical study of bagging predictors for different learning algorithms," *Proc. - AAAI'11*, pp. 1802–1803, 2011.