

**Sakarya University**

# **Journal of Computer and Information Sciences**

e-ISSN 2636-8129

**VOLUME 6 ISSUE 3**

**DECEMBER 2023**





# SAUCIS

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

ISSN 2636-8129

**December 2023**

**Volume 6 Issue 3**

## Editor-in-Chief

Ahmet ZENGİN, Sakarya University, Türkiye, [azengin@sakarya.edu.tr](mailto:azengin@sakarya.edu.tr)

## Associate Editors

Hessam SARJOUGHIAN, Arizona State University, USA, [hessam.sarjoughian@asu.edu](mailto:hessam.sarjoughian@asu.edu)

Muhammed Fatih ADAK, Sakarya University, Türkiye, [fatihadak@sakarya.edu.tr](mailto:fatihadak@sakarya.edu.tr)

Muhammed KOTAN, Sakarya University, Türkiye, [mkotan@sakarya.edu.tr](mailto:mkotan@sakarya.edu.tr)

Mustafa AKPINAR, Higher Collages of Technology, United Arab Emirates, [mustafaa@hct.ac.ae](mailto:mustafaa@hct.ac.ae)

Unal CAVUSOGLU, Sakarya University, Türkiye, [unalc@sakarya.edu.tr](mailto:unalc@sakarya.edu.tr)

A F M Suaib AKHTER, Sakarya Applied Science University, Türkiye, [suaibakhter@subu.edu.tr](mailto:suaibakhter@subu.edu.tr)

Selman HIZAL, Sakarya Applied Science University, Türkiye, [selmanhizal@subu.edu.tr](mailto:selmanhizal@subu.edu.tr)

## Editorial Assistants – Secretary

Deniz BALTA, Sakarya University, Türkiye, [ddural@sakarya.edu.tr](mailto:ddural@sakarya.edu.tr)

Gozde Yolcu OZTEL, Sakarya University, Türkiye, [gyolcu@sakarya.edu.tr](mailto:gyolcu@sakarya.edu.tr)

Ibrahim DELIBASOGLU, Sakarya University, Türkiye, [ibrahimdelibasoglu@sakarya.edu.tr](mailto:ibrahimdelibasoglu@sakarya.edu.tr)

Sumeyye KAYNAK, Sakarya University, Türkiye, [sumeyye@sakarya.edu.tr](mailto:sumeyye@sakarya.edu.tr)

Fatma AKALIN, Sakarya University, Türkiye, [fatmaakalin@sakarya.edu.tr](mailto:fatmaakalin@sakarya.edu.tr)

Nur Yasin PEKER, Sakarya Applied Science University, Türkiye, [yasinpeker@subu.edu.tr](mailto:yasinpeker@subu.edu.tr)

## Editorial Board

Aref YELGHI, Istanbul Ayvansaray University, Türkiye, [ar.yelqi@gmail.com](mailto:ar.yelqi@gmail.com)

Ayhan ISTANBULLU, Balikesir University, Türkiye, [iayhan@balikesir.edu.tr](mailto:iayhan@balikesir.edu.tr)

Bahadir KARASULU, Canakkale Onsekiz Mart University, Türkiye, [bahadirkarasulu@comu.edu.tr](mailto:bahadirkarasulu@comu.edu.tr)



# SAUCIS

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES

ISSN 2636-8129

**December 2023**

**Volume 6 Issue 3**

## Editorial Board (Cont)

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

Ibrahim TURKOGLU, Firat University, Türkiye, iturkoglu@firat.edu.tr

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

Nuri YILMAZER, Texas A&M University, USA, nuri.yilmazer@tamuk.edu

Nejat YUMUŞAK, Sakarya University, Türkiye, nyumusak@sakarya.edu.tr

Okan ERKAYMAZ, National Defense University, Naval Academy, Türkiye, oerkaymaz@dho.edu.tr

Ömer Hulusi DEDE, Sakarya Applied Science University, Türkiye, ohdede@subu.edu.tr

Priyadip RAY, Lawrence Livermore National Laboratory, USA, priyadipr@gmail.com

Resul DAS, Firat University, Türkiye, rdas@firat.edu.tr

## Language Editor




A F M Suaib AKHTER, Sakarya Applied Science University, Türkiye, suaibakhter@subu.edu.tr

## CONTENTS

No	Author(s), Paper Title	Pages
1	Remzi GÜRFİDAN, Şerafettin ATMACA, Tuncay YİĞİT, “Real-Time Intelligent Anomaly Detection and Prevention System” (RESEARCH ARTICLE)	160-171
2	Pakize ERDOĞMUŞ, Abdullah Talha KABAKUŞ, Enver KÜÇÜKKÜLAHLI, Büşra TAKGİL, Ezgi KARA TİMUÇİN, “A Novel Gender Classification Model based on Convolutional Neural Network through Handwritten Text and Numeral” (RESEARCH ARTICLE)	172-188
3	İbrahim ÖZ, Cevat ÖZARPA, “Conjoint Analysis of GPS Based Orbit Determination among Traditional Methods” (RESEARCH ARTICLE)	189-197
4	Rifat Volkan ŞENYUVA, “Estimation of Uplink Channels for Multiple Users Using Tensor Modeling in RIS-Aided MISO Communication” (RESEARCH ARTICLE)	198-207
5	Kenan BAYSAL, Deniz TAŞKIN, “High-Capacity Data Processing with FPGA-Based Multiplication Algorithms and the Design of a High-Speed LUT Multiplier” (RESEARCH ARTICLE)	208-217
6	Aamo IORLIAM, “A Novel Additive Internet of Things (IoT) Features and Convolutional Neural Network for Classification and Source Identification of IoT Devices” (RESEARCH ARTICLE)	218-225
7	Erkan AKKUR, “Prediction of Cardiovascular Disease Based on Voting Ensemble Model and SHAP Analysis” (RESEARCH ARTICLE)	226-238
8	Ahmed ABBAS, Nebras IBRAHİM, Farah KHORSHEED. “A Systematic Review for Misuses Attack Detection based on Data Mining in NFV” (RESEARCH ARTICLE)	239-252
9	İnci UMAKOĞLU, Mustafa NAMDAR, Arif BAŞGÜMÜŞ, “Performance Evaluation of OTFS-NOMA Scheme for High Mobility Users” (RESEARCH ARTICLE)	253-260



# Real-Time Intelligent Anomaly Detection and Prevention System

Remzi Gürfidan<sup>1</sup> , Şerafettin Atmaca<sup>2</sup> , Tuncay Yiğit<sup>3</sup> 

<sup>1</sup> Isparta University of Applied Science, Yalvac Technical Sciences Vocational School, Isparta, Türkiye

<sup>2</sup> Isparta University of Applied Sciences, Isparta, Türkiye

<sup>3</sup> Süleyman Demirel University, Computer Engineering Department, Isparta, Türkiye



## Corresponding author:

Remzi Gürfidan, Isparta University of Applied Science, Yalvac Technical Sciences Vocational School, Isparta, Türkiye

## E-mail address:

[remzigurfidan@isparta.edu.tr](mailto:remzigurfidan@isparta.edu.tr)

**Submitted:** 12 May 2023

**Revision Requested:** 06 September 2023

**Last Revision Received:** 21 September 2023

**Accepted:** 24 September 2023

**Published Online:** 28 September 2023

**Citation:** Gürfidan R. Atmaca Ş. Yiğit T.(2023). Real-Time Intelligent Anomaly Detection and Prevention System. *Sakarya University Journal of Computer and Information Sciences*. 6 (3) <https://doi.org/10.35377/saucis...1296210>

## ABSTRACT

Real-time anomaly detection in network traffic is a method that detects unexpected and anomalous behavior by identifying normal behavior and statistical patterns in network traffic data. This method is used to detect potential attacks or other anomalous conditions in network traffic. Real-time anomaly detection uses different algorithms to detect abnormal activities in network traffic. These include statistical methods, machine learning, and deep learning techniques. By learning the normal behavior of network traffic, these methods can detect unexpected and anomalous situations. Attackers use various techniques to mimic normal patterns in network traffic, making it difficult to detect. Real-time anomaly detection allows network administrators to detect attacks faster and respond more effectively. Real-time anomaly detection can improve network performance by detecting abnormal conditions in network traffic. Abnormal traffic can overuse the network's resources and cause the network to slow down. Real-time anomaly detection detects abnormal traffic conditions, allowing network resources to be used more effectively. In this study, blockchain technology and machine learning algorithms are combined to propose a real-time prevention model that can detect anomalies in network traffic.

**Keywords:** Anomaly behavior detection, intrusion detection, machine learning, blockchain

## 1. Introduction

The detection of an outlier that is outside the normal value that may occur in a business process is called anomaly detection. In anomaly cases, unusual or unique patterns may occur in the dataset that deviate from the expected values of the predicted behavior. Anomaly detection is a serious problem in many different fields, including cybersecurity, manufacturing problem detection, and fraud detection in the financial sector. Statistical-based methods and machine learning-based methods are the two main detection techniques for anomaly detection. While statistical methods use variables such as mean and standard deviation, machine learning-based approaches use supervised or unsupervised learning methods to identify spam. Spam refers to electronic messages sent via electronic mail or cell phone messages, sometimes individually and sometimes collectively, without the consent of the users, harassing them. Spam messages can harass users in many different categories. Figure 1 shows an image in which spam messages are classified.

In order to distinguish spam e-mails from others, it is useful to know some tips. These clues can be very useful in the preliminary diagnostic process to help users differentiate between spam e-mails and real ones. Figure 2 shows some of the clues that can be used to identify spam e-mails.

Spam emails and messages put users in a very difficult situation because they slow down routine workflow, bloat the inbox, and pose a security risk by exposing them to phishing scams or malicious links. Anomalies and spam attempts are all caused



by cyber-attacks. Although their strength and effectiveness vary depending on the nature, the main purpose of cyber-attacks is to compromise user security and exploit security vulnerabilities.

<b>SPAM MESSAGES</b>	Advertising messages promoting products or services
	Phishing scams for revealing sensitive information
	Malicious messages with links to viruses or other malware
	Chain letters or other types of pyramid schemes
	Messages of political or religious content that you did not request

Figure 1 Spam message categories

<b>TIPS USED IN SUSPECTING</b>	A misleading or irrelevant subject line
	Sender you don't know
	An unfamiliar email address
	A general greeting such as "Dear friend" or "Hello"
	Poor grammar or spelling mistakes
	Personal or sensitive information requests

Figure 2 Tips used in Suspecting spam

Cyber-attacks on web pages are carried out to gain unauthorized access to the pages, to obtain users' sensitive information, or to disrupt the normal functioning of the web page. Some cyber-attack actions targeting web pages are shown in Figure 3.

<b>CYBER ATTACK METHODS</b>	SQL Injection: A method involving injecting malicious code by accessing the database of a web application containing a database.
	Cross-Site Scripting (XSS): by injecting malicious code into a web application, it is executed in the victim user's browser by navigating to the compromised site.
	Distributed Denial of Service (DDoS): aims to keep a heavy-traffic web application busy, making it inaccessible to users.
	Phishing: aims to create a fake website or email that appears to come from a trusted source to trick users into revealing sensitive information such as passwords or credit card numbers.
	Malware: aims to infect a web application with malware such as viruses or Trojan horses that can compromise the security of the site and the computers of its visitors.

Figure 3 Cyber-attack methods

In cases where cyber-attacks on websites are successful, there are serious consequences such as theft of sensitive information of users, disruption of commercial activities of companies, and damage to the reputation of organizations. To stay safe from such attacks, it is vital to regularly update and maintain your website's security software and implement robust security measures such as intrusion detection systems and encryption.

The motivation and salient features of this work are listed below.

- An artificial intelligence intelligence-based model is proposed to detect real-time network anomalies.
- Six different machine learning models are trained for the proposed model and the training results are presented with different metrics and the most successful one is selected.
- The situations that cause anomalies are collected in a secure and transparent blacklist structure thanks to the blockchain structure.
- A smart contract is prepared to manage the registration process to the blockchain structure.
- The performances of all transactions are meticulously measured and tested for real-time operation.

## 2. Related Works

Walling and Lodh developed a univariate selection-based IDS model that can be applied with machine learning algorithms such as decision trees, kNN, SVM, and logistic regression. The developed IDS model was applied on the NSL-KDD dataset and performance improvements were demonstrated [1]. Sreenivasula and Sathya presented a NIDS model based on machine

learning methods that can detect and prevent various types of attacks. The NSL-KDD dataset was used to measure the classification performance of various ML classifiers based on different attributes. It was shown that the developed NIDS model achieved better results than existing single ML methods [2]. Aktar and Nur presented a new model for deep learning learning-based intrusion detection focusing on DoS and DDoS attacks. The performance of the proposed model is evaluated using three different datasets (CIC-DDoS2019, CIC-IDS2017, and NSL-KDD). The developed model has shown that it can achieve up to 97.58% accuracy in anomaly detection in the system [3]. In their study, Özalp and Albayrak, unlike other studies in the literature, examined the effect of the weights of the attributes in the dataset on the detection of cyber attacks on computer networks using the NSL-KDD dataset [4]. Fernandes et al. conducted a comprehensive research study on related techniques, systems, and analysis for the detection of network anomalies. They analyzed anomaly detection under five categories: categories of intrusion detection systems, network traffic anomalies, detection methods and systems, network data types, and open issues [5]. In their study, Dutta et al. used Deep Neural Network (DNN) and Long Short Term Memory (LSTM) deep models in combination with a meta-classifier (logistic regression) following the principle of mass generalization. The proposed approach is twofold. In the first step, a DSAE is used for data preprocessing and feature engineering. In the second step, a stacking ensemble learning approach is used for classification. The effectiveness of the method is evaluated on various datasets including IoT-23, LITNET-2020, and NetML-2020 collected in an IoT environment [6]. The methodology presented by Hawawreh and Rawashdeh proposes an approach to detect the presence of anomalies in the hypervisor layer. This approach is designed to deter Distributed Denial of Service (DDoS) activities between virtual machines. The proposed method for the detection and classification of traffic between virtual machines is executed through an evolutionary neural network. This network seamlessly combines particle swarm optimization with neural network to achieve its goal. The approach to detect and categorize DDoS attacks in a cloud environment detects and classifies DDoS attacks with a high success rate [7]. Hoque et al. proposed a real-time approach to detect DDoS attacks using an innovative correlation metric. The effectiveness of the technique is evaluated using three different network datasets, namely CAIDA DDoS 2007, MIT DARPA, and TUIDS. In addition, the proposed technique is executed on FPGA to evaluate its effectiveness. The detection accuracy of this method is extremely high and the FPGA implementation of this process can identify the attack in less than a microsecond [8]. Gurina and Eliseev investigate the detection of network attacks targeting web servers. The study focuses on two common types of attacks, "denial of service" and "code injection". Multiple techniques for detecting attacks are evaluated. A novel approach based on the recognition of the dynamic response of the web server during request processing is proposed to detect attacks as anomalies. After implementing the detection algorithm, its effectiveness is measured and the advantages and disadvantages of the proposed methodology are evaluated [9]. Alsamiri and Alsubhi aimed to contribute to the existing literature by evaluating various machine learning algorithms that can quickly and efficiently identify network attacks targeting IoT devices. Various detection algorithms were evaluated using a newly created dataset called Bot-IoT. In the implementation phase, seven different machine learning algorithms were used, most of which exhibited high performance. After the implementation of the Bot-IoT dataset, new features were derived and compared with previous research studies. The comparison revealed better results showing the superiority of the new features [10].

### 3. Method

The main purpose of this study is to detect attacks such as Probe, DoS, R2L, and U2R and to create a decentralized blacklist blockchain structure. For this purpose, machine learning infrastructure is prepared as a decision-making mechanism. A decentralized blacklist and request validator blockchain infrastructure that executes actions with the outputs of the decision-making mechanism has been prepared. These two infrastructures work together to create a real-time, reliable, and objective security structure. These infrastructures working together realize a secure network operation by detecting whether the request in the network is an anomaly or not and taking precautions.

#### 3.1 Dataset Description

KDD'99 dataset by Salvatore J. Stolfo et al. [11] has been one of the most widely used datasets for evaluating anomaly detection since 1999. The KDD training dataset consists of about 4,000,000 single-link vectors [12]. Each vector has 42 attributes. His 42nd attribute in the record is the class attribute, which indicates whether the link is an attack or a normal link. Class attributes are divided into five classes, one normal class and four attacks (probe, DoS, R2L, and U2R). The categories in which attacks occur are listed below [13].

- DoS attack: An attacker can cause a computer or memory resource to become sufficiently busy or full that it cannot process legitimate requests or deny access to the computer by legitimate users[14].
- User to Root Attack (U2R): Attackers attempt to gain root access to a system by accessing regular user accounts on the system and exploiting vulnerabilities (through password sniffing, dictionary attacks, or social engineering) [15].
- Remote-to-local attacks (R2L): An attacker could send packets over the network to a computer that does not have an account. However, by exploiting some vulnerability he gains his access locally as a user on this computer. An R2L attack is

unauthorized access from a remote computer. As R2L attack types he can specify Imap, Ftp Write, Phf, and Warezmaster [16].

- Probing Attack: It is a type of attack against computer networks or systems that aims to gather information about the network or systems. A probing attack assumes that the attacker can access individual components of a device, such as CPU/GPU/ASIC, RAM, non-volatile storage, or data paths, but cannot perform the invasive attacks necessary to access the internals of the device.

The first 41 attributes in the dataset can be categorized into four groups according to their characteristics: Basic (T), Content (C), Traffic (TT), and Host (H) attributes. The attributes of individual TCP connections refer to Basic attributes, attributes within a connection refer to Content attributes, attributes calculated using a two-second time window refer to Traffic attributes, and attributes designed to evaluate attacks lasting longer than two seconds refer to Host attributes.

### 3.2 Machine Learning Module

The Machine Learning Module applies a machine learning approach to the NSL-KDD dataset to determine whether requests that occur in the network are anomalies. We evaluated machine learning algorithms that model with the highest accuracy in anomaly detection [17]. There are many machine learning algorithms as supervised and unsupervised learning. We examined the advantages and disadvantages of these algorithms because of the literature survey. We defined some rules for selecting the machine learning algorithms used in this study. These rules are listed below:

1. Providing algorithm diversity
2. Use of algorithms in current studies
3. Algorithms have the potential for anomaly detection

### 3.3 Classifier Selection

Machine learning algorithms have been described in detail in many survey studies. Therefore, we have chosen to briefly describe the machine learning algorithms used in this study. Figure 4 shows the selection of the best-performing algorithm.

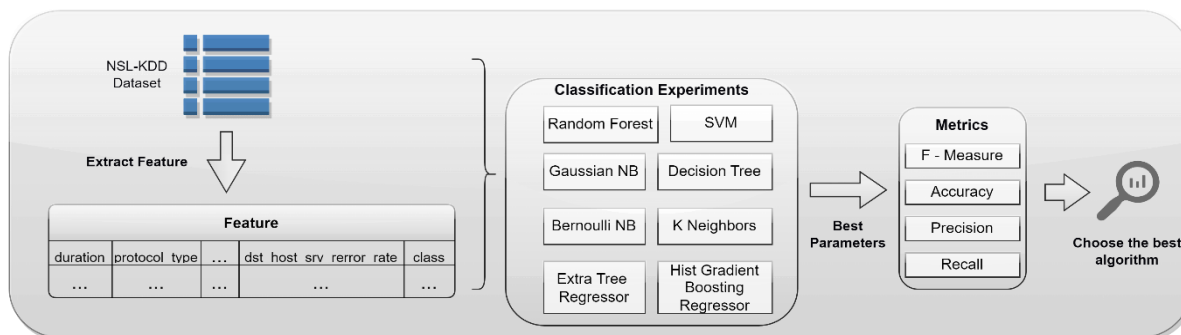


Figure 4 Machine learning algorithm selection

### 3.4 Random Forest Algorithm

The Random Forest Algorithm (RF) [18], first introduced by Leo Breiman, is a popular tool for ensemble learning. Trees in a forest learn to use a subset of feature variables. While RF works efficiently with large data sets, the generated forests or trees can be stored for later use [19]. It can handle data sets with outliers and noisy data while providing insight into the influence of variables in classification [20]. Tree-based ensemble learning algorithms are used in many industries and services such as healthcare [21], agriculture [22], transportation [23], and energy [24].

### 3.5 Support Vector Machine Algorithm

Support Vector Machines, developed by Vapnik et al. in the 1900s as one of the supervised learning methods, are used to classify linear or nonlinear data [25]. SVM is a popular machine learning algorithm that creates hyperplanes for the separation of data consisting of multiple classes in the data set [26]. With the developing technology, the amount of data obtained today is increasing. While this may seem beneficial, more data means more possibilities to identify meaningful data. This can create memory and time complexity for SVM training [27]. SVM has significant advantages in classification as it reduces the error



during training by using structural risk minimization [28]. As a result of its success in classification, SVM has been applied in many different fields such as human action recognition [29], text classification [30], and financial application [31].

### 3.6 Decision Tree Algorithm

The decision tree classifier is one of the most popular machine learning techniques. Decision trees built based on knowledge acquisition are used to classify test data [32]. A decision tree is a structure containing decision nodes and leaf nodes. Decision nodes are associated with a test  $X$  on a particular attribute of the input data and have branches that process the results of the  $X$  tests. Each leaf node represents a class with a decision outcome of the situation [33].

### 3.7 KNN Algorithm

The nearest neighbor algorithm (KNN) is a nonparametric supervised classification algorithm that produces efficient results with simple but effective performance [34]. The KNN classifier finds and analyses the nearest neighbors of sample  $x$  and classifies  $x$  into the class that has the most representatives among the neighbors. KNN calculates all distances for each state in the training set. This may not be practical for large datasets, as the growth of the dataset may incur time costs in calculating distances [35]. It has been successfully applied in many fields such as text classification [36], health [37], and economics [38].

### 3.8 Gaussian NB & Bernoulli Algorithm

Bayesian method is a statistical method used to calculate the probability of an event occurring based on its observed effects. Naive Bayes is a simple probabilistic classification technique based on the Bayes theorem with strong independence assumptions [39]. Gaussian NB assumes that when the attributes are continuous, the values associated with the classes are sampled according to a Gaussian distribution, i.e., a normal distribution. Bernoulli NB assumes that each of the multiple features is a binary-valued (present-absent, normal-attack) variable [30].

### 3.9 Extra Trees Regressor

A refinement of the Random Forest algorithm, the Extremely Random Tree (or Extra Tree) algorithm is a relatively new machine learning method that is less prone to overfit a dataset [40]. Similar to random forests, extra trees (ET) train each base predictor using a random selection of features. But to divide the nodes, it chooses at random the best characteristics and related values. Each regression tree is trained by ET using the whole training dataset. To train the model, RF employs bootstrap copies [41].

### 3.10 Gradient Boosting Regressor

Another sort of ensemble model is a gradient boosting regressor (GBR), which is an iterative collection of sequentially ordered tree models that allows the following model to learn from the errors of the preceding model. By "boosting" an ensemble of weak predictive models (often decision trees) to produce a more reliable model, this machine learning model delivers predictions [42].

### 3.11 Discussion and Analysis

Data were classified using machine learning for anomaly detection. The NSL-KDD dataset was trained and classified with machine learning algorithms. There are four attack types in the dataset: Probe, DoS, R2L and U2R. There are 67342 DoS, 11656 Probe, 995 R2L, and 52 U2R attacks in the dataset. Information on the number of attacks and normal cases in the dataset is given in Figure 5. The dataset was randomly mixed as 80% training data and 20% test data to obtain test and training datasets.

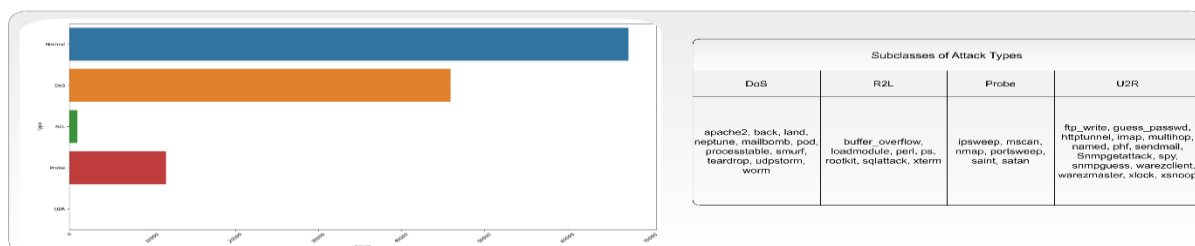


Figure 5 Number and subclasses of attack types in the dataset

Random Forest, SVM, Decision Tree, K Neighbours, Extra Trees Regressor, Gradient Boosting Regressor, Gaussian Naive Bayes, and Bernoulli Naive Bayes algorithms were used to classify the data. The parameters used in classification and classification results are given in Table 1.

Table 1. NSL-KDD Data set properties

Algorithms	Classification Parameters	Accuracy Percentage	Classification Result				
<b>Random Forest</b>	n_estimators:10, max_features:sqrt, criterion: entropy	0.99857	precision	recall	f1-score	support	
			DoS	1.00	1.00	1.00	9302
			Normal	1.00	1.00	1.00	13396
			Probe	1.00	0.99	1.00	2285
			R2L	0.99	0.96	0.97	203
			U2R	0.71	0.56	0.63	9
<b>SVM</b>	kernel='rbf', gamma=0.001, C=1000	0.99571	precision	recall	f1-score	support	
			DoS	1.00	1.00	1.00	9147
			Normal	0.99	1.00	1.00	13463
			Probe	0.99	0.98	0.99	2358
			R2L	0.98	0.93	0.95	218
			U2R	0.40	0.22	0.29	9
<b>Desicion Tree</b>	criterion: entropy, splitter: best, max_depth: None	0.99781	precision	recall	f1-score	support	
			DoS	1.00	1.00	1.00	9302
			Normal	1.00	1.00	1.00	13396
			Probe	0.99	0.99	0.99	2285
			R2L	0.96	0.99	0.97	203
			U2R	0.78	0.78	0.78	9
<b>K-Neighbors</b>	weights: distance, algorithm: auto	0.99293	precision	recall	f1-score	support	
			DoS	0.99	1.00	0.99	9302
			Normal	1.00	1.00	1.00	13396
			Probe	0.97	0.97	0.97	2285
			R2L	0.93	0.97	0.95	203
			U2R	0.75	0.67	0.71	9
<b>Bernualli NB</b>	alpha: 0.5, fit_prior: True	0.81282	precision	recall	f1-score	support	
			DoS	0.96	0.76	0.85	9302
			Normal	0.91	0.91	0.91	13396
			Probe	0.28	0.51	0.36	2285
			R2L	0.29	0.42	0.35	203
			U2R	0.15	0.67	0.25	9
<b>Gaussian NB</b>	var_smoothing: 1.0	0.53169	precision	recall	f1-score	support	
			DoS	0.00	0.00	0.00	9302
			Normal	0.53	1.00	0.69	13396
			Probe	0.50	0.00	0.00	2285
			R2L	0.00	0.00	0.00	203
			U2R	0.00	0.00	0.00	9
<b>Extra Trees Regressor</b>		0.97872	precision	recall	f1-score	support	
			DoS	1.00	1.00	1.00	9231
			Normal	1.00	0.99	0.99	2344
			Probe	0.35	0.53	0.42	15
			R2L	0.27	0.98	0.43	195
			U2R	1.00	0.96	0.98	13410
<b>HistGradient Boositng Regressor</b>		0.57963	precision	recall	f1-score	support	
			DoS	0.89	0.92	0.90	9169
			Normal	0.99	0.56	0.71	2397
			Probe	0.06	0.57	0.11	7
			R2L	0.02	0.84	0.04	198
			U2R	1.00	0.34	0.51	13424

Random Forest, SVM, Decision Tree, and K-Neighbors classifiers achieved approximately 99 percent classification success, while Bernoulli Naive Bayes achieved 81 percent and Gaussian Naive Bayes achieved 53 percent classification success. The error matrices of the classifications are given in Figure 6.

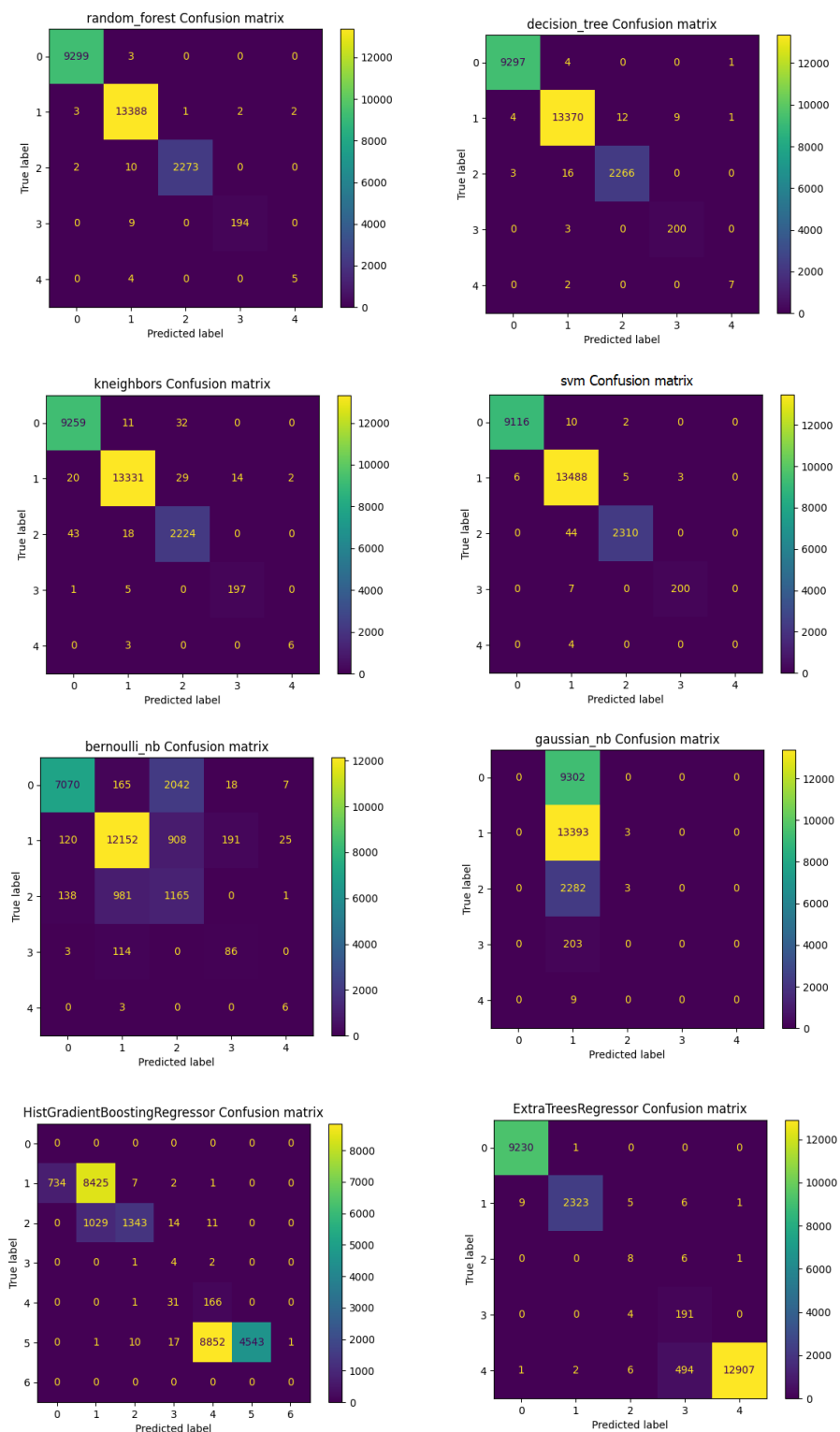


Figure 6. Confusion Matrix for Machine Learning algorithms

In detecting attacks, factors such as machine learning algorithms and data pre-processing change the success rates. Table 2 summarizes the results obtained by different researchers.

Table 2 Comparison of the study with similar studies

Ref.	Year	Research Paper Title	Algorithm used in preprocessing / Model core	Accuracy %
[43]	2015	Research on NSL-KDD data set of intrusion detection system based on classification algorithm	CFS J48 SVM Naive Bayes	Varies between 70.1 and 99.8 for different attack types and algorithms
[44]	2016	Anomaly-based intrusion detection system through feature selection analysis and construction of hybrid efficient models	SMOTE CANN	98.99
[45]	2016	A hybrid data mining approach for intrusion detection in the imbalanced NSL-KDD dataset.	Hybrid comprising of J48 Random Tree Naïve Bayes	99.81
[46]	2022	A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems	Hybrid SVM Model	89.7
[47]	2023	Hybrid Statistical-Machine Learning for Real-Time Anomaly Detection in Industrial Cyber-Physical Systems	Hybrid LSTM Model	95
This Work	2023	Real-Time Intelligent Anomaly Detection and Prevention System	Random Forest	99.85

Algorithm 1 Blacklist Smart Contract Pseudo Code

```

1  func Initialize ()
2      configure DistrubutedLedgerRules ()
3      configure DistrubutedLedgerStandarts ()
4  func CreateBlackListAsset (ctx, params) ←Ip, Mac, Timestamp, Request Address
5      if exist (ctx in BlackList) == true then
6          return rejection.
7      else
8          add StandartLogList (id ← params)
9          return (obj ⊃ [params])
10
11 func GetBlackList (ctx)
12     if exist (ctx in BlackList) == true then
13         while! result. done
14             var res = GetAllList (ctx, id) then
15                 return result
16     else
17         add StandartLogList (id ← params)
18         return (obj ⊃ [params])
19
20 func GetAllLogList (ctx, id)
21     do
22         static allLogListResult = []
23         while! result. done then.
24             allLogListResult.Push → Key: result.val.key, Record: params
25             result ← await. iterator. next ()
26         return allResult end

```

In the prepared smart contract, the initial rules and settings of the distributed ledger structure are executed with the Initialize method. The classification information from machine learning is integrated into our smart contract as an asset. This asset is

represented as "ctx" in the Pseudo code. To complete the security process, the smart contract checks the machine-learning results of the request in the blockchain and executes forked transactions according to the process. When creating the blacklist ledger, some information about the requesting request is requested and its status in the list is checked. Depending on the returned result, the blacklist process is managed. When a positive response is received, a new object is created, and a new record is returned at the end of the process. The GetBlackList or GetAllList method is executed to read the records. After checking the necessary permissions, the data saved in the ledger can be read and listed with the help of an iterator. Algorithm 1 shows the pseudo-code of the generated smart contract.

#### 4. Findings

Each network created in blockchain systems has a limit of transactions per second that it can process. This limit is referred to as TPS (Transfer Per Second). TPS is an acronym that stands for how many transactions per second blockchain networks can confirm and validate. In Figure 7, the error rates at different TPS values are measured with 10 different threads performing simultaneous tasks for a fixed duration of 20 seconds. In these measurements, the TPS value varies between 20-1000. In light of the findings obtained, it is seen that the blockchain successfully manages the requests received from 10 different threads with small-valued error rates up to 800 TPS values. As the TPS value increases above 800, it is seen that the error values start to increase linearly. In this sense, it can be said that the upper limit of the performance of the proposed blockchain system is the TPS value of 800.

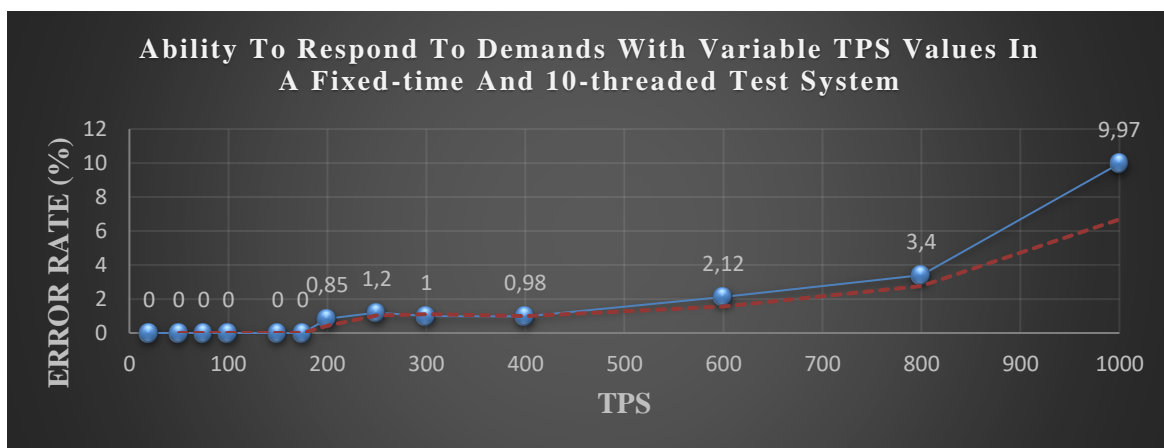


Figure 7 Ability to respond to demands with variable TPS values in a fixed-time and 10-threaded test system.

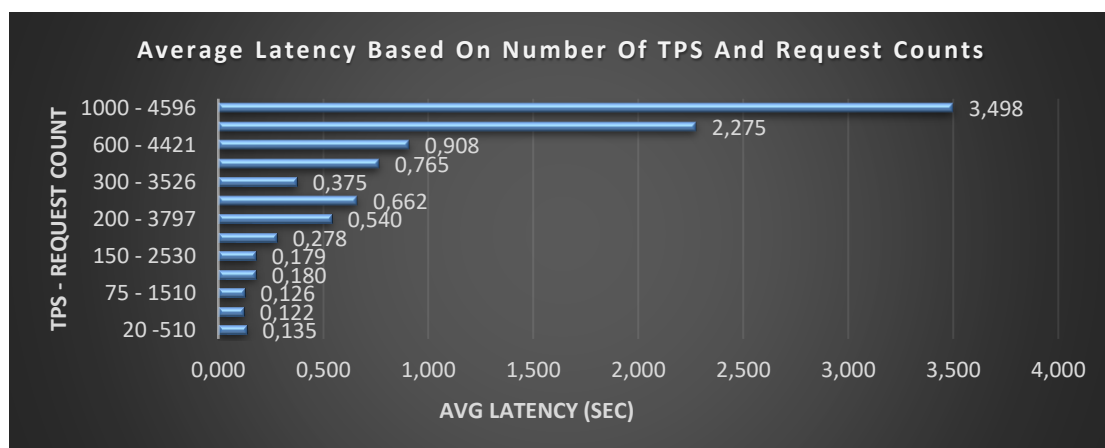


Figure 8 Average latency based on the number of TPS and Request Counts

It was envisaged that the temporal performance of the system should be evaluated by measuring the error rates during its active operation. For this reason, in Figure 8, the number of requests generated according to TPS values ranging from 20-1000 were combined and the average completion time of the blockchain process was measured. With the findings obtained, the average delay time experienced in the blockchain system until the TPS value reaches 800 varies between 0.135 seconds and 0.908 seconds. Although this latency is considered acceptable for a blockchain system with strong verification and

logging processes, it is observed that the average latency suddenly reaches 2.227 and gradually increases as the TPS value exceeds 800.

Considering the findings obtained from Figure 7 and Figure 8, it can be said that the upper limit of the performance of the prepared blockchain structure has a value of 800 TPS. The fact that the results obtained in these two graphs confirm each other shows the objectivity of the measurement processes performed.

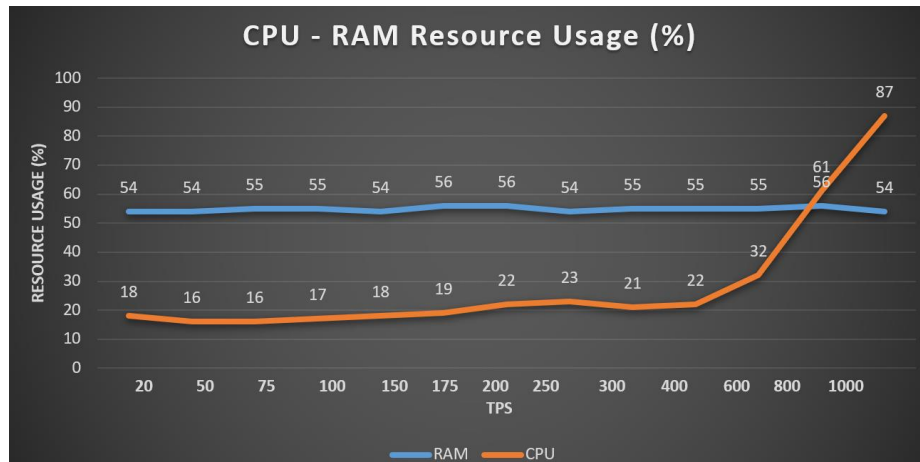


Figure 9 CPU - RAM Resource Usage (%)

To complete the performance evaluation of the proposed blockchain structure, it was deemed necessary to determine how it utilizes computer resources during its operation. The measurements were performed on a computer equipped with an Intel(R) Xeon(R) E-2236 CPU @ 3.40GHz 3.41 GHz and 32 GB of RAM. For this reason, in Figure 9, the CPU and RAM resource utilization rates of the computer during the execution of blockchain transactions are measured and graphed. In the measurement, was aimed to measure the upper limits of the system by increasing the amount of work demand per unit time (TPS value). In the findings obtained, it was interpreted that the RAM capacity did not show a significant change and therefore remained constant. It was found that the processor power remained at similar values until 600 TPS and increased linearly after 700 TPS. These findings show us that the optimal upper limit of the system in terms of hardware is 800 TPS, just like in Figure 6 and Figure 7.

## 5. Conclusion

In this study, blockchain technology and machine learning algorithms are combined to propose a real-time prevention model that can detect anomalies that may occur in network traffic. The classification criteria, success values, and classification results of the algorithms used in the training are explained and demonstrated in detail. According to the results obtained, the Decision Tree algorithm has the most successful classification results among the tested algorithms. In the prepared blockchain structure, anomalies detected with the help of smart contracts are transferred to the blacklist chain. Standard requests continue their processes in the usual flow of network traffic. The performance measurements of these transactions have been meticulously measured and resource utilization has been measured and shown in the study. As the TPS value exceeded 500, an increase in error conditions, response delay times, and resource utilization was observed. When the security and decentralization contributions provided by the system are evaluated, it can be said that the results obtained are satisfactory. In future studies, we plan to improve the resource utilization and time performance of this system. We aim to minimize error rates by including optimization methods in our proposed model.

## References

- [1] S. Walling and S. Lodh, "Performance Evaluation of Supervised Machine Learning Based Intrusion Detection with Univariate Feature Selection on NSL KDD Dataset," Feb. 2023, doi: 10.21203/RS.3.RS-2537820/V1.
- [2] T. S. Reddy and R. Sathya, "Ensemble Machine Learning Techniques for Attack Prediction in NIDS Environment," Iraqi Journal For Computer Science and Mathematics, vol. 3, no. 2, pp. 78–82, Mar. 2022, doi: 10.52866/IJCSM.2022.02.01.008.
- [3] S. Aktar and A. Yasin Nur, "Towards DDoS attack detection using deep learning approach," Comput Secur, vol. 129, p. 103251, Jun. 2023, doi: 10.1016/J.COSE.2023.103251.
- [4] A. N. Özalp and Z. Albayrak, "Detecting Cyber Attacks with High-Frequency Features using Machine Learning Algorithms," Acta Polytechnica Hungarica, vol. 19, no. 7, pp. 213–233, 2022, doi: 10.12700/APH.19.7.2022.7.12.
- [5] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey

- on network anomaly detection,” *Telecommunication Systems* 2018 70:3, vol. 70, no. 3, pp. 447–489, Jul. 2018, doi: 10.1007/S11235-018-0475-8.
- [6] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, “A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection,” *Sensors* 2020, Vol. 20, Page 4583, vol. 20, no. 16, p. 4583, Aug. 2020, doi: 10.3390/S20164583.
- [7] A. Rawashdeh, M. Alkasassbeh, and M. Al-Hawawreh, “An anomaly-based approach for DDoS attack detection in cloud environment,” *International Journal of Computer Applications in Technology*, vol. 57, no. 4, pp. 312–324, 2018, doi: 10.1504/IJCAT.2018.093533.
- [8] N. Hoque, H. Kashyap, and D. K. Bhattacharyya, “Real-time DDoS attack detection using FPGA,” *Comput Commun*, vol. 110, pp. 48–58, Sep. 2017, doi: 10.1016/J.COMCOM.2017.05.015.
- [9] A. Gurina and V. Eliseev, “Anomaly-Based Method for Detecting Multiple Classes of Network Attacks,” *Information* 2019, Vol. 10, Page 84, vol. 10, no. 3, p. 84, Feb. 2019, doi: 10.3390/INFO10030084.
- [10] J. Alsamiri and K. Alsubhi, “Internet of Things Cyber Attacks Detection using Machine Learning,” *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019, Accessed: May 10, 2023. [Online]. Available: [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- [11] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, “Cost-based modeling for fraud and intrusion detection: Results from the JAM project,” *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000*, vol. 2, pp. 130–144, 2000, doi: 10.1109/DISCEX.2000.821515.
- [12] “UCI Machine Learning Repository: KDD Cup 1999 Data Data Set.” <https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data> (accessed Mar. 29, 2023).
- [13] M. Tavallae, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, Dec. 2009, doi: 10.1109/CISDA.2009.5356528.
- [14] R. Vishwakarma and A. K. Jain, “A survey of DDoS attacking techniques and defence mechanisms in the IoT network,” *Telecommun Syst*, vol. 73, no. 1, pp. 3–25, Jan. 2020, doi: 10.1007/S11235-019-00599-Z/TABLES/5.
- [15] D. Sklavounos, “Statistical Process Control Method for Cyber Intrusion Detection (DDoS, U2R, R2L, Probe),” *International Journal of Cyber-Security and Digital Forensics*, vol. 8, no. 1, pp. 82–88, 2019, doi: 10.17781/P002560.
- [16] M. Amini, R. Jalili, and H. R. Shahriari, “RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks,” *Comput Secur*, vol. 25, no. 6, pp. 459–468, Sep. 2006, doi: 10.1016/J.COSE.2006.05.003.
- [17] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, “Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review,” *Journal of Cybersecurity and Privacy* 2022, Vol. 2, Pages 527-555, vol. 2, no. 3, pp. 527–555, Jul. 2022, doi: 10.3390/JCP2030027.
- [18] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324/METRICS.
- [19] K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification,” *Augmented Human Research* 2020 5:1, vol. 5, no. 1, pp. 1–16, Mar. 2020, doi: 10.1007/S41133-020-00032-0.
- [20] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, “An assessment of the effectiveness of a random forest classifier for land-cover classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, no. 1, pp. 93–104, Jan. 2012, doi: 10.1016/J.ISPRSJPRS.2011.11.002.
- [21] C. Iwendi et al., “COVID-19 patient health prediction using boosted random forest algorithm,” *Front Public Health*, vol. 8, p. 357, Jul. 2020, doi: 10.3389/FPUBH.2020.00357/BIBTEX.
- [22] J. Magidi, L. Nhamo, S. Mpandeli, and T. Mabhaudhi, “Application of the Random Forest Classifier to Map Irrigated Areas Using Google Earth Engine,” *Remote Sensing* 2021, Vol. 13, Page 876, vol. 13, no. 5, p. 876, Feb. 2021, doi: 10.3390/RS13050876.
- [23] X. Cheng and B. Huang, “A center-based secure and stable clustering algorithm for VANETs on highways,” *Wirel Commun Mob Comput*, vol. 2019, 2019, doi: 10.1155/2019/8415234.
- [24] D. Liu and K. Sun, “Random forest solar power forecast based on classification optimization,” *Energy*, vol. 187, p. 115940, Nov. 2019, doi: 10.1016/J.ENERGY.2019.115940.
- [25] M. A. Chandra and S. S. Bedi, “Survey on SVM and their application in image classification,” *International Journal of Information Technology (Singapore)*, vol. 13, no. 5, pp. 1–11, Oct. 2021, doi: 10.1007/S41870-017-0080-1/TABLES/1.
- [26] S. Dong, “Multi class SVM algorithm with active learning for network traffic classification,” *Expert Syst Appl*, vol. 176, p. 114885, Aug. 2021, doi: 10.1016/J.ESWA.2021.114885.
- [27] J. Nalepa and M. Kawulok, “Selecting training sets for support vector machines: a review,” *Artificial Intelligence Review* 2018 52:2, vol. 52, no. 2, pp. 857–900, Jan. 2018, doi: 10.1007/S10462-017-9611-1.
- [28] M. Tanveer, T. Rajani, R. Rastogi, Y. H. Shao, and M. A. Ganaie, “Comprehensive review on twin support vector machines,” *Ann Oper Res*, pp. 1–46, Mar. 2022, doi: 10.1007/S10479-022-04575-W/TABLES/8.
- [29] S. Agarwal, D. Tomar, and Siddhant, “Prediction of software defects using twin support vector machine,” *Proceedings of the 2014 International Conference on Information Systems and Computer Networks, ISCON 2014*, pp. 128–132, Nov. 2014, doi: 10.1109/ICISCON.2014.6965232.

- [30] N. Rezaeian and G. Novikova, "Persian Text Classification using naive Bayes algorithms and Support Vector Machine algorithm," *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 8, no. 1, pp. 178–188, Mar. 2020, doi: 10.52549/IJEI.V8I1.1696.
- [31] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega (Westport)*, vol. 29, no. 4, pp. 309–317, Aug. 2001, doi: 10.1016/S0305-0483(01)00026-3.
- [32] I. D. Mienye, Y. Sun, and Z. Wang, "Prediction performance of improved decision tree-based algorithms: a review," *Procedia Manuf*, vol. 35, pp. 698–703, Jan. 2019, doi: 10.1016/J.PROMFG.2019.06.011.
- [33] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision tree classifier for network intrusion detection with GA-based feature selection," *Proceedings of the Annual Southeast Conference*, vol. 2, pp. 2136–2141, 2005, doi: 10.1145/1167253.1167288.
- [34] S. Hota, S. P.-Int. J. Eng. Technol, and undefined 2018, "KNN classifier based approach for multi-class sentiment analysis of twitter data," *scholar.archive.org*, vol. 7, no. 3, pp. 1372–1375, 2018, doi: 10.14419/ijet.v7i3.12656.
- [35] F. Moreno-Seco, L. Micó, and J. Oncina, "A modification of the LAESA algorithm for approximated k-NN classification," *Pattern Recognit Lett*, vol. 24, no. 1–3, pp. 47–53, Jan. 2003, doi: 10.1016/S0167-8655(02)00187-3.
- [36] S. Tan, "An effective refinement strategy for KNN text classifier," *Expert Syst Appl*, vol. 30, no. 2, pp. 290–298, Feb. 2006, doi: 10.1016/J.ESWA.2005.07.019.
- [37] A. Murugan, S. A. H. Nair, and K. P. S. Kumar, "Detection of Skin Cancer Using SVM, Random Forest and kNN Classifiers," *J Med Syst*, vol. 43, no. 8, pp. 1–9, Aug. 2019, doi: 10.1007/S10916-019-1400-8/FIGURES/6.
- [38] Imandoust SB and Bolandraftar M. *Int. Journal of Engineering Research and Applications*. Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610
- [39] J. Bains, K. Kaki, K. S.-I. J. of Computer, and undefined 2013, "Intrusion detection system with multi layer using Bayesian networks," *Citeseer*, vol. 67, no. 5, pp. 975–8887, 2013, Accessed: Mar. 29, 2023
- [40] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63, 3-42.
- [41] John, V., Liu, Z., Guo, C., Mita, S., & Kidono, K. (2016). Real-time lane estimation using deep features and extra trees regression. In *Image and Video Technology: 7th Pacific-Rim Symposium, PSIVT 2015, Auckland, New Zealand, November 25-27, 2015, Revised Selected Papers 7* (pp. 721-733). Springer International Publishing.
- [42] Otchere, D. A., Ganat, T. O. A., Ojero, J. O., Tackie-Otoo, B. N., & Taki, M. Y. (2022). Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions. *Journal of Petroleum Science and Engineering*, 208, 109244.
- [43] D. H. Deshmukh, T. Ghorpade, and P. Padiya, "Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset," in *Proceedings - 2015 IEEE International Conference on Communication, Information and Computing Technology, ICCICT 2015*, 2015.
- [44] K. Rai, M. S. Devi, and A. Guleria, "Decision Tree Based Algorithm for Intrusion Detection," vol. 2834, pp. 2828–2834, 2016.
- [45] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, 2016.
- [46] D. Velásquez et al., "A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems," in *IEEE Access*, vol. 10, pp. 72024–72036, 2022, doi: 10.1109/ACCESS.2022.3188102.
- [47] W. Hao, T. Yang and Q. Yang, "Hybrid Statistical-Machine Learning for Real-Time Anomaly Detection in Industrial Cyber-Physical Systems," in *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 32–46, Jan. 2023, doi: 10.1109/TASE.2021.3073396.

#### Author(s) Contributions

All three authors contributed equally to the study.

#### Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

#### Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

#### Availability of data and material

Not applicable

#### Plagiarism Statement

This article has been scanned by iThenticate™.





# A Novel Gender Classification Model based on Convolutional Neural Network through Handwritten Text and Numeral

Pakize Erdoğan<sup>1</sup>, Abdullah Talha Kabak<sup>1</sup>, Enver Küçüközlü<sup>1</sup>, Büşra Takıl<sup>1</sup>, Ezgi Kara Timuçin<sup>1</sup>

<sup>1</sup> Duzce University, Faculty of Engineering, Department of Computer Engineering, Düzce, Türkiye



## Corresponding author:

Pakize Erdoğan, Düzce University,  
Faculty of Engineering, Department of  
Computer Engineering, Düzce, Türkiye

## E-mail address:

[pakizeerdogmus@duzce.edu.tr](mailto:pakizeerdogmus@duzce.edu.tr)

Submitted: 04 August 2023

Revision Requested: 06 September 2023

Last Revision Received: 10 September 2023

Accepted: 28 September 2023

Published Online: 30 September 2023

Citation: Erdoğan P. et al. (2023).

A Novel Gender Classification Model  
based on Convolutional Neural Network  
through Handwritten Text and Numeral.

*Sakarya University Journal of Computer  
and Information Sciences*. 6 (3)

<https://doi.org/10.35377/saucis...1337649>

## ABSTRACT

Human handwriting is used to investigate human characteristics in various applications, including but not limited to biometric authentication, personality profiling, historical document analysis, and forensic investigations. Gender is one of the most distinguishing characteristics of human beings. From this point forth, we propose a novel end-to-end model based on Convolutional Neural Network (CNN) that automatically extracts features from a given handwritten sample, which contains both handwritten text and numerals unlike the related work that uses only handwritten text and classifies its owner's gender. In addition to proposing a novel model, we introduce a new dataset that consists of 530 gender-labeled Turkish handwritten samples since, to the best of our knowledge, there does not exist a public gender-labeled Turkish handwriting dataset. Following an exhaustive process of hyperparameter optimization, the proposed CNN featured the most optimal hyperparameters and was both trained and evaluated on this dataset. According to the experimental result, the proposed novel model obtained an accuracy as high as 74.46%, which overperformed the state-of-the-art baselines and is promising on such a task that even humans could not have achieved highly-accurate results for, as of yet.

**Keywords:** Handwriting, gender classification, convolutional neural network, computer vision, forensic science

## 1. Introduction

Today's rapidly advancing technological world does indeed bring forth numerous security challenges. These technological advancements serve not only benign consumers but also, unfortunately, enable malicious entities. Authentication technologies are employed to address security concerns by verifying or recognizing a person's identity through various factors, including passwords and facial features. Biometric authentication is one of the authentication methods using inherent factors such as fingerprints, DNA (DeoxyriboNucleic Acid), face, or retina. The utilization of computers for human recognition based on physical and behavioral traits traces its origins to the digital computer revolution of the 1960s [1]. But even today, after more than 60 years, biometric studies remain fresh since new technologies require using more secure applications. Gender detection, the ability to detect an individual's gender based on distinct physiological features and patterns, is also one of the biometric factors for authentication and forensic applications. In addition to this, human handwriting is used to investigate human characteristics in various applications, including but not limited to personality profiling [2], [3], historical document analysis [4], and forensic investigations [5], [6]. In areas like artificial intelligence (AI), computer vision, and human-computer interaction, precise gender detection becomes pivotal in crafting personalized and inclusive user experiences. Furthermore, gender detection can play a vital role in promoting fair representation and addressing biases, particularly within domains such as criminal justice and employment. For criminal justice, gender detection provides,



including but not limited to (i) equitable treatment, (ii) risk assessment, (iii) victim identification, (iv) investigation and profiling, and (v) evidence handling. Gender stands as the primary physiological and physical distinction among individuals. It influences how people perceive themselves and each other [7]. So, from hand use to brain functions, there are fundamental differences between genders [8]– [10]. The reasons behind its importance for human distinction can be listed as follows: (i) social identity and cultural context, (ii) self-perception and identity formation, (iii) social interaction and communication, (iv) access to opportunities, (v) representation, (vi) advocacy, (vii) psychological well-being, and (viii) historical significance.

Drawing is the oldest communication tool, including the alphabet, digits, and traffic signs and has been used for centuries by various cultures as a means to convey ideas, stories, and information visually. A drawing consists of a lot of information about its creator, such as graphology, expressiveness, attention to detail, creativity, imagination, emotions, feelings, cultural and social influences, and communication style. Even having *Alzheimer's* or *Parkinson's disease* [11] is one of these conveyed information. Drawing differs between genders. Sex differences in drawings have been discussed by a number of researchers [12], [13]. Handwriting, as a biometric modality, offers an unobtrusive means of inferring gender-related attributes without the need for direct personal interaction. This capacity has sparked interest in a wide range of fields, including but not limited to psychology, forensics, linguistics, and AI. Examining the connection between handwriting patterns and the identification of gender offers valuable insights into how individuals encapsulate their identities within the very act of writing. The motivation behind this study is to explore the potential presence of gender differences in handwriting drawings. When we have reviewed the literature, we have found that there exist some studies on this topic [14]–[17].

In this paper, we explore the relationship between handwriting and gender. In other words, we classify an individual's gender through handwriting. In order to verify the relation between handwriting and gender, we collected both handwritten Turkish sentences consisting of the whole letters in the Turkish alphabet and the numbers from zero to nine. After collecting the raw data, we digitized it and constructed a dataset for both handwritten numbers and text. Some image pre-processing methods, such as denoising and morphological operations, have been applied to the dataset. After the pre-processing of the dataset, we have proposed a novel Convolutional Neural Network (CNN) model that accepts the handwritten text and numerals as the input and generates the detected gender, which covers two classes, namely, (i) *male*, and (ii) *female*. The deliberate design choice for the proposed model was to adopt a CNN structure, given that CNNs have consistently demonstrated state-of-the-art performance across a spectrum of computer vision tasks. These tasks encompass a wide range, such as image classification, object detection, object tracking, medical image analysis, autonomous driving, facial recognition, and document analysis, among others. The main contributions of this article can be summarized as follows:

- We propose a novel CNN model combining both handwritten text and numerals features as input. To the best of our knowledge, this is the first study that makes gender classification through a combination of handwritten text and numerals.
- Thanks to proposing a model based on DNN, neither manual feature extraction nor manual feature selection was required. Instead, an end-to-end solution was proposed.
- Since there does not exist a public gender-labeled Turkish handwritten dataset, we introduce a new dataset that consists of 530 gender-labeled Turkish handwritten samples as another contribution to the research field.
- A comprehensive range of values for each hyperparameter was evaluated in automated manner to discover the most optimal combination of hyperparameters that yield the highest classification performance for the proposed model.
- A wide range of widely used traditional Machine Learning (ML), and DNN models was employed as the baseline of the proposed CNN model. According to the experimental result, the proposed CNN model that yields both handwritten text and numerals obtained better accuracy, an accuracy as high as 74.46%, than the state-of-the-art baselines on a task that even humans could not have achieved highly-accurate results for, as of yet [18].

The remaining of the paper is organized as follows: Section 2 briefly reviews the related work. Section 3 outlines the materials and methods employed in this study. In Section 4, we present the experimental results and engage in discussions surrounding them. Lastly, Section 5 encapsulates the paper by drawing conclusions and suggesting potential avenues for future exploration.

## 2. Related Work

There exist studies that deal with gender classification through English, French, and Arabic text, while studies dealing with gender classification for handwritten Turkish text lack in the research field. To the best of our knowledge, there does not exist a study that makes gender classification from handwritten Turkish text. In an early study, *Koppel et al.* [19] introduced a method rooted in a variant of Exponential Gradient for gender classification using documents sourced from the British National Corpus (BNC). Each individual document extracted was delineated by a feature vector encapsulating distinctive characteristics. The dimensionality of these feature vectors was reduced by the elimination of the irrelevant features. According to the experimental result, the proposed model obtained an accuracy of approximately 80%.

Liwicki et al. [20] proposed a model for detecting gender and handedness from online handwriting. In terms of gender detection, they covered two classes, namely, (i) *male*, and (ii) *female*. Regarding handedness detection, they covered left- and right-handedness. To this end, they employed two models: (i) The proposed first model employed *Support Vector Machine (SVM)*, and (ii) the other model employed *Gaussian Mixture Model (GMM)*. These proposed models were trained and evaluated on the *IAM-OnDB*, an English handwriting dataset consisting of more than 200 writers with eight handwritten texts per writer which were acquired from a whiteboard. Despite that, the authors used only 100 of them for the training of the gender classifier and 30 of them for the training of the handedness classifier. The gender detection model was evaluated on a set of 50 writers. The obtained accuracy values on this subset were 67.06% and 62.19% when *GMM* and *SVM* were employed for the classification, respectively. The handedness classification model was evaluated on a set of 30 writers. The obtained accuracy values on this subset were 62.57% and 84.66% when *GMM* and *SVM* were employed for the classification, respectively. The limitations of this study are as follows: (i) Both classifiers were trained on a small dataset despite having a relatively large dataset, and (ii) more complex ML models such as Deep Neural Networks (DNNs) were not employed in addition to the employed traditional ML models.

Gattal et al. [21] proposed a handwriting analysis-based gender classification model using *Cloud of Line Distribution (COLD)* and Hinge features, which were coupled with two SVM classifiers. The proposed model was evaluated on a subset of the *QUWI* dataset, which consisted of 1,000 samples. The constructed subset was split as follows: 500 samples were used for the training, 250 samples were used for the validation, and the remaining 250 samples were used for the testing. The proposed model obtained an accuracy of 73.60% on the test set.

Morera et al. [22] introduced a CNN-based model for gender and handedness classification. This model was applied to two publicly available handwriting datasets: (i) the *IAM* dataset, containing English text, and (ii) the *KHATT* dataset, containing Arabic text. The experimental findings revealed that the proposed model achieved an accuracy of 80.72% for gender classification on the *IAM* dataset. As for the *KHATT* dataset, the accuracy of the proposed model was calculated at 68.90%.

Rabaev et al. [23] proposed a DNN for gender classification from handwriting images. In this study, they investigated cross-domain transfer learning with *ImageNet* [24] pre-training. The experiments were carried out on two datasets, namely, (i) the *QUWI* dataset, and (ii) a new dataset of documents in Hebrew script. They experimented with various DNNs and demonstrated that advanced DNNs outperformed traditional ML algorithms.

Al Maadeed and Hassaine [25] proposed a gender classification approach from offline documents using two approaches as follows: (i) All subjects wrote the same text, and (ii) each subject wrote a different text. They extracted a number of features based on shape, including but not limited to curvatures, chain codes, and stroke orientations. They employed two classification algorithms, namely, (i) *Random Forest*, and (ii) *Kernel Discriminant Analysis (KDA)*. The proposed model underwent evaluation on the *QUWI* dataset through a series of diverse experiments involving Arabic texts, English texts, and a merged amalgamation of the two. According to the experimental result, an accuracy of 72.30%, was obtained when the proposed model employed *KDA* when documents from both languages were combined and the handwritten texts of subjects were the same.

Bouadjenek et al. [26] introduced a gender classification methodology utilizing handwriting samples. Their approach was founded on a fusion of *Histogram of Oriented Gradients (HOG)* and *SVM*. While *HOG* was employed to extract relevant features, *SVM* was leveraged for classification purposes. This proposed model underwent evaluation using two distinct datasets: (i) the *IAM* dataset and (ii) the *KHATT* dataset. Based on experimental findings, the proposed model achieved a precision of 75.45% on the *IAM* dataset and a precision of 68.89% on the *KHATT* dataset.

Siddiqi et al. [27] proposed a gender classification of handwriting based on slant/orientation, roundedness/curvature, neatness/legibility, and writing texture features. They employed ANN and *SVM* for the classification. The proposed classifiers were evaluated on the *QUWI* and *MSHD* datasets. According to the experimental result, the proposed model that employed *SVM* obtained an accuracy of 68.75% for the *QUWI* dataset and an accuracy of 73.02% for the *MSHD* dataset when slant and curvature features were used.

Akbari et al. [28] introduced a gender classification methodology using handwriting images. This method initially transforms each image into a texture representation, which is then decomposed into multiple subbands at different levels. These subbands are subsequently utilized to create Probabilistic Finite State Automata (PFSA) for generating feature vectors. For classification purposes, they applied both *SVM* and NNs. The proposed models were trained and evaluated on the *QUWI* and *MSHD* datasets. According to the experimental result, the proposed NN obtained the best accuracy, an accuracy of 79.30% on the *QUWI* dataset. When it comes to the *MSHD* dataset, the proposed model based on *SVM* obtained the best accuracy, an accuracy of 79.90%.

Bouadjenek et al. [29] proposed a gender classification model for handwriting images. They employed *HOG* and *Local Binary Patterns (LBP)* as feature extractors and *SVM* as the classifier. The experiments were carried out on the *IAM* dataset. According to the experimental result, the proposed model based on *HOG* obtained an accuracy of 74%.

Youssef et al. [30] proposed a gender classification model based on the combination of *Wavelet Domain Local Binary Patterns*

(*WD-LBP*) and *SVM*. They trained and evaluated their model on a subset of the *QUWI* dataset, which consists of documents in English, and Arabic. According to the experimental result, the proposed model obtained an accuracy of 74.30%.

*Illouz et al.* [18] proposed a CNN-based model for gender classification using handwriting data. This model comprised six layers: Four convolutional layers, succeeded by a *Dense* layer and a *softmax* output layer. The proposed classifier underwent evaluation on their proprietary dataset, namely, *HEBIU*, encompassing a total of 810 samples in Hebrew and English collected from 405 subjects. Through a series of experiments, the highest accuracy achieved was 82.89%. Notably, this peak accuracy was attained when the model was trained on Hebrew samples and evaluated on English samples. Unlike our model, this model yielded 200 patches extracted from each handwriting sample.

*Maken and Gupta* [31] proposed an ensemble approach that employed *SVM*, *Logistic Regression (LR)*, and *k-Nearest Neighbor (kNN)* for automated classification of gender from handwriting using the landmarks of differences between genders. They used the shape of the visual appearance of the handwriting for extracting features of the handwriting such as slantness (direction), area, and perimeter. The proposed model was evaluated on the dataset of the *ICDAR 2013 Gender Prediction Competition*, which comprised 282 writers with 2 samples per writer. According to the experimental result, the proposed model obtained an accuracy of 65.71%.

Table 1 lists a comparison of the related work in terms of employed technique(s), used dataset(s), covered language(s), content type, and obtained gender classification accuracy.

Table 1 A comparison of the related work

Related Work	Employed Technique(s)	Used Dataset(s)	Covered Language(s)	Content Type	Classification Accuracy
[19]	Exponential Gradient	A subset of <i>BNC</i>	English	Handwritten text	~80%
[20]	<i>SVM</i> , and <i>GMM</i>	<i>IAM-OnDB</i>	English	Handwritten text	67.06%
[21]	<i>COLD</i> and Hinge features coupled with <i>SVM</i>	A subset of <i>QUWI</i>	English, and Arabic	Handwritten text	73.60%
[22]	CNN	<i>IAM</i> , and <i>KHATT</i>	English, and Arabic	Handwritten text	80.72% ( <i>IAM</i> ) 68.90% ( <i>KHATT</i> )
[23]	CNN	<i>QUWI</i> , and a dataset of documents in Hebrew script	English, Arabic, and Hebrew	Handwritten text	N/A
[26]	<i>HOG</i> , and <i>SVM</i>	<i>IAM</i> , and <i>KHATT</i>	English, and Arabic	Handwritten text	75.45% ( <i>IAM</i> ) 68.89% ( <i>KHATT</i> )
[27]	ANN, and <i>SVM</i>	<i>QUWI</i> , and <i>MSHD</i>	English, French, and Arabic	Handwritten text	68.75% ( <i>QUWI</i> ) 73.02% ( <i>MSHD</i> )
[28]	PFSA, <i>SVM</i> , and NN	<i>QUWI</i> , and <i>MSHD</i>	English, French, and Arabic	Handwritten text	79.30% ( <i>QUWI</i> ) 79.90% ( <i>MSHD</i> )
[30]	<i>WD-LBP</i> , and <i>SVM</i>	A subset of <i>QUWI</i>	English, and Arabic	Handwritten text	73.40%
[29]	<i>HOG</i> , <i>LBP</i> , and <i>SVM</i>	<i>IAM-OnDB</i>	English	Handwritten text	74%
[18]	CNN	<i>HEBIU</i>	Hebrew, and English	Handwritten text	82.89%
[31]	<i>SVM</i> , <i>LR</i> , and <i>kNN</i>	<i>ICDAR 2013</i>	English, and Arabic	Handwritten text	65.71%
[32]	CNN	<i>ICDAR 2013</i> , <i>IAM-OnDB</i> , and <i>KHATT</i>	English, and Arabic	Handwritten text	71.8% ( <i>ICDAR 2013</i> ), 76.1% ( <i>IAM</i> ), 74.1% ( <i>KHATT</i> )
<b>Proposed work</b>	<b>Image preprocessing, and CNN</b>	<b>Own dataset, and <i>IAM-OnDB</i></b>	<b>Turkish, and English</b>	<b>Handwritten text and numeral</b>	<b>74.46% (proprietary dataset) 68.11% (<i>IAM-OnDB</i>)</b>

Xue et al. [32] proposed *ATP-DenseNet*, an attention-based two-pathway Densely Connected Convolutional Neural Network to identify the gender of handwriting. More specifically, they proposed three models based on this architecture as follows: (i) *ATP-DenseNet-121*, (ii) *ATP-DenseNet-169*, and (iii) *ATP-DenseNet-201*. The proposed models were evaluated on three widely used datasets, namely, (i) *ICDAR 2013*, (ii) *IAM*, and (iii) *KHATT*. According to the experimental results, *ATP-DenseNet-169* obtained accuracy scores of 71.8%, 76.1%, and 74.1% on the *ICDAR 2013*, *IAM-OnDB*, and *KHATT* datasets, respectively.

Unlike the related work, the proposed model (i) utilizes both handwritten text and numeral while the related work utilizes only handwritten text, (ii) was finalized through an extensive task of optimization task, and (iii) to the best of our knowledge, is the only study that covers Turkish.

### 3. Material and Method

In this section, we describe the details of (i) how the used dataset was constructed and which preprocessing techniques were employed, (ii) the software stack used for the implementation of the proposed models, (iii) the proposed novel models for gender classification, and (iv) the employed evaluation metrics to evaluate the performance of the proposed models.

#### 3.1 Dataset Preparation

To the best of our knowledge, there does not exist a public handwritten Turkish text/numerals dataset that is labeled with the corresponding gender. Therefore, we constructed our own dataset through a designed form that was filled out by each volunteer. We have collected a total of 530 handwriting samples. Of these samples, 330 were male, 195 were female, and gender was not specified in 5 of them. Each form collected from volunteers is a single shape of paper that has a size of A4. While ordering the samples, a single type of research form was replicated from the same device was used. Each volunteer filled out the form with a single type of pen with a 0.9 mm 2B tip while sitting on a chair on a table. The parts that were required to be filled in handwriting in the form consist of two parts, namely, (i) text, and (ii) numbers, which were separated from each other by frames. The details regarding the information requested from the volunteers via the provided research form are listed in Table 2. Some sample handwritten text and numerals from the constructed dataset are presented in Figure 1, and Figure 2, respectively.

Table 2 The details regarding the information requested from the volunteers via the provided research form.

Information	Data Type	Values
Age range	Categorical	15 – 17, 18 – 21, 22 – 29, 30 – 50, and 50 +
Educational status	Categorical	primary education, secondary education, high school, associate degree, undergraduate, and graduate

Pijamalı hasta, yağiz şoföre cabucak gönderdi

Pijamalı hasta, yağiz şoföre cabucak gönderdi

Pijamalı hasta, yağiz şoföre cabucak gönderdi

Pijamalı hasta, yağiz şoföre cabucak gönderdi

Figure 1 Some handwritten text samples from the constructed dataset

0 1 2 3 4 5 6 7 8 9  
 0 1 2 3 4 5 6 7 8 9  
 0 2 2 3 4 5 6 7 8 9  
 0 1 2 3 4 5 6 7 8 9

Figure 2 Some handwritten numeral samples from the constructed dataset

Additionally, volunteers were asked to fill in the text and numbers in the frame with their handwriting in a single line. The sample text used in this study consists of a sentence containing all 29 letters in Turkish. Regarding the numerical data to be filled, the volunteers were asked to write all the numbers in a single line. In the next step, each form was given an id number according to the collection order. These forms were scanned by 600 DPI TA Triumph-Adler 4555i printer, in color mode and PDF, at a high resolution of 1654 x 2338. Each page in the PDF has been converted to PNG format and named according to their id numbers and folders according to gender. To separate the text and numbers of each image in the folder, the coordinates were determined and cut by an implemented Python script that employed *pdf2image* [33], *OpenCV* [34], and *pandas* [35], [36] libraries to this end. The information in the form (such as age, and gender) was labeled with the id values and turned into categorical data. Then, the data were categorized and folded, and made ready for the preprocessing steps thanks to another implemented Python script. The process of constructing and preparing the novel dataset to be ready to be yielded into the proposed neural network is presented in Figure 3.

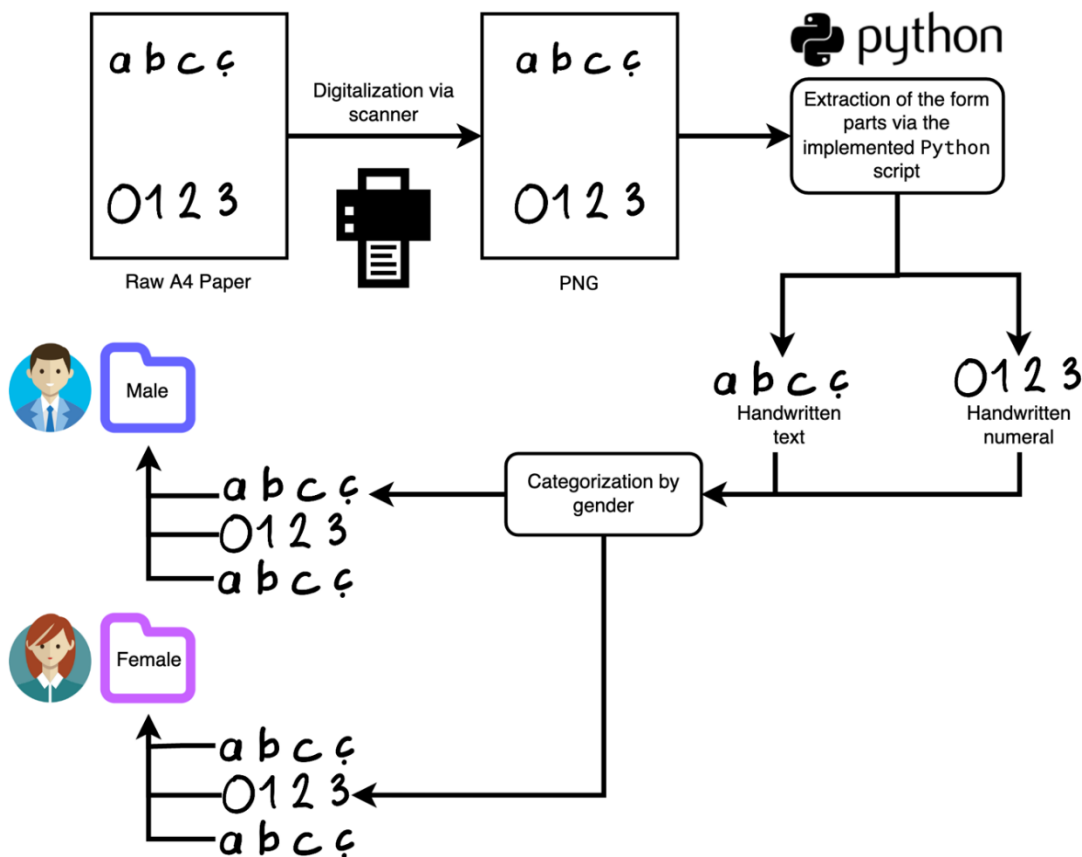


Figure 3 The process of constructing and preparing the novel dataset

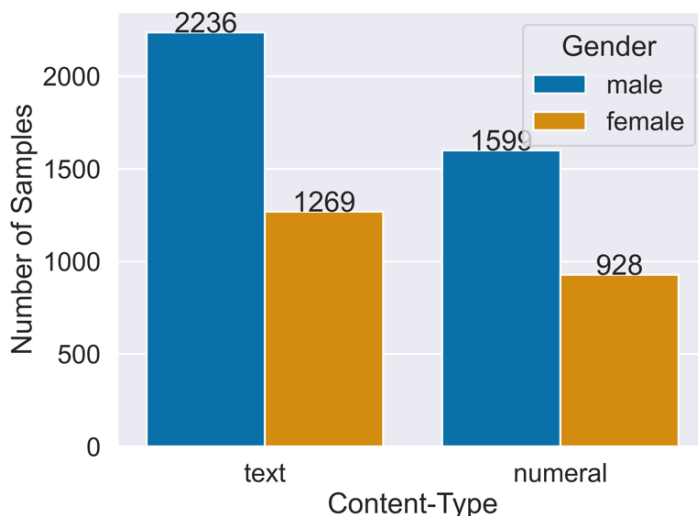


Figure 4 The distribution of collected handwriting samples by gender and content type

Similar to the approach by *Illouz et al.* [18], each handwriting sample was divided into square-shaped patches of  $100 \times 100$  pixels thanks to the implemented Python script as the widely-used pre-trained CNNs such as *ResNet50V2*, *InceptionV3*, and *MobileNetV2* do work with the square-shaped (e.g.,  $32 \times 32$ ,  $75 \times 75$ ,  $299 \times 299$ ) input, too. This operation also helped to keep the computational effort feasible. Then, each patch was mapped with the gender of the handwriting sample. Because of this process, the novel dataset was constructed. The distribution of the constructed dataset by gender and content type is presented in Figure 4.

Two rules were followed during the construction of the balanced dataset from the constructed dataset: (i) A handwritten text sample and a handwritten numerals sample were collected for each volunteer, and (ii) the same number of samples per gender were collected to construct a balanced dataset. Consequently, the constructed balanced dataset consisted of 928 handwritten text and 928 handwritten numerals per gender. The distribution of the constructed balanced dataset is given in Table 3.

Table 3 The distribution of the constructed balanced dataset

Gender	Text	Numerals	Total
Male	928	928	1,856
Female	928	928	1,856
Total	1,856	1,856	3,712

### 3.2 Software Stack

The entire software used for this study was implemented in the Python programming language and was powered by open-source technologies. *Keras* [37] was opted for the implementation of the proposed DNNs since being a high-level interface for the implementation of DNNs. Other advantages of *Keras* can be listed as follows: (i) seamless integration with other state-of-the-art data science frameworks, (ii) support to wide range of applications, including but not limited to computer vision, natural language processing, and plotting, and (iii) transferability as the models constructed using *Keras* can be easily transferred to various deep learning framework thanks to its modular design. The up-to-date version of *Keras* supports two deep learning backends, namely, (i) *TensorFlow* [38], and (ii) *Theano*. The selection of *TensorFlow* as *Keras*' backend arose from the developer's (*Keras*' creator) recommendation [39], owing to its ability to deliver high-performance and scalable capabilities. *NumPy* [40] and *pandas* [36], two widely-used Python libraries that *TensorFlow* depends on, were employed for the data manipulation and analysis of multi-dimensional matrices and numerical tables, respectively. In managing dataset operations such as partitioning into subsets based on the predefined ratio, data preprocessing tasks, and assessing the classification performance of the proposed models, a widely-used Python library, namely, *scikit-learn* [41] was employed. *Matplotlib* [42] was employed for the visualization of the experimental result. The details of the used software stack are listed in Table 4.

Table 4 The details of the used software stack

Software	Version
<i>Operating System</i>	<i>macOS Monterey 12.5</i>
<i>Python</i>	3.8.13
<i>Keras</i>	2.8.0
<i>TensorFlow</i>	2.8.0
<i>NumPy</i>	1.21.5
<i>pandas</i>	1.4.2
<i>scikit – learn</i>	1.0.2

### 3.3 Proposed Model

We propose two novel CNN models for the gender prediction problem through the given handwriting samples: (i) The first model makes gender prediction through the given both handwritten text and numerals, and (ii) the second model makes gender prediction through the given handwritten text. Each proposed novel CNN model is described in the following subsections.

#### 3.3.1 Proposed Two-Channel Model

The proposed two-channel model yields grayscale handwritten text and numerals in order to output its writer's gender into two classes, namely, (i) *male*, and (ii) *female*. To the best of our knowledge, this is the first gender prediction model that inputs handwritten numerals alongside handwritten text. The proposed two-channel model consisted of 21 layers as follows: Each channel is identical and consists of 9 layers. The model started with a *Convolutional* (denoted with *Conv2D*) layer, tasked with performing convolution operations on the provided input. This layer employed 16 filters and utilized a kernel size of (3,3). Then, a *Batch Normalization* (denoted with *Batch Norm.*) layer was employed to normalize the activations of

previous layers. Subsequently, a *Max Pooling* layer with a pool size of (2,2) was applied to gradually diminish the spatial dimensions of the representation, ultimately leading to a reduction in the network's parameter count and computational workload [43]. Following this *Max Pooling* layer, a *Dropout* [44] layer with a dropout rate of 0.5 was employed to randomly drop neurons from the network, which eventually helps to prevent the well-known problem of DNNs, namely, the “overfitting”. The second *Conv2D* layer with 32 filters and a kernel size of (5, 5) succeeded the *Dropout* layer. Similar to the first Convolutional layer, a *Batch Norm.*, a *Max Pooling* with a pool size of (2, 2), and a *Dropout* layer with a dropout rate of 0.5 succeeded the second *Conv2D* layer. As the final layer of each channel, a *Global Max Pooling* layer was utilized as the final layer, serving to consolidate activations across spatial locations and generate a vector of fixed size, which is common in several state-of-the-art CNNs [45]. A *Concatenation* layer was employed to concatenate the outputs of the channels. Another *Dropout* layer, but with a dropout rate of 0.6 followed the concatenation operation. Finally, a *Dense* layer, which is a deeply (fully) connected neural network component, with the sigmoid activation function was employed to output the predicted gender. The *Rectified Linear Unit (ReLU)* [46] was employed as the activation function of the *Conv2D* layers to avoid the vanishing gradient problem as a result of some other activation functions [47]. The default kernel initialization option of *Keras*, namely, *Glorot (a.k.a. Xavier) Uniform*, was employed as the kernel initializer of the employed *Conv2D* layers. Given that the handled problem is a binary classification task, the *Binary Cross-Entropy* was employed as the loss function of the model to calculate the loss after each epoch. The *Adadelta* [48] was employed as the optimization algorithm of the model to minimize the obtained loss by adjusting the attributes, namely, (i) weight, and (ii) bias. An overview of this model is presented in Figure 5. Table 5 lists each layer of the proposed two-channel model along with the hyperparameters that were utilized.

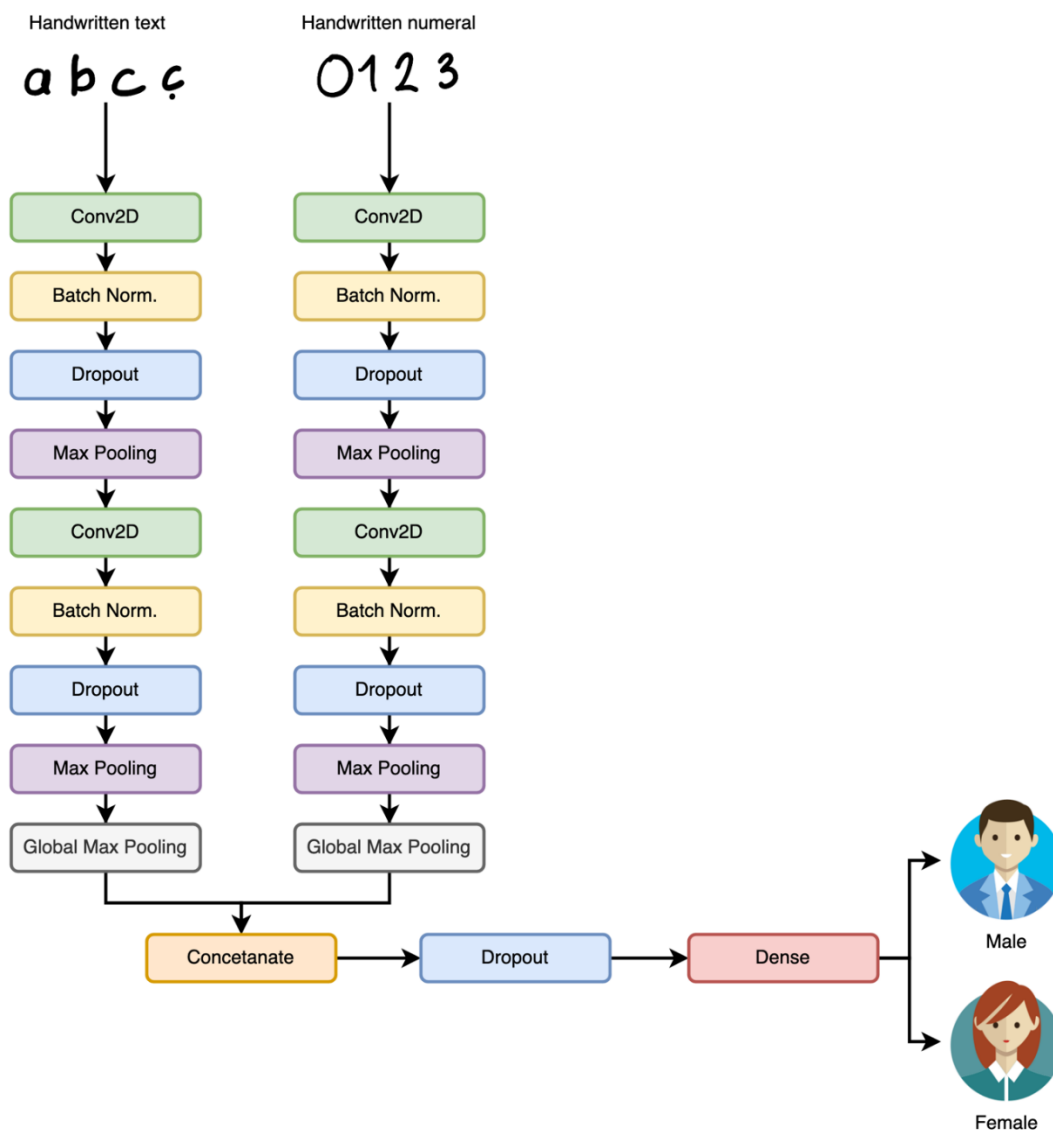


Figure 5 An overview of the proposed novel two-channel CNN model



Table 5 The layers of the proposed novel two-channel CNN model along with their corresponding hyperparameters

#	Layer Type	Hyperparameters
1	<i>Conv2D</i>	- Number of filters: 16 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: <i>Glorot Uniform</i> - Padding: <i>valid</i> - Activation function: <i>ReLU</i>
2	<i>Batch Norm.</i>	<i>N/A</i>
3	<i>Dropout</i>	- Dropout rate: 0.6
4	<i>Max Pooling</i>	- Pool size: (2, 2)
5	<i>Conv2D</i>	- Number of filters: 32 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: <i>Glorot Uniform</i> - Padding: <i>valid</i> - Activation function: <i>ReLU</i>
6	<i>Batch Norm.</i>	<i>N/A</i>
7	<i>Dropout</i>	- Dropout rate: 0.6
8	<i>Max Pooling</i>	- Pool size: (2, 2)
9	<i>Global Max Pooling</i>	<i>N/A</i>
10	<i>Conv2D</i>	- Number of filters: 16 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: <i>Glorot Uniform</i> - Padding: <i>valid</i> - Activation function: <i>ReLU</i>
11	<i>Batch Norm.</i>	<i>N/A</i>
12	<i>Dropout</i>	- Dropout rate: 0.6
13	<i>Max Pooling</i>	- Pool size: (2, 2)
14	<i>Conv2D</i>	- Number of filters: 32 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: <i>Glorot Uniform</i> - Padding: <i>valid</i> - Activation function: <i>ReLU</i>
15	<i>Batch Norm.</i>	<i>N/A</i>
16	<i>Dropout</i>	- Dropout rate: 0.6
17	<i>Max Pooling</i>	- Pool size: (2, 2)
18	<i>Global Max Pooling</i>	<i>N/A</i>
19	<i>Concatenate</i>	<i>N/A</i>
20	<i>Dropout</i>	- Dropout rate: 0.6
21	<i>Dense</i>	- Number of units: 1 - Activation function: <i>sigmoid</i>

### 3.3.2 Proposed Single-Channel Model

This model was intentionally proposed to be able to benchmark a CNN on a gold standard dataset as, to the best of our knowledge, there does not exist a handwritten gender dataset that contains both text and numeral. The proposed two-channel model consisted of 9 layers as follows: The model starts with a *Conv2D* layer with 32 filters and a kernel size of (3, 3). Then, a *Batch Norm.*, a *Dropout* layer with a dropout rate of 0.5, and a *Max Pooling* layer with a pool size of (2, 2) was employed, respectively. A second *Conv2D* layer followed this *Max Pooling* layer. Then, similar to the first *Conv2D* layer, a *Dropout* layer with a dropout rate of 0.5, and a *Max Pooling* layer with a pool size of (2, 2) was employed, respectively. Then, a *Global Max Pooling* layer was employed to aggregate the activations of spatial locations. Then, another *Dropout* layer with a dropout rate of 0.5 was employed. Finally, a *Dense* layer with a unit size of 1 and the *sigmoid* activation function was employed for the gender classification. Similar to the proposed two-channel model, (i) *ReLU* was employed as the activation function, (ii) the optimization of the model through the backpropagation was carried out on the *Adadelta* optimization algorithm, and (iii) the *Binary Cross-Entropy* was employed as the loss function of the model. An overview of this single-channel model is presented in Figure 6. Table 6 lists each layer of the proposed single-channel model along with the hyperparameters that were utilized.

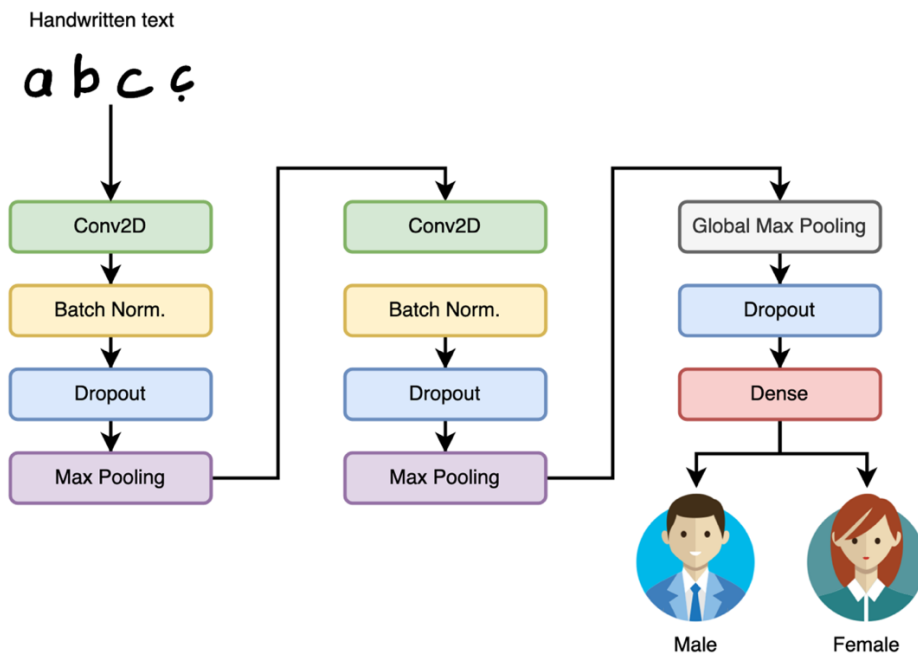


Figure 6 An overview of the proposed novel single-channel CNN model

Table 6 The layers of the proposed novel single-channel CNN model along with their corresponding hyperparameters

#	Layer Type	Hyperparameters
1	Conv2D	- Number of filters: 32 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: Glorot Uniform - Padding: valid - Activation function: ReLU
2	Batch Norm.	N/A
3	Dropout	- Dropout rate: 0.3
4	Max Pooling	- Pool size: (2, 2)
5	Conv2D	- Number of filters: 64 - Kernel size: (2, 2) - Strides: (1, 1) - Kernel initializer: Glorot Uniform - Padding: valid - Activation function: ReLU
6	Batch Norm.	N/A
7	Dropout	- Dropout rate: 0.3
8	Max Pooling	- Pool size: (2, 2)
9	Global Max Pooling	N/A
10	Dropout	- Dropout rate: 0.3

### 3.4 Evaluation Metrics

*De – facto* standard metrics to evaluate the performance of classifiers, namely, *accuracy*, *precision*, *recall* (a.k.a. *sensitivity*), and *F1 – score* were employed to assess the classification performance of the proposed model. Let *P* denote *positives*, referring to the samples labeled with the target class, and *N* represent *negatives*, signifying the samples labeled with the complementary class of the target. *TP*, *TN*, *FP*, and *FN* denote correctly predicted *positives*, correctly predicted *negatives*, *positives* incorrectly predicted as *negative*, and *negatives* incorrectly predicted as *positive*, respectively. *Accuracy* is the proportion of the correctly predicted samples to all samples. *Precision* is defined as the proportion of accurately predicted positive instances to the total number of instances predicted as positive. *Recall* is the ratio of correctly predicted

positive instances to the total number of actual positive instances.  $F1 - score$  is the harmonic mean of the  $precision$  and  $recall$  and is more useful than accuracy when the used dataset is imbalanced. The equations of  $accuracy$ ,  $precision$ ,  $recall$ , and  $F1 - score$  are given in Eq. 1.

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{P + N} \\
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN} \\
 F1 - score &= 2 \times \frac{Precision \times Recall}{Precision + Recall}
 \end{aligned} \tag{1}$$

When it comes to the evaluation of the proposed model, we employed a confusion matrix, which is a specific table that visualizes the classification performance of classifiers. In the confusion matrix, every row corresponds to the count of samples in the true class, and each column corresponds to the count of samples in the predicted class.

#### 4. Experimental Results and Discussion

In the following subsections, the hyperparameter optimization, the training and evaluation of the proposed models, and the discussion in the light of experimental results are described.

##### 4.1 Hyperparameter Optimization

Hyperparameters are the parameters of a DNN model that impact the learning process and are determined through empirical tuning [39], [49]. More specifically, the hyperparameter optimization task provides various improvements to the neural network such as performance enhancement, generalization improvement, faster convergence, resource efficiency, robustness, and exploratory analysis. Both proposed models were trained under the same hyperparameters which were finalized as a result of automatized hyperparameter optimization task. Throughout this task, an extensive set of values for each hyperparameter was assessed to uncover the optimal combination of hyperparameters as they are listed in Table 7, where the obtained best value for each hyperparameter is given in bold. Despite encompassing a wide range of hyperparameters, the employed hyperparameter optimization task remained efficient and streamlined due to its automated nature. To be more specific, the proposed models utilized a widely-recognized technique known as *Hyperband* [50] as the optimization algorithm. The optimization objective was defined as the accuracy achieved on the validation set. A subset of 20% from the training set was allocated for use as the validation set. As listed in Table 7, several widely-used optimization algorithms, namely, (i) *Adaptive Moment Estimation (Adam)* [51], (ii) *Root Mean Square Propagation (RMSprop)* [52], (iii) *Stochastic Gradient Descent (SGD)* [53], and (iv) *Adadelta* were evaluated as the optimization algorithms. The *Adadelta* was employed as the optimization of the proposed model as a result of the employed hyperparameter optimization. Several widely-used activation functions, namely, (i) *Rectified Linear Unit (ReLU)*, (ii) *Exponential Linear Unit (eLU)*, (iii) *Parametric ReLU (PReLU)*, (iv) *Leaky ReLU*, (v) *tanh*, and (vi) *softmax* were evaluated as the activation functions. The *ReLU* was employed as the activation algorithm of the proposed models as a result of the employed hyperparameter optimization. Regarding the *dropout rate*, an assessment encompassed a set of 0.2, 0.3, 0.4, 0.5, and 0.6. The range of 3 to 10 was scrutinized for the number of folds ( $k$  value). Additionally, evaluations were conducted for both *kernel regularization penalty* and *bias regularization penalty*, considering the set of  $1xe^{-5}$ ,  $1xe^{-6}$ ,  $1xe^{-7}$ , and  $1xe^{-8}$ . The *batch size* underwent evaluation using the set of 16, 32, 64, 128, and 256.

Table 7 The evaluated values of the employed hyperparameters during the optimization. The obtained best values were given in bold

Model	Evaluated Values
<i>Dropout rate for Conv. layers</i>	0.2, 0.3, 0.4, 0.5, <b>0.6</b>
<i>Dropout rate for the Dense layer prior to final</i>	0.2, 0.3, 0.4, 0.5, <b>0.6</b>
<i>Activation function</i>	<b>ReLU</b> , eLU, PReLU, Leaky ReLU, tanh, softmax
<i>Optimization algorithm</i>	Adam, RMSprop, SGD, <b>Adadelta</b>
<i>Kernel size</i>	<b>3</b> , 5, 7, 9
<i>Kernel regularization penalty</i>	$1xe^{-5}$ , $1xe^{-6}$ , <b><math>1xe^{-7}</math></b> , $1xe^{-8}$
<i>Bias regularization penalty</i>	$1xe^{-5}$ , $1xe^{-6}$ , <b><math>1xe^{-7}</math></b> , $1xe^{-8}$
<i>Learning rate</i>	<b><math>1xe^{-3}</math></b> , $5xe^{-3}$ , $1xe^{-4}$ , $1xe^{-5}$ , $1xe^{-6}$
<i>Batch size</i>	16, 32, <b>64</b> , 128, 256
<i>Number of folds</i>	3, 4, <b>5</b> , 6, 7, 8, 9, 10

## 4.2 Model Training and Evaluation

The training setup of a neural network is crucial to ensure that the network is optimized for achieving its highest performance potential. Its proper design and configuration influence every aspect of neural network's performance, from accuracy and efficiency to generalization and robustness. A properly configured training can make the difference between a neural network that struggles to learn and one that excels at its intended task. A well-trained NN should neither overfit nor underfit. In the following subsections, the training and evaluation of the proposed models are described.

### 4.2.1 Two-Channel Model

The training of each proposed model was started with the *Early Stopping* callback, which is responsible for stopping the training when the model stops learning. In pursuit of this objective, two parameters are utilized in the following manner: (i) The monitored criterion, and (ii) the number of epochs that the callback waits before cessation, also known as "*patience*." Specifically, the achieved validation loss was designated as the monitored criterion, and a patience value of 10 epochs was established.

A subset comprising 20% of the entire dataset, totaling 372 samples, was set aside as the test set. This subset was employed to evaluate the classification performance of the proposed model. The remaining dataset was employed for both training and validation using the *Stratified k-Fold Cross Validation* technique, a specialized form of *k-Fold Cross Validation* that divides the entire dataset into  $k$  folds while maintaining the proportional distribution of samples for each class. The value of  $k$  was determined as 5 based on experimental results from the employed hyperparameter optimization. This indicates that the training set was divided into 5 folds, with the initial fold serving as the validation set and the remaining 4 folds being utilized for training purposes. This process was repeated 5 times to utilize the entire dataset for both training and validation purposes. Under this configuration, the training of the two-channel model was continued for 97 epochs until the employed *Early Stopping* callback stopped the training. An accuracy as high as 74.46% was obtained on the test set. The accuracy values achieved for both the training and validation sets during the training of the proposed two-channel model were plotted in Figure 7. According to this experimental result, it is safe to conclude that the model neither overfit nor underfit. The obtained confusion matrix for evaluating the test set is presented in Figure 8.

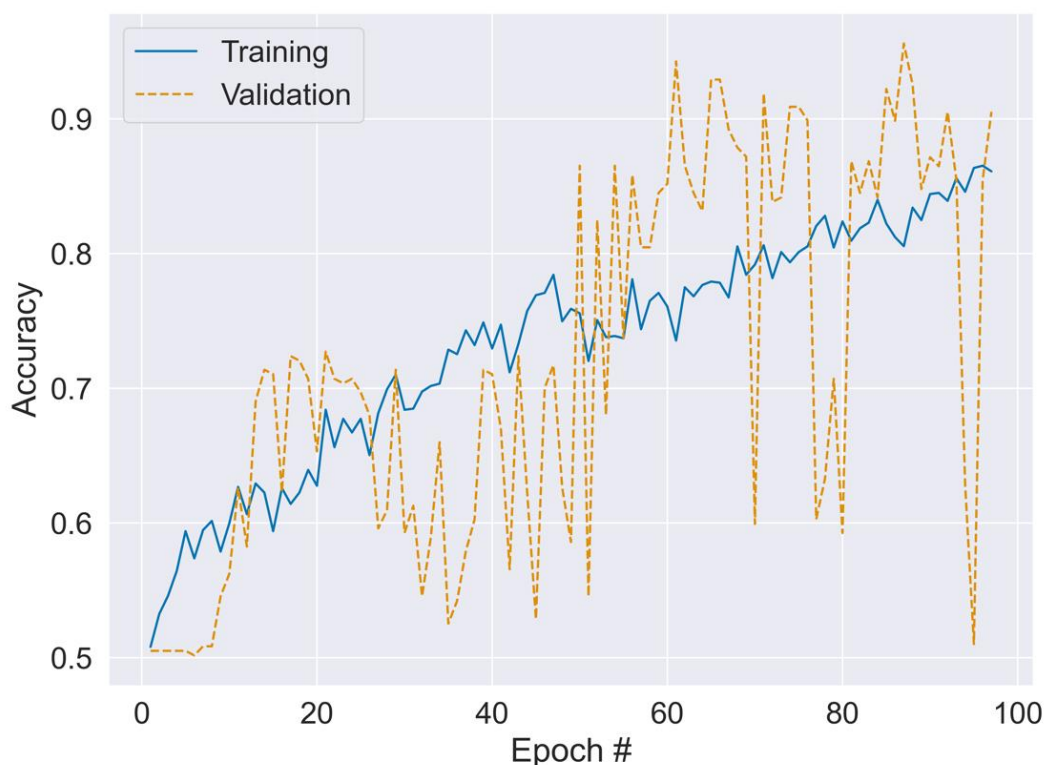


Figure 7 The accuracy values obtained for both the training and validation sets during the training process of the proposed two-channel model.

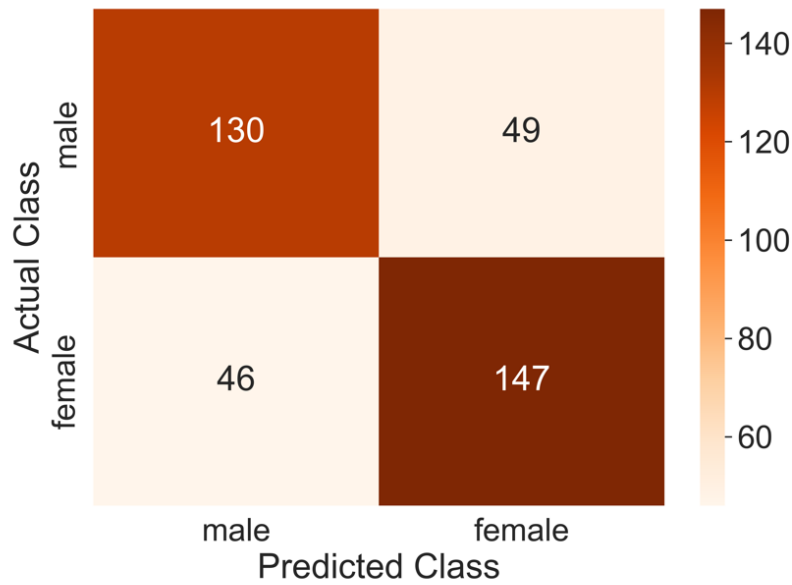


Figure 8 The obtained confusion matrix of the proposed two-channel model upon evaluation on the test set

#### 4.2.2 Single-Channel Model

Under the same training configuration as the two-channel model, the single-channel model had been trained for 74 epochs until the employed *Early Stopping* callback stopped the training. An accuracy as high as 72.33% was obtained on the test set. The accuracy values achieved for both training and validation sets during the training of the proposed two-channel model were plotted in Figure 9. According to this experimental result, it is safe to conclude that the model neither overfit nor underfit. The obtained confusion matrix for evaluating the test set is presented in Figure 10.

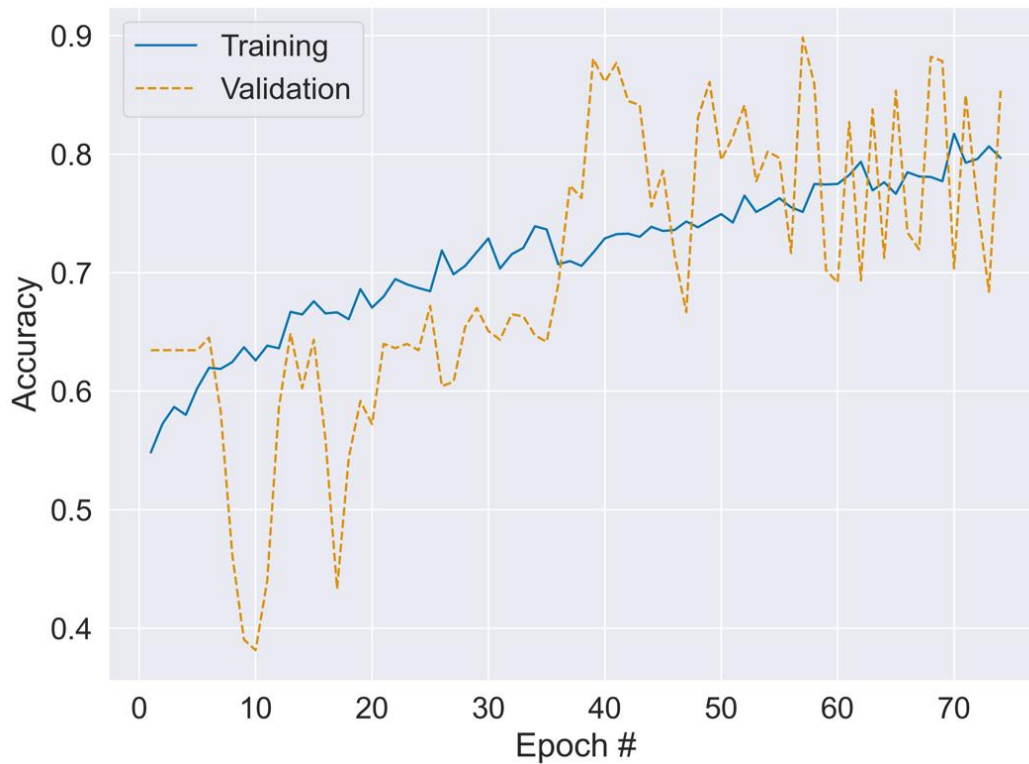


Figure 9 The accuracy values obtained for both the training and validation sets during the training process of the proposed single-channel model.

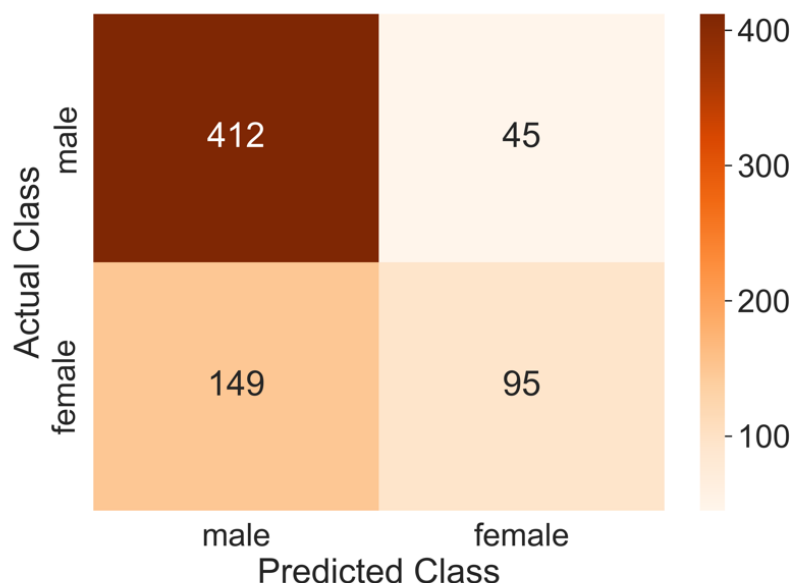


Figure 10 The obtained confusion matrix of the proposed single-channel model upon evaluation on the test set

We have also experimented with gender detection through the numerals only using the same single-channel model. This time, an accuracy of 64.03% was obtained. Alongside the proposed DNNs, (1) the widely-used traditional ML algorithms, namely, (i) *SVM*, (ii) *Logistic Regression*, (iii) *Naïve Bayes*, (iv) *Random Forest*, (v) *Decision Tree*, (vi) *k-Nearest Neighbors*, (vii) *Light Gradient Boosting Machine (LGBM)*, and (viii) *eXtreme Gradient Boosting (XGBoost)*, and (2) the widely-used pre-trained DNNs, namely, (i) *ResNet50V2* [54], (ii) *InceptionV3* [55], and (iii) *MobileNetV2* [56] through the transfer-learning were employed. For the pre-trained DNNs, each channel was replaced with the pre-trained DNN, including the weights calculated for the *ImageNet* dataset. The same layers of the proposed two-channel CNN were applied after the concatenation. Similarly, the pre-trained DNNs were trained under the same hyperparameters. As a final model, the proposed two-channel CNN was employed as the feature extractor, and *SVM* was employed as the classifier. According to the experimental result, the proposed two-channel CNN that yields both handwritten text and numerals provided the best accuracy among all models. The proposed single-channel model that yields handwritten text followed that. This experimental result demonstrates that yielding both handwritten text and numerals provides better accuracy than yielding only handwritten text or numerals. Another conclusion in the light of the experimental results is that the DNN models provided better accuracy than the traditional ML models for gender classification through handwriting. The obtained classification accuracy scores of the employed traditional ML algorithms and proposed CNN models on the test set of the novel dataset are listed in Table 8.

Table 8 The obtained classification accuracy scores of the employed traditional ML algorithms and proposed CNN models on the test set of the novel dataset

Model	Accuracy
<i>SVM</i> (Text)	53.49%
<i>Logistic Regression</i> (Text)	51.88%
<i>Naïve Bayes</i> (Text)	51.08%
<i>Random Forest</i> (Text)	58.33%
<i>Decision Tree</i> (Text)	52.42%
<i>kNN</i> ( $k=2$ ) (Text)	52.96%
<i>LGBM</i> (Text)	55.11%
<i>XGBoost</i> (Text)	56.45%
<i>ResNet50V2</i> (Text and Numeral)	47.58%
<i>InceptionV3</i> (Text and Numeral)	50.54%
<i>MobileNetV2</i> (Text and Numeral)	51.88%
Proposed Single-channel CNN (Text)	72.33%
Proposed Single-channel CNN (Numeral)	64.03%
<b>Proposed Two-channel CNN (Text and Numeral)</b>	<b>74.46%</b>
Proposed Two-channel CNN-SVM (kernel= <i>linear</i> ) (Text and Numeral)	68.55%
Proposed Two-channel CNN-SVM (kernel= <i>rbf</i> ) (Text and Numeral)	70.70%
Proposed Two-channel CNN-SVM (kernel= <i>poly</i> ) (Text and Numeral)	71.51%

## 5. Conclusion

Gender is one of the most distinguishing characteristics of human beings. Drawing is the oldest communication tool of human beings and consists of a lot of information about its owner. From this point forth, we have proposed gender classification models through the given handwriting samples, which can be text, numerals, or both text and numerals. The proposed models were based on CNNs, which have provided the state-of-the-art for many classification problems whether it can be text classification or image classification. To the best of our knowledge, there does not exist a public Turkish handwriting dataset, which is labeled with genders. Therefore, we constructed our own dataset, which consists of 530 handwriting samples. The proposed CNN models were trained and evaluated on this dataset. According to the experimental result, the best accuracy, an accuracy as high as 74.46%, was obtained by the proposed CNN model that yields both handwriting text and numerals. The obtained accuracy is higher than the compared state-of-the-art techniques and is promising on such a task that even humans could not have achieved highly-accurate results for, as of yet. This experimental result demonstrates that yielding both handwriting text and numerals provides better accuracy compared to yielding only handwriting text or numerals. Another key finding in the light of the experimental result is that the models based on CNNs provided better accuracy compared to the models that employ the traditional ML algorithms as well as the combination of CNN and SVM.

As a future work, the authors would like to extend the constructed dataset by combining it with other datasets to improve the learning ability of the proposed model.

## References

- [1] J. Wayman, A. Jain, D. Maltoni, and D. Maio, *Biometric Systems*, 1st ed. London: Springer London, 2005. doi: 10.1007/b138151.
- [2] R. N. King and D. J. Koehler, "Illusory correlations in graphological inference," *J Exp Psychol Appl*, vol. 6, no. 4, pp. 336–336, 2000, doi: 10.1037/1076-898X.6.4.336.
- [3] V. Shackleton and S. Newell, "European Management Selection Methods: A Comparison of Five Countries," *International Journal of Selection and Assessment*, vol. 2, no. 2, pp. 91–102, 1994, doi: 10.1111/j.1468-2389.1994.tb00155.x.
- [4] M. Ahmed, A. G. Rasool, H. Afzal, and I. Siddiqi, "Improving handwriting based gender classification using ensemble classifiers," *Expert Syst Appl*, vol. 85, pp. 158–168, 2017, doi: 10.1016/j.eswa.2017.05.033.
- [5] N. Bouadjenek, H. Nemmour, and Y. Chibani, "Local Descriptors to Improve Off-line Handwriting-based Gender Prediction," in *Proceedings of the 2014 6th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2014)*, Tunis, Tunisia: IEEE, pp. 43–47, 2014. doi: 10.1109/SOCPAR.2014.7007979.
- [6] N. Bouadjenek, H. Nemmour, and Y. Chibani, "Age, Gender and Handedness Prediction from Handwriting using Gradient Features," in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR '15)*, Washington, DC, United States: IEEE, 2015, pp. 1116–1120. doi: 10.1109/ICDAR.2015.7333934.
- [7] "What is gender? What is sex?," *Canadian Institutes of Health Research*, 2020 [Online]. Available: <https://cihr-irsc.gc.ca/e/48642.html>. [Accessed: 10-Aug-2022].
- [8] L. A. M. Galea and D. Kimura, "Sex differences in route-learning," *Pers Individ Dif*, vol. 14, no. 1, pp. 53–65, 1993, doi: 10.1016/0191-8869(93)90174-2.
- [9] E. Coluccia, G. Iosue, and M. Antonella Brandimonte, "The relationship between map drawing and spatial orientation abilities: A study of gender differences," *J Environ Psychol*, vol. 27, no. 2, pp. 135–144, 2007, doi: 10.1016/j.jenvp.2006.12.005.
- [10] R. S. Astur, A. J. Purton, M. J. Zaniwski, J. Cimadevilla, and E. J. Markus, "Human sex differences in solving a virtual navigation problem," *Behavioural Brain Research*, vol. 308, pp. 236–243, 2016, doi: 10.1016/j.bbr.2016.04.037.
- [11] D. A. Cahn-Weiner, K. Williams, J. Grace, G. Tremont, H. Westervelt, and R. A. Stern, "Discrimination of Dementia with Lewy bodies from Alzheimer disease and Parkinson disease using the Clock Drawing Test," *Cognitive and Behavioral Neurology*, vol. 16, no. 2, pp. 85–92, 2003, doi: 10.1097/00146965-200306000-00001.
- [12] K. Amunts et al., "Gender-specific left-right asymmetries in human visual cortex," *Journal of Neuroscience*, vol. 27, no. 6, pp. 1356–1364, 2007, doi: 10.1523/JNEUROSCI.4753-06.2007.
- [13] A. C. Hurlbert and Y. Ling, "Biological components of sex differences in color preference," *Current Biology*, vol. 17, no. 16, pp. 623–625, 2007, doi: 10.1016/j.cub.2007.06.022.
- [14] S. Bradley, "Handwriting and Gender: A multi-use data set," *Journal of Statistics Education*, vol. 23, no. 1, pp. 1–15, 2015, doi: 10.1080/10691898.2015.11889721.
- [15] G. Cordasco, M. Buonanno, M. Faundez-Zanuy, M. T. Riviello, L. Likforman-Sulem, and A. Esposito, "Gender Identification through Handwriting: an Online Approach," in *Proceedings of the 11th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2020)*, Mariehamn, Finland: IEEE, pp. 197–202, 2020. doi: 10.1109/CogInfoCom50765.2020.9237863.
- [16] S. Saha, M. Asif, B. Khaled, M. S. Islam, N. Saha Puja, and M. Hasan, "Detecting Sex From Handwritten

- Examples; Detecting Sex From Handwritten Examples,” in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India: IEEE, pp. 1–7, 2018.
- [17] S. Upadhyay, J. Singh, and S. K. Shukla, “Determination of Sex Through Handwriting Characteristics,” *Int J Curr Res Rev*, vol. 9, no. 13, pp. 11–18, 2017, doi: 10.7324/ijcrr.2017.9133.
- [18] E. Illouz, E. Omid David, and N. S. Netanyahu, “Handwriting-Based Gender Classification Using End-to-End Deep Neural Networks,” in *Proceedings of the 27th International Conference on Artificial Neural Networks (ICANN 2018)*, Island of Rhodes, Greece: Springer Verlag, pp. 613–621, 2018. doi: 10.1007/978-3-030-01424-7\_60.
- [19] M. Koppel, S. Argamon, and A. R. Shimoni, “Automatically Categorizing Written Texts by Author Gender,” *Literary and Linguistic Computing*, vol. 17, no. 4, pp. 1–13, 2002, doi: 10.1093/lc/17.4.401.
- [20] M. Liwicki, A. Schlappbach, P. Loretan, and H. Bunke, “Automatic Detection of Gender and Handedness from On-Line Handwriting,” in *Proceedings of the 13th Biennial Conference of the International Graphonomics Society (IGS2007)*, Melbourne, Australia, pp. 179–183, 2007.
- [21] A. Gattal, C. Djeddi, A. Bensefia, and A. Ennaji, “Handwriting Based Gender Classification Using COLD and Hinge Features,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, pp. 233–242, 2020. doi: 10.1007/978-3-030-51935-3\_25.
- [22] Á. Morera, Á. Sánchez, J. F. Vélez, and A. B. Moreno, “Gender and Handedness Prediction from Offline Handwriting Using Convolutional Neural Networks,” *Complexity*, vol. 2018, pp. 1–14, 2018, doi: 10.1155/2018/3891624.
- [23] I. Rabaev, M. Litvak, S. Asulin, and O. H. Tabibi, “Automatic Gender Classification from Handwritten Images: A Case Study,” in *Proceedings of the 19th International Conference on Computer Analysis of Images and Patterns (CAIP 2021)*, Virtual, 2021, pp. 329–339, 2021. doi: 10.1007/978-3-030-89131-2\_30.
- [24] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proceeding of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, FL, USA: IEEE, pp. 248–255, 2009. doi: 10.1109/cvpr.2009.5206848.
- [25] S. Al Maadeed and A. Hassaine, “Automatic prediction of age, gender, and nationality in offline handwriting,” *EURASIP J Image Video Process*, vol. 2014, no. 10, pp. 1–10, 2014, doi: 10.1186/1687-5281-2014-10.
- [26] N. Bouadjenek, H. Nemmour, and Y. Chibani, “Histogram of Oriented Gradients for Writer’s Gender, Handedness and Age prediction,” in *Proceedings of the 2015 International Symposium on Innovations in Intelligent Systems and Applications, Proceedings (INISTA 2015)*, pp. 1–5, 2015. doi: 10.1109/INISTA.2015.7276752.
- [27] I. Siddiqi, C. Djeddi, A. Raza, and L. Souici-meslati, “Automatic analysis of handwriting for gender classification,” *Pattern Analysis and Applications*, vol. 18, pp. 887–899, 2015, doi: 10.1007/s10044-014-0371-0.
- [28] Y. Akbari, K. Nouri, J. Sadri, C. Djeddi, and I. Siddiqi, “Wavelet-based gender detection on off-line handwritten documents using probabilistic finite state automata,” *Image Vis Comput*, vol. 59, no. C, pp. 17–30, 2017, doi: 10.1016/j.imavis.2016.11.017.
- [29] N. Bouadjenek, H. Nemmour, and Y. Chibani, “Writer’s Gender Classification Using HOG and LBP Features,” in *Proceedings of the International Conference on Electrical Engineering and Control Applications (ICEECA 2016)*, Kuala Lumpur, Malaysia, pp. 1–5, 2016. doi: 10.1007/978-3-319-48929-2\_24.
- [30] A. E. Youssef, A. S. Ibrahim, and A. Lynn Abbott, “Automated Gender Identification for Arabic and English Handwriting,” in *Proceedings of the 5th International Conference on Imaging for Crime Detection and Prevention (ICDP 2013)*, London, UK, pp. 1–6. Doi, 2013: 10.1049/ic.2013.0274.
- [31] P. Maken and A. Gupta, “A method for automatic classification of gender based on text- independent handwriting,” *Multimed Tools Appl*, vol. 80, no. 16, pp. 24573–24602, Jul. 2021, doi: 10.1007/s11042-021-10837-9.
- [32] G. Xue, S. Liu, D. Gong, and Y. Ma, “ATP-DenseNet: a hybrid deep learning-based gender identification of handwriting,” *Neural Comput Appl*, vol. 33, pp. 4611–4622, May 2021, doi: 10.1007/s00521-020-05237-3.
- [33] E. Belval, “pdf2image: A python (3.6+) module that wraps pdftoppm and pdftocairo to convert PDF to a PIL Image object,” 2021. [Online]. Available: <https://github.com/Belval/pdf2image> [Accessed: 10-Aug-2022].
- [34] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, vol. 120, pp. 122–125, 2000.
- [35] The pandas development team, “pandas: Python Data Analysis Library,” 2020. [Online]. Available: <https://pandas.pydata.org>. [Accessed: 12-Jul-2022].
- [36] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference (SCIPY 2010)*, Austin, Texas, pp. 56–61, 2010. doi: 10.25080/majora-92bf1922-00a.
- [37] F. Chollet, “Keras: the Python deep learning API,” 2015. [Online]. Available: <https://keras>. [Accessed: 2-Dec-2022].
- [38] M. Abadi et al., “TensorFlow: A System for Large-Scale Machine Learning,” in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016)*, Savannah, GA, USA, pp. 265–283, 2016.
- [39] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.
- [40] C. R. Harris et al., “Array Programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
- [41] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12,



- pp. 2825–2830, 2011.
- [42] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Comput Sci Eng*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.
- [43] “CS231n Convolutional Neural Networks for Visual Recognition,” *Stanford University*, 2020. <https://cs231n.github.io/convolutional-networks> (accessed Dec. 02, 2022).
- [44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [45] V. Christlein, L. Spranger, M. Seuret, A. Nicolaou, P. Král, and A. Maier, “Deep Generalized Max Pooling,” in *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR 2019)*, Sydney, Australia, pp. 1–7, 2019.
- [46] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, Madison, WI, USA, pp. 807–814, 2010.
- [47] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV 2016)*, Lake Placid, NY, USA: IEEE, pp. 1–10, 2016. doi: 10.1109/WACV.2016.7477450.
- [48] D. C. Johnny *et al.*, “ADADELTA: An Adaptive Learning Rate Method,” *IEEE Access*, vol. 7, no. November, 2018.
- [49] X. Zhang, X. Chen, L. Yao, C. Ge, and M. Dong, “Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning,” in *International Conference on Neural Information Processing (ICONIP 2019)*, Sydney, NSW, Australia: Springer, pp. 287–295, 2019. doi: 10.1007/978-3-030-36808-1\_31.
- [50] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2018.
- [51] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, California, USA, pp. 1–15, 2015.
- [52] G. Hinton, N. Srivastava, and K. Swersky, “Neural Networks for Machine Learning,” 2012.
- [53] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951, doi: 10.1214/aoms/1177729586.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, Amsterdam, The Netherlands, Oct. 2016. doi: 10.1007/978-3-319-46493-0\_38.
- [55] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Seattle, Washington, USA: IEEE, pp. 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.
- [56] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, Salt Lake City, UT, USA: IEEE, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.

### Acknowledgments

The authors would like to thank a large number of anonymous subjects from Duzce University who voluntarily participated in this study by providing the handwriting samples.

### Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

### Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

### Availability of data and material

Not applicable

### Plagiarism Statement

This article has been scanned by iThenticate™.



# Conjoint Analysis of GPS-Based Orbit Determination among Traditional Methods

İbrahim Öz<sup>1</sup> , Cevat Özarpa<sup>2</sup> 

<sup>1</sup> Ankara Yıldırım Beyazıt University, Technology Transfer Office; Ankara, Türkiye

<sup>2</sup> University of Karabük, Mechanical Engineering Department; Karabük, Türkiye



## Corresponding author:

İbrahim Öz, Ankara Yıldırım Beyazıt University, Technology Transfer Office; Ankara, Türkiye  
E-mail address:  
[ibrahimoz@gazi.edu.tr](mailto:ibrahimoz@gazi.edu.tr)

Submitted: 9 December 2022  
Revision Requested: 1 September 2023  
Last Revision Received: 13 September 2023  
Accepted: 30 September 2023  
Published Online: 30 September 2023

Citation: Öz İ. Özarpa C. (2023). Conjoint Analysis of GPS-Based Orbit Determination among Traditional Methods. *Sakarya University Journal of Computer and Information Sciences*. 6 (3) <https://doi.org/10.35377/saucis...1215689>

## ABSTRACT

This Satellite operators rely on accurate satellite orbit estimation to ensure safe orbital operations, considering the influence of external forces. Traditional methods, such as single station angles and range (AZEL), along with range-to-range (RNG) techniques, have been widely employed by operators. However, the use of GPS signals for determining the orbits of geostationary communication satellites (GEO) has gained popularity due to its effectiveness. Extensive research has validated the reliability and efficiency of GPS-based GEO orbit determination. In this study, the performance of the GPS-based method is evaluated by comparing it with flight-proven techniques. Three GEO communication satellites located at different longitudes were analyzed using GPS-based, RNG-based, and AZEL-based methods. The results indicated that the GPS-based determined orbit had a root mean square error (RMSE) of 75.887 m, 372.420 m, and 768.223 m for Satellites A, B, and C, respectively, when compared with the RNG-based determined orbit. Similarly, the RMSE between the GPS-based and AZEL-based determined orbits was 133.287 m, 242.076 m, and 764.866 m for Satellites A, B, and C, respectively. These findings strongly support using GPS-based orbit determination, as it aligns with the results obtained from flight-proven RNG and AZEL methods. The study demonstrates the reliability and accuracy of the GPS-based orbit estimation method. Consequently, it encourages satellite operators to adopt GPS-based navigation for precise determination of communication satellite orbits. The comparison between AZEL vs. GPS and RNG vs. GPS methods reinforces the advantages of utilizing GPS-based navigation.

**Keywords:** Orbit determination, GPS based orbit, navigation, GEO orbit

## 1. Introduction

This Geostationary (GEO) satellites seem fixed from the Earth; however, satellite orbit motion deviates from theoretical orbital motion due to perturbing forces. Those are the gravitational forces of the sun and the moon, the earth's non-uniform mass distribution, solar pressure, and other small forces. Maneuvers balance perturbing forces act on a GEO satellite [1]– [3]. It is mandatory to determine the orbit of a satellite for operators. There are various types of data collection, observation, and orbit determination methods for orbit estimation; subsequently, the orbit has always been subject to change due to external forces. The two most common ways of orbit determination are range measurement, bi-static range measurement from two ground stations, and single station tracking based on azimuth elevation [4], [5]. Global Positioning System (GPS) is becoming a gorgeous method for the orbit determination of GEO satellites. However, GEO satellite operators mainly use traditional ground-based measurement systems for orbit determination [6]. In GPS-based orbit estimation, the data acquisition system is inside the GPS in low earth orbit (LEO) and ground receiver cases. Nevertheless, in the GEO satellite case, the orbit is beyond the GPS constellation. GPS satellites' altitudes (~22000km) are lower than GEO satellite's altitude (~35786 km), and the earth shadows the GPS signals most of the time. However, utilizing GPS signals for accurate orbit estimation of GEO satellites is still promising.

GPS is a satellite navigation system that can provide highly accurate position and timing information in all weather conditions worldwide. The onboard satellite GPS receiver calculates the pseudo-range distance between the GEO satellite (user) and the recognized GPS satellite. The GPS signal is subject to factors that degrade signal quality and cause GEO satellite position inaccuracies, such as clock errors, multipath propagation, ephemeris uncertainty, and ionosphere and troposphere delay. The number of visible GPS satellites and satellite geometry from the user's point of view also affect the accuracy [7].



There are many studies on GPS-based orbit determination in different aspects. According to some researchers, real-time onboard GPS orbit determination was developed to provide a very accurate orbit. The reliability of uncertainty was one of the essential parameters in orbit determination (OD). The characterization of GPS uncertainty was analyzed in different aspects. The uncertainties were analyzed, and the effect of factors was estimated for the GEO orbit [8, 9].

GPS-based orbit determination of GEO satellites is becoming an attractive approach. The accuracy of GNSS systems was studied, such as the GEO satellite called JS-2 equipped with high gain GNSS antenna, amplifiers, and high sensitivity receivers. Weak GPS signals and the onboard orbit determination filters were investigated to improve OD performance. The analysis of carrier-to-noise ratio density (C/N0), position dilution of precision (PDOP), availability of signal, and characteristics result in excellent OD performance [10, 11].

There are various articles about GEO satellite orbit determination using GPS receivers. GEO orbit determination accuracy is about 20 m, according to the GPS receiver and orbital filter performance assessment study. The precision requirements of GEO satellites were identified with a simulator of GPS signals and a single-frequency receiver.

A European project demonstrated an on-board receiver that acquire weak signal to increase the number of visible GPS satellites. Flight performance was demonstrated by signal processing and onboard orbit determination.

GPS-based navigation for lunar missions is an emerging field with several publications. GNSS flight experiments show beneficial results for lunar navigation applications [11].

The GEO orbit is used mainly for telecommunication purposes and is unique. GPS-based navigation methods offer some advantages over ground-based methods. Capuano Vincenzo et al. studied the best GNSS signal for GEO navigation and achieved reliable performances [12].

Jun Zhu et al. investigated GPS-based navigation performance for GEO satellite telecommunication to determine the signal quality effect on OD. The results provide sub-meter-level precision [13].

The Extended Kalman Filter (EKF) method was developed for real-time and onboard OD by Chiaradia Ana et al. They analyzed the model's accuracy and performed simple and relatively accurate orbit determination. The obtained velocity and position errors vary in a reasonable range along a day [14, 15].

Researchers established and analyzed GPS and GEO-based integrated networks to find a GPS receiver's user position or position coordinates in another work. They developed a new approach for determining the minimum dilution of precision with an integrated network. [7]

There are various studies on GEO satellite orbit determination based on GPS navigation. In particular, no study, to our knowledge, has validated GPS-based OD by comparing traditional flight-proven, frequently used RNG and AZEL methods. The GEO satellite operators and manufacturers need encouragement to use GPS-based orbit determination. Providing evidence about the performance of GPS-based OD by showing consistent results with flight-proven and frequently used methods would be very appreciated. Our research aims to assess the GPS-based OD with flight-proven methods. This study investigates a GPS-based orbit determination performance for GEO communication satellites by comparing the GPS with the classical angle and range measurement.

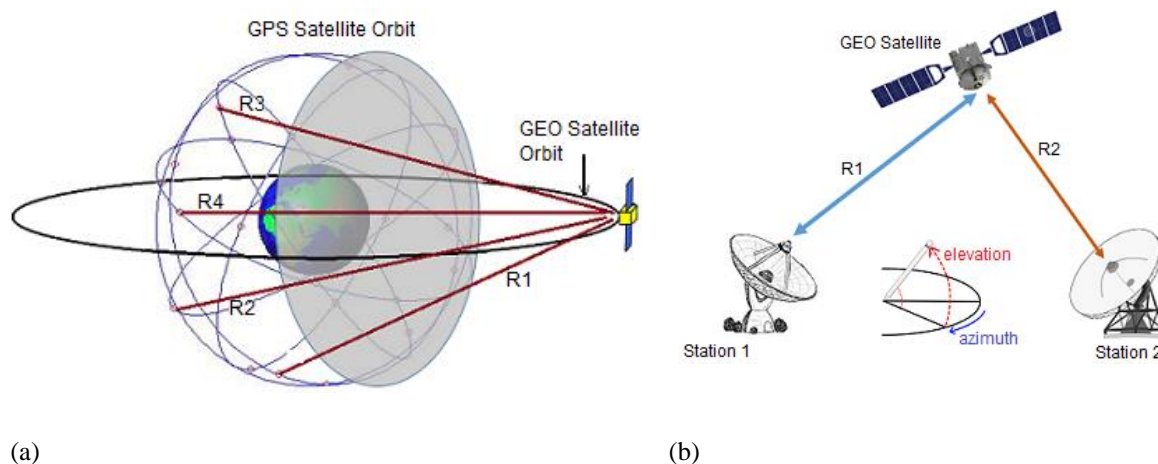


Figure 1 (a) GPS-based orbit determination for GEO orbit (b) AZEL and RNG-based orbit determination methods.

## 2. Observation and Orbit Determination Methods

There are many observation methods to gather orbital data for orbit estimation. In this work, two commonly utilized methods among satellite operators, single station tracking and measurement of azimuth, elevation, and range (AZEL) and the distance measurement from the ground station to satellite called ranging (RNG) methods were utilized to collect data for orbit estimation. Since those are flight-proven and frequently utilized methods, comparing GPS-based OD with these two methods would be more meaningful for the satellite operators.

Figure 1 (a) shows pseudo-range measurements between GEO and GPS satellites. The GEO satellite cannot receive the signals of GPS satellites in gray shaded [5]. Figure 1(b) shows range and angle measurements [6]. In the range-to-range (RNG) method, the distance between ground Station 1, the GEO satellite and ground Station 2, and the GEO satellites are measured simultaneously. The antenna azimuth, elevation angle to the GEO satellite and the range are measured simultaneously in azimuth elevation (AZEL) type observation.

### 2.1 Azimuth Elevation and Range Method (AZEL)

The single-station tracking method is the most traditional way to gather orbital data for orbit determination. In this method, a ground station antenna follows a GEO satellite, and azimuth-elevation angle and range data were gathered to estimate the orbit. This method is mainly utilized and flight-proven methods among satellite operators.

In this method, a single station position vector of is defined as an  $R_{GS}$  in earth-centered earth fixed (ECEF) coordinate. The satellite position vector,  $R_{sat}$ , can also be expressed in the ECEF coordinate system. The range vector of the distance between the ground station and the satellite is shown in Equations 1.

$$\rho = \|R_{sat} - R_{GS}\| + \Delta\rho + v_\rho \tag{1}$$

Here,  $\Delta\rho$  is the range offset, and  $v_\rho$  show the range noise. We represent the station to satellite vector of the topocentric frame using a transformation of coordinate; Topo-centric ECEF can be defined in Equation 2 as,

$$\rho_{Topocentric} = C_{ECEF}^{Topocentric} (R_{sat} - R_{GS}) \tag{2}$$

The angles-tracking data, azimuth, and elevation are obtained from the combination of each range, as shown in Equations 3 and 4 [6].

$$Az = atan2(\rho_y/\rho_x) \tag{3}$$

$$El = acos(\rho_z/\rho) \tag{4}$$

The Keplerian orbital parameters in Table 1 were calculated using range-range observation data for three satellites, Sat A, Sat B, and Sat C.

Table 1 Table Classical (Keplerian) orbital parameters of the considered satellite orbits obtained using the AZEL method.

Satellite/ Method	<b>SMA (km)</b>	<b>Ecc</b>	<b>Incl (deg)</b>	<b>RAAN (deg)</b>	<b>ArgPer (deg)</b>	<b>TrueAn (deg)</b>
Sat A / AZEL	42165.049	9.12E-05	0.048062	282.527	342.499	353.358
Sat B / AZEL	42165.056	9.35E-05	0.048915	302.152	331.043	4.181
Sat C / AZEL	42164.533	7.32E-05	0.047396	258.838	355.294	341.764

Those data were collected using the AZEL observation method. The sequential processing technique was utilized to obtain the classical orbital parameters.

### 2.2 Ranging Method (RNG)

The RF signal emitted from the ground station is received and re-transmitted from the satellite. The re-transmitted signal is received via the ground station. After performing the necessary process, the range between the ground station and the satellite is obtained as range data [16].

The range from a ground station to a satellite can be defined in the following Equation 1.

$$\rho_{i1} = |R_{sat} - R_{GSi}| + c\tau_{delay} + \Delta d_{trop} + \Delta d_{ion} + \varepsilon \tag{5}$$

Where;  $\rho$ : station to satellite distance,  $R_{SAT}$ : satellite position vector,  $R_{GS}$ : ground station position vector,  $c$ : speed of the light,  $\tau$ : ground station and transponder time delay,  $\Delta d_{trop}$ : tropospheric delay,  $\Delta d_{ion}$ : ionospheric delay,  $\varepsilon$ : other errors

Table 2 Table Classical (Keplerian) orbital parameters of the considered satellite orbits obtained using the RNG method.

Satellite/ Method	<b>SMA (km)</b>	<b>Ecc</b>	<b>Incl (deg)</b>	<b>RAAN (deg)</b>	<b>ArgPer (deg)</b>	<b>TrueAn (deg)</b>
Sat A / RNG	42165.055	9.12E-05	0.048068	282.529	342.392	353.463
Sat B / RNG	42165.056	9.35E-05	0.048915	302.152	331.043	4.181
Sat C / RNG	42164.558	7.33E-05	0.047476	258.815	354.894	342.186

The range observation data was utilized to estimate Keplerian orbital parameters. The calculated classical parameters are shown in Table 2, and those values will be a reference to assess GPS-based measurement results.

### 2.3 GPS-based method

GEO satellites can receive GPS signals from the main or side lobe, although geo orbit is higher than GPS orbit. In this method, onboard GPS receivers acquire the signal from known GPS satellites and process raw data. This work uses C/A (clear/acquisition) code pseudo-range measurement to calculate the range between GEO satellites and GPS satellites.

GPS satellite's C/A signal code pseudo-range in L1 frequency can be expressed in Equation 6,

$$\rho_c = \rho + c[\Delta t_{GPS}(t) - \Delta t_U(t)] \quad (6)$$

where  $\rho_c$ : CA code pseudo-range in L1,  $c$ : speed of light,  $\Delta t_{GPS}$ : clock offset of GPS satellite,  $\Delta t_U$ : clock offset of a receiver,  $t$ : instant observation time,  $\rho$ : 3D distance between onboard GEO satellite receiver and GPS satellite.

$$\rho = \sqrt{(x_{GPS} - x)^2 + (y_{GPS} - y)^2 + (z_{GPS} - z)^2} \quad (7)$$

Where  $x, y, z$ : position of the GEO satellites,  $X_{GPS}, Y_{GPS},$  and  $Z_{GPS}$ : position of the GPS satellite.

Table 3 provides calculated Keplerian parameters of three GEO satellites using GPS pseudo-range data. This method is called GPS-based navigation.

Table 3 Classical (Keplerian) orbital parameters of the considered satellite orbits.

Satellite/ Method	<b>SMA (km)</b>	<b>Ecc</b>	<b>Incl (deg)</b>	<b>RAAN (deg)</b>	<b>ArgPer (deg)</b>	<b>TrueAn (deg)</b>
Sat A / GPS	42165.055	9.12E-05	0.048068	282.529	342.392	353.463
Sat B / GPS	42165.059	9.40E-05	0.048825	302.187	331.137	4.052
Sat C / GPS	42164.537	7.32E-05	0.047394	258.832	355.206	341.858

The orbital parameters in Table 1 and Table 2 were used to assess GPS-based navigation method performance. It is expected to have quite identical orbits with orbits obtained using other AZEL and RNG methods [17, 18].

In this work, considering communication satellites' Keplerian parameters are expressed in J2000, the earth-centered inertial (ECI) coordinate system. The X points toward the mean vernal equinox of the Earth on 1 January 2000, at 12:00:00:00 UTC, in the J2000 system. The satellites are assumed to have 2500 kg mass,  $c_d=0.2$   $c_r=1.3$ , and a solar pressure area of 60 m<sup>2</sup>. The collected data for one satellite is 144 samples for each method. The total collected data is 1296 samples.

### 2.4 Orbit Determination and Analysis Method

GPS, AZEL, and RNG observation data were used to calculate the Keplerian orbit parameters, also known as classical orbital parameters. The same orbit determination method, namely the Sequential Process Method, was applied to calculate the satellite orbits for all three types of observation data. A Sequential Process (SP) Kalman filter was employed consistently across the three methods to analyze the impact of the observation data on orbit determination. The resulting orbital parameters are presented in Table 1, Table 2, and Table 3.

To evaluate the differences in the obtained orbits, the orbits were propagated for 48 hours with a time interval of 20 minutes. Statistical analyses were conducted to assess the root mean square error (RMSE) and standard deviation (Std Dev) of the orbit differences. These analyses were performed using the propagated orbital data from each method, with a high significance level. Additionally, data analysis studies involving RMSE and Std Dev were carried out, and graphical representations were created for all three methods, considering three satellites located at different orbital positions.

### 3. Results and Discussion

This paper compares using a GPS receiver in communication satellite orbit determination with operators' widely used methods. We evaluated classical (Keplerian) orbital elements of three observation data sets for three satellites using the GPS, RNG, and AZEL methods. Those parameters were propagated for 48 hours with a 20-minute interval for each method starting from an epoch. All methods produced their results, and there are some differences between them. The principal focus of this work was to calculate the similarity of orbits obtained using each method. The GPS-based navigation method results are compared with traditional orbit determination methods RNG and AZEL.

This work outlines the performance of a GPS receiver usage for orbit determination of communication satellites by comparing it with traditional single-station tracking (AZEL azimuth, elevation range measurement) and two-station range-to-range (RNG) measurement. Three GEO satellites at different orbital locations were utilized in this work, and classical orbital element are given in Table 1, Table 2, and Table 3.

The performance of the GPS method can be evaluated by analyzing the differences between the RNG and AZEL methods. The results obtained using each data set have been compared in terms of radial, along-track, cross-track, and range (3D distance).

The orbits are compared, and the differences have been calculated in all spatial directions to analyze the obtained orbital parameters in three data collection methods. Table 4 provides differences between the GPS versus the RNG method and the GPS versus the AZEL method in the radial, along-track, and cross-track direction for three satellites. The maximum value of RMSE is 768.173 m in Sat C along-track direction. Similarly, the worst standard deviation is 49.159 m in the Sat B cross-track direction.

The results below represent that using GPS receivers proposes good enough performance for orbit determination.

Table 4 Sat A, Sat B and Sat C spatial (RAC) position differences between GPS and RNG and GPS and AzEL.

Satellite Name	Statistics	RNG - GPS			AZEL - GPS		
		Radial	Along Track	X Track	Radial	Along Track	X Track
Sat A	StDev	5.419	30.843	3.596	12.420	47.855	37.437
	RMSE	7.702	75.410	3.583	13.659	127.228	37.308
Sat B	StDev	9.246	18.943	43.724	15.089	41.658	49.159
	RMSE	9.238	369.747	43.573	15.429	236.564	48.989
Sat C	StDev	2.183	10.974	9.129	12.593	38.552	2.746
	RMSE	3.052	768.173	9.098	13.085	764.749	2.737

Figure 2 a. and b. show detailed radar views of GPS vs. RNG and GPS vs. AZEL method in radial, along-track, and cross-track directions differences for 48 hours and Sat A. The prediction difference in RMSE in the radial, along-track, and cross-track directions are 7.702 m, 75.410 m, and 3.583 m, respectively, for the Sat A GPS-RNG method. GPS-AZEL method position differences for Sat A are similar to GPS-RNG method position differences. The graph has shown promising results in all spatial directions.

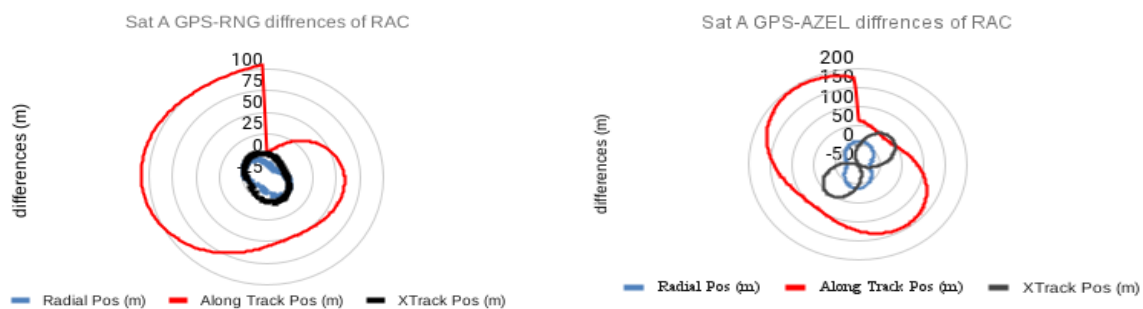


Figure 2 (a) The details of differences of orbits in RAC directions, obtained from GPS-based and RNG based method for Sat A (b) The details of differences of orbits in RAC directions, obtained from GPS-based and AZEL-based method for Sat A, in radar view.

Figure 3 a) and Table 4 present the differences between the GPS-based obtained orbit and RNG-based obtained orbit for Sat B. The left vertical axis in red color shows details of along-track position differences. The right vertical axis in blue and black shows radial and cross-track differences, and the horizontal axis shows time in an hour for both Figures 3a) and 3 b). The RMSE errors are 9.238 m, 369.747 m, and 43.724 m. simultaneously, standard deviation values are 9.246, 18.943, and 43.724 in radial, along-track, and cross-track directions, respectively. The maximum difference between GPS-based and AZEL-based calculated orbit is 236.564 m in the along-track direction. The differences between models are due to measurement errors and the accuracy of the dynamic satellite model.

These results are in line with expectations and less than the maximum allowed error of 1582 m value.

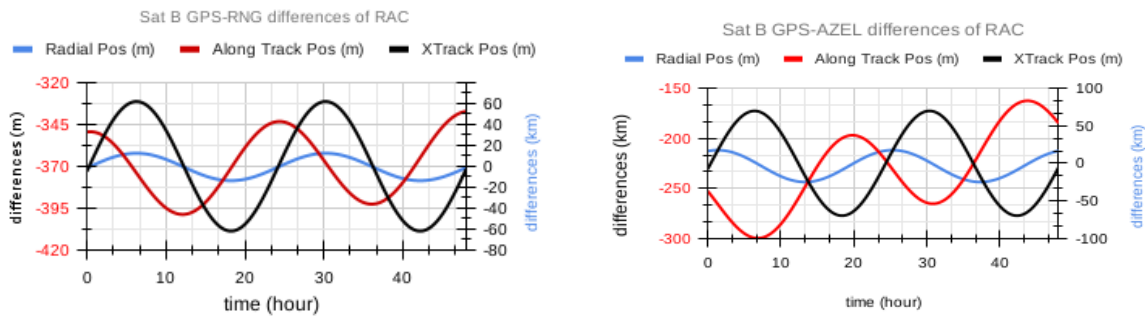


Figure 3 (a) The details of differences of orbits in RAC directions, obtained from GPS-based and RNG based method for Sat B (b) The details of differences of orbits in RAC-directions, obtained from GPS-based and AZEL-based method for Sat B, in the time axis.

Table 4 and Figure 4 a) compare calculated orbits using GPS-based and RNG-based measurements for Sat C. The left vertical axis in red color shows details of along-track position differences. The right vertical axis in red shows cross-track differences, the left vertical axis in blue color shows radial position, and the horizontal axis shows along-track position differences for both Figure 4 (a) and Figure 4 8b). Figure 4 (a) shows position differences in radial and cross-track directions between GPS-based and RNG-based orbits. Similarly, Figure 4 (b) shows position differences in radial and cross-track directions for Sat C between GPS-based and RNG-based orbits. The differences in all directions for the two methods are less than 1582 m success criteria and about 780 m. Standard deviations are slight, and the distribution of data is at an acceptable level. Consequently, evaluating GPS-based determination using Sat C orbits as a sample shows a perfect correlation between flight-proven and GPS-based orbits.

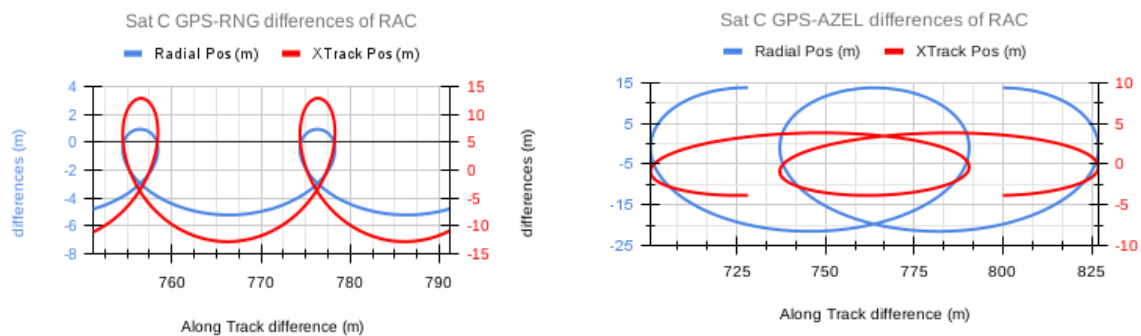


Figure 4 (a) The details of differences of orbits in radial and cross-track directions, obtained from GPS-based and RNG based method for Sat B (b) The details differences of orbits in radial and cross-track directions, obtained from GPS-based and AZEL based method for Sat B, in distributed view and x-axis shows along-track differences

As seen from Figures 4 a and b, the variations in RAC directions have a low level of fluctuation. Still, the fluctuation in the along-track direction is relatively higher. The fluctuation is primarily due to solar pressure and the accuracy of dynamic satellite models of Sat A.

The differences in Figures 2, 3, and 4 are less than 1 km. The results are auspicious and validate GPS-based navigation with flight-proven and widely used methods among satellite operators.

Table 5. Statistical summary of 3D position differences of the argued methods for all three satellites

Sat Name	RMSE (RNG- GPS)	RMSE (AZEL-GPS)	Stdev (RNG- GPS)	Stdev (AZEL- GPS)
Sat A	75.887	133.287	29.873	44.467
Sat B	372.420	242.076	18.996	40.481
Sat C	768.233	764.866	10.981	38.548

The actual physical distance (3D) between the based method and the other two methods was investigated for three satellites. 3D differences in RMSE and standard deviation values are calculated and tabulated in Table 5.

GPS-based and RNG-based determined orbit RMSE of 3D differences are 75.887 m, 372.420m, and 768.223 m for Sat A, Sat B, and Sat C, respectively. GPS versus AZEL orbit 3D differences are small and similar to GPS vs RNG-based 3D orbit differences. Similarly, AZEL-based and GPS-based determined orbit RMSE of 3D position differences are 133.287 m, 242.076 m, and 764.866 m for Sat A, Sat B, and Sat C, respectively. Small standard deviation values imply that the orbits are identical to each other.

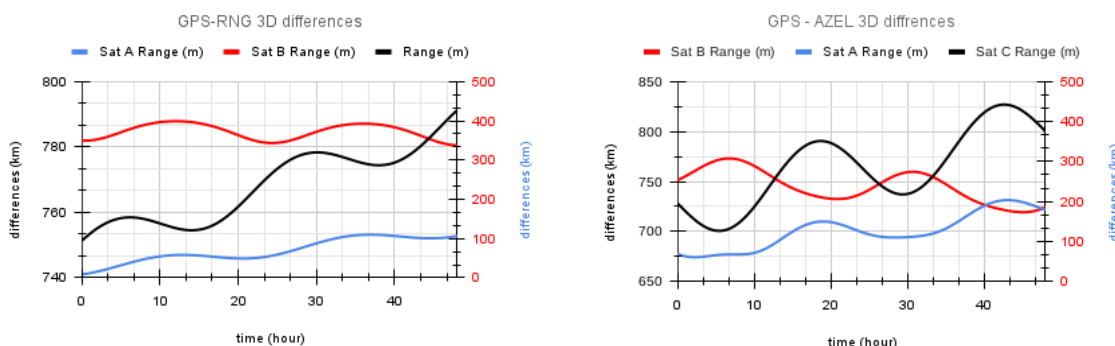


Figure 5 (a) The details of differences of orbits in 3D, obtained from GPS-based and RNG-based method for Sat C (b) The details of differences of orbits in 3D, obtained from GPS-based and AZEL-based method for Sat C, in the time axis.

Figure 5 (a) compares the orbits of three GEO satellites at different longitudes. The left vertical axis in black color shows details of 3D position differences of Sat C—the right vertical axis in red and blue shows 3D differences between Sat A and Sat B.

Table 5 values and Figure 5 graphics are in line with expectations, and errors are less than the success criteria [19, 20].

The literature contains several studies resembling GPS-based orbit determination from various perspectives. One such study is "Real-Time Multi-GNSS Precise Orbit Determination Based on the Hourly Updated Ultra-Rapid Orbit Prediction Method" [21]. This research focuses on evaluating accuracy through a frequent data-receiving approach. It delves into the analysis of both BDS (BeiDou Satellite System) and GPS side-lobe observation quality, providing insights into the impact of side-lobe effects on-orbit accuracy.

Another relevant work is "Orbit Determination with a GEO Satellite Onboard Receiver" [22]. This study evaluates orbit determination by employing a GEO satellite onboard receiver. It particularly investigates how the presence of such a receiver influences orbit accuracy, shedding light on the intricacies of using GPS in this context.

Furthermore, the research titled "Orbit Determination for All-Electric GEO Satellites Based on Space-Borne GNSS Measurements" [23] is another noteworthy contribution. This study explores the utilization of space-borne GNSS (Global Navigation Satellite System) measurements for orbit determination, emphasizing its relevance for all-electric GEO (Geostationary Earth Orbit) satellites.

These studies collectively demonstrate the diverse applications of GPS in satellite orbit determination, each offering unique insights into its use from different angles and contexts. This body of research highlights the versatility and efficacy of GPS-based methods in advancing our understanding of satellite orbits and enhancing their precision.



The present research validates and reinforces strong support for GPS-based orbit determination. The well-established flight-proven RNG and AZEL methods, known and trusted by satellite operators, provided compelling evidence in favor of the GPS-based orbit determination method.

#### 4. Conclusion

The evolution in data collecting and processing methods in OD has enforced the satellite operators to look for unconventional OD methods. OD methods should provide satellite operators with precise orbit parameters and cost-effective, reliable, and sustainable solutions.

The accuracy of GPS-based range measurement via onboard GEO satellites was investigated. Related estimated orbit accuracy is discussed by comparing traditional frequently utilized flight-proven single station tracking and range-range methods in this work. This research suggests that GPS-based OD provides a reliable solution compared to single-station tracking and range-range methods. The results from three satellite longitudes indicate that satellite operators can utilize GPS-based navigation for orbit determination. The results agree with flight-proven AZEL and RNG method's orbit parameters.

Finally, our comparison between the AZEL vs. GPS and RNG vs. GPS methods has confirmed the viability of GPS-based navigation for accurately estimating the orbit of communication satellites.

#### References

- [1] H. Li, "Geostationary satellites collocation," Springer, pp. 10-98, 2014.
- [2] I. Oz, and U. C. Yılmaz, "Determination of Coverage Oscillation for Inclined Communication Satellite," *Sakarya University Journal of Science*, 24.5: 973-983, 2020.
- [3] I. Oz, "Coverages stabilization of an inclined orbit communication satellite with two axis biases," *Journal of The Faculty of Engineering and Architecture of Gazi University*, 38.1: 219-229, 2022.
- [4] B. Schutz, T. Byron, and H. B. George, "Statistical orbit determination," Elsevier, pp. 1-210, 2004.
- [5] I., Oz, U. C. Yılmaz, and U. Guler, "Performance Assessment of a Turn Around Ranging in Communication Satellite Orbit Determination," *Sakarya University Journal of Computer and Information Sciences* 4.1, 73-83., 2021.
- [6] M. Wang, et al. "GNSS-based orbit determination method and flight performance for geostationary satellites," *Journal of Geodesy* 95.8: 1-15., 2021.
- [7] U. K. Acharjee, A. Anis, and R. Shahida, "Performance analysis of navigation by the integration of GPS-24 with LEO & GEO," *2007 10th international conference on computer and information technology*. IEEE, 2007.
- [8] A. Cano, et al. "Covariance Determination for Improving Uncertainty Realism in Orbit Determination and Propagation," *Advances in Space Research*, 2022.
- [9] A. Cano, et al., "Improving Orbital Uncertainty Realism Through Covariance Determination in GEO," *The Journal of the Astronautical Sciences*, 69.5: 1394-1420., 2022.
- [10] A. Chiaradia, P. M. Ana, K. K. Hélio, and F. Antonio, "Onboard and real-time artificial satellite orbit determination using GPS," *Mathematical Problems in Engineering*, 2013.
- [11] M. Guan, T. Xu, M. Li, F. Gao, D. Mu, "Navigation in GEO, HEO, and Lunar Trajectory Using Multi-GNSS Sidelobe Signals," *Remote Sensing*, 14(2), 318., 2022.
- [12] V. Capuano, et al. "High accuracy GNSS based navigation in GEO." *Acta Astronautica* 136: 332-341., 2017.
- [13] Z. Jun, et al. "High Accuracy Navigation for Geostationary Satellite TTS-II via Space-borne GPS." *2020 39th Chinese Control Conference (CCC)*. IEEE, 2020.
- [14] Y. Hwang, et al., "Orbit determination accuracy improvement for geostationary satellite with single station antenna tracking data," *ETRI journal*, 30.6: 774-782., 2008.
- [15] D. Fasbender, et al. "A Simple Similarity Index for the Comparison of Remotely Sensed Time Series with Scarce Simultaneous Acquisitions," *Remote Sensing* 11.13: 1527, 2019.
- [16] M. Wang, et al., "GNSS-based orbit determination method and flight performance for geostationary satellites," *Journal of Geodesy*, 95.8: 1-15., 2021.
- [17] U. K. Acharjee, A. Anis; R. Shahida, "Performance analysis of navigation by the integration of GPS-24 with LEO & GEO," *2007 10th international conference on computer and information technology*. IEEE, p. 1-6., 2007.
- [18] I. Oz, "GEO satellite orbit determination using spaceborn onboard receiver," *Politeknik Dergisi*, 1-1., 2022.
- [19] I. Oz, U.C. YILMAZ, U. Guler, "Tdoa Based Tracking Measurement for Geo Satellites Orbit Determination: Evaluation for The Satellite Operators," *Eskişehir Technical University Journal of Science and Technology A-Applied*

*Sciences and Engineering*, 23.1: 137-148., 2022.

- [20] S. Pessina, et al. "Operational Concepts Refinement for The Orbit Determination of Meteosat Third Generation," *International Symposium on Space Flight Dynamics (ISSFD)*. 2017.
- [21] B. Tan, et al. "Real-Time Multi-GNSS Precise Orbit Determination Based on the Hourly Updated Ultra-Rapid Orbit Prediction Method.", *Remote Sensing*, 14(17), 4412, 2022.
- [22] W. Li, K., et al." BDS and GPS side-lobe observation quality analysis and orbit determination with a GEO satellite onboard receiver.", *GPS Solutions*, 27(1), 18, 2023.
- [23] W. Lu, H. Wang, G. Wu, Y. Huang, Y. "Orbit determination for all-electric geo satellites based on space-borne GNSS measurements.", *Remote Sensing*, 14(11), 2627, 2022.

#### **Conflict of Interest Notice**

The authors declare that there is no conflict of interest regarding the publication of this paper.

#### **Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

#### **Availability of data and material**

Not applicable

#### **Plagiarism Statement**

This article has been scanned by iThenticate™.



# Estimation of Uplink Channels for Multiple Users Using Tensor Modeling in RIS-Aided MISO Communication

Rifat Volkan Şenyuva<sup>1</sup>

<sup>1</sup>Maltepe University, Faculty of Engineering and Natural Sciences, Department of Electrical and Electronics Engineering; İstanbul, Türkiye



**Corresponding author:**

Rifat Volkan Şenyuva  
Maltepe University, Faculty of Engineering  
and Natural Sciences, Department of Electrical  
and Electronics Engineering; İstanbul, Türkiye  
**E-mail address:**  
[rifatvolkansenyuva@maltepe.edu.tr](mailto:rifatvolkansenyuva@maltepe.edu.tr)

**Submitted:** 07 September 2023  
**Revision Requested:** 05 November 2023  
**Last Revision Received:** 07 November 2023  
**Accepted:** 09 November 2023  
**Published Online:** 30 November 2023

**Citation:** Şenyuva R.V. (2023). Estimation of Uplink Channels for Multiple Users Using Tensor Modeling in RIS-Aided MISO Communication. *Sakarya University Journal of Computer and Information Sciences*. 6 (3) <https://doi.org/10.35377/saucis...1356872>

## ABSTRACT

In this paper estimation of uplink channels using tensor modeling is addressed for multiple users in a reconfigurable intelligent surface (RIS)-aided multiple-input single-output (MISO) communication. The coherence interval is divided into structured frames of pilot symbols transmitted by the users and pattern of phase shifts applied by the RIS in order to estimate the base station (BS)-RIS channels and the RIS-user's channels. Estimation methods that use tensor modeling including Khatri-Rao Factorization (KRF) and bilinear alternating least squares (BALS) are applied to the signal model. Numerical results show that both KRF and BALS are superior to the LS estimator by 10 dB SNR for the correlated Rayleigh fading channel model.

**Keywords:** Reconfigurable Intelligent Surfaces, Multi-User MISO, Channel Estimation, Least Squares, Khatri-Rao Factorization, Parallel Factor Decomposition

## 1. Introduction

Energy consumption is an important concern for the emerging wireless networks such as 5G [1-2], 6G [2], the Internet of Things (IoT) [2], geostationary (GEO) satellite communications (SatCom) [3-4]. A massive number of devices such as mobile phones, sensors [2], and smart sockets [5] that require uninterrupted connectivity and increased quality of service (QoS) [6] are expected to be deployed in IoT. Thus, these wireless networks must be energy efficient to be realized [1]. One of the solutions is providing some control over the propagation environment via the concept of a smart radio environment [2].

A reconfigurable intelligent surface (RIS) is a candidate technology for making these emerging networks energy efficient. The RIS comprises many low-cost antennas or metamaterials on a 2D surface with integrated circuits that can passively shape an incoming electromagnetic field in desired ways [1-2]. The phase shift of each element of the RIS can be tuned so that the reflected signals can be coherently combined such that the whole incoming signal is amplified. The energy consumption of the RIS is much less compared to that of an Amplify-and-Forward (AF) relay transceiver since the RIS does not employ power amplifiers [7]. The RIS hardware can be easily deployed in a communication environment since they do not take up much space.

One of the fundamental challenges in RIS-employed systems is obtaining the state information of the base station (BS)-RIS channel and the RIS-user channel. However, RIS with many elements means the system will have many channel links that must be estimated. In addition to this, the channels must be estimated at the receiver since no signal processing can be done

at the RIS due to the passive elements. The channel estimation problem is tackled by several previous works in the literature [8-16]. [8] shows the optimal selection for the activation pattern of the RIS elements using a minimum variance unbiased (MVU) estimator. [9] proposes a minimum mean squared error (MMSE) estimator for a deterministically scattered BS-RIS channel. A comparison of the MVU and the MMSE estimators is shown for a single user in [10]. A three-phase channel estimation protocol is given in [11] to deal with the BS-user channel or the direct channel by using the first phase to estimate the direct channel and then applying interference cancellation in the second phase. [12] applies the Khatri-Rao factorization (KRF) and bilinear alternating least squares (BALS) methods to estimate the downlink channels for a single user with multiple antennas. Channel estimation for multiple users using BALS is investigated in [13-15]. The tensor-based channel estimation in a system with double RIS is studied by [16].

We investigate the uplink channel estimation for multiple users in a RIS-aided multiple-input single-output (MISO) communication operating in a time division duplex (TDD). Compared to the conference paper [10], where there is only one user, we consider channel estimation for multiple users in this work. We focus on applying the tensor-based channel estimation methods, which decouple the estimation of the BS-RIS and the RIS-user's channels with the direct channel between the BS and users considered unavailable, unlike the signal model in [10]. Our signal model employs the pilot symbols and the RIS phase shift structure of [9,12]. We apply the two tensor-based channel estimation methods, KRF and BALS, to our multiple-user signal model. Our KRF implementation differs from that of [12] since it's built upon the least squares (LS) filtered signal and does not require the bilinear filtering step given in [12]. The performances of the algorithms are numerically evaluated for a multiple-user scenario with the channels modeled more realistically according to correlated Rayleigh fading, unlike the results given in [12-15] for uncorrelated Rayleigh fading.

The contributions of the paper can be summarized as follows:

- To the best of our knowledge, our formulation for the KRF method is the first in the literature that uses the output of the LS estimator as its input rather than the bilinear filtered signal given in [12].
- The proposed KRF method with the LS estimator is computationally efficient compared to the KRF method using bilinear filtering [12] since the LS estimator can be applied as a Fast Fourier Transform (FFT).
- We use the more practical correlated Rayleigh fading as our channel model. Our numerical results differ from the rest of the literature [12-15] in showing the impact of spatial correlation of the channels concerning RIS elements on the performance of the channel estimation methods using tensor modeling.

The remaining parts of this paper are organized in the following way: the discrete-time baseband received signal model of the MISO system is introduced in Section 2. Then, the derivations of the channel estimation methods, including the LS, KRF, and BALS, are given in Section 3. The performances of the investigated methods are compared in Section 4. Finally, Section 6 presents the conclusions of the paper.

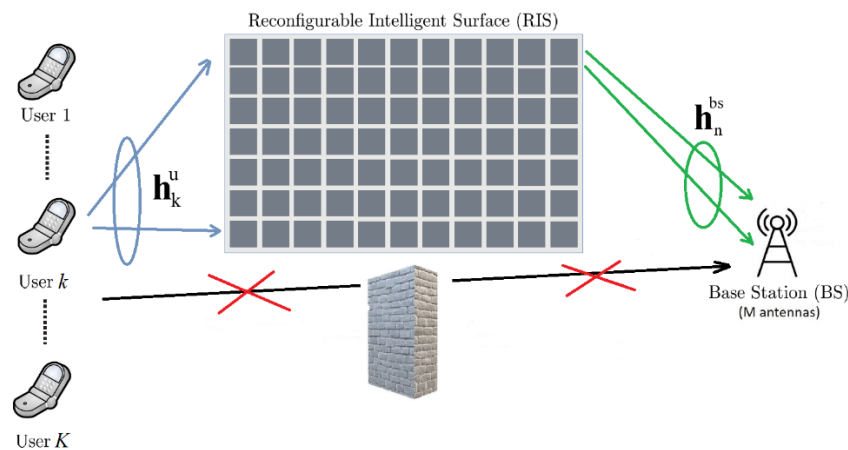


Figure 1  $K$  single antenna users being communicating with an  $M$  -antenna BS in a RIS-assisted multi-user MISO system. Blue and green lines show the uplink channel vectors that must be estimated. The BS-users channel or the direct channel (black line with red cross) is ignored because of high attenuation.

## 2. System Model

We consider a narrow band MISO communication system with a BS equipped with  $M$  antennas serving  $K$  single-antenna users simultaneously, as seen in Figure 1. A RIS with  $N$  passive reflecting elements, which can only shift the phases of the impinging waves, is deployed to assist the BS. The RIS is attached to a surrounding building's façade, and the phase shift of

each RIS element can be adjusted by the BS over a backhaul link. The direct channels between the BS and any  $K$  users are ignored due to high attenuation or can be estimated by turning off the RIS elements.

The MISO system shown in Figure 1 operates in half-duplex TDD mode where the channel between antennas is the same in both directions within the coherence interval. Once the BS learns the uplink channel from uplink pilots sent by the users in the training step, it also automatically has an estimate of the downlink channel. So, a quasi-static block fading channel model, where the channels are constant within the coherence interval of  $T_C$  time slots, is assumed. The total channel training time  $T_C$  is divided into  $S$  sub-blocks where each sub-block has  $T$  time slots so that  $T_C = ST$ . While the RIS phase shifts are kept fixed for the duration the  $s$ -th sub-block that is  $T$  time slots, the users transmit the same pilot sequence across the  $S$  sub-blocks [9,12]. The received baseband signal at the  $t$ -th time slot of the  $s$ -th sub-block  $\mathbf{y}_{s,t} \in \mathbb{C}^{M \times 1}$ , is given as

$$\mathbf{y}_{s,t} = \mathbf{H}^{\text{bs}}(\boldsymbol{\phi}_s \odot \mathbf{H}^{\text{u}}\mathbf{x}_t) + \mathbf{n}_{s,t} \quad (1)$$

where the channels between the RIS and BS are shown as  $\mathbf{H}^{\text{bs}} = [\mathbf{h}_1^{\text{bs}} \dots \mathbf{h}_N^{\text{bs}}] \in \mathbb{C}^{M \times N}$ , the channels between the users and the RIS are represented as  $\mathbf{H}^{\text{u}} = [\mathbf{h}_1^{\text{u}} \dots \mathbf{h}_K^{\text{u}}] \in \mathbb{C}^{N \times K}$ ,  $\mathbf{x}_t \in \mathbb{C}^{K \times 1}$  show the orthogonal pilot sequence transmitted by the users, and  $\mathbf{n}_{s,t} \in \mathbb{C}^{M \times 1}$  shows the complex additive white Gaussian noise random vector with single-sided power spectral density of  $N_0$ , i.e.  $\mathbf{n}_{s,t} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I}_M)$  where  $\mathbf{I}_M$  is the  $M \times M$  identity matrix. The RIS phase shift vector applied at the  $s$ -th sub-block is given as  $\boldsymbol{\phi}_s = [e^{i\theta_{1,s}}, \dots, e^{i\theta_{N,s}}]^T \in \mathbb{C}^{N \times 1}$  where  $\theta_{n,s} \in (\mathbf{0}, 2\pi]$ , and  $\odot$  is the element-wise multiplication, i.e. the Hadamard product shown as with  $\mathbf{H}^{\text{u}}\mathbf{x}_t$  in Equation 1.  $\mathbf{h}_n^{\text{bs}}$  and  $\mathbf{h}_k^{\text{u}}$  are modeled as correlated Rayleigh channels.

$$\mathbf{h}_n^{\text{bs}} = \sqrt{\beta_n^{\text{bs}}} \mathbf{K}_n^{1/2} \mathbf{g}_n \quad (2)$$

$$\mathbf{h}_k^{\text{u}} = \sqrt{\beta_k^{\text{u}}} \mathbf{K}_k^{1/2} \mathbf{g}_k \quad (3)$$

where  $\mathbf{K}_n^{1/2}$  and  $\mathbf{K}_k^{1/2}$  are the correlation matrices at the BS and IRS respectively.  $\mathbf{g}_n \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_M)$  and  $\mathbf{g}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$  are the fast-fading components while  $\beta_n^{\text{bs}}$  and  $\beta_k^{\text{u}}$  are the path loss factors. The received signals for the  $s$ -th sub-block,  $\mathbf{Y}_s = [\mathbf{y}_{s,1}, \dots, \mathbf{y}_{s,T}] \in \mathbb{C}^{M \times T}$ , can be written as

$$\mathbf{Y}_s = \mathbf{H}^{\text{bs}} \text{diag}\{\boldsymbol{\phi}_s\} \mathbf{H}^{\text{u}} \mathbf{X} + \mathbf{N}_s \quad (4)$$

where  $\text{diag}\{\boldsymbol{\phi}_s\}$  is the matrix with the elements of  $\boldsymbol{\phi}_s$  on its diagonal,  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{C}^{K \times T}$  is the pilot sequence matrix across the  $T$  slots, and the noise matrix for the  $s$ -th sub-block is denoted as  $\mathbf{N}_s = [\mathbf{n}_{s,1}, \dots, \mathbf{n}_{s,T}] \in \mathbb{C}^{M \times T}$ . The receiver first despreads the received signal in Equation 4 by multiplying with Hermitian transpose of  $\mathbf{X}$ , i.e.,  $\mathbf{X}^{\text{H}}$ , resulting in

$$\tilde{\mathbf{Y}}_s = \mathbf{H}^{\text{bs}} \text{diag}\{\boldsymbol{\phi}_s\} \mathbf{H}^{\text{u}} + \tilde{\mathbf{N}}_s \quad (5)$$

where  $\tilde{\mathbf{Y}}_s = \mathbf{Y}_s \mathbf{X}^{\text{H}}$ ,  $\tilde{\mathbf{N}}_s = \mathbf{N}_s \mathbf{X}^{\text{H}}$ , and the size of both matrices is  $M \times K$ . Due to the pilot sequence matrix,  $\mathbf{X}$ , being a unitary matrix, i.e.,  $\mathbf{X} \mathbf{X}^{\text{H}} = \mathbf{I}$ , the distribution of the noise vectors in  $\tilde{\mathbf{N}}_s$  do not change.

### 3. Channel Estimation Methods

Channel estimation schemes take the received signal  $\tilde{\mathbf{Y}}_s$  in Equation 5 as input and output an estimate of either each of the channel matrices  $\mathbf{H}^{\text{bs}}$  and  $\mathbf{H}^{\text{u}}$  or the cascade channel matrix which is the product of the individual channel matrices. The least squares method estimates the cascade channel matrix while the KRF and the BALS method estimate  $\mathbf{H}^{\text{bs}}$  and  $\mathbf{H}^{\text{u}}$  separately. Estimating the channel matrices separately is subject to complex scaling ambiguity, but the scaling factors cancel each other when the separate channel matrices are multiplied to calculate the cascade channel.

#### 3.1 Least Squares Channel Estimation

Least squares (LS) estimation is a benchmark method of estimating the cascade channel matrix. If both sides of the Equation 5 are vectorized, since  $\text{diag}\{\boldsymbol{\phi}_s\}$  is a diagonal matrix, it can be rewritten by using the properties  $\text{vec}\{\mathbf{ABC}\} = (\mathbf{C}^{\text{T}} \diamond \mathbf{A}) \text{vec}\{\mathbf{B}\}$  as

$$\mathbf{r}_s = (\bar{\mathbf{H}}^{\text{u}} \diamond \mathbf{H}^{\text{bs}}) \boldsymbol{\phi}_s + \mathbf{w}_s \quad (6)$$

where  $\mathbf{r}_s = \text{vec}\{\tilde{\mathbf{Y}}_s\} \in \mathbb{C}^{MK \times 1}$ ,  $\mathbf{w}_s = \text{vec}\{\tilde{\mathbf{N}}_s\} \in \mathbb{C}^{MK \times 1}$ ,  $\bar{\mathbf{H}}^{\text{u}} = [\mathbf{H}^{\text{u}}]^{\text{T}} \in \mathbb{C}^{K \times N}$  that is the transpose of  $\mathbf{H}^{\text{u}}$ ,  $\diamond$  denotes the Khatri-Rao product. If  $\mathbf{R} = [\mathbf{r}_1 \dots \mathbf{r}_S] \in \mathbb{C}^{MK \times S}$  and  $\mathbf{W} = [\mathbf{w}_1 \dots \mathbf{w}_S] \in \mathbb{C}^{MK \times S}$  are defined, then  $\mathbf{R}$  is equal to

$$\mathbf{R} = \mathbf{H}^{\text{cascade}} \boldsymbol{\Phi} + \mathbf{W} \quad (7)$$

where  $\mathbf{H}^{\text{cascade}} = [\bar{\mathbf{H}}^{\text{u}} \diamond \mathbf{H}^{\text{bs}}] \in \mathbb{C}^{MK \times N}$  and  $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_S] \in \mathbb{C}^{N \times S}$ . Applying another vectorization to both sides of Equation 7 and then using the property  $\text{vec}\{\mathbf{ABC}\} = (\mathbf{C}^{\text{T}} \otimes \mathbf{A}) \text{vec}\{\mathbf{B}\}$  yields

$$\mathbf{r}' = (\bar{\Phi} \otimes \mathbf{I}_{MK})\text{vec}\{\mathbf{H}^{\text{cascade}}\} + \mathbf{z} \tag{8}$$

where  $\mathbf{r}' = \text{vec}\{\mathbf{R}\} \in \mathbb{C}^{MKS \times 1}$ ,  $\mathbf{z} = \text{vec}\{\mathbf{W}\} \in \mathbb{C}^{MKS \times 1}$ ,  $\bar{\Phi} = \Phi^T \in \mathbb{C}^{S \times N}$  that is the transpose of  $\Phi$  and  $\otimes$  denotes the Kronecker product. Equation 8 can be rewritten as

$$\mathbf{r}' = \mathbf{Q}\mathbf{h}^{\text{cascade}} + \mathbf{z} \tag{9}$$

where  $\mathbf{Q} = [\bar{\Phi} \otimes \mathbf{I}_{MK}] \in \mathbb{C}^{MKS \times MKN}$  and  $\mathbf{h}^{\text{cascade}} = \text{vec}\{\mathbf{H}^{\text{cascade}}\} \in \mathbb{C}^{MKN \times 1}$ . The LS estimate is obtained from

$$\hat{\mathbf{h}}^{\text{cascade}} = \arg \min_{\mathbf{h}^{\text{cascade}}} \|\mathbf{r}' - \mathbf{Q}\mathbf{h}^{\text{cascade}}\|_2^2 \tag{10}$$

where the solution is equal to  $\hat{\mathbf{h}}^{\text{cascade}} = \mathbf{Q}^\dagger \mathbf{r}'$  provided that  $S \geq N$  holds and  $\mathbf{Q}^\dagger$  is the Moore-Penrose left inverse of  $\mathbf{Q}$ . The LS estimate can be simplified to

$$\hat{\mathbf{h}}^{\text{cascade}} = [(\bar{\Phi}^H \bar{\Phi})^{-1} \bar{\Phi}^H \otimes \mathbf{I}_{MK}] \mathbf{r}' \tag{11}$$

where  $\bar{\Phi}^H$  is the conjugate transpose of  $\bar{\Phi}$ . [8-9] shows that constructing  $\bar{\Phi}$  from the  $S$  leading columns of the  $N \times N$  DFT matrix,  $\mathbf{F} \in \mathbb{C}^{N \times S}$ , as

$$[\bar{\Phi}]_{s,n} = [\mathbf{F}]_{s,n} = \exp\left[-i \frac{2\pi(s-1)(n-1)}{S}\right], \quad s = 1, \dots, S, n = 1, \dots, N \tag{12}$$

minimizes the LS error. Plugging Equation 12 into Equation 11 simplifies the LS estimator expression further as

$$\hat{\mathbf{h}}^{\text{cascade}} = (1/S)[\mathbf{F}^H \otimes \mathbf{I}_{MK}] \mathbf{r}' \tag{13}$$

where  $\mathbf{F}^H$  is the conjugate transpose of  $\mathbf{F}$ . The LS estimator in Equation 13 can be calculated in total  $\mathcal{O}(MKS \log S)$  operations due to the application of  $MK$  inverse DFT which can be implemented as FFT in  $\mathcal{O}(S \log S)$  operations [8].

### 3.2 Khatri-Rao Factorization Based Channel Estimation

Once the LS estimator in Equation 13 is applied, we obtain

$$\tilde{\mathbf{r}} = \mathbf{h}^{\text{cascade}} + \tilde{\mathbf{z}} \tag{14}$$

where  $\tilde{\mathbf{r}} = (1/N)[\mathbf{F}^H \otimes \mathbf{I}_{MK}] \mathbf{r}'$ , and  $\tilde{\mathbf{z}} = (1/N)[\mathbf{F}^H \otimes \mathbf{I}_{MK}] \mathbf{z}$ . If the filtered signal,  $\tilde{\mathbf{r}}$ , in Equation 14 is reshaped into a  $MK \times N$  matrix, then it can be written using as

$$\tilde{\mathbf{R}} = \bar{\mathbf{H}}^u \diamond \mathbf{H}^{\text{bs}} + \tilde{\mathbf{Z}} \tag{15}$$

where  $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1 \dots \tilde{\mathbf{r}}_N] \in \mathbb{C}^{MK \times N}$  and  $\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_1 \dots \tilde{\mathbf{z}}_N] \in \mathbb{C}^{MK \times N}$ . The Khatri-Rao least squares problem

$$\min_{\bar{\mathbf{H}}^u, \mathbf{H}^{\text{bs}}} \|\tilde{\mathbf{R}} - \bar{\mathbf{H}}^u \diamond \mathbf{H}^{\text{bs}}\|_F^2 \tag{16}$$

deals with the solving both  $\bar{\mathbf{H}}^u$  and  $\mathbf{H}^{\text{bs}}$  in Equation 15 [12]. An efficient solution of the Khatri-Rao least squares problem in Equation 16 is the KRF algorithm which is shown in Algorithm 1. The  $n$ -th column of the Khatri-Rao product  $\tilde{\mathbf{R}} \approx \bar{\mathbf{H}}^u \diamond \mathbf{H}^{\text{bs}}$  is defined as  $\tilde{\mathbf{r}}_n \approx \bar{\mathbf{h}}_n^u \otimes \mathbf{h}_n^{\text{bs}}$  which is a collection of all pair-wise product of its elements. This collection of products can be reshaped into a rank-one matrix,  $\text{vec}\{\tilde{\mathbf{R}}_n\} = \tilde{\mathbf{r}}_n$ , that is the outer product of two vectors that is  $\tilde{\mathbf{R}}_n = \mathbf{h}_n^{\text{bs}} (\bar{\mathbf{h}}_n^u)^T$ . The best rank-one approximation is known to be given by the truncated singular value decomposition (SVD). The KRF algorithm (Algorithm 1) cannot give a unique solution since there exists one non-zero complex number which results in a scaling ambiguity per column that is  $\bar{\mathbf{h}}_n^u \otimes \mathbf{h}_n^{\text{bs}} = (\alpha_n \bar{\mathbf{h}}_n^u) \otimes \left(\frac{1}{\alpha_n} \mathbf{h}_n^{\text{bs}}\right) \forall \alpha_n \in \mathbb{C}_{\neq 0}$ .

The computational complexity of the KRF is determined by the fourth step in Algorithm 1, which calculates the truncated SVD.  $N$  number of truncated SVD can be calculated in  $\mathcal{O}(MKN)$  operations which is the complexity of the KRF algorithm [12]. The flow chart of Algorithm 1 is shown in Figure 2.

Algorithm 1 Khatri-Rao Factorization (KRF)

1	input $\tilde{\mathbf{R}}$
2	for $n = 1, \dots, N$
3	reshape $n$ -th column of $\tilde{\mathbf{R}}$ into $\tilde{\mathbf{R}}_n \in \mathbb{C}^{M \times K}$ such that $\text{vec}\{\tilde{\mathbf{R}}_n\} = \tilde{\mathbf{r}}_n$
4	calculate the SVD of $\tilde{\mathbf{R}}_n$ as $\tilde{\mathbf{R}}_n = \mathbf{U}_n \Sigma_n \mathbf{V}_n^H$
5	calculate the best rank-one approximations by truncating the
6	SVD as $\hat{\mathbf{h}}_n^u = \sqrt{\sigma_1} \mathbf{v}_1^*$ and $\hat{\mathbf{h}}_n^{\text{bs}} = \sqrt{\sigma_1} \mathbf{u}_1$ where $\sigma_1$ is the largest singular
7	value and $\mathbf{v}_1$ and $\mathbf{u}_1$ are the first columns of $\mathbf{V}_n$ and $\mathbf{U}_n$
8	end
9	end
10	output $\hat{\mathbf{H}}^u = [\hat{\mathbf{h}}_1^u \dots \hat{\mathbf{h}}_N^u]$ and $\hat{\mathbf{H}}^{\text{bs}} = [\hat{\mathbf{h}}_1^{\text{bs}} \dots \hat{\mathbf{h}}_N^{\text{bs}}]$

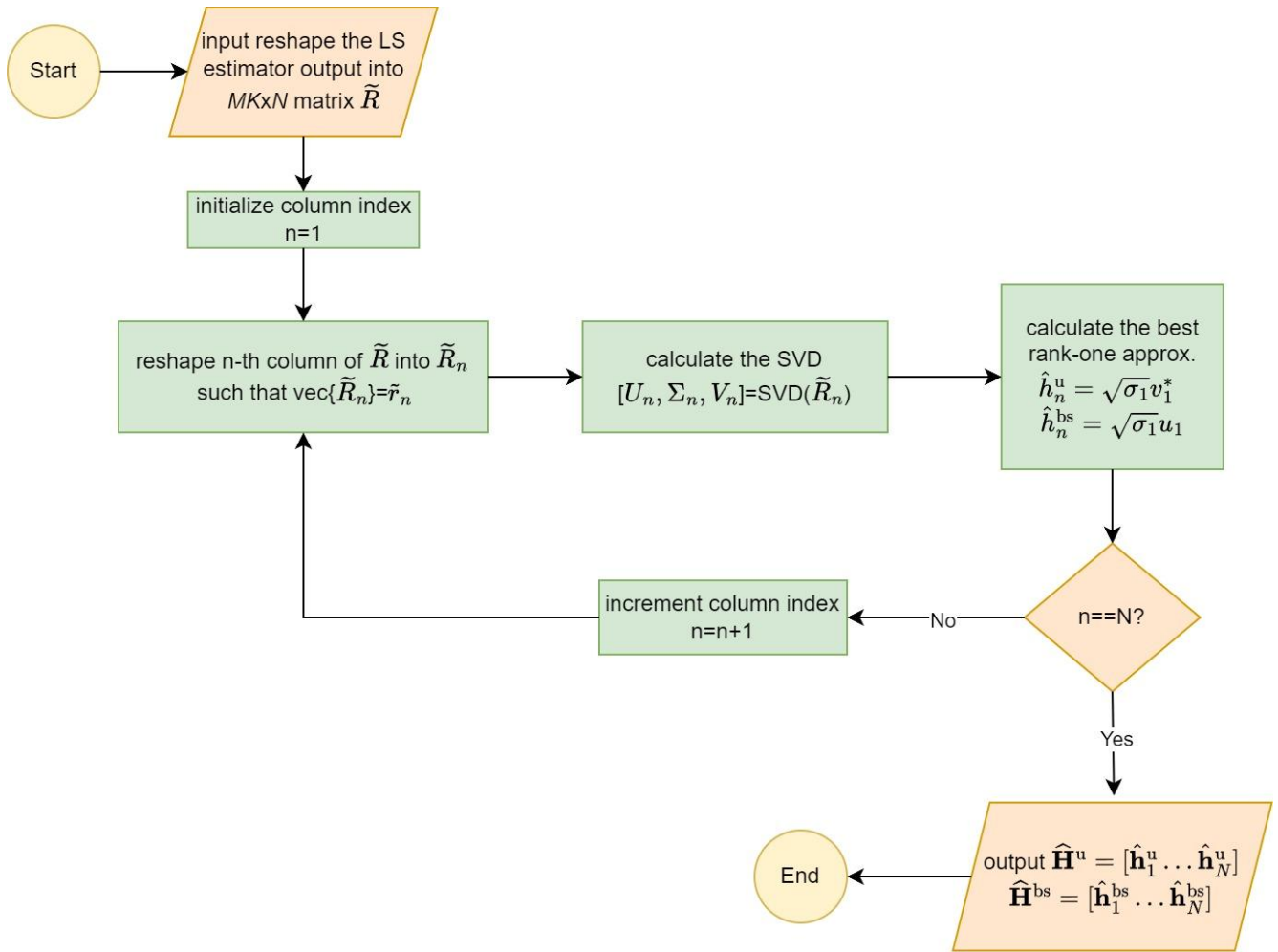


Figure 2 The flow chart of Algorithm 1.

### 3.3 Bilinear Alternating Least Squares Channel Estimation

The signal part of the received signal in Equation 5 is given as

$$\mathbf{Y}'_s = \mathbf{H}^{bs} \text{diag}\{\boldsymbol{\Phi}_s\} [\bar{\mathbf{H}}^u]^T \tag{17}$$

where  $\mathbf{Y}'_s \in \mathbb{C}^{M \times K}$ . The matrix,  $\mathbf{Y}'_s$ , is the  $s$ -th frontal slice of a three-way signal tensor  $\mathcal{Y}' \in \mathbb{C}^{M \times K \times S}$ . Using the canonical parallel factor (PARAFAC) decomposition, the signal tensor  $\mathcal{Y}'$  can be factorized into a sum of rank-one tensors [17-18] as

$$\mathcal{Y}' = \llbracket \mathbf{H}^{bs}, \bar{\mathbf{H}}^u, \bar{\boldsymbol{\Phi}} \rrbracket = \sum_{n=1}^N \mathbf{h}_n^{bs} \circ \bar{\mathbf{h}}_n^u \circ \bar{\boldsymbol{\Phi}}_n \tag{18}$$

where  $\circ$  denotes the outer product.  $\mathcal{Y}'$  in Equation 18 can be written in three matricized forms or mode- $n$  unfoldings [17-18] as

$$\mathbf{Y}'_{(1)} = \mathbf{H}^{bs} (\bar{\boldsymbol{\Phi}} \diamond \bar{\mathbf{H}}^u)^T \tag{19}$$

$$\mathbf{Y}'_{(2)} = \bar{\mathbf{H}}^u (\bar{\boldsymbol{\Phi}} \diamond \mathbf{H}^{bs})^T \tag{20}$$

$$\mathbf{Y}'_{(3)} = \bar{\boldsymbol{\Phi}} (\bar{\mathbf{H}}^u \diamond \mathbf{H}^{bs})^T \tag{21}$$

where  $\mathbf{Y}'_{(1)} = [\mathbf{Y}'_1, \dots, \mathbf{Y}'_S] \in \mathbb{C}^{M \times KS}$ ,  $\mathbf{Y}'_{(2)} = [\bar{\mathbf{Y}}'_1, \dots, \bar{\mathbf{Y}}'_S] \in \mathbb{C}^{K \times MS}$ ,  $\bar{\mathbf{Y}}'_s = [\mathbf{Y}'_s]^T$ , and  $\mathbf{Y}'_{(3)} = [\text{vec}\{\mathbf{Y}'_1\}, \dots, \text{vec}\{\mathbf{Y}'_S\}]^T \in \mathbb{C}^{S \times MK}$ . When noise is added to the signal tensor in Equation 18, the received signal tensor is given as

$$\tilde{\mathcal{Y}} = \mathcal{Y}' + \tilde{\mathcal{N}} \tag{22}$$

where the noise tensor is shown as  $\tilde{\mathcal{N}}$ . The mode- $n$  unfoldings of the received signal tensor in Equation 22 are

$$\tilde{\mathbf{Y}}_{(l)} = \mathbf{Y}'_{(l)} + \tilde{\mathbf{N}}_{(l)} \quad (23)$$

where  $l = 1, 2, 3$  and  $\tilde{\mathbf{N}}_{(l)}$  is the corresponding unfolding for the noise tensor,  $\tilde{\mathcal{N}}$ . The bilinear alternating least squares (BALS) estimation shown in Algorithm 2 is applied to the noisy versions of Equation 19 and Equation 20 which are  $\tilde{\mathbf{Y}}_{(1)}$  and  $\tilde{\mathbf{Y}}_{(2)}$  respectively. BALS requires  $\bar{\mathbf{H}}^u$  to be initialized and this is achieved by calculating the SVD of  $\mathbf{Y}'_{(2)}$  and setting  $\bar{\mathbf{H}}^u$  to  $N$  leading left singular vectors of  $\mathbf{Y}'_{(2)}$ . Since the RIS coefficients matrix,  $\bar{\Phi}$ , is known and so does not need to be fixed, BALS first fixes  $\bar{\mathbf{H}}^u$  to solve for  $\mathbf{H}^{bs}$  by calculating

$$\hat{\mathbf{H}}^{bs} = \min_{\mathbf{H}^{bs}} \|\tilde{\mathbf{Y}}_{(1)} - \mathbf{H}^{bs}(\bar{\Phi} \diamond \bar{\mathbf{H}}^u)^T\|_F^2 = \tilde{\mathbf{Y}}_{(1)} [(\bar{\Phi} \diamond \bar{\mathbf{H}}^u)^T]^\dagger \quad (24)$$

where  $[(\bar{\Phi} \diamond \bar{\mathbf{H}}^u)^T]^\dagger$  is the Moore-Penrose right inverse  $(\bar{\Phi} \diamond \bar{\mathbf{H}}^u)^T$  and then fixes  $\mathbf{H}^{bs} = \hat{\mathbf{H}}^{bs}$  to solve for  $\bar{\mathbf{H}}^u$  by calculating

$$\hat{\mathbf{H}}^u = \min_{\bar{\mathbf{H}}^u} \|\tilde{\mathbf{Y}}_{(2)} - \bar{\mathbf{H}}^u(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs})^T\|_F^2 = \tilde{\mathbf{Y}}_{(2)} [(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs})^T]^\dagger \quad (25)$$

where  $[(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs})^T]^\dagger$  is the Moore-Penrose right inverse of  $(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs})^T$  [12-13]. The necessary condition for obtaining unique solutions to Equations 24 and 25 is that the matrices  $(\bar{\Phi} \diamond \bar{\mathbf{H}}^u) \in \mathbb{C}^{SK \times N}$  and  $(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs}) \in \mathbb{C}^{SM \times N}$  must have full column rank which means  $SK \geq N$  and  $SM \geq N$ . Satisfying these two inequalities at the same time yields  $S \min(K, M) \geq N$  [12]. This condition is not sufficient to guarantee the uniqueness of the BALS estimates. For the PARAFAC decomposition in Equation 18 to be identifiable  $M, K \geq N$  [13-14]. Both the number of users and the number of BS antennas of a practical MISO system is going to be less than the number of RIS elements. The identifiability condition may be satisfied by partitioning the RIS into groups of non-overlapping cells with the number of elements in each cell less than  $M$  and  $K$  [13]. The algorithm stops either when  $\|\varepsilon_j - \varepsilon_{j-1}\| < \epsilon$  that is the error calculated at the  $j$ -th iteration of the algorithm,  $\varepsilon_j = \|\tilde{\mathcal{Y}} - \hat{\mathcal{Y}}_j\|_F^2$ , is less than a threshold,  $\epsilon$ , or a maximum number of iterations,  $J$ , has been completed.

Algorithm 2 BALS

1	input $\bar{\Phi}$
2	initialize $\bar{\mathbf{H}}^u$ by calculating SVD of $\mathbf{Y}'_{(2)}$ and set $\bar{\mathbf{H}}^u$ to $N$ leading
3	left singular vectors of $\mathbf{Y}'_{(2)}$
4	for $j = 1, \dots, J$
5	find an estimate of $\mathbf{H}^{bs}$ by calculating $\hat{\mathbf{H}}^{bs} = \mathbf{Y}'_{(1)} [(\bar{\Phi} \diamond \bar{\mathbf{H}}^u)^T]^\dagger$
6	find an estimate of $\bar{\mathbf{H}}^u$ by calculating $\hat{\mathbf{H}}^u = \mathbf{Y}'_{(2)} [(\bar{\Phi} \diamond \hat{\mathbf{H}}^{bs})^T]^\dagger$
7	if $\ \varepsilon_j - \varepsilon_{j-1}\  < \epsilon$
8	break
9	end
10	end
11	output $\hat{\mathbf{H}}^u$ and $\hat{\mathbf{H}}^{bs}$
12	
13	

The computationally involved steps of the BALS estimation shown in Algorithm 2 are the fifth and sixth steps in which are the calculation of two right pseudo-inverses. The estimates of  $\hat{\mathbf{H}}^{bs}$  and  $\hat{\mathbf{H}}^u$  can be calculated in  $\mathcal{O}(N^3 + 4N^2MS - NMS)$  and  $\mathcal{O}(N^3 + 4N^2KS - NKS)$  operations [14]. Thus, the complexity of the BALS for a maximum number of iterations is equal to  $\mathcal{O}(2N^3J + 4N^2SJ(M + K) - NSJ(M + K))$ . The flow chart of Algorithm 2 is shown in Figure 3.

#### 4. Numerical Results

The numerical results are calculated in a MATLAB environment of R2023b release with version number 23.2.0.2409890 and 3rd update installed on a personal computer (PC). The PC's operating system is 64-bit Windows 11 Pro with AMD Ryzen 5 3600 6-Core Processor at 3.60 GHz and 16 GB RAM. The normalized mean square error (NMSE) is used to compare the performances of the investigated estimators. NMSE can be calculated in

$$\text{NMSE}(\hat{\mathbf{H}}^o) = \frac{1}{R} \sum_{r=1}^R \frac{\|\mathbf{H}_r - \hat{\mathbf{H}}_r^o\|_F^2}{\|\mathbf{H}_r\|_F^2} \quad (26)$$

where  $o \in \{\text{bs}, \text{u}, \text{cascade}\}$ ,  $\hat{\mathbf{H}}_r^o$  is the corresponding channel matrix estimated at the  $r$ -th iteration,  $R$  is the total number of runs and  $\|\cdot\|_F^2$  shows the square of the Frobenius norm. The signal-to-noise ratio (SNR) is given as



$$\text{SNR} = 10 \log_{10} \left( \frac{\|\mathbf{Y}'\|_F^2}{\|\tilde{\mathcal{N}}\|_F^2} \right) \tag{27}$$

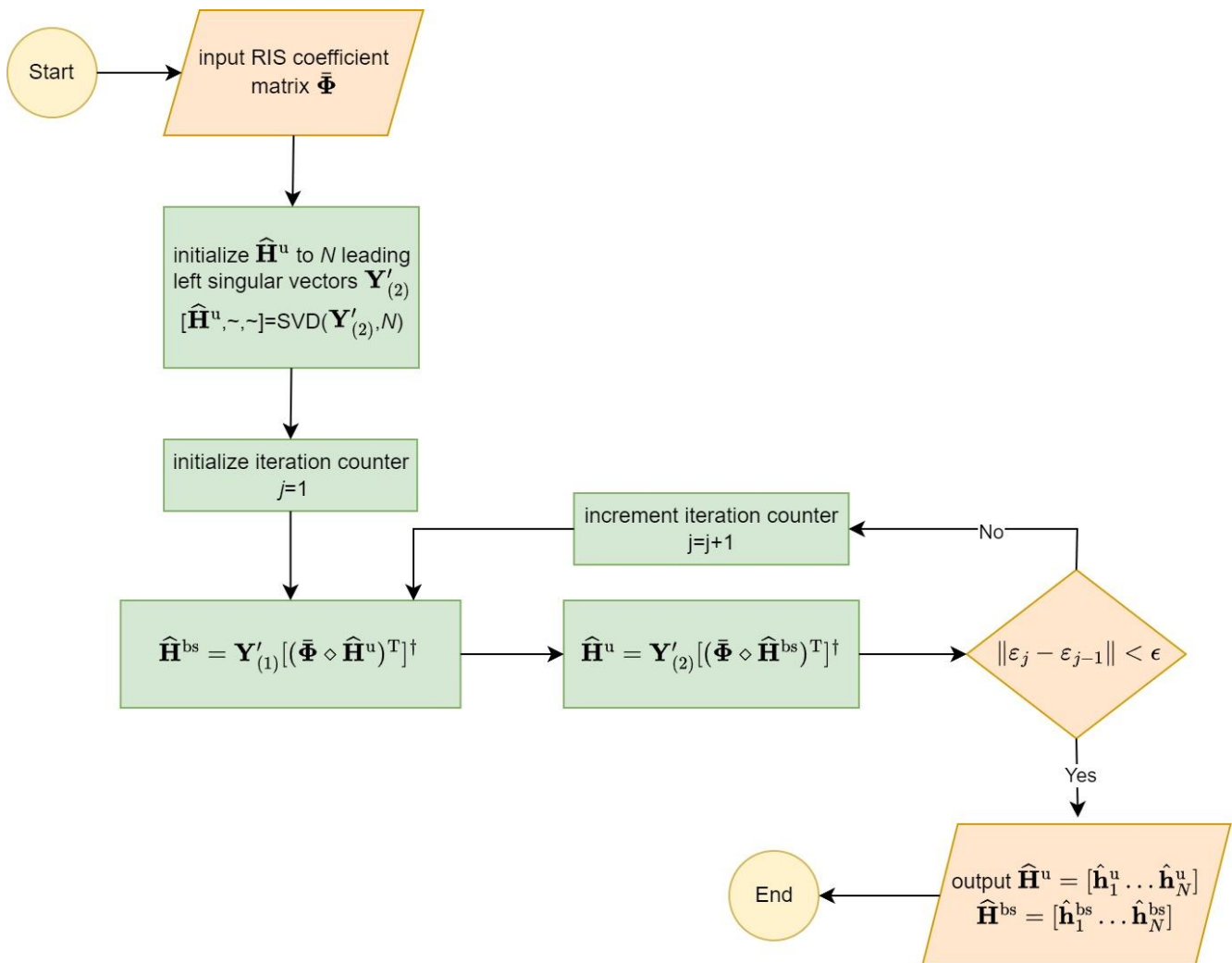


Figure 3 The flow chart of Algorithm 2.

where  $\|\tilde{\mathcal{N}}\|_F^2 = MTSN_0$ . The channel matrices  $\mathbf{H}^{bs}$  and  $\mathbf{H}^u$  are generated as Rayleigh fading channels with correlation matrices  $[\mathbf{K}_n]_{m,m'} = \eta^{|m-m'|}$  and  $[\mathbf{K}_k]_{n,n'} = \eta^{|n-n'|}$  respectively with correlation coefficient set to  $\eta = 0.95$  in each run. NMSE curves are averaged over  $R = 1000$  iterations for parameters  $M = K = T = 16$ ,  $S = 32$ ,  $N \in \{8,16\}$  and the SNR is within  $[0,30]$  dB. Figure 4 gives the NMSE curves for the estimation of  $\mathbf{H}^u$  and  $\mathbf{H}^{bs}$  separately by the KRF and BALS algorithms. The performance of the LS estimator is not available in this scenario since it can only estimate the cascade channel. The scaling ambiguity that is inherent in both the KRF and BALS algorithms separate estimation of  $\mathbf{H}^u$  and  $\mathbf{H}^{bs}$  is dealt with normalizing the first column of  $\mathbf{H}^u$  to an all-ones vector so that the first column  $\hat{\mathbf{H}}^u$  gives the scaling coefficients [13-14]. We can see from Figure 4 that the performances of the KRF and the BALS algorithm are very close to each other except for  $N = 16$  at 0 dB SNR where the BALS is better than the KRF in estimating  $\mathbf{H}^u$ . The estimation accuracy for  $\mathbf{H}^{bs}$  is better than that of  $\mathbf{H}^u$  for both tensor-based approaches. Doubling of  $N$  from 8 to 16 causes approximately a 2.5 dB loss in SNR (Figure 4) due to the increased dimension of the channel coefficient vector.

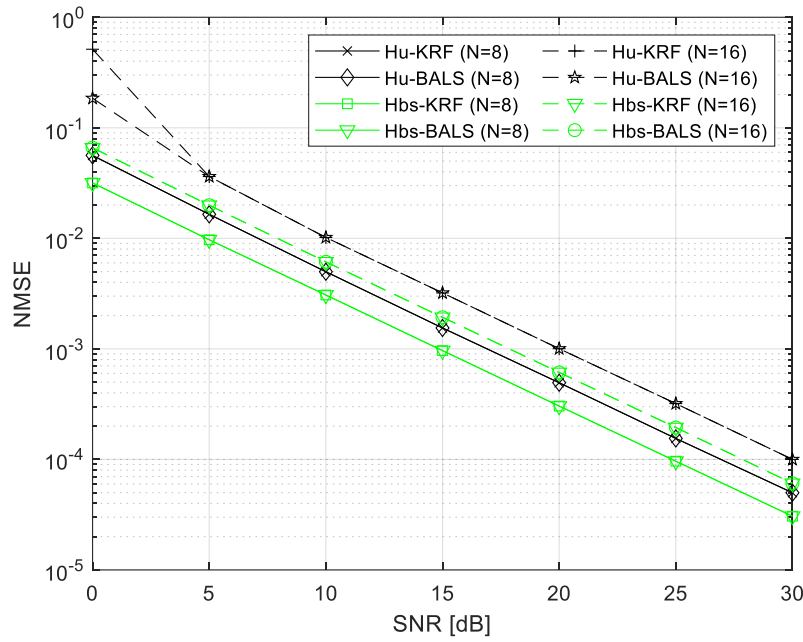


Figure 4 NMSE of  $\hat{\mathbf{H}}^{bs}$  and  $\hat{\mathbf{H}}^u$  for  $M = K = T = 16, S = 32, N \in \{8, 16\}$ .

The NMSE plots for the estimation of the cascade channel,  $\mathbf{H}^{cascade}$ , are shown in Figure 5. The LS estimator, computationally less involved, performs the worst compared to KRF or BALS. Both tensor-based channel estimation methods improve the LS estimator by a 10 dB SNR margin. The performances of the KRF and BALS algorithms are indistinguishable regarding the cascade channel NMSE. Increasing  $N$  from 8 to 16 results in approximately 2.5 dB loss in SNR for both the LS and the tensor-based approaches.

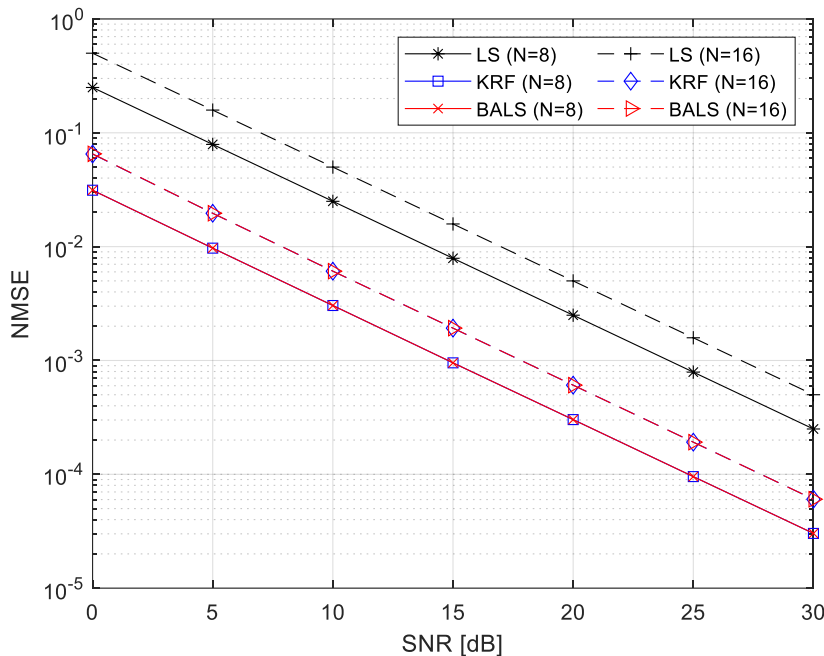


Figure 5 NMSE of  $\hat{\mathbf{H}}^{cascade}$  for  $M = K = T = 16, S = 32, N \in \{8, 16\}$ .

### 5. Discussion

Since the LS estimator can be implemented using the FFT algorithm, our proposed KRF method, which takes the LS estimator as its input, has a computational advantage over the KRF method of [12], which requires a bilinear filtering step. While the

numerical results of [12] are given for only a single-user scenario, our numerical results, like those in [11,13-15], are evaluated for multiple users. We use the correlated Rayleigh fading model, which is a better model for the spatial correlation due to the RIS elements in the uplink channels. Our numerical results show its impact on the channel estimation performance unlike the rest of the literature on tensor modeling methods [12-15], which include only the results for the uncorrelated Rayleigh fading. Also, the numerical results of [13] and [15] only focus on applying the BALS algorithm and do not include the KRF method. [16] investigates the application of tensor-based channel estimation methods for double RIS-aided systems, unlike the single RIS-aided systems considered by our paper and the rest of the literature [12-15]. Compared to [14], we do not cover the achievable sum rates for the downlink communication using the channels estimated by the proposed methods in our paper. As a future work, the rate of each user can be investigated for different RIS matrix designs and precoding schemes such as maximum ratio transmission, zero forcing, and minimum mean square error schemes.

## 6. Conclusion

We presented applying two tensor-based channel estimation methods, KRF and BALS, to the uplink of a RIS-aided multi-user MISO communication system. The derivations, identifiability conditions, and complexities of the algorithms are given in detail. The performances of the tensor-based algorithms are numerically compared against the baseline conventional LS estimation for the correlated Rayleigh fading channel model. Numerical results show that while the performances of both tensor-based are on the same level concerning each other, both improve upon the LS estimator by a 10-dB margin. It is observed that all estimators lose 2.5 dB in SNR when the number of RIS elements is doubled.

## References

- [1] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah and C. Yuen, "Reconfigurable Intelligent Surfaces for Energy Efficiency in Wireless Communication," in *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157-4170, Aug. 2019.
- [2] M. Di Renzo *et al.*, "Smart Radio Environments Empowered by Reconfigurable Intelligent Surfaces: How It Works, State of Research, and The Road Ahead," in *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450-2525, Nov. 2020.
- [3] W. U. Khan, E. Lagunas, A. Mahmood, B. M. ElHalawany, S. Chatzinotas and B. Ottersten, "When RIS Meets GEO Satellite Communications: A New Sustainable Optimization Framework in 6G," *IEEE VTC2022-Spring*, Helsinki, Finland, 2022, pp. 1-6.
- [4] YILMAZ Ü, Güler Ü "On Orbit Demonstration of Pointing Accuracy of Ground Antennas by a Flying GEO Satellite." *Sakarya University Journal of Computer and Information Sciences (Online)*, 6, ss.32 - 36, 2023.
- [5] Elorbany K, BAYILMIS C, BALTA S "A Smart Socket Equipped With IoT Technologies for Energy Management of Electrical Appliances." *Sakarya University Journal of Computer and Information Sciences (Online)*, 4, ss.347 - 353, 2021.
- [6] ARAT F, Demirci S "Experimental Analysis of Energy Efficient and QoS Aware Objective Functions for RPL Algorithm in IoT Networks." *Sakarya University Journal of Computer and Information Sciences (Online)*, 4, ss.192 - 203, 2021.
- [7] M. Di Renzo *et al.*, "Reconfigurable Intelligent Surfaces vs. Relaying: Differences, Similarities, and Performance Comparison," in *IEEE Open J. Commun. Soc.*, vol. 1, pp. 798-807, 2020.
- [8] T. L. Jensen and E. De Carvalho, "An Optimal Channel Estimation Scheme for Intelligent Reflecting Surfaces Based on a Minimum Variance Unbiased Estimator," in *Proc. IEEE ICASSP*, Barcelona, Spain, 2020, pp. 5000-5004.
- [9] Q. -U. -A. Nadeem, H. Alwazani, A. Kammoun, A. Chaaban, M. Debbah and M. -S. Alouini, "Intelligent Reflecting Surface-Assisted Multi-User MISO Communication: Channel Estimation and Beamforming Design," in *IEEE Open J. Commun. Soc.*, vol. 1, pp. 661-680, 2020.
- [10] R. V. Şenyuva, "Channel Estimation for RIS aided MISO System," *SIU*, Istanbul, Turkiye, 2023, pp. 1-4.
- [11] Z. Wang, L. Liu and S. Cui, "Channel Estimation for Intelligent Reflecting Surface Assisted Multiuser Communications," in *IEEE WCNC*, Seoul, Korea (South), 2020, pp. 1-6.
- [12] G. T. de Araújo, A. L. F. de Almeida and R. Boyer, "Channel Estimation for Intelligent Reflecting Surface Assisted MIMO Systems: A Tensor Modeling Approach," in *IEEE J. Sel. Topics in Signal Process.*, vol. 15, no. 3, pp. 789-802, April 2021.
- [13] L. Wei, C. Huang, G. C. Alexandropoulos and C. Yuen, "Parallel Factor Decomposition Channel Estimation in RIS-Assisted Multi-User MISO Communication," in *IEEE SAM*, Hangzhou, China, 2020, pp. 1-5.
- [14] L. Wei, C. Huang, G. C. Alexandropoulos, C. Yuen, Z. Zhang and M. Debbah, "Channel Estimation for RIS-Empowered Multi-User MISO Wireless Communications," in *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 4144-4157, June 2021.
- [15] C. Beldi, A. Dziri, F. Abdelkefi and H. Shaiek, "PARAFAC Decomposition based Channel Estimation for RIS-aided Multi-User MISO Wireless Communications," in *IWCMC*, Marrakesh, Morocco, 2023, pp. 1537-1542.
- [16] K. Ardah, S. Gherekhloo, A. L. F. de Almeida and M. Haardt, "Double-RIS Versus Single-RIS Aided Systems: Tensor-

- Based MIMO Channel Estimation and Design Perspectives," in *Proc. IEEE ICASSP*, Singapore, 2022, pp. 5183-5187.
- [17] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455-500, 2009.
- [18] P. Comon, X. Luciani, and A.L.F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometrics*, vol. 23, no. 7-8, pp. 393-405, 2009.

**Conflict of Interest Notice**

The author declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

**Availability of data and material**

Not applicable

**Plagiarism Statement**

This article has been scanned by iThenticate™.



# High-Capacity Data Processing with FPGA-Based Multiplication Algorithms and the Design of a High-Speed LUT Multiplier

Kenan Baysal <sup>1</sup> , Deniz Taşkın <sup>2</sup> 

<sup>1</sup> Information Management, Hayrabolu Vocational High School, Tekirdağ Namık Kemal University Tekirdağ/Türkiye

<sup>2</sup> Computer Engineering, Engineering Faculty, Trakya University Edirne/Türkiye



## Corresponding author:

Kenan Baysal, Tekirdağ Namık Kemal University, Hayrabolu Vocational High School, Tekirdağ/Türkiye  
E-mail address:  
[kbaysal@nku.edu.tr](mailto:kbaysal@nku.edu.tr)

Submitted: 04 January 2023  
Revision Requested: 16 October 2023  
Last Revision Received: 04 November 2023  
Accepted: 07 November 2023  
Published Online: 07 November 2023

Citation: Baysal K. and Taşkın D. (2023). High-Capacity Data Processing with FPGA-Based Multiplication Algorithms and the Design of a High-Speed LUT Multiplier *Sakarya University Journal of Computer and Information Sciences*. 6 (3) <https://doi.org/10.35377/saucis...1229353>

## ABSTRACT

Encryption algorithms work with very large key values to provide higher security. To process high-capacity data in real time, we need advanced hardware structures. Today, compared to previous design methods, hardware solutions can be designed more easily using Field-Programmable Gate Arrays (FPGAs). Over the past decade, FPGA speeds, capacities, and design tools have been improving. Thus, the hardware that can process high-capacity data can be designed and produced with lower costs. This study aims to create the components of a high-speed arithmetic unit that can process high-capacity data and be used for FPGA encoding algorithms.

In this study, multiplication algorithms were analyzed. High-capacity adders that constitute high-speed multiplier and look-up tables were designed using Very High-Speed Integrated Circuit Hardware Description Language (VHDL). The designed circuit/multiplier was synthesized with ISE Design Suite 14.7 software. Simulation results were obtained using the ModelSIM and ISIM programs.

**Keywords:** FPGA, VHDL, look up table, multiplication, adder

## 1. Introduction

As the computers reach the physical limits of the power of arithmetic operations, the command structures of the processors and how they process these commands have become significant [1]. The operations performed by the processors are based on arithmetic and logic commands [2]. The speed at which arithmetic operations, such as addition and multiplication, directly affect the processor's data processing capacity.

Since FPGA has been used for electronic circuit designs created using transistors, complex circuit designs have become easily and quickly achievable. Since companies like Altera, Xilinx, Actel, etc., started FPGA production and developed more easy-to-use design and simulation tools and equipment in the mid-1980s, the lower-level designers began to program their own FPGA hardware on tighter budgets.

In this study, we analyzed the high-speed multiplication methods, which have a significant role in the data processing capacity and speed of the computers, to create sub-units of the high-speed arithmetic unit. A high-speed expandable adder was designed. A high-speed LUT multiplier, expandable through look-up tables, was designed using a high-speed adder. A 32x32-bit long LUT multiplier was modeled with VHDL and applied to FPGA hardware.

The study aims to design efficient and cost-effective hardware solutions for processing high-capacity data in real time, particularly in encryption algorithms requiring large key values. The primary objective of the study is to create the



components of a high-speed arithmetic unit that can process high-capacity data, which are designed to be used in FPGA-based encryption algorithms. The study focuses on analyzing multiplication algorithms and designing high-capacity adders for building high-speed multipliers. Additionally, look-up tables are designed using Very High-Speed Integrated Circuit Hardware Description Language (VHDL). The entire circuit and multiplier design is synthesized using ISE Design Suite 14.7 software, and simulation results are obtained through ModelSIM and ISIM programs. The study aims to advance the field of hardware design for encryption algorithms, offering efficient and cost-effective solutions for processing large key values in real time. The study's findings can contribute to developing more secure and efficient encryption algorithms, essential for protecting sensitive data in various healthcare, finance, and government applications.

## 2. Literature Review

Abd-Elkader et al. developed a design model to improve the performance of the hardware structure of the Montgomery Modular Multiplier. They used VHDL to code their proposed model and found that it consumed fewer resources on the FPGA. Their study showed that the proposed model could operate more efficiently than the existing hardware structure [3].

Behl et al. developed a multiplier circuit that reduces carry propagation time and performs faster calculations using the Redundant Binary Signed Digit number system. They encoded this circuit in VHDL and applied it on an FPGA. Using the VIVADO multiplier in their design, they observed a significant reduction in the number of Look-up tables used. This approach can potentially improve the efficiency of digital electronics, such as binary multipliers, and could be useful in various applications. [4].

Sakali et al. have introduced a new error analysis approach to reduce hardware redundancy in existing fault-tolerant techniques. This approach is particularly useful for reducing the additional hardware resources that Triple Modular Redundancy (TMR) requires. They applied the proposed approach to a multiplier circuit and found a 48% reduction in hardware resource usage. [5].

Malathi et al. have explored a new approach to enhancing image quality and resolution on FPGA using deep learning and Fast Fourier Transform (FFT) techniques. They tackled this approach in three main stages: noise reduction, segmentation, and resolution enhancement. This method achieved low power consumption, minimal latency, and high efficiency. This approach can potentially improve the quality of images in various applications, including medical imaging and surveillance. Deep learning and FFT techniques allow for greater flexibility and customization in image processing, making it a valuable tool for image enhancement. Applying this approach to FPGA allows for efficient and fast processing of images, making it a promising technology for real-time image processing. [6].

Bianchi et al. have proposed an architecture for a new Vedic multiplier that employs the 'Urdhava-tiryakbhyam' method and implemented it on an FPGA. They evaluated this approach regarding hardware performance, LUT (Look-Up Table) sizes, and propagation delay. The results have shown performance equal to or better than existing approaches in the literature. [7].

Özcan et al. have proposed a fast Montgomery multiplier design for modern FPGAs. Their designs implemented on Virtex-7 have provided competitive performance and significant savings in FPGA logic resources.[8].

Morales-Sandoval et al. discussed using Field-Programmable Gate Arrays (FPGAs) to implement Montgomery Multiplication in public-key encryption algorithms like RSA and Elliptic Curve Cryptography (ECC). Their study focused on area-performance trade-offs, tested different architectures, and compared their efficiencies with previous FPGA Montgomery multipliers. [9].

## 3. Multiplication Algorithms

All arithmetic operations made on a computer are based on addition. Multiplication, the basis of many scientific practices, such as encoding, decoding, signal processing, etc., is also realized based on shifting and addition. The structure of multiplication algorithms is the most significant factor that affects the computer's performance, especially when we work on high-capacity data in scientific programs. Approximately 9% of this scientific program consists of multiplication [10]. Thus, a wide array of multiplication algorithms has been developed in the years following computer technology developments. The multiplication algorithm, which will be selected according to the size of the data and suitability of the algorithm, can increase the speed of the process without enhancing the performance of the computer's processor.

For an  $n$  bit-long number of  $X$  and  $m$  bit-long number  $Y$  in an ordinary multiplication, as seen in Equation 1, the equation can indicate multiplication [5].

$$P(m+n) = X(n)Y(m) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} x(i)y(j)2^{i+j} \quad (1)$$

### 3.1. Sequential Shift and Add Multiplication

In this method, we start with the lowest-weight bit rate of the multiplier and obtain results by shifting the multiplicand to the left and summing up at each clock stroke, depending on whether the bit rate is either 1 or 0. Although this method is based on a very simple logic, the performance is adversely affected if the data size is too large.

### 3.2. Booth Algorithm

In the booth algorithm, the numbers in both marked bases are multiplied based on the add-shift principle by comparing adjacent bits through a reciprocating operation of 2. The advantage of this method, when compared to other multiplication algorithms, is that there are fewer additions and multiplications. [11].

### 3.3. Wallace Tree

Wallace tree multiplication is an effective method that may be preferred at hardware-level multiplication of two unmarked integer numbers. This method was developed by computer scientist Chris Wallace in 1964 [12].

In the Wallace Tree method, the partial fractions are added up as a tree until they reach the last two partial fraction rows that will be added at the final stage. The speed of this algorithm is inversely proportional to the number of bits to be processed. The waste of unused space and its complex structure are some of the primary problems of this algorithm [13]. Its structure is not suitable for rapid transit logic in hardware structure. It is also not as fast as the transit structure within modern FPGAs [14].

### 3.4. Array Multiplication

It is a common multiplication algorithm with a parallel index structure based on the add-shift principle. Partial result rates are obtained by multiplying the respective 1-bit digits of the multiplier by the multiplicand and adding up by shifting per the bit order. Although its structure is slow as an algorithm, this method is often preferred for its parallel-moving index structure is systematic and easy to position on the hardware [15].

### 3.5. Karatsuba Multiplication Algorithm

This is an effective multiplication method for multiplying two large numbers. The numbers to be multiplied are divided into sub-groups. The results are obtained by adding the results obtained by multiplying these sub-groups. This method provides a great advantage in multiplying large numbers [16]. While the length of the operation is  $O(n^2)$  in traditional methods, the length becomes  $O(n^{\log(2/3)})$  with the Karatsuba method [17].

## 4. Material and Methods

### 4.1. Look-Up Table Multiplication

This efficient and fast multiplication method is suitable for hardware structures with strong memory units. Look-up table multipliers are generated using block memory units, which store multiplication results corresponding to all input values. The results are obtained in a shorter period since no real multiplications are performed through this method. However, the biggest disadvantage of this method is that the look-up table size increases incrementally as the number increases. Since the memory unit includes all multiplication possibilities, some data may take up space in memory even though they are never used. However, in cases where continuous and unstable signal routing, such as encoding, decoding, image processing etc., is present, the real-time multiplication performance is very high.

If the input data length of the look-up table is k-bit in multiplication, the potential number of results contained in the look-up table would be  $2^{2k}$ . However, when at least one input is zero,  $2^{2k}/(2 \cdot 2^k - 1)$  number of results would be zero. This enables us to send a zero value directly to the output without taking up space in the look-up table. Thus, depending on the state of the input bit length from the memory unit, it would be possible to save one  $2^{2k}/(2 \cdot 2^k - 1)$  of space.

Multiplying directly with the look-up table in large data sizes may not be practical due to the space the look-up table takes up in the memory unit. In this case, the duration of the standard multiplication may be shortened by using a partial look-up table.

The number is divided into factions by the bit length of the look-up table, and partial results are obtained according to the result of the look-up table. K-bit left-shift is performed according to the bit length and the number of steps in the look-up table when the partial results are obtained. While the multiplication is performed in the standard shift-add form, the result

can be seen directly in the look-up table instead of obtaining results through performing the multiplication physically. With this method, the k-bit length of the operational cycle would be saved.

### 4.2. n Bit High-Speed LUT Multiplier Design

To multiply n bit-long numbers A and B, the numbers are divided into k bit-long  $\frac{n}{k}$  fractions. Being  $i \leq \frac{n}{k}$  and  $j \leq \frac{n}{k}$ , partial results are obtained by using  $A_j$  and  $B_i$  number fractions look-up table at each t time. The partial results are summed up, and the multiplication result is found. The adder used at the multiplier is obtained by increasing and expanding a 1-bit adder accordingly. It is now possible to perform a 2n bit-long addition at a single clock stroke with the advantage it provides.

In a multiplication performed with a lookup table by using an expandable adder, the relationship between the multiplication by the number of operational steps (t) and the bit length (n) of the multiplicands and fraction bit-length (k) can be expressed by the Equation 2.

$$t = 2^{2[\log_2(n) - \log_2(k)]} \tag{2}$$

The look-up table size should be calculated by considering the capacity of the FPGA chip where the multiplier will be applied. The number of possibilities of the look-up table can be expressed as in Equation 3.

$$LUT \text{ possibility number} = 2^{2k} \tag{3}$$

Because the output will be zero if the input is zero regardless of the state of the other input, the number of possibilities may be reduced as seen in Equation 4.

$$LUT \text{ possibility number} = 2^{2k} - 2.2k - 1 \tag{4}$$

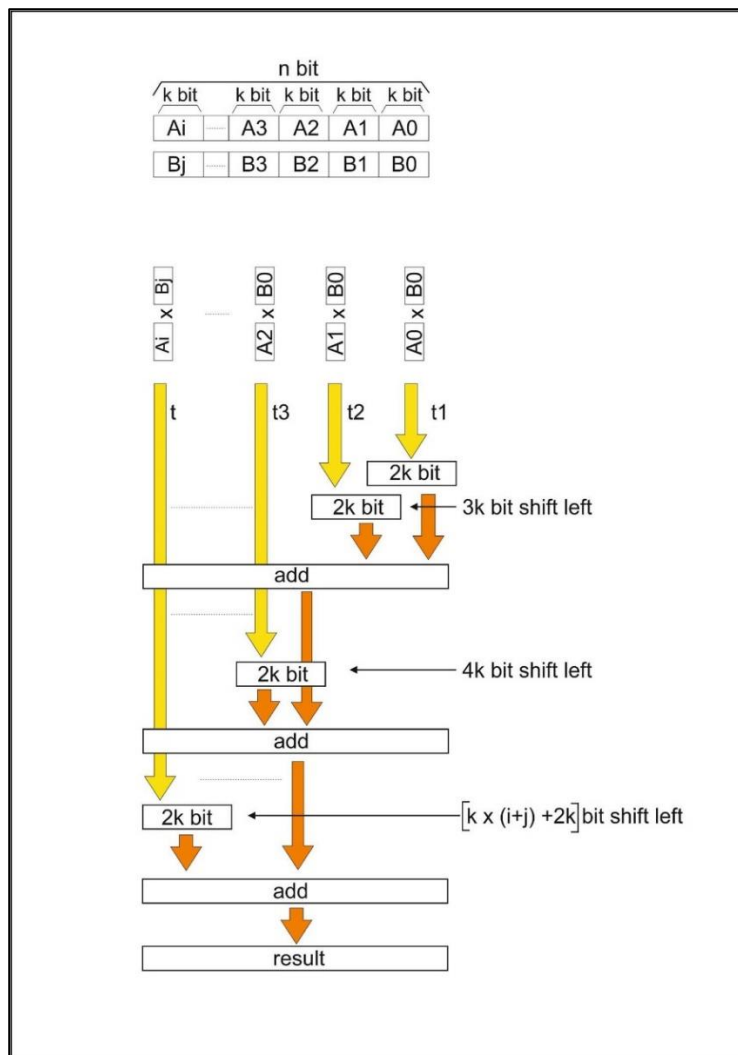


Figure 1 Shifting and adding.



Figure 1 shows how many bits of the  $A_j$  and  $B_i$  fractions were shifted to the left according to the current condition before the addition.

Being  $t_a \leq t$ , according to  $t_a$  current condition, the relationship between fraction  $j$  of  $k$  bit-long  $n$ -bit  $A$  multiplicand data and  $i$  fraction of  $n$ -bit  $B$  multiplier data at LUT inputs can be expressed as in the Equation 5 and the Equation 6.

$$j = t_a \left( \text{mod } \frac{n}{k} \right) \quad (5)$$

$$i = \left[ t_a - t_a \left( \text{mod } \frac{n}{k} \right) \right] \left( \text{mod } \left( \frac{n}{k} - 1 \right) \right) \quad (6)$$

Thus, in the case of  $t_a$ , the multiplicand fractions of  $A$  and  $B$  data can be expressed as in Equation 7 and Equation 8.

$$St_a \rightarrow A_j \times B_i \quad (7)$$

$$St_a \rightarrow A \left[ t_a \left( \text{mod } \frac{n}{k} \right) \right] \times B \left[ t_a - t_a \left( \text{mod } \frac{n}{k} \right) \left( \text{mod } \left( \frac{n}{k} - 1 \right) \right) \right] \quad (8)$$

Table 1 Look-up table and number of steps in different fraction lengths for 1024 bits of data input

k	8	32	128	512
LUT	65536	$1.8 \times 10^{19}$	$1.15 \times 10^{77}$	$1.79 \times 10^{308}$
Clock Cycle	16384	1024	64	4

### 4.3. Performing 32-Bit High-Speed LUT Multiplier with Vhdl

As seen in Figure 2, the LUT multiplier that will be created through VHDL has three main parts.

1. A 64-bit adder that has been created with the expansion of 1-bit adders,
2. ROM that consists of a look-up table,
3. A basic circuit that divides the input data into fractions reads the results of the multiplication of relevant fractions on the look-up table and sends them to the adder.

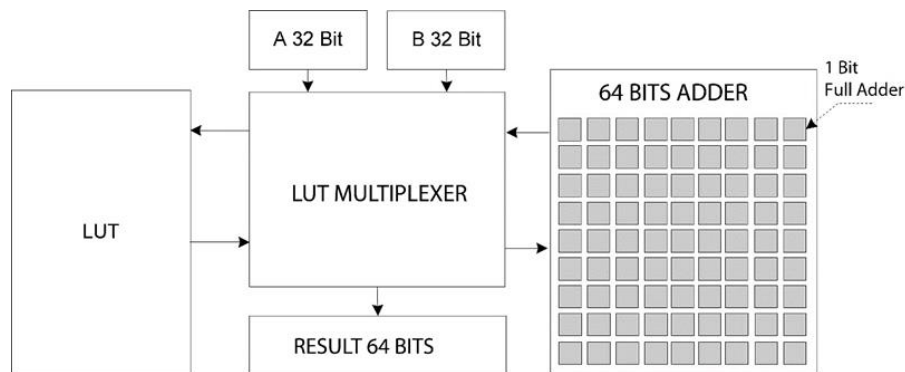


Figure 2 Basic block diagram of 32-bit multiplier created with VHDL.

### 4.4. 64-Bit High-Speed Adder Design

Since multiple addition is the basis of multiplication, the adder's speed is critical in designing a multiplier. In a standard addition, the duration of the operation increases as the number of digits increases. In this multiplier, designed to multiply high-capacity numbers quickly, a standard adder cannot provide desired results in larger data sizes. To solve this problem that we encounter in adding large data, an adder was designed by properly expanding 1-bit full adders by the input data size.

The biggest advantage of this adder is that it can add input and output at a single clock stroke using 1-bit full adders connected in parallel. Another advantage of this circuit is that its size can be expanded to the desired capacity. While its capacity can be expanded to the desired size depending on the size of the FPGA hardware, the addition can be performed within the same period, i.e., within a single clock stroke.

Since the amount of data that can be processed within the same processing time can be expanded, the size of the multiplier can be increased to the desired level.

Figure 3 shows simulation results of a 1024-bit adder created with the same method. It is observed that the result is obtained within a single clock pulse. As long as the FPGA hardware has sufficient capacity, the input sizes can be increased to a desired amount without changing the time spent for addition. In this case, the sizes of designed multipliers can be expanded to desired sizes.

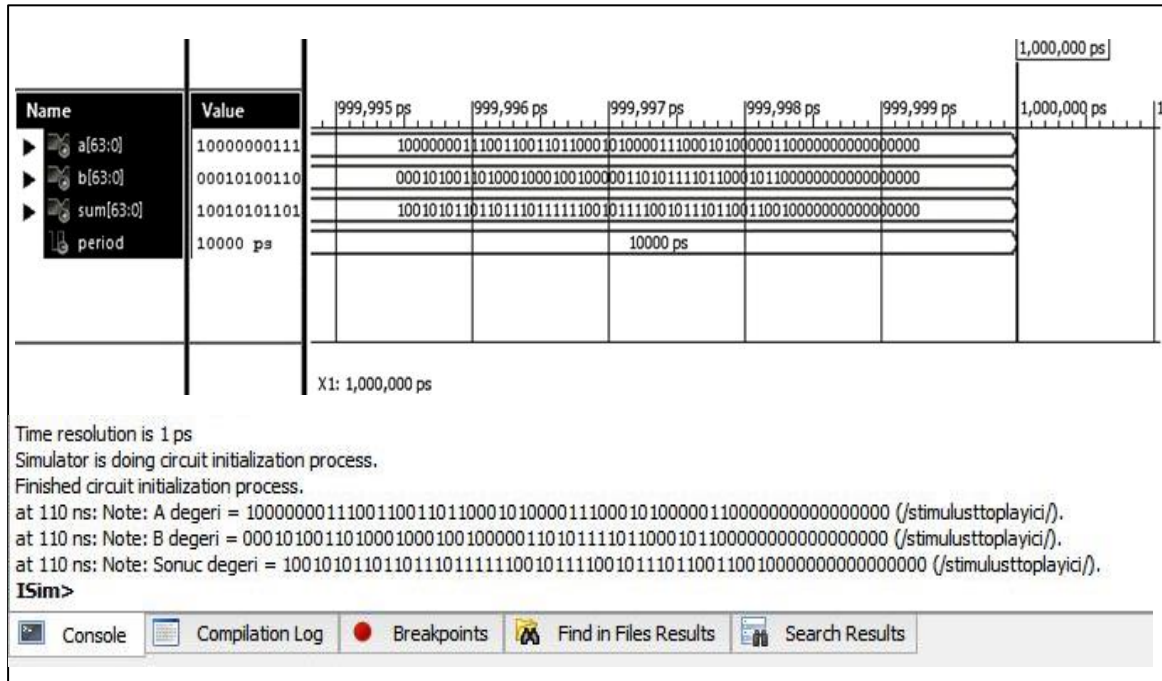


Figure 3 Results of simulation obtained through ISIM program of 64-bit adder.

#### 4.5. Look-Up Table Design

The look-up table stores all potential results of previously calculated multiplier and multiplicand numbers. In this multiplier, this circuit takes up the most space on FPGA. Thus, the capacity of the FPGA hardware to be used when designing the look-up table should be considered.

If we want to design a 32-bit input-long LUT to multiply two 32-bit binary numbers as an integral, the LUT possibility number will be 264.

In compliance with the capacity of Xilinx Virtex 5 XC5VLX50T FPGA hardware, 32 bit-long inputs may be divided into  $k=8$  bit-long fractions. In this case, the LUT possibility number will be 216.

In this case, the time equation required for obtaining the result of the multiplication can be found from Equation 9 as;

$$t = 2^{2^{\lceil \log_2(32) - \log_2(8) \rceil}} = 2^{2^{[5-3]}} = 2^4 = 16 \text{ steps.} \quad (9)$$

The entire  $2^{2^k}$  possibility that may be implemented on  $k$  bit-long inputs of the circuit has been tested in  $2^{2^k}$  of time. According to the results of the simulation, the random input values of the circuit showed that it gave results within a single clock pulse.

Since no physical multiplication was performed with the LUT circuit, the results can be obtained without additional addressing circuits according to the values applied to their inputs.

#### 4.6. High-Speed LUT Multiplier Design

In a multiplier designed to multiply two 32-bit numbers, an 8-bit-input look-up table and a 64-bit adder that will add the values from the look-up table at each step were used.

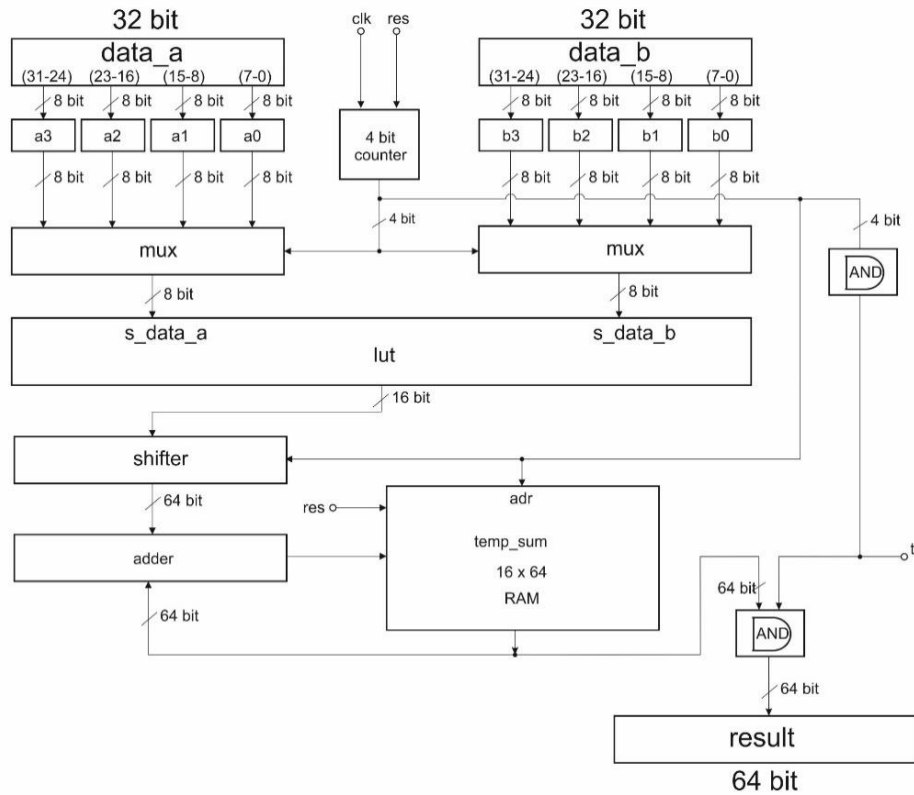


Figure 4 Block diagram of 32-bit LUT Multiplier

Algorithm 1 Expandable multiplication circuit structure with LUT

```

1  case c_state is
2  when sr =>
3      result <= (others=>'0');
4      t<='0';
5      b_signal <= (others=>'0');
6      a_signal <= (others=>'0');
7      s_data_a <= data_a(k downto 0);
8      s_data_b <= data_b(k downto 0);
9      n_state <= S0;
10 when Sta =>
11     s_data_a<=data_a [k.ta(mod  $\frac{n}{k}$ ) + (k - 1)  downto  k.ta(mod  $\frac{n}{k}$ )];
12     s_data_b<=data_b [k.[ta - ta(mod  $\frac{n}{k}$ )](mod ( $\frac{n}{k} - 1$ )) + (k - 1)  downto  k.[ta - ta(mod  $\frac{n}{k}$ )](mod ( $\frac{n}{k} - 1$ ))] ;
13     a_signal[ k(i+j)+2k  downto  k(i+j) ] <= lut_result;
14     b_signal <= total_signal;
15     t='0';
16     n_state<= Sta + 1;
17 when Sta(max) =>
18     result <= total_signal;
19     t<='1';
20     b_signal <= (others=>'0');
21     a_signal <= (others=>'0');
22     s_data_a <= (others=>'0');
23     s_data_b <= (others=>'0');
24     n_state <= sr;
25 when others =>
26     b_signal <= (others=>'0');
27     a_signal <= (others=>'0');
28     s_data_a <= (others=>'0');
29     s_data_b <= (others=>'0');
30     result <= (others=>'0');
31     t<='0';
32 end case;

```



The FPGA chip and the location of the circuit on the FPGA chip determine the circuit's performance at high speeds. When the transistors within FPGA delay data transmission, this has a negative impact on the performance of the circuit at high speeds. Accordingly, the maximum frequency of the circuit is determined by the longest distance the data between input and output will follow. The register was attached to the input and outputs of the circuit when analyzing timing. The maximum frequency was determined based on the time it takes for data to travel between two registers.

Table 2 shows synthesizing processes performed on various FPGA chips with ISE, Quartus and Vivado software and the maximum frequency values.

Table 2 The timing analysis chart of the circuit between two recorders

Program	ISE 14.7			Quartus II 14.1			Vivado v2014.4	
	Virtex 5	Virtex 6	Kintex 7	Cyclone IV	Cyclone V	Cyclone V	Kintex 7	Artix 7
FPGA	xc5vlx50t-2ff1136	xc6vlx75t-2ff784	xc7k70t-fbg676	ep4cgx150d-f3117ad	5cgxfc7d7-f27c8	5cgxfc7d7f-31c8	xc7k160t-ffg676-2l	xc7a200t-fg1156-3
RR (ns)	1.498	1.171	0.995	1.975	2.763	1.640	1.409	1.592
Fmax (Mhz)	667.557	853.97	1005.03	506.33	361.93	609.76	709.72	628.14

## 5. Discussion and Conclusion

In this study, a high-speed multiplier was designed by using look-up tables. The look-up circuit stored all previously calculated multiplication results for two 8-bit numbers, and the logic circuit that multiplied two 32-bit long numbers was synthesized on Virtex 5 xc5vlx50t FPGA hardware. In the results of the simulation performed on ModelSIM and ISIM, it was observed that the multiplication of two 32-bit numbers gave results in 16 cycles through multiplication by division into partial fractions. Since the high-speed expandable adder, designed to add partial results, could add the partial results in a single cycle, it allows the circuit to perform arithmetic operations quickly. The maximum frequency value of the circuit was calculated as 667.557 Mhz in time analyses performed via the ISE program.

This circuit can be used as a sub-unit of an arithmetic unit or as a circuit sub-unit in encoding applications. It can provide high-speed processing for larger capacities in real-time signal processing. In addition, since the circuit uses look-up tables, it consumes less power as it does not perform physical multiplication.

Table 3 Comparison of Time Complexity

	Time Complexity	Number of steps for two 32-bit numbers
Standard Multiplication	$O(n^2)$	1024
Karatsuba	$O(n^{1.585})$	243
Shift/Add	$O(n)$	32
High-Speed LUT	$O(2^{2(\log_2(n) - \log_2(k))})$	16

Although the algorithm built for disintegration numbers in this circuit is similar to the Karatsuba algorithm, they differ in the multiplication of fractions and addition of partial results.

Depending on the unit where the multiplier will be used, LUT fractions can be expanded, and a look-up table can be recreated again. However, the biggest problem of the circuit here is the size of LUT. As FPGA capacity continues to grow in the coming years, it will be possible to perform multiplications in shorter time frames by creating larger lookup tables.

This study presents a new, innovative approach to high-speed multiplication using look-up tables and FPGA hardware. This approach's numerous benefits include increased speed, power efficiency, versatility, scalability, and algorithmic innovation. These benefits can have a significant impact on various applications in the field of digital design and arithmetic units.

## References

- [1] R. W. Keyes., "Physical Limits of Silicon Transistors and Circuits", *Reports on Progress in Physics*, vol. 68, no. 12, 2005, doi: 10.1088/0034-4885/68/12/R01
- [2] B. Parhami, *Computer Arithmetic Algorithms and Hardware Designs Secon Edition*, Oxford University Press, New York USA, 2010, ISBN 978-0-19-532848-6
- [3] A. A. H. Abd-Elkader, M. Rashdan, E. A. M. Hasaneen and H. F. A. Hamed, "Efficient implementation of Montgomery modular multiplier on FPGA," *Computers and Electrical Engineering*, vol. 97, 2022, doi: <https://doi.org/10.1016/j.compeleceng.2021.107585>
- [4] A. Behl, A. Gokhale, N. Sharma, "Design and Implementation of Fast Booth-2 Multiplier on Artix FPGA", *Procedia*

- Computer Science*, vol. 173, pp. 140-148, 2020, doi: <https://doi.org/10.1016/j.procs.2020.06.018>
- [5] R. K. Sakali, S. Veeramachaneni, N. M. Sk, "Preferential fault-tolerance multiplier design to mitigate soft errors in FPGAs", *Integration*, vol. 93, 2023, doi: <https://doi.org/10.1016/j.vlsi.2023.102068>
- [6] L. Malathi, A. Bharathi, A.N. Jayanthi, "FPGA design of FFT based intelligent accelerator with optimized Wallace tree multiplier for image super resolution and quality enhancement", *Biomedical Signal Processing and Control*, vol. 88, part B, 2024, doi: <https://doi.org/10.1016/j.bspc.2023.105599>
- [7] V. Bianchi, I. D. Munari, "A modular Vedic multiplier architecture for model-based design and deployment on FPGA platforms", *Microprocessors and Microsystems*, vol. 76, 2020, doi: <https://doi.org/10.1016/j.micpro.2020.103106>
- [8] E. Özcan, S. S. Erdem, "A fast digit based Montgomery multiplier designed for FPGAs with DSP resources", *Microprocessors and Microsystems*, vol. 62, pp. 12-19, 2018, doi: <https://doi.org/10.1016/j.micpro.2018.06.015>
- [9] M. Morales-Sandoval, C. Feregrino-Urbe, P. Kitsos, R. Cumplido, "Area/performance trade-off analysis of an FPGA digit-serial GF(2<sup>m</sup>) Montgomery multiplier based on LFSR", *Computers & Electrical Engineering*, vol. 32, i. 2, pp. 542-549, 2013, doi: <https://doi.org/10.1016/j.compeleceng.2012.08.010>
- [10] R. S. Özbey and A. Sertbaş, "Klasik Çarpma Algoritmalarının Donanımsal Simülasyonları ve Performans Değerlendirmesi", *Inter. Conf. on Electrical and Electronics Engineering (ELECO 2004)*, pp. 303-308, 2004
- [11] A. D. Booth, "A Signed Binary Multiplication Technique", *The Quarterly Journal of Mechanics and Applied Mathematics. Math. Oxford University Press*, vol. 4, no. 2, pp. 236-240, 1951, doi: <https://doi.org/10.1093/qjmam/4.2.236>
- [12] C. S. Wallace, "A Suggestion for a Fast Multiplier", *IEEE Transactions on Electronic Computers*, vol. 13, no. 1, pp. 14-17, 1964, doi: 10.1109/PGEC.1964.263830
- [13] M. R. Kumar and G. P. Rao, "Design and Implementation Of 32 Bit High Level Wallace Tree Multiplier", *International Journal of Technical Research and Applications*, vol. 1, no. 4, pp. 86 - 90, 2013, Accessed : 29 October 2023 [Online]. Available: <https://api.semanticscholar.org/CorpusID:13022315>
- [14] J. Kulisz, J. Mikucki, "An IP-Core Generator for Circuits Performing Arithmetic Multiplication", *IFAC Proceedings Volumes*, vol. 46, i. 28, 2013, doi: <https://doi.org/10.3182/20130925-3-CZ-3023.00006>
- [15] A. J. Al-Khalili, *Digital Design and Synthesis Lecture Notes (2019)*, Accessed : 29 October 2023 [Online]. Available: [https://users.encs.concordia.ca/~asim/COEN\\_6501/elec650.html](https://users.encs.concordia.ca/~asim/COEN_6501/elec650.html)
- [16] S. Mishra and M. Pradhan, "Implementation of Karatsuba Algorithm Using Polynomial Multiplication", *Indian Journal of Computer Science and Engineering*, ISSN: 0976-5166, vol. 3, no. 1, pp 88 - 93, 2012.
- [17] R. T. Kneusel, *Numbers and Computers*, Springer, USA, pp. 136, 2015, ISBN: 978-3-319-17260-6

#### Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

#### Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

#### Availability of data and material

Not applicable

#### Plagiarism Statement

This article has been scanned by iThenticate™.



# A Novel Additive Internet of Things (IoT) Features and Convolutional Neural Network for Classification and Source Identification of IoT Devices

Aamo Iorliam<sup>1</sup> 

<sup>1</sup>Department of Mathematics & Computer Science, Benue State University, Makurdi, Nigeria



**Corresponding author:**  
Aamo Iorliam, Department of Mathematics  
& Computer Science, Benue State  
University, Makurdi, Nigeria  
**E-mail address:**  
[aamiorliam@gmail.com](mailto:aamiorliam@gmail.com)

**Submitted:** 04 September 2023

**Accepted:** 15 November 2023

**Published Online:** 15 November 2023

**Citation:** Iorliam A. (2023).  
A Novel Additive Internet of Things (IoT)  
Features and Convolutional Neural Network  
for Classification and Source Identification  
of IoT Devices. *Sakarya University Journal  
of Computer and Information Sciences*. 6 (3)  
<https://doi.org/10.35377/saucis...1354791>

## ABSTRACT

The inter-class classification and source identification of IoT devices have been studied by several researchers recently due to the vast amount of available IoT devices and the huge amount of data these IoT devices generate almost every minute. As such there is every need to identify the source where the IoT data is generated and also separate an IoT device from the other using the data they generate. This paper proposes novel additive IoT features with the CNN system for the purpose of IoT source identification and classification. Experimental results show that indeed the proposed method is very effective achieving an overall classification and source identification accuracy of 99.67 %. This result has a practical application to forensics purposes due to the fact that accurately identifying and classifying the source of an IoT device via the generated data can link organizations/persons to the activities they perform over the network. As such ensuring accountability and responsibility by IoT device users.

**Keywords:** Internet of Things (IoT), Additive IoT Features, Inter-class classification, Source identification.

## 1. Introduction

The concept of inter-class classification was described by Iorliam, Ho, Waller, and Zhao [1] to mean the classification of biometric images that are not closely related and are generated by different devices. This concept is extended to the Internet of Things (IoTs) in order to perform the inter-class classification and source identification of IoT devices based on the data they generate.

IoT device source identification is concerned with determining which device has produced particular IoT data. Source identification of devices is very important because it has the tendency to identify devices within an organization and also unauthorized devices that are connected to the network of such an organization [2,3].

The concept of “Additive IoT features” is motivated by the concept of flow size difference proposed by Iorliam [4] as a network traffic feature for the analysis and deductions from network traffic data. Flow size difference took into consideration the absolute values achieved by subtracting two adjacent flows. For the fact that subtraction and addition are associative, this paper extends this concept into the Additive IoT features, where two adjacent IoT values of a feature are added for the purpose of classification and source identification for the first time.

Convolutional Neural Network (CNN) is a powerful machine learning technique that has applications in images, network traffic analysis, document analysis, and Internet of Things, amongst several other applications. Based on its huge capabilities, it is adapted for usage in this paper. In this paper, the novel use of Additive IoT features and CNN for inter-class classification and source identification of IoT devices based on the benign data they generate is proposed.

Studies such as Bai *et al.* [5], Cvitić, Peraković, Periša, and Gupta [6], Zahid *et al.* [7], Zarzoor, Al-Jamali, and Al-Saedi [8], and Koball *et al.* [3] have proposed methods aimed at classifying IoT, however, my novel approach proposes a novel



“Additive IoT features” and achieves an accuracy that is similar or greater than the existing state-of-the-art proposed methods.

This paper contributes to the area of IoT device classification and source identification as follows:

- i. To the best of the researchers' knowledge, this is the first time Additive IoT features are proposed as a stable IoT metric that could be utilized for classification purposes.
- ii. This paper proposes the novel device classification and source identification of IoT devices based on Additive IoT features and CNN.
- iii. The novel proposed approach is very simple and free from the overhead of feature engineering.

## 2. Literature Review

Classification and source identification of IoT devices have attracted huge attention recently. Most of the literature is focused on identifying and proposing new features that can effectively be used for classification and source identification purposes. While some literature is focused on developing/utilizing machine learning approaches in performing classification and source identification of IoT devices.

In this paper, a detailed review is performed based on two areas, namely: feature extraction approaches for classification and source identification of IoT devices and Machine learning approaches for classification and source identification of IoT devices.

Bai *et al.* [5] used the flows from 15 devices categorized into 4 classes for the purpose of classifying seen and unseen IoT devices. They used the LSTM-CNN technique for the classification of IoT devices and achieved an accuracy of 74.8%.

Cvitić, Peraković, Periša, and Gupta [6] used 13 network traffic features to perform the classification of IoT devices. These devices were classified into 4 major classes using their proposed multiclass classification model and achieved an accuracy of 99.79%.

Kotak, and Elovici [2] used grayscale snapshots of payloads of TCP sessions that are exchanged between IoT devices as features. The authors used the deep learning technique to identify known IoT devices and unknown IoT devices. They achieved an accuracy of 99% for identifying known devices and 99% for detecting unknown devices using the proposed deep learning technique.

Zahid *et al.* [7] achieved optimal features by performing recursive feature elimination and utilized features of interest for their experiments. They used the hierarchical deep neural networks with the utilized features and achieved a classification accuracy of 91% for the classification of Internet of Things devices from devices that are not Internet of Things, and a classification accuracy of 91.33% for the classification of only IoT devices within a heterogeneous network.

Zaroor, Al-Jamali, and Al-Saedi [8] utilized features such as packet intermediate time among two sequential packet receptions, packet length, IP source address, IP destination address, protocol utilized by the flow, source port number, destination port number, window size, source MAC address and heights number of hop that required for each packet to reach destination. The authors proposed a spike neural network to classify IoT devices. They showed that the proposed model consumed less energy and was able to perform IoT classification with a Precision of .98, a Recall value of 0.97, and an F1-score of 0.98.

Koball *et al.* [3] used 242 features from 8 IoT devices and achieved the highest classification accuracy of 96.5% using unsupervised machine learning techniques.

From the above-reviewed literature, this is the first time additive IoT features will be proposed and fed as inputs into CNN to perform inter-class classification and source identification of IoT devices.

## 3. Methodology

This section first describes the dataset used and the preprocessing performed on the dataset. It further vividly describes the proposed Additive IoT Features for IoT device classification and source identification (AIFID). Furthermore, it explains the proposed model architecture and the evaluation metrics used in this paper.

### 3.1 Dataset and Dataset Pre-Processing

First, the study utilized the N-BaIoT dataset is made of 9 IoT devices. The 9 devices include Danmini Doorbell, Ennio Doorbell, Ecobee Thermostat, Philips B120N/10 Baby Monitor, Provision PT-737E Security Camera, Provision PT-838 Security Camera, SimpleHome XCS7-1002-WHT Security Camera, SimpleHome XCS7-1003-WHT Security Camera, and Samsung SNH 1011 N Webcam produced benign data to include 49548, 3910, 13113, 17524, 62154, 98514, 46585, 1952, and 52150 instances, respectively [9]. This traffic data collected using 9 IoT devices has also infected the dataset with Mirai



and BASHLITE. The benign N-BaIoT dataset is suitable for experimenting with the proposed AIFID model because it can aid us in performing IoT device classification and source identification.

For the pre-processing, all NULL values were dropped using the Python “dropna” method.

The “MinMaxScaler()” Python command is used to scale each element of the features used in this experiment. The preprocessed dataset is split into 70 % train and 30% test sets. The train set is used to train the CNN learning model. While the test set serves as input to test the performance of the model. The performance outcome of the model is then evaluated, and the results are presented as a confusion matrix, F1-score, accuracy, precision, and recall.

### 3.2 Additive IoT Features for IoT Device Classification and Source Identification

The additive IoT features are defined as the numeric sum of two consecutive adjacent IoT-generated data as illustrated in Table 1.

Table 1: Sample Data for Additive IoT Features

S/No	Additive_ MI_dir_L5 _weight	MI_dir_L5_ weight	Additive_ MI_dir_L5_mean	MI_dir_L5 _mean	Additive_ MI_dir_L5_variance	MI_dir_L5_v ariance
1.	2	1	414	60	0	0
2.	2.857878541	1	714.4589798	354	35.78933753	0
3.	2.857878541	1.857878541	697.4589798	360.4589798	35.78933753	35.78933753
4.	2.680222861	1	509.1409171	337	18487.44875	0
5.	4.284299211	1.680222861	306.2104017	172.1409171	32200.47372	18487.44875
6.	5.707896692	2.60407635	253.9405041	134.0694846	23432.06087	13713.02497

Additive IoT features have a background from the flow size difference proposed by Iorliam [4]. It has been proven from the literature that the flow size difference (flow subtraction) is a stable feature for network traffic classification and intrusion detection [4,] Iorliam *et al.* [10]. In our study, the “additive IoT features” are introduced for the first time for IoT classification and source identification purposes. This is inspired by the fact that addition and subtraction both share a closure property.

For that reason, if Iorliam [4] and Sethi *et al.* [11] used the flow size difference as features for network traffic analysis and intrusion detection purposes, and achieved their targeted goal of intrusion detection, then our proposed additive IoT features for IoT device classification and source identification would be very efficient and effective.

### 3.3 CNN for IoT Device Classification and Source Identification

Generally, CNN in terms of performance is very efficient in solving machine learning tasks [12].

CNN has proven over the years to be very effective in classification tasks especially when the datasets are huge. In our study, we chose the CNN due to the fact that it has the tendency to automatically select the best features in a particular dataset and has proven to achieve high accuracies.

The steps are as follows:

- i. Get ALL the 115 statistical features from the IoT device dataset,
- ii. Calculate the IoT features addition (additive IoT features),
- iii. Feed values from (ii) into the CNN classifier and
- iv. Perform classification.

The 9-class classification and source identification are performed by merging the 115 benign IoT features for all the 9 IoT devices and labeling them from 0 to 8 as class labels. These features are fed into the CNN as shown below:

- i. 70% of the IoT dataset is used for training, while 30% of the dataset is used for testing.
- ii. The first layer used in this experiment is the sequential model “sequential ()” which allows the network to be built layer by layer and it’s well suited for our experiment.

- iii. 480 neurons were used in the first hidden layer with 115 input parameters. The rectified linear activation function (ReLU) is first chosen due to its ability to achieve higher performance and again it is non-linear.
- iv. Other two dense layers were added which had 240 and 120 neurons, accordingly.
- v. The model ended with 9 dense layers, no activation, and a sigmoid activation function.
- vi. The model is compiled using binary cross entropy as loss, the adam as an optimizer, and accuracy as the metrics.
- vii. 1000 epochs were used in this experiment with a batch size of 128.

### 3.4 Model Architecture

The proposed Additive IoT Features for IoT Device Classification and Source Identification (AIFID) are presented in Figure 1.

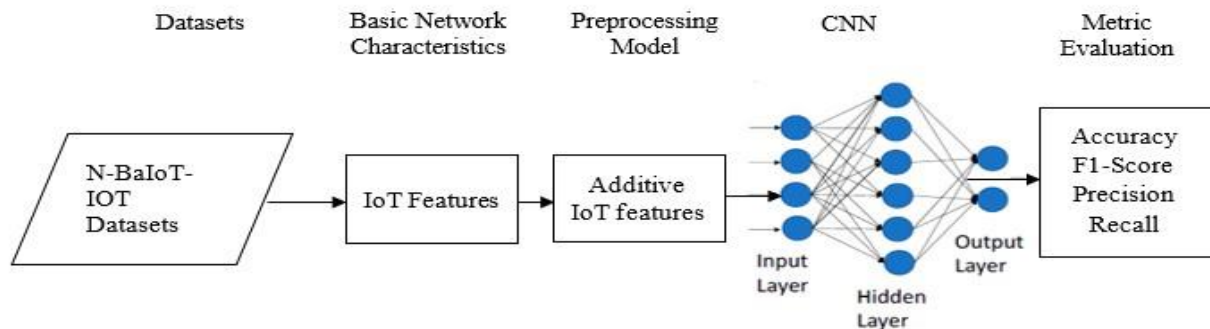


Figure 1: IoT-Based Additive Features for Classification and Source Identification Architecture

In Figure 1, the framework consists of five phases which include: (i) Selecting the suitable dataset (N-Balot-IoT Datasets) for the experiments; (ii) Utilizing the basic network characteristics (IoT Features) for experimentation; (iii) Proposed preprocessing model (Additive IoT Features) from the IoT features; (iv) Adapt the CNN Model for IoT classification and source identification; and (v) Metric Evaluation (Accuracy, F1-Score, Precision and Recall). These phases are carefully followed to implement the proposed AIFID model.

### 3.5 Evaluation Measures

This study leverages the strengths of Accuracy, F1 score, Precision, and Recall metrics to evaluate the effectiveness of the proposed Additive IoT Features for IoT device classification and source identification (AIFID). These metrics are briefly discussed as follows.

#### i. Accuracy metric

Mathematically, accuracy is given as;

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

#### ii. Precision metric

It is mathematically expressed as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

#### iii. The recall metric

It is mathematically expressed as:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

#### iv. F1-Measure metric

It is mathematically expressed as:

$$F1 - Measure = \frac{2 \times Pr \times Rc}{Pr + Rc} \quad (4)$$

Where: TP = True Positive, TN = True Negative, FN = False Negative, FP = False Positive, Pr = Precision, and Rc = Recall.

#### 4. Results/Discussions

This section of the study presents and discusses the experimental outcomes of the proposed IoT device classification and source identification model. The CNN model was trained and tested using the N-Balot-IoT dataset. These results are presented with clear discussion from two perspectives as shown below.

##### 4.1 Performance Results of the CNN Classifier

First, Figure 2 depicts the training loss vs Epochs for the CNN classification and identification of IoT devices. It could be observed that epochs after 200 achieved relatively low loss values. When the loss values become very low, it means our proposed model learned properly.

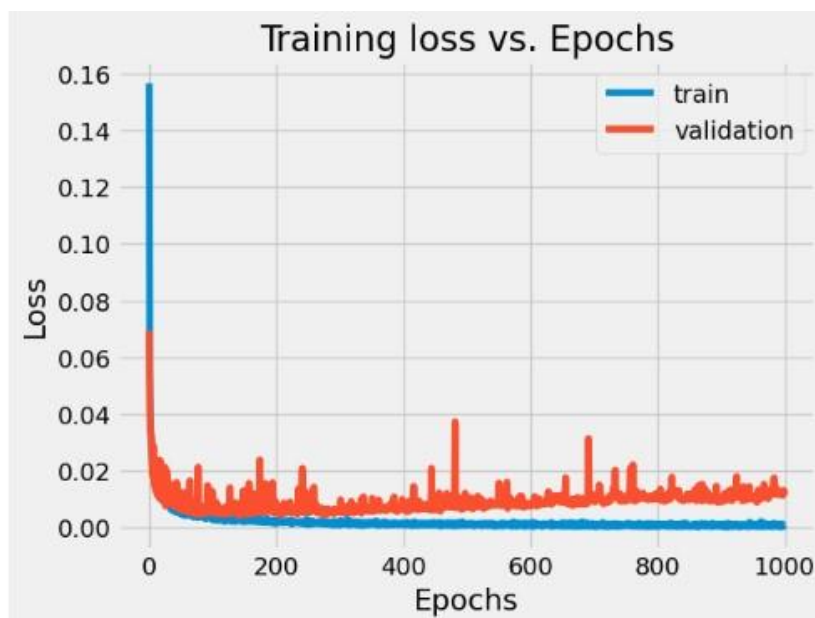


Figure 2: The Training Loss Vs Epochs of the CNN Model on N-Balot-IoT Datasets.

In Figure 3, as the Epochs increased especially after 200, the training and validation accuracy increased closely to 1.00 (100%). An accuracy very close to 100% shows that the proposed model was correctly trained.

The performance of the proposed model in terms of the confusion matrix is depicted in Figure 4. The CNN algorithm was fed with the 115 features of the N-Balot-IoT dataset for experimental purposes. The 9-class confusion matrix comprising 9 IoT devices confirms that the CNN model achieved excellent identification and classification results for the IoT devices with an overall accuracy of **99.67 %**.

From Figure 4, it is clear that devices such as Danmini Doorbell (d1), Ecobee Thermostat (d2), Enio doorbell (d3), Philips baby monitor (d4), Samsung webcam (d7), wht security camera (d8), and wht security camera2 (d9) were all correctly identified and classified at an accuracy of 100 percent. Whereas Pt Security camera1 (d5), and Pt Security camera2 (d6) were all identified and classified at an accuracy of 99.0 percent.

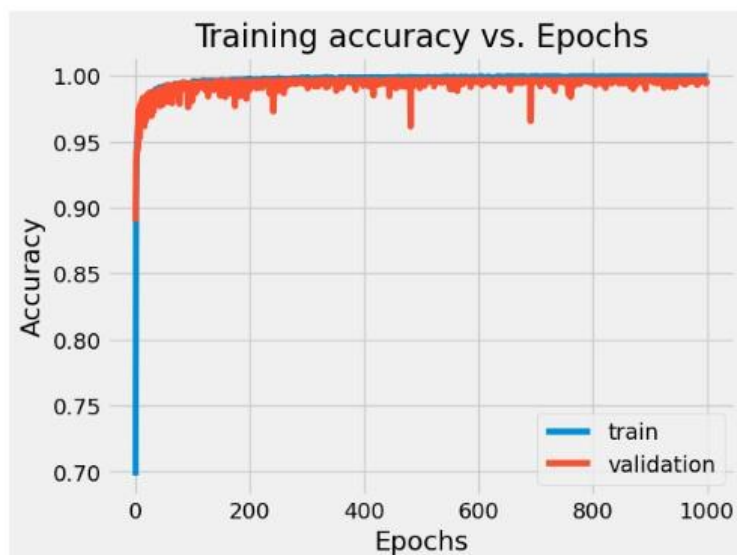


Figure 3: The Training Accuracy Vs Epochs of the CNN Model

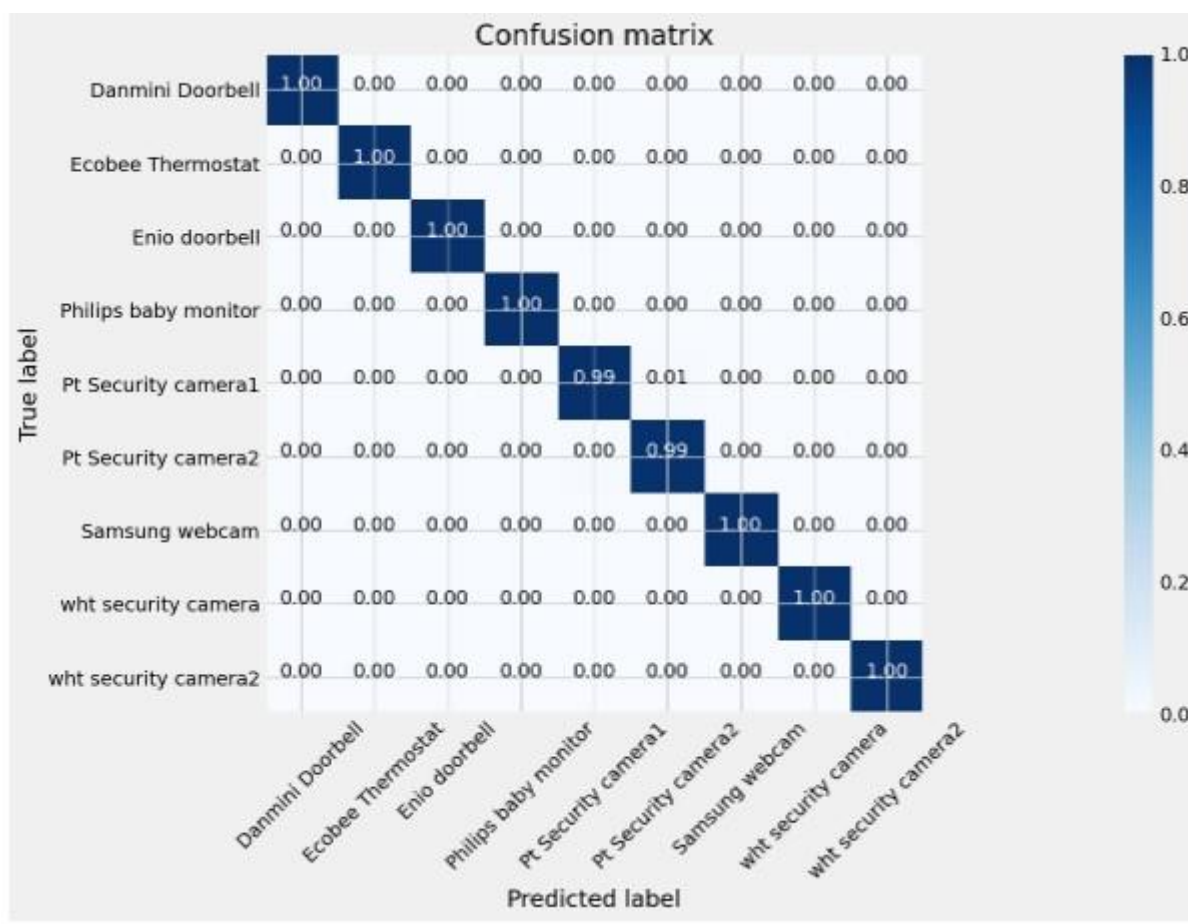


Figure 4: Confusion Matrix for the CNN Model

Furthermore, Table 2 provides a comprehensive summary of Precision, Recall, and F1-score results for the various IoT devices considered in this study. These results clearly demonstrate that the CNN model unambiguously understood the N-Balot-IoT dataset utilized in the study and accurately identified and classified them.

Comparatively, the proposed AIFID with an overall accuracy of **99.67 %** performs at par with existing state-of-the-art models such as Cvitić, Peraković, Periša, and Gupta [6] where they achieved the highest IoT device classification accuracy of 99.79%. This analysis illustrates that the proposed model understands the N-Balot-IoT dataset, and it can effectively and efficiently perform IoT device classification and source identification.

Table 2: Results Summary of the Other Evaluation Metrics

IoT Device ID	IoT Devices	Precision	Recall	F1-Score
d1	Danmini Doorbell	1.00	1.00	1.00
d2	Ecobee Thermostat	1.00	1.00	1.00
d3	Enio doorbell	1.00	1.00	1.00
d4	Philips baby monitor	1.00	1.00	1.00
d5	Pt Security camera1	0.99	0.99	0.99
d6	Pt Security camera2	0.99	0.99	0.99
d7	Samsung webcam	1.00	1.00	1.00
d8	wht security camera	1.00	1.00	1.00
d9	wht security camera2	1.00	1.00	1.00

## 5. Conclusion

In this study, a novel Additive IoT Feature for IoT device classification and source identification (AIFID) is presented. This model leveraged the features of the N-Balot-IoT dataset. The dataset was fed to the CNN learning model. Usually, evaluation metrics are used to assess the effectiveness of a model. Thus, the study employed Accuracy, F1-Measure, and Precision including Recall to measure the efficiency of the proposed CNN model. The performance results of the proposed AIFID were presented, discussed, and compared to the state-of-the-art IoT device classification technique proposed by Cvitić, Peraković, Periša, and Gupta [6]. The experimental performance results of the AIFID model perform favorably well with existing models. This study has shown that the Additive IoT Features for IoT device classification and source identification are very effective. The study addresses the rarity of a model to classify and identify device sources. In the future, the researcher hopes to experiment and get the best features for IoT device classification and improve on the performance accuracies as well.

## References

- [1] A. Iorliam, A.T.S. Ho, A. Waller, and X. Zhao. "Using benford's law divergence and neural networks for classification and source identification of biometric images." In *Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15*, pp. 88105. Springer International Publishing, 2017.
- [2] J. Kotak, and E. Yuval. "IoT device identification using deep learning." *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020) 12*. Springer International Publishing, 2021.
- [3] C. Koball, P.R. Bhaskar, W. Yong, S. Tyler, and F. Connor "IoT Device Identification Using Unsupervised Machine Learning." *Information* 14.6, 2023
- [4] A. Iorliam, A. *Application of power laws to biometrics, forensics, and network traffic analysis*. University of Surrey (United Kingdom), 2016.
- [5] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang. "Automatic device classification from network traffic streams of internet of things." *2018 IEEE 43rd conference on local computer networks (LCN)*. IEEE, 2018.
- [6] I. Cvitić, D. Peraković, M. Periša, and B. Gupta. "Ensemble machine learning approach for classification of IoT devices in smart home." *International Journal of Machine Learning and Cybernetics* 12.11 (2021): 3179-3202.

- [7] H. M. Zahid, Y. Saleem, F. Hayat, F. Z. Khan, R. Alroobaea, F. Almansour, M. Ahmad, and I. Ali. "A framework for identification and classification of iot devices for security analysis in heterogeneous network." *Wireless Communications and Mobile Computing 2022* (2022).
- [8] A. R. Zarzoor, N.A.S. Al-Jamali, and I.R.K. Al-Saedi. "Traffic Classification of IoT Devices by Utilizing Spike Neural Network Learning Approach." *Mathematical Modelling of Engineering Problems* 10.2 (2023).
- [9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici. "N-baiot—networkbased detection of iot botnet attacks using deep autoencoders." *IEEE Pervasive Computing* 17.3 (2018): 12-22.
- [10] A. Iorliam, A., S. Tirunagari, A.T. Ho, S. Li, A. Waller, and N. Poh. "Flow Size Difference" Can Make a Difference: Detecting Malicious TCP Network Flows Based on Benford's Law." *arXiv preprint arXiv:1609.04214* (2016).
- [11] K. Sethi, E. Sai Rupesh, R. Kumar, P. Bera, and Y. Venu Madhav "A context-aware robust intrusion detection system: a reinforcement learning-based approach." *International Journal of Information Security* 19 (2020): 657678.
- [12] S. Albawi, T.A.M. Mohammed, and S. Al-Zawi. "Understanding of a convolutional neural network." *2017 international conference on engineering and technology (ICET)*. IEEE, 2017.

**Conflict of Interest Notice**

The author declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

**Availability of data and material**

Not applicable

**Plagiarism Statement**

This article has been scanned by iThenticate™.



# Prediction of Cardiovascular Disease Based on Voting Ensemble Model and SHAP Analysis

Erkan Akkur<sup>1</sup>

<sup>1</sup> Turkish Medicines and Medical Devices Agency, Çankaya, Ankara, Türkiye



**Corresponding author:**

Erkan Akkur, Turkish Medicines and Medical Devices Agency, Çankaya, Ankara, Türkiye

**E-mail address:**

[eakkur@gmail.com](mailto:eakkur@gmail.com)

**Submitted:** 27 September 2023

**Revision Requested:** 15 October 2023

**Last Revision Received:** 11 November 2023

**Accepted:** 15 November 2023

**Published Online:** 15 November 2023

**Citation:** Akkur E.(2023).

Prediction of Cardiovascular Disease Based on Voting Ensemble Model and SHAP Analysis  
*Sakarya University Journal of Computer and Information Sciences*.6 (3)

<https://doi.org/10.35377/saucis...1367326>

## ABSTRACT

Globally, cardiovascular diseases (CVD) account for a large number of deaths. Early detection plays a critical role in reducing the mortality rate. Early detection can be achieved by utilizing machine learning algorithms on existing data of patients. Ensemble learning methods are one of the techniques applied to improve the classification performance of ML algorithms. This study suggests a prediction model based on voting ensemble learning for the prediction of CVD. The hyperparameters of classification algorithms are optimized by using grid search. The results of each model are validated by using a 10-fold cross-validation schema. The IEEE Data port dataset is used for all experiments Furthermore, the SHAP technique is employed to interpret the proposed prediction model, including the risk factors that play a role in detecting this disease The proposed model for CVD prediction achieved an accuracy of 0.937 and an AUC-ROC score of 0.936. The model presented in this study has a high classification rate compared to previous similar studies.

**Keywords:** Cardiovascular disease, Machine learning, Voting ensemble model, SHAP value

## 1. Introduction

Cardiovascular diseases (CVD) are considered conditions that negatively affect the heart and blood vessel system. Risk factors such as smoking, obesity, diabetes, lack of exercise, and unhealthy diet can cause these diseases. According to global statistics, CVD causes a large number of deaths worldwide. The high mortality rates also emphasize the importance of diagnostic methods [1-2]. In available clinical data, finding the difference between healthy and heart-diseased individuals has been a powerful approach in classification studies. The ability to classify CVD is a critical basis for the diagnosis of patients. In recent years, machine learning (ML) algorithms have played a significant role in research like the detection of CVD. The ML algorithms are one of the approaches applied to predict the status of CVD based on clinical data. These algorithms can make predictions by training with existing data sets. Models built with such methods with high classification rates can be used for diagnosis in new patient records. ML algorithms can help doctors with accurate prognostic predictions based on the patient's clinical data [3-5].

It is possible to increase the performance of ML algorithms by applying different methods. Ensemble learning (EL) techniques are one of the methods used to improve model performance by eliminating the disadvantages of classifiers. These algorithms aim to build models using multiple classifiers instead of a single classifier. When multiple classifiers are applied to train the input data, the actual predictions may outperform the result obtained by a single classifier [6]. The “black box” nature of ML algorithms also poses some challenges regarding the interpretability of the prediction models developed. An interpretable forecasting model is significant from a medical perspective as it enables people to understand the rationale

behind the predictions and decisions made by the model. The SHapley Additive exPlanations (SHAP) method can illustrate the effect of attributes in the dataset on the final prediction. It can also effectively refine and explicate model predictions [7].

This study aims to improve the performance of classifiers for the prediction of CVD using voting techniques. The voting EL prediction model has been built using two different approaches, hard and soft voting. The SHAP analysis method is utilized to interpret the prediction model presented in this study.

## 2. Related Works

ML models have been proven to be effective in predicting CVD in numerous studies. The UCI Heart Disease dataset in the UCI Machine Learning Repository is publicly accessible and is a widely used dataset in this research area [8].

Mohapatra et al [5] applied a stacked EL model on Cleveland Heart Disease for CVD prediction. Ten different classifiers were used as base learners. The classification performance of the suggested EL model was compared with the base learner classifiers. The suggested model in the study achieved an accuracy of 92%. The study indicated that ensemble learning algorithms improve classification performance. Sangya et al [9] presented a comparative analysis of different ML algorithms including Logistic Regression (LR), K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF) algorithms for predicting CVD on the Cleveland dataset. Data preprocessing steps were applied to fill in missing data and remove noise from the dataset. As a result of the experiments, the SVM algorithm achieved the best result with an accuracy of 89.34%. Shah et al. [10] compared different ML algorithms to predict CVD. NB, DT, K-NN, and RF algorithms were used classification process. The experiments were carried out on the Cleveland dataset. Data preprocessing stages were performed on the dataset before the classification process. As a result of the comparisons, the K-NN algorithm achieved the best classification rate with 90.78 % accuracy. Rajdhan et al. [11] employed various classification algorithms containing NB, RF, LR, and RF. The experiments were carried out on the Cleveland dataset. The dataset was divided into 80% training data and the rest test data. The RF algorithm outperformed with an accuracy of 90.16%. Poorani and Hemalatha [12] carried out the comparison of SVM, DT, MLP, RF, and J48 algorithms to predict CVD. The NB algorithm outperformed with 90.33% accuracy. Ozhan and Kucukakcali [13] utilized the XGBoost (XGB) model to estimate the risk prediction of CVD. A 10-fold CV was applied to measure the performance of the classifier. The suggested model was achieved with an accuracy of 89.4%. Das and Sinha [14] suggested a voting-based EL model to predict CVD. The experiments were implemented on the Statlog Heart Disease dataset. The proposed model yielded 90.74% accuracy comparing K-NN, SVM, NB, DT, LR, and ANN algorithms. The study showed that EL models provide higher success rates than classical classifiers.

Akyol and Atilla [15] conducted a study comparing Gradient Boosting Machines, RF and NB algorithms for CVD detection. Recursive Feature Elimination with a cross-validation (CV) technique was applied to select the most discriminative features. The experiments were performed on the Statlog Heart Disease dataset and the SPECT dataset. In both datasets, the NB algorithm achieved the highest classification rate with accuracy of 86.42% and 77.78%. Jan et al. [16] proposed a voting EL model for CVD prediction using SVM, ANN, NB, RF, and Regression Analysis algorithms. The proposed model is tested on the dataset created by combining the Cleveland and Hungary datasets. The Weka Data Mining Tool is used for the analysis. The proposed method achieved an accuracy of 93%. Tiwari et al. [17] suggested an ensemble approach containing the stacked model for the prediction process. EXC, SVM, RF, and XGB algorithms were utilized as a base classifier. The results obtained by the suggested model compared with basic classifiers. The suggested model showed 92.34% accuracy on the Heart Disease Dataset (IEEE DataPort).

Yilmaz and Yagin [18] suggested a predictive model containing SVM, LR, and RF algorithms for CVD prediction. The performance of models was evaluated on the Heart Disease Dataset (IEEE DataPort). The hyperparameters of the ML algorithms were tuned using a 10-fold repeated CV. RF algorithm yielded an accuracy rate of 92.9%. Doppala et al. [19] utilized the EL approach for the prediction of CVD. NB, RF, SVM, and XGB algorithms were adopted as base classifier algorithms. The Majority Voting technique was utilized as an EL approach using the Cleveland, IEEE Dataport, and Mendeley Data Center datasets, respectively. The presented EL method demonstrated accuracy rates of 88.24%, 93.39%, and 96.75%. The suggested model achieved higher classification rates than classical classifiers. García-Ordás et al. [20] utilized a Conventional Neural Network (CNN) algorithm for CVD prediction. A 10-fold CV approach was utilized to avoid randomness. The proposed model achieved a better result with an accuracy rate of 90.09% compared to ML algorithms.

Table 1 summarizes some recent studies on CVD prediction in the literature. Although machine learning is used for CVD prediction, the majority of studies have been conducted with individual classifiers. However, the number of studies using EL approaches is quite small. When determining the hyperparameters of classifiers, hyperparameter optimization is not commonly utilized. To address these limitations in the literature, a CVD prediction model based on EL is proposed in this study. The Grid Search technique is used for hyperparameter tuning. The SHAP analysis is also used, which allows machine learning algorithms to make interpretations on models by removing black box features.



Table 1: Some recent studies on CVD prediction in the literature

Author	Dataset	Techniques used	Claimed outcome	Accuracy	Limitations/Gaps
Sangya et al. [9]	Cleveland	LR, SVM, K-NN, RF, NB, DT	SVM	89.34 %	Small dataset is used. Hyperparameter tuning is not used.
Shah et al. [10]	Cleveland	NB, DT, K-NN, RF	K-NN	90.78 %	Small dataset is used. Hyperparameter tuning is not used.
Rajdhan et al. [11]	Cleveland	DT, RF, LR, NB	RF	90.16 %	Small dataset is used. Data pre-processing is not specified.
Poorani and Hemalatha [12]	Cleveland	DT, RF, NB, MLP, SVM	NB	90.33 %	Small dataset is used. Data pre-processing is not specified. Hyperparameter tuning is not used.
Ozhan and Kucukakcali [13]	Cleveland	-	XGB	89.4 %	Small dataset is used. Data pre-processing is not specified. A single ML algorithm is used.
Das and Sinha [14]	Statlog	K-NN, SVM, NB, DT, LR, ANN, Voting EL	Voting EL	90.74 %	Small dataset is used. Hyperparameter tuning is not utilized.
Akyol and Atilla [15]	Statlog	GBM, NB, RF	NB	86.42 %	Small dataset is used. Hyperparameter tuning is not utilized.
Jan et al. [16]	Cleveland+ Hungarian	SVM, ANN, NB, RF	Voting EL	93.00 %	Hyperparameter tuning is not utilized.
Tiwari et al. [17]	IEEE Dataport	EXC, SVM, RF, XGB Stacked EL	Stacked EL	92.34 %	Hyperparameter tuning is not utilized.
Yilmaz and Yagin [18]	IEEE Dataport	SVM, LR, and RF	RF	92.9 %	Data pre-processing is not specified.
Doppala et al. [19]	Cleveland	NB, RF, SVM, XGB	Majority Voting	88.24 %	Hyperparameter tuning is not utilized.
	IEEE Dataport			93.39 %	

## 2. Material and Method

### 2.1. Dataset Description

The IEEE Data Port is used to obtain the CVD dataset. The dataset was built by merging five previously individually available CVD datasets identified as Hungarian, Cleveland, Long Beach VA, Switzerland & Statlog. The dataset contains 1190 instances of 12 features, including 11 attribute values and one target variable [21]. The target variable of the dataset is composed of 561 samples without CVD (0) and 629 samples with CVD (1). Table 2 shows the attributes and their descriptions in the dataset.

### 2.2 Data Preprocessing

Data preprocessing is one of the essential steps before building predictive models. This process includes the steps that ensure that datasets are suitable for prediction models. Well-processed and organized data can significantly determine the effectiveness of the models designed [22]. There is no missing data in the data set. Data visualization provides an easier understanding of datasets. Figure 1 and Figure 2 indicate a visual representation of the categorical and numeric variables in the dataset.

Table 2: The attributes of the CVD dataset

No	Attributes	Description	Unit	Types of Attributes
1	age	in years	28-77	numerical
2	sex	gender (0=female, 1=male)	0-1	categorical
3	chest pain type	typical angina (1), atypical angina (2), non-anginal pain (3), asymptomatic pain (4)	1-4	categorical
4	resting bp s	resting blood pressure in mmHg	0-200	numerical
5	cholesterol	serum cholesterol in mg/dl	0-603	numerical
6	fasting blood sugar	(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)	0-1	categorical
7	resting ECG	normal (0), ST-T wave abnormality (1), LV Hypertrophy (2)	0-2	categorical
8	max. heart rate	max. heart rate achieved	60-202	numerical
9	exercise-induced angina	yes (1), no (0)	0-1	categorical
10	oldpeak	ST depression	-2.6-6.2	numerical
11	ST slope	the slope of the peak exercise ST segment upsloping (1), flat (2), downsloping (3)	1-3	categorical
12	target	heart disease (1), no heart disease (0)	0-1	categorical

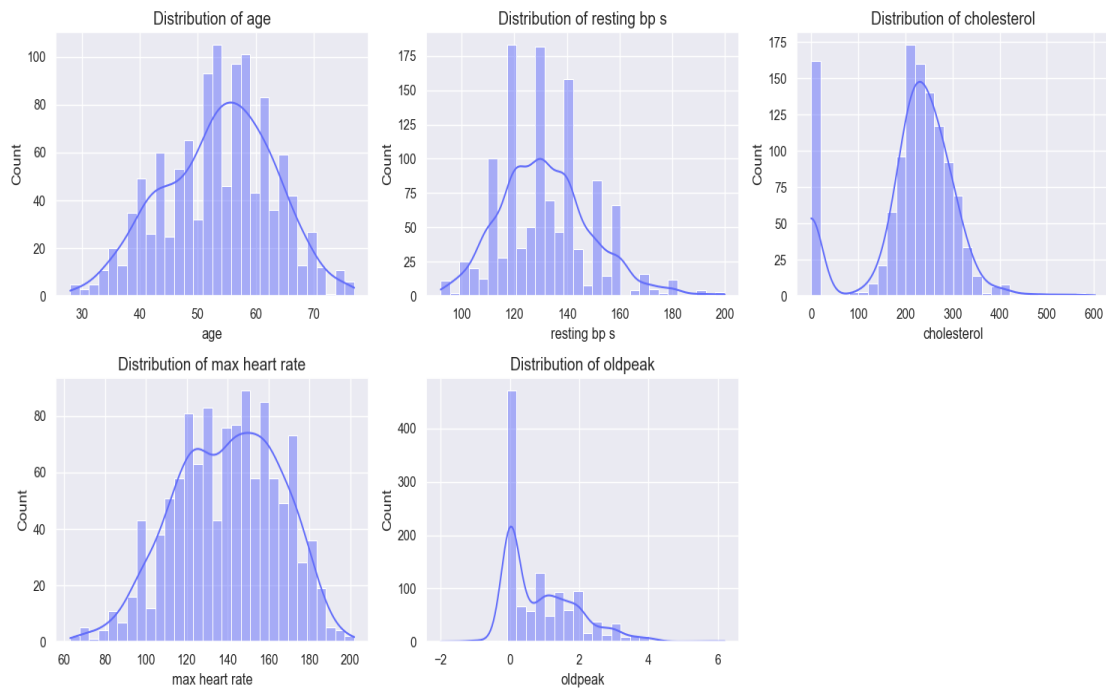


Figure 1: Numerical features

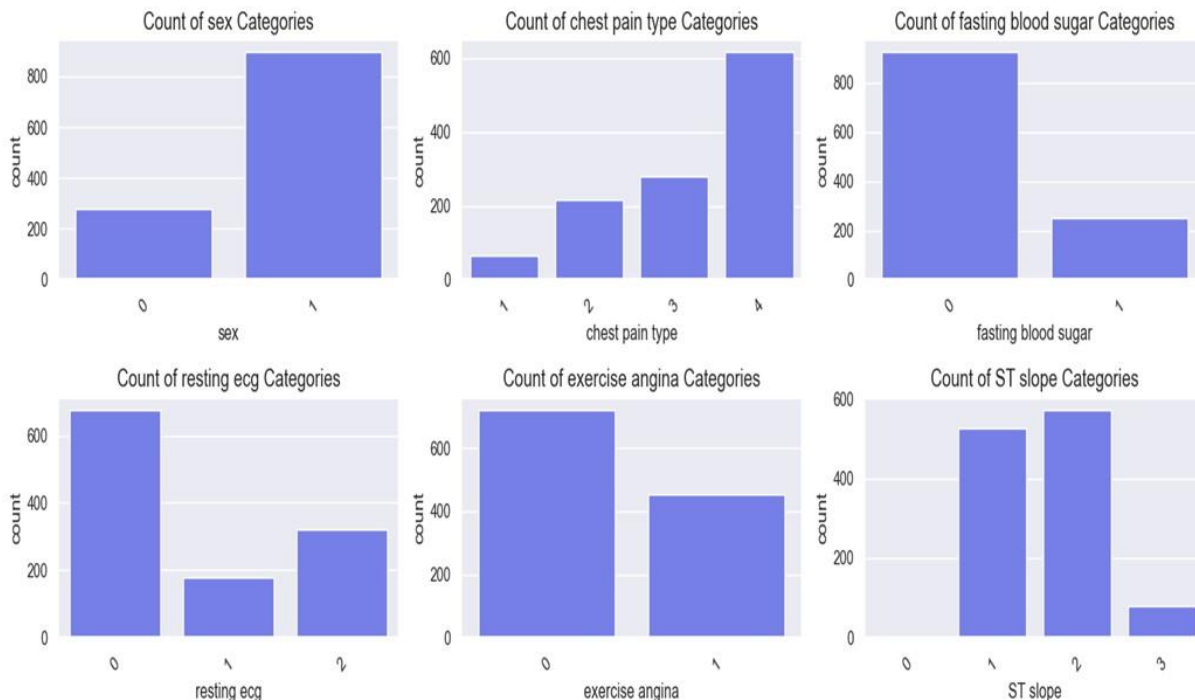


Figure 2: Categorical features

When the figures are analyzed, the following conclusions can be drawn about the dataset.

- There are five numerical and six categorical variables.
- Numerical variables in the dataset have a normal distribution.
- People with heart disease in the dataset are most frequently male.
- The most common type of chest pain is asymptomatic pain.
- Fasting blood levels are mostly below 120 mg/dl.
- ECG values are in the normal range
- Most people do not have angina.
- Most people have a flat ST slope.

In the next stage, a data transformation process is applied. The aim of this is to preserve the quality of the data and improve the data structure. The attributes are rescaled utilizing the Min-max technique between [0-1]. This technique leverages Equation 1 for rescaling. In equation 1,  $x_{scaled}$  depicts the rescaled value,  $x$  denotes the attribute value, and  $x_{max}$  and  $x_{min}$  depict the maximum and minimum attribute values [23].

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

The correlation heatmap graphically expresses the strength of relationships between numerical variables in data sets. It assists in analyzing which variables are correlated and the power of that relationship [24]. Figure 3 illustrates the correlation heatmap. Figure 4 shows the correlations observed between the target variable and the independent variables in the dataset. Accordingly, ST slope, exercise angina, chest pain type, and gender variables have a positive correlation. However, max. heart rate and cholesterol are negatively correlated with the target variable.

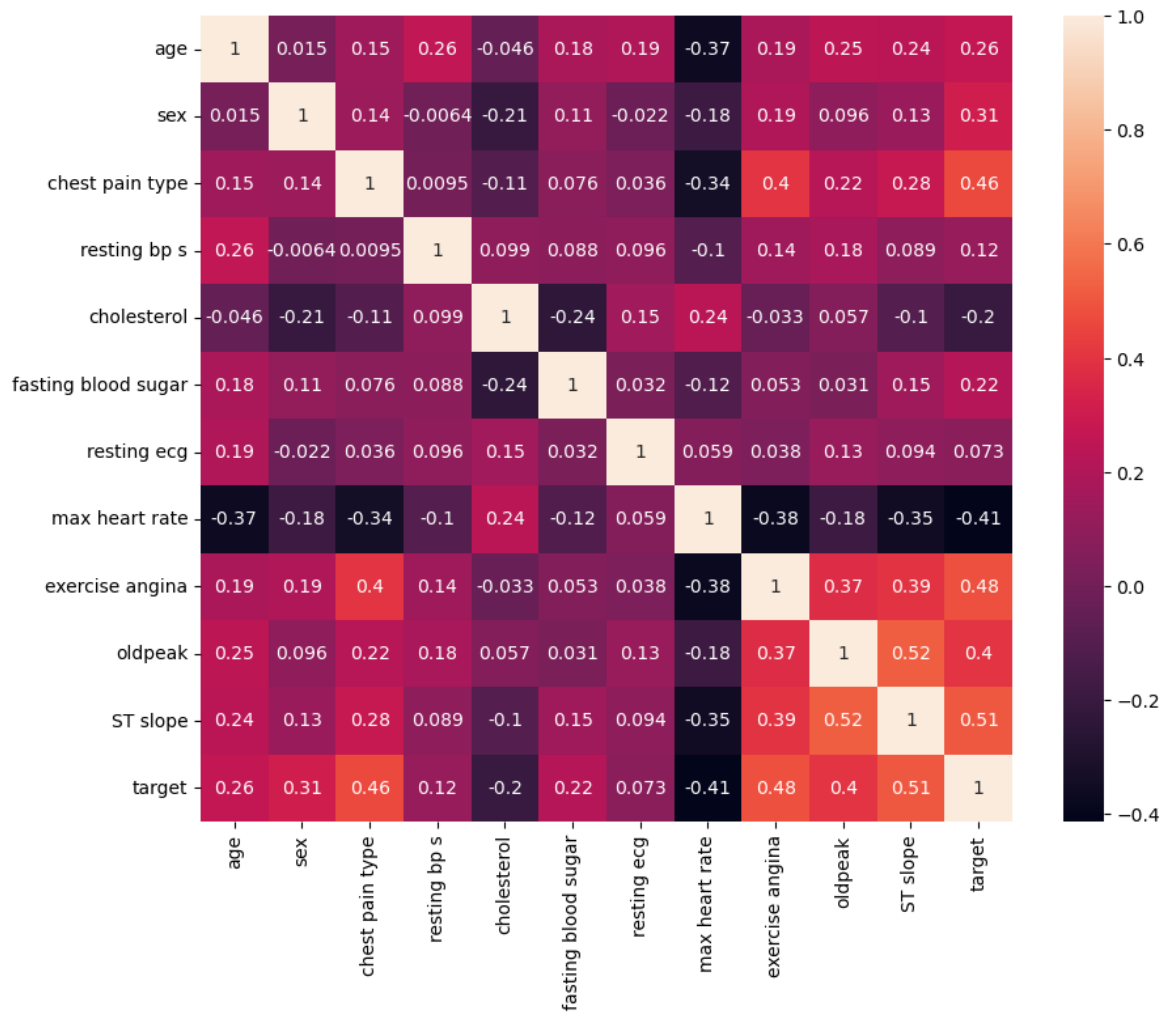


Figure 3: The correlation heatmap of all attributes in the dataset.

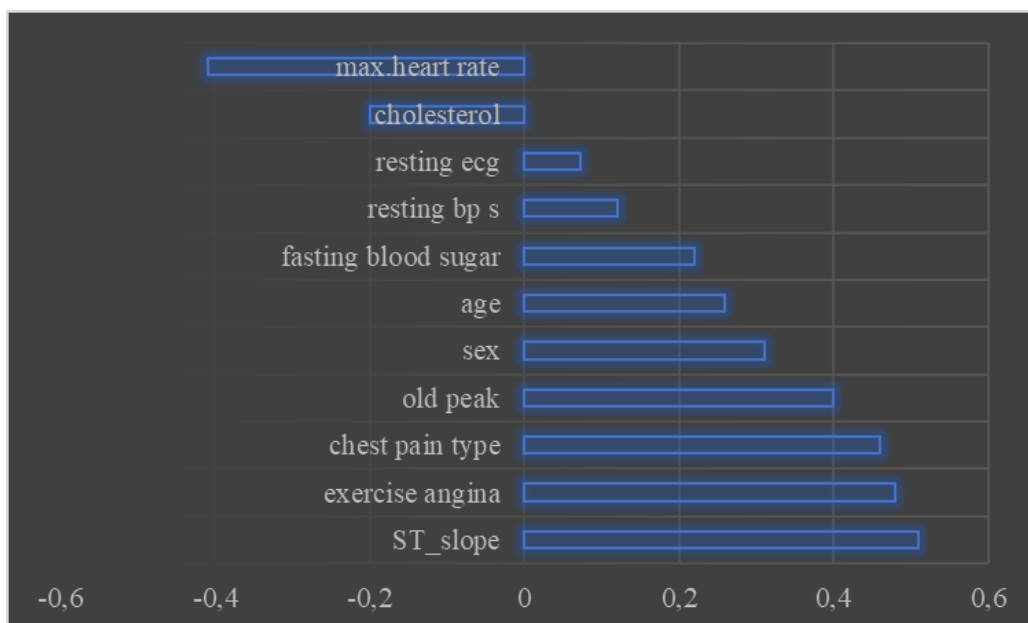


Figure 4: Correlation with target attribute

### 2.3 Ensemble Learning Approach

An EL technique usually acquires training data and trains models. After the training process, this technique provides testing data to the models and then each model predicts a class label for each sample in the test data. Then, each sample is put through a voting process for prediction. The voting ensemble model aims to predict an outcome based on the maximum probability of the output label that is trained on an ensemble of multiple classifiers and selected as output. The basic approach in this technique is to combine the predictions of each classifier passed to the Voting Classifier and predict the resulting label depending on the highest voting majority. Instead of building separate custom models and trying to find the accuracy for each of them, this approach is to build a single model that is trained with these models and estimates the output based on the combined majority vote for each output label. This approach provides flexibility in the combination strategy and helps to achieve the maximum possible classification accuracy. In general, two forms of voting techniques exist: Hard Voting (HV) and Soft Voting (SV). In the HV technique, each model is voted for each test case and the one that receives more than half of the votes is the final output prediction. It can be concluded that the ensemble approach does not provide a stable prediction for this problem if none of the predictions gets more than half of the votes. Instead of building discrete models and calculating accuracy for each one, a single model is built that trains on a group of multiple models and predicts the output for each output label based on the total majority of votes. In the SV technique, each classifier assigns a probability value that a given data point falls into a particular class. The predictions are then weighted by the importance of the classifier and summed. The target with the highest weighted probability sums the label wins the vote [25, 26]. In this study, soft and hard voting ensemble models are implemented on a set of algorithms including Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (K-NN), Support Vector Machines (SVM), AdaBoost (ADAB) and XGBoost (XGB) for CVD prediction. Figure 5 demonstrates the concept of a Voting Classifier, one of the EL methods that unifies multiple ML algorithms.

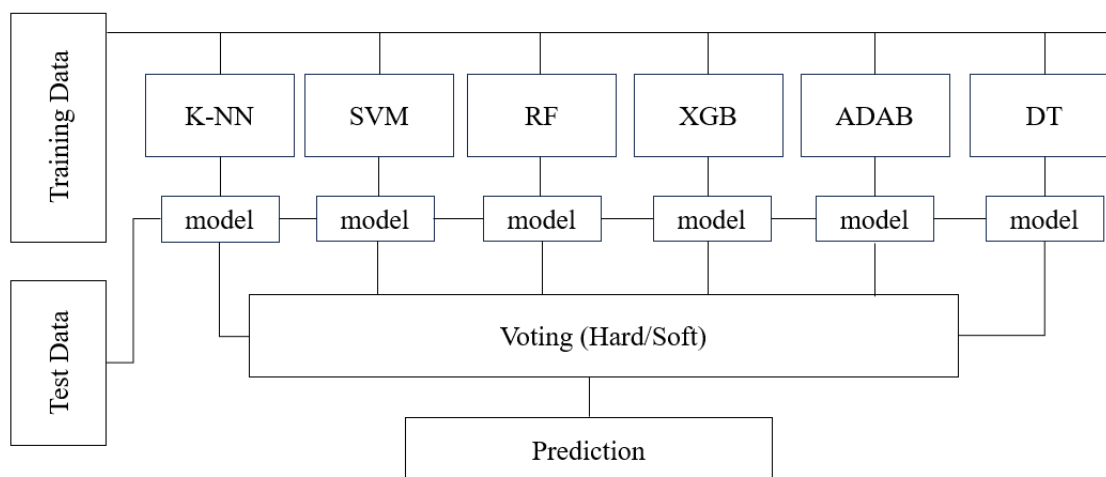


Figure 5: The workflow of the Voting Classifier

### 2.4 Data partition

The input dataset is partitioned into training sets and test sets utilizing a ten-fold cross-validation (CV) technique. The CV technique minimizes the chances of over-fitting and under-fitting models. The ten-fold CV technique randomly divides the dataset into ten equal subsets. Nine subsets are used for the training set and the rest for the test set. This process is iterated until each subset represents the test set. The final accuracy of each ML algorithm is decided by the mean of the accuracies derived through iterations in the model [27].

### 2.5 The hyperparameters of ML algorithms

The hyperparameters are parameters for ML algorithms whose values are set before training the model. Hyperparameter tuning denotes the tuning of the parameters of the model, which is a long process. This process improves the performance of ML algorithms. In this study, the Grid Search (GS) technique was used for hyperparameter tuning of the classifiers. It subdivides the space of hyperparameters into a separate grid. The performance of the model is then appraised for each combination of hyperparameters through k-fold CV. A 10-fold CV technique is used for the evaluation process. The grid point that maximizes the mean value in CV is the optimal combination of values for the hyperparameters [28]. Table 2 summarizes the hyperparameters of each ML algorithm.

Table 2: Hyperparameter tuning summary

Algorithm	Hyperparameters	Search Range	The best hyperparameter
RF	N_estimators	[10-500]	200
	Min_samples_split	[2-10]	3
	Max_depth	[2-10]	2
	Min_samples_leaf	[1-10]	2
XGB	N_estimators	[10-300]	150
	Min_samples_split	[2-10]	3
	Max_depth	[2-10]	2
	Min_samples_leaf	[1-10]	2
ADAB	N_estimators	[10-200]	[100]
	Learning_rate	[0.001-0.5]	[0.1]
K-NN	N_neighbors	[1-31]	5
DT	Max_feature	[auto, sqrt, log2]	log2
	Ccp_alpha	[0.001- 0.1]	0.01
	Max_depth	[2-10]	2
	Criterion	entropy, Gini	Gini
SVM	C	[1-1000]	100
	Gamma	[0.001-1]	0.1
	Kernel	[rbf, linear]	rbf

## 2.6 Model interpretation and feature importance

It is a challenging issue to comment on these models as ML algorithms are often considered a black box. SHapley Additive Explanations (SHAP) takes these algorithms out of the black box and allows comments to be made on the model. With this technique, it is possible to interpret attribute importance scores obtained after a training process and make interpretable predictions for a test instance. It provides an annotated representation of the feature value of each variable that is influential in determining the output of a ML model. It also offers the possibility of determining whether the contribution of each input characteristic is positive or negative. In this study, the Tree SHAP algorithm was used to calculate the SHAP values. For a model with prediction function  $f(x)$  and  $m$  attributes, SHAP values are obtained by Equation 2.

$$\phi_i = \sum_{p \subseteq N \setminus \{i\}} \frac{|p|! (m - |p| - 1)!}{m!} [f_x(p \cup \{i\}) - f_x(p)] \quad (2)$$

The formula expressed in Equation 1 is the sum of all possible subsets ( $p$ ) of all attribute values except the  $i_{th}$  attribute value.  $|p|!$  is the number of permutations of attributes that precede the  $i_{th}$  attribute value.  $(m-|p|-1)!$  is the number of permutations of attributes that follow the  $i_{th}$  attribute value. The difference operation in the equation expresses the marginal contribution of adding the  $i_{th}$  attribute value to  $p$  [29].

## 3. Results and Discussion

The results and analysis of the suggested framework for the prediction of CVD have been presented in this section. The suggested framework is performed on a 64-bit machine with an 8th Generation Intel i7 CPU (16 GB DDR3 - 1 TB Hard drive 256 GB SSD). The experiments have been performed utilizing Jupyter Notebook 3.8.16 in Python containing packages such as Numpy, Pandas, Matplotlib, Seaborn, and Scikit-Learn. The Pandas 1.5.3 package was used for reading and analyzing the dataset, and The Numpy 1.23.5 package was chosen to realize mathematical functions with multidimensional arrays and matrices. The Matplotlib 3.6.3 package was used to gain insight into how attributes are distributed in the dataset. The Seaborn 0.12.2 package was used to provide appealing and instructive data visualization graphics, especially the distribution and heatmap functions. The Scikit-Learn 1.2.2 package was utilized for splitting the dataset, model selection, and calculating statistical measures to evaluate the performance of the models. A confusion matrix is employed to measure the performance of the classifiers created within the scope of the study. Using this matrix, the performance metrics such as accuracy, precision,

sensitivity, and F1-score can be calculated. The necessary mathematical expressions for these metrics are presented in Equations 3-6.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3}$$

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

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

$$\text{F1 - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

Table 3 and Figure 6 summarize the results of the performance comparison of ML algorithms concerning the four performance metrics. When the classification rates of the individual classifiers are compared, the XGB and RF algorithms achieved higher classification rates with accuracies of 0.9185 and 0.9084. To improve classification performance, soft and hard voting ensemble learning models are generated by combining individual classifiers. The hard voting ensemble classifier (HVE) and soft voting ensemble classifier (SVE) outperform the baseline classifiers with accuracy of 0.9278 and 0.937, precision of 0.9387 and 0.9459, recall of 0.9253 and 0.9355 and F1-score of 0.9319 and 0.9407. Analyzing the performance of the two proposed voting ensemble models, it is seen that SVE performed better with an accuracy of 0.9370.

Table 3: Comparison of suggested voting-based models with the baseline classifier

Classifiers	Accuracy	Precision	Recall	F1-Score
K-NN	0.8431	0.8553	0.8472	0.8513
SVM	0.8580	0.8696	0.8628	0.8662
ADB	0.8706	0.8808	0.8752	0.8780
DT	0.8815	0.8919	0.8849	0.8884
RF	0.9084	0.9173	0.9101	0.9137
XGB	0.9185	0.9269	0.9196	0.9232
<b>HVE</b>	<b>0.9278</b>	<b>0.9387</b>	<b>0.9253</b>	<b>0.9319</b>
<b>SVE</b>	<b>0.9370</b>	<b>0.9459</b>	<b>0.9355</b>	<b>0.9407</b>

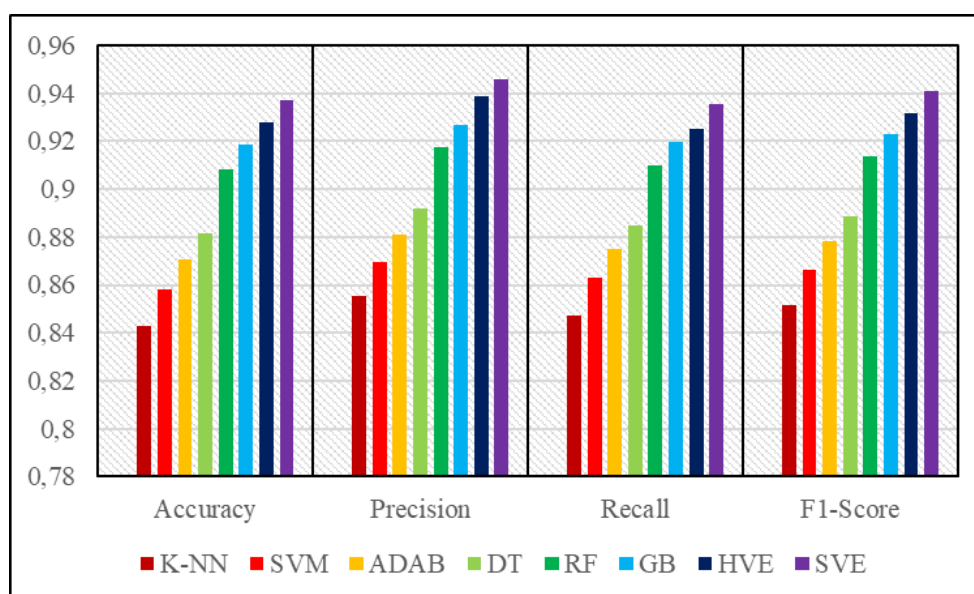


Figure 6: The comparison of classification rates in terms of performance metrics

The confusion matrix summarizes the prediction results of the proposed SVE and HVE models and is given in Figure 7 and Figure 8 respectively. Besides these metrics, the AUC-ROC curve can also be leveraged to evaluate the performance of classifiers. Figure 9 presents the AUC-ROC scores of the suggested voting ensemble techniques comparatively. Comparing the results of voting EL techniques, SVE has a higher performance rate than HVE with an AUC value of 0.936.

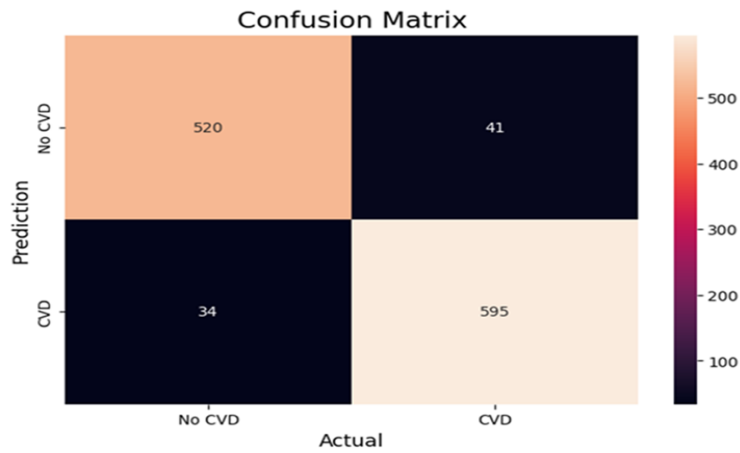


Figure 7: The confusion matrix of the suggested soft-voting ensemble model

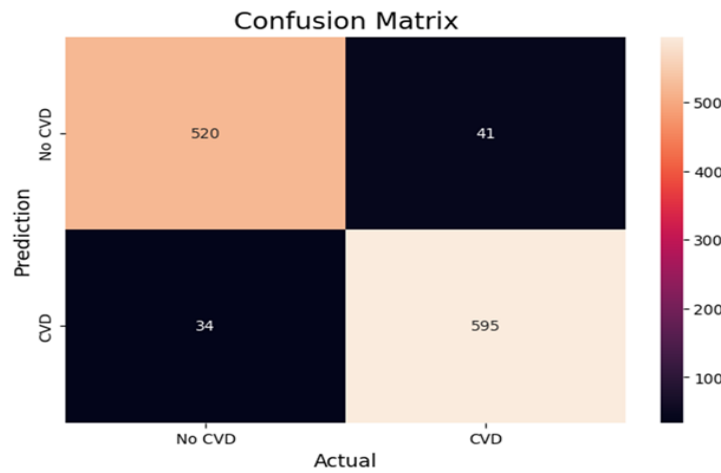


Figure 8: The confusion matrix of the suggested hard-voting ensemble model

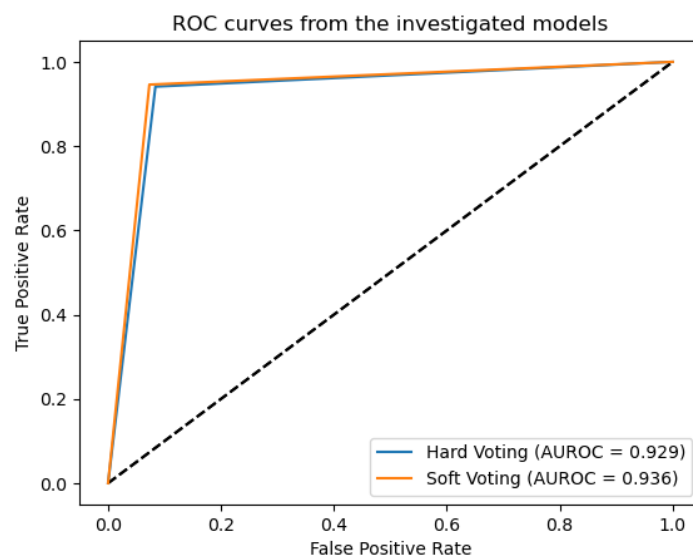


Figure 9: AUC-ROC curve for suggested voting ensemble models



With the help of Shap analysis, it is possible to learn the importance of attributes in CVD prediction and the contribution of each attribute to the accuracy of the model. Since the Tree SHAP algorithm is used, SHAP values are calculated for the XGB algorithm, which achieves the highest accuracy among tree-based algorithms. XGB is an innovative algorithm based on a decision tree and uses the method of gradient boosting in its computations. Figure 10 shows the variables evaluated by their mean absolute SHAP values. Figure 11 shows the variables in order of importance. SHAP values that hurt the predictions have a negative sign, whilst those that have a positive impact have a positive sign. The ST slope is the most significant risk factor for this disease when both graphs are examined. This means that the presence of the ST slope variable in a patient increases the patient's risk of suffering from heart disease. Moreover, the attributes with the least contribution to the CVD prediction model are "resting ecg" and "resting bp s".

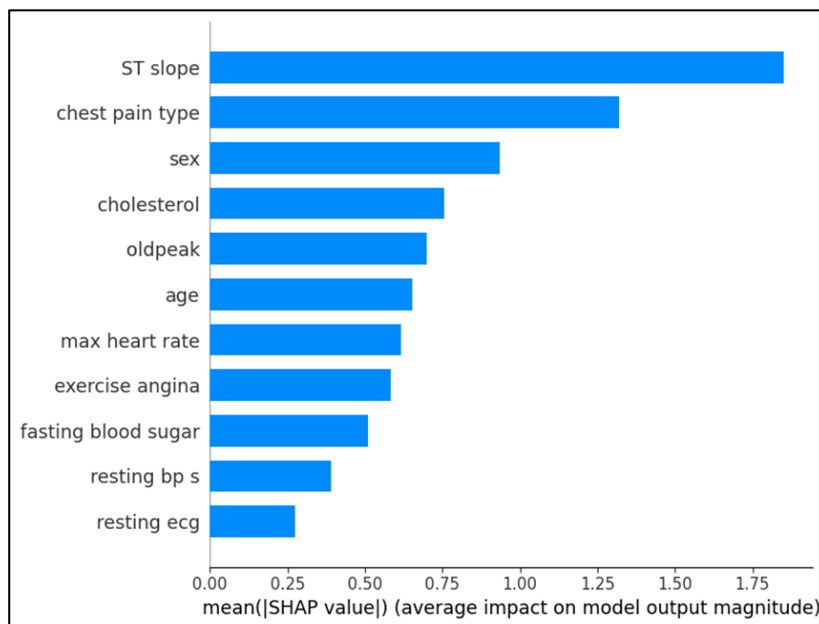


Figure 10: The attribute importance ranking according to the mean |SHAP| value



Figure 11: The attribute importance with the stability and interpretation

Table 4 provides a comparative analysis with previous similar studies for CVD prediction. As a result of the comparison with similar studies, it can be said that the SVE has a good classification rate.

Table 4: Comparison of the suggested model with some existing studies using the CVD dataset

References	Year	Dataset	Methods	Accuracy	Precision	Recall	F1-Score
Shah et al. [10]	2020	Cleveland	K-NN	0.9078	-	-	-
Rajdhan et al. [11]	2020	Cleveland	RF	0.9016	0.937	0.882	
Poorani and Hemalatha [12]	2021	Cleveland	NB	0.9033	-	-	-
Ozhan and Kucukakcah [13]	2022	Cleveland	XGB	0.894	-	0.894	0.884
Das and Sinha [14]	2023	Statlog	Voting EL	0.9074	-	-	0.9230
Akyol and Atilla [15]	2019	Statlog	NB	0.8642	-	0.7188	-
Tiwari et al. [17]	2022	IEEE Data port	Stacked EL	0.9234	0.92	0.9349	0.9274
Yilmaz & Yagin [18]	2022	IEEE Data port	RF	0.929	-	0.928	0.928
Doppala et al. [19]	2022	Cleveland	Majority Voting	0.8824	0.85	0.90	0.88
		IEEE Data port		0.9339	0.99	0.88	0.90
<b>Our suggested model</b>		IEEE Data port	<b>Soft Voting</b>	<b>0.9370</b>	<b>0.9459</b>	<b>0.9355</b>	<b>0.9407</b>

#### 4. Conclusion

In conclusion, this study presents a prediction model based on the voting ensemble technique for the detection of cardiovascular disease. Two different models, hard and soft voting, have been developed as voting ensemble models. Six different classifiers are selected as baseline classifiers. In the proposed method, GSCV is utilized to obtain the best hyperparameters of the classifiers. Moreover, the presented EL approaches have better classification rates when compared to the baseline classifiers. Among the voting techniques, the proposed SVE model achieved the highest classification rate with 0.9370 in accuracy, 0.9459 in precision, 0.9355 in sensitivity, 0.9407 in F1-score, and 0.936 AUC-ROC values. In addition, the SHAP technique is used to extract the black box structure of classifiers and to investigate the effects of variables in the dataset on the model. According to the results of the analysis, the ST slope variable is found to be the most important risk factor for this disease. Although voting EL is not the optimal solution for all problems, it provides a higher classification rate than individual classifiers. In the future, we intend to evaluate and test the proposed model on different datasets. Machine learning algorithms face major challenges owing to the limited amount of data. If hospitals and other data-generating organizations collaborate to obtain a larger amount of high-quality medical data, more study and research can be done.

#### References

- [1] F. Coronado, S. C. Melvin, R. A. Bell and G. Zhao, "Global Responses to Prevent, Manage, and Control Cardiovascular Diseases." *Prev Chronic Dis*, 2022, 8:19:E84.
- [2] R. Hajar, "Risk Factors for Coronary Artery Disease: Historical Perspectives." *Heart Views*, 2017; 18(3), 109-114.
- [3] J. Azmi, M. Arif, M.T. Nafis, M. A. Alam, S. Tanweer, G. Wang, "A systematic review on machine learning approaches for cardiovascular disease prediction using medical big data." *Medical Engineering & Physics*, 2022, 105, 103825.
- [4] K. P. Kresoja, M. Unterhuber, R. Wachter, H. Thiele, P. Lurz, "A cardiologist's guide to machine learning in cardiovascular disease prognosis prediction." *Basic research in cardiology*, 2023, 118(1), 10.
- [5] S. Mohapatra, S. Maneesha, S. Mohanty, P. K. Patra, S.K. Bhoi, K. S. Sahoo and A.H. Gandomi. "A stacking classifiers model for detecting heart irregularities and predicting cardiovascular disease." *Healthcare Analytics*, 2023, 3, 100133.
- [6] I.D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects." *IEEE Access*, 2022, 10, 99129-99149
- [7] K. Wang et al. "Interpretable prediction of 3-year all-cause mortality in patients with heart failure caused by coronary

- heart disease based on machine learning and SHAP.” *Computers in Biology and Medicine*, 2021, 137, 104813.
- [8] M. Ahsan and Z. Siddique, “Machine learning-based heart disease diagnosis: A systematic literature review.”, *Artificial Intelligence in Medicine*, 2022, 128, 102289.
- [9] Sangya W., Shanu KR, C. Bharat., Heart Attack Prediction by using Machine Learning Techniques. *International Journal of Recent Technology and Engineering* 2020;8(5):1577–80.
- [10] D. Shah, S. Patel, S.K. Bharti. “Heart disease prediction using machine learning techniques.” *SN COMPUT. SCI.* 2020, 1:345.
- [11] Rajdhan A, Agarwal A, Sai M, Ravi D, Ghuli P. Heart disease prediction using machine learning. *International Journal of Research and Technology* 2020;9(04): 659–62.
- [12] Poorani S, Hemalatha D. Machine Learning Techniques for Heart Disease Prediction. *Journal of Cardiovascular Disease Research* 2021;12(1):93–6.
- [13] O. Ozhan and Z. Kuçukakcali, “Estimation of risk factors related to heart attack with XGBoost that machine learning model.” *Middle Black Sea Journal of Health Science*, 2022, 8(4), 582-591.
- [14] T. Das and B. B. Sinha, "A comprehensive study on machine learning methods for predicting heart disease: a comparative analysis," *8th International Conference on Computing in Engineering and Technology (ICCET 2023)*, Hybrid Conference, Patna, India, 2023, pp. 205-210.
- [15] K. Akyol and U. Atilla, “A study on performance improvement of heart disease prediction by attribute selection methods.”, *Academic Platform Journal of Engineering and Science*, 2019; 7-2, 174-179.
- [16] M. Jan, AA Awan, MS Khalid & Salman Nisar, Ensemble approach for developing a smart heart disease prediction system using classification algorithms, *Research Reports in Clinical Cardiology*, 2018; 9: 33-45.
- [17] A. Tiwari, A. Chugh, A. Sharma, “Ensemble framework for cardiovascular disease prediction.” *Computers in Biology and Medicine*, 2022, 146, 105624.
- [18] R. Yilmaz and F.H. Yagin, “Early detection of coronary heart disease based on machine learning methods.” *Medical Records*, 2022, 4(1), 1-6.
- [19] BP. Doppala, D. Bhattacharyya D, M. Janarthanan, N. Baik, “A reliable machine intelligence model for accurate identification of cardiovascular diseases using ensemble techniques.” *J Healthc Eng.* 2022, 8:2022:2585235
- [20] MT. García-Ordás, M. Bayón-Gutiérrez, C. Benavides et al. “Heart disease risk prediction using deep learning techniques with feature augmentation.”, *Multimed Tools Appl* 2023, 82, 31759–31773.
- [21] M. Siddhartha, November 5, 2020, "Heart Disease Dataset (Comprehensive)", IEEE Dataport, doi: <https://dx.doi.org/10.21227/dz4t-cm36>. (Accessed -10.09.2023).
- [22] S. Garcia, S. Ramírez-Gallego, J. Luengo, J.M. Benítez & F. Herrera, “Big data preprocessing: methods and prospects.”, *Big Data Analytics*, 2016, 1(1), 1-22.
- [23] SGK Patro and KK Sahu, Normalization: A preprocessing stage. *arXiv preprint arXiv: 2015, 1503.06462*.
- [24] BC. Haarman, RF. Riemersma-Van der Lek, WA Nolen, R. Mendes, HA. Drexhage, H. Burger. “Feature-expression heat maps--a new visual method to explore complex associations between two variable sets.” *J Biomed Inform.* 2015, 53:156-61.
- [25] S. Tewari, U.D. Dwivedi. “A comparative study of heterogeneous ensemble methods for the identification of geological lithofacies.” *J Petrol Explor Prod Technol.* 2020, 10, 1849–1868.
- [26] N. Chandrasekhar, S. Peddakrishna, “Enhancing heart disease prediction accuracy through machine learning techniques and optimization.” *Processes* 2023, 11, 1210.
- [27] Y. Xie, C. Zhu, W. Zhou, Z. Li, X. Liu, T. Tu. “Evaluation of machine learning methods for formation lithology identification: a comparison of tuning process and model performance.” *J Pet Sci Eng* 2018, 60:182–193.
- [28] D.M. Belete, M. D. Huchaiah. “Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results.”, *International Journal of Computers and Applications*, 2022, 44:9, 875-886.
- [29] S.M. Lundberg and S.I. Lee, “A unified approach to interpreting model predictions.” *Advances in neural information processing systems*, 2017, 30.

### Conflict of Interest Notice

The author declare that there is no conflict of interest regarding the publication of this paper.

### Ethical Approval and Informed Consent

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

### Availability of data and material

Not applicable

### Plagiarism Statement

This article has been scanned by iThenticate™.



# A Systematic Review for Misuses Attack Detection based on Data Mining in NFV

Nebras Jalel Ibrahim<sup>1</sup> , Ahmed K. Abbas<sup>2</sup> , Farah Hatem Khorsheed<sup>3</sup> 

<sup>1</sup> Computer Center, University of Diyala, Diyala / Iraq

<sup>2</sup> Collage of Education for pure science, University of Diyala, Diyala / Iraq

<sup>3</sup> Computer Center, University of Diyala, Diyala / Iraq



## Corresponding author:

Ahmed K. Abbas, College of Education  
for Pure Science, Diyala University, Diyala-Iraq  
E-mail address:  
[dr.ahmed.k.abbas@uodiyala.edu.iq](mailto:dr.ahmed.k.abbas@uodiyala.edu.iq)

Submitted: 20 October 2023

Revision Requested: 11 December 2023

Last Revision Received: 19 December 2023

Accepted: 20 December 2023

Published Online: 27 December 2023

Citation: N. Ibrahim, A. Abbas, and  
F. Khorsheed, (2023). A Systematic Review for  
Misuses Attack Detection based on Data Mining I  
n NFV. *Sakarya University Journal of  
Computer and Information Sciences*, 6 (3)  
<https://doi.org/10.35377/saucis...1379047>

## ABSTRACT

Network Function Virtualization could be a quickly advancing innovation that guarantees to revolutionize the way networks are planned, sent, and overseen. However, as with any modern innovation, there are potential security risk that must be tended to guarantee the security of the network. Misuses attacks are one such risk that can compromise the security and integrity of NFV frameworks.

In recently years , data mining has risen as a promising approach for recognizing misuses attacks in NFV systems. The novelty of this systematic mapping study is ponders points to supply an overview of the existing research on misuses attack detection based on data mining in NFV. Particularly, the study will recognize and analyze the research conducted in this region, counting the sorts of data mining methods utilized, the types of misuses attacks identified, and the assessment strategies utilized.

The results of this study will give experiences into the current state of investigate on misuses attack detection based on data mining in NFV, as well as recognize gaps and openings for future research in this range. Also, the study will serve as an important asset for analysts and professionals looking for to create successful and effective methods for recognizing misuses attacks in NFV frameworks

**Keywords:** Misuses attack detection, Data mining, Network Function Virtualization (NFV), Systematic mapping

## 1. Introduction

Network Function Virtualization (NFV) is a technology that enables the deployment of network functions as software-based services that can run on standard servers and cloud infrastructure. NFV promises to reduce costs, improve network flexibility, and accelerate service delivery. However, the use of NFV also introduces new security challenges that need to be addressed [1].

One of the primary security concerns in NFV systems is the threat of misuses attacks. Misuses attacks occur when an attacker misuses a legitimate access point or privilege to gain unauthorized access to the network or its resources [2]. These attacks can result in data breaches, service disruptions, and other serious security incidents. To address the threat of misuses attacks in NFV systems, researchers have explored the use of data mining techniques for detecting such attacks. Data mining is a process of discovering patterns and knowledge from large datasets using statistical and computational techniques [3].

The use of data mining for detecting misuses attacks in NFV systems has several advantages. It allows for the detection of previously unknown attacks, can identify complex attack patterns, and can be used to analyze large amounts of network data in real-time [4].

This systematic mapping study aims to provide an overview of the existing research on misuses attack detection based on data mining in NFV [5]. The study will identify the types of data mining techniques used, the types of misuses attacks detected, and the evaluation methods employed in previous research [6]. The results of this study will help researchers and practitioners to develop more effective and efficient techniques for detecting misuses attacks in NFV systems, thereby enhancing the security and resilience of these systems.



## 2. Literature Review

Misuses attacks are one of the most significant security threats in NFV systems. As NFV systems are designed to be flexible and scalable, they are vulnerable to a wide range of misuses attacks that can compromise their security and integrity. Therefore, researchers have been exploring various techniques to detect misuses attacks in NFV systems.

Shilan S. Hameed et al 2021 designed a systematic review that explores the role of machine learning approaches in addressing the security requirements of IoT devices and systems. The authors created a list of research questions, the authors searched for relevant papers from different databases including IEEE, Web of Science, Springer Link, Scopus, and Science Direct. The most specific and relevant papers were extracted to answer the research questions. Later on, the selected papers were comprehensively screened and analyzed. Finally, the results were presented using different methods [7].

In another study, Zhang et al. (2020) proposed a misuses attack detection system for NFV based on ensemble learning techniques. The proposed system combined multiple classifiers to improve the accuracy of misuses attack detection in NFV[8].

Additionally, Mohamed Amine Ferrag, Lei Shu, Hamouda Djallel, and Kim-Kwang Raymond Choo discuss the importance of implementing effective intrusion detection systems in the agriculture industry to prevent Distributed Denial of Service (DDoS) attacks[9].

In [10] Nadra Guizani and Arif Ghafoor from Purdue University (2020) discussed a network function virtualization system for detecting malware in large IoT based networks and addressed the challenges posed by the exponential growth of IoT devices and the need for effective software-based security systems.

Abdullah Emir Çil et al in 2021 proposed the use of a deep neural network (DNN) model to detect and classify DDoS attacks based on captured network traffic. The experiments conducted on a dataset of DDoS attacks showed a 99.99% success rate in detecting attacks and a 94.57% accuracy rate in classifying attack types. The study concludes that deep learning models, such as DNN, can be effectively used to combat DDoS attacks. Previous studies have also utilized deep learning models, such as Deep Belief Network (DBN), Stacked Autoencoders (SAE), Long Short-Term Memory (LSTM), and Deep Convolutional Neural Network (DCNN), for DDoS intrusion detection with high accuracy [30].

Overall, the literature suggests that data mining techniques have considerable potential for misuses attack detection in NFV systems. In [11] Sulaiman, N. S. et al. (2021) provide a comprehensive overview of various techniques used in detecting and preventing unauthorized access to computer systems. However, there is a need for further research to develop more effective and efficient techniques that can be applied to real-world NFV systems. The results of this systematic mapping study will help to identify gaps and opportunities for future research in this area.

## 3. Research Questions

The following are research questions that could guide a systematic mapping study for misuses attack detection based on data mining in NFV:

**Q1**\\ What are the databases that used in this study? And what are the models that are used to build different perspectives?

**Q2**\\ What Classification schemes have been used to assess the effectiveness of misuses attack detection based on data mining in NFV systems?

**Q3**\\ What types of misuses attacks have been detected using data mining techniques in NFV systems?

**Q4**\\ What are the types of data mining techniques that have been used for misuses attack detection in NFV systems?

### 3.1 Search Statement

The following is a search statement for a systematic mapping study on misuse attack detection based on data mining in Network Function Virtualization (NFV):

```
((("misuse attack" OR "misuse detection") AND ("data mining" OR "machine learning" OR "deep learning" OR "artificial intelligence")) AND ("network function virtualization" OR "NFV")) AND ("systematic mapping" OR "systematic review" OR "systematic literature review" OR "mapping study"))
```

This search statement includes keywords related to misuse attack detection, data mining, machine learning, artificial intelligence, and NFV. The search statement also includes terms related to systematic mapping studies, which will help identify relevant research in this area.

### 3.2 Search in databases

There are many different databases and platforms used by publishers to manage their content and information. However, some of the most widely used publisher databases include:

1. **Scopus:** A bibliographic database of scientific literature, including journals, books, and conference proceedings, published by Elsevier [12]
2. **ACM digital library:** A digital library that provides access to thousands of academic journals, books, and primary sources in the humanities, social sciences, and sciences [13].
3. **ProQuest:** A provider of digital information and research tools, including databases of academic journals, newspapers, and dissertations [14].
4. **IEEE Xplore:** A digital library of scientific and technical content published by the Institute of Electrical and Electronics Engineers (IEEE) [15].
5. **Springer:** is an international publisher that offers a wide range of opportunities for authors, customers, and partners. Springer is a leading scientific publisher that publishes in various fields [16].

We collected the papers in this study depending on the databases above (Appendix A).

### 4. Screening of Papers

In a systematic mapping review, the screening process typically involves several stages to identify relevant papers that will be included in the review [17]. The following are the general steps involved in the screening process. The figure below explains these steps:

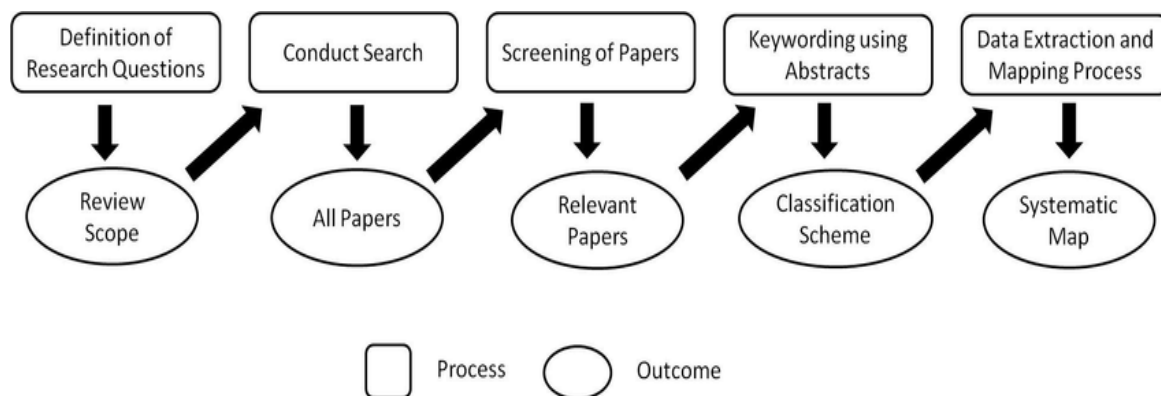


Figure 1. Systematic review process

#### 4.1 Use various models to build different perspectives

We can explain any schema or description of any topic by constructing schemas. Define an overall vision for the article on each topic and approach it with some options. In this article, we show how to use these scenarios as we explain below.

##### A. Distribution of studies according to years

This graph shows the distribution of the number of studies per year and the percentage of publications per year, it focuses on which papers have full pages or short pages.

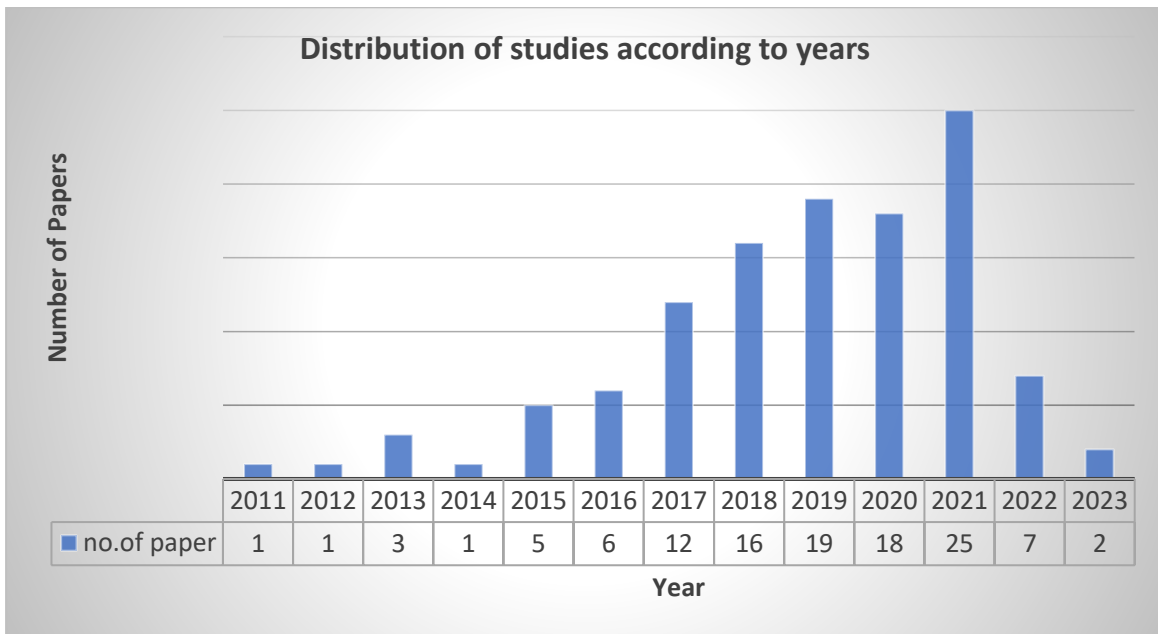


Figure 2. The distribution of studies in each year

**B. Distribution of studies according to Publication type**

The chart offers researchers a different perspective. Distribute papers by year, number of short or full-page papers, and paper type for conferences and journals.

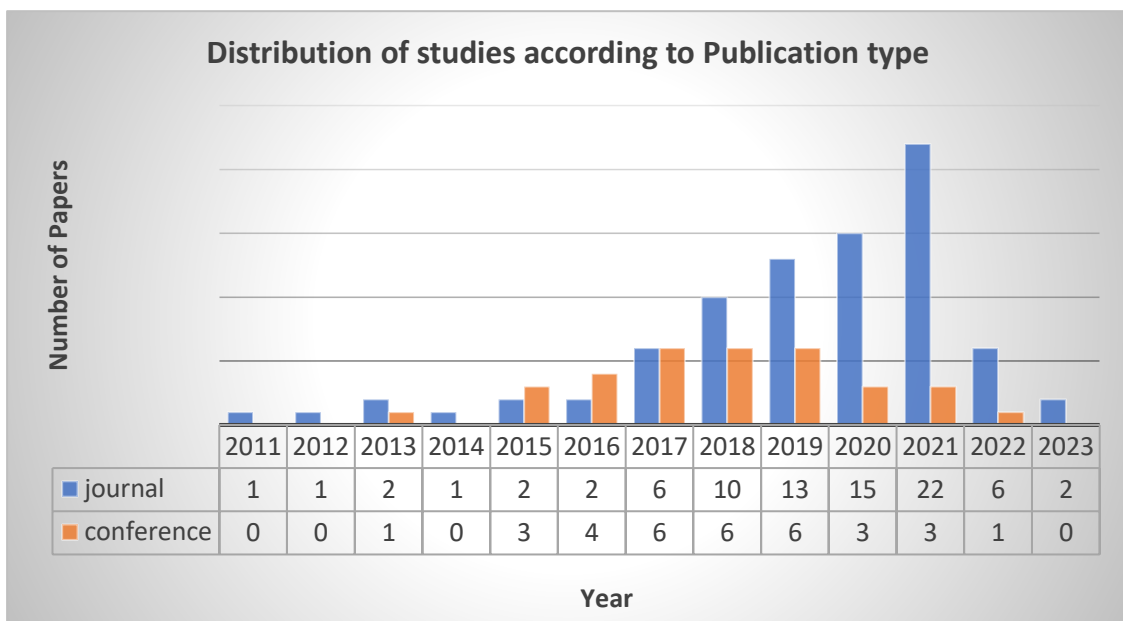


Figure 3. The Distribution of studies according to publication type

**C. Distribution of studies according to Country**

This chart shows the distribution of the number of studies per country.

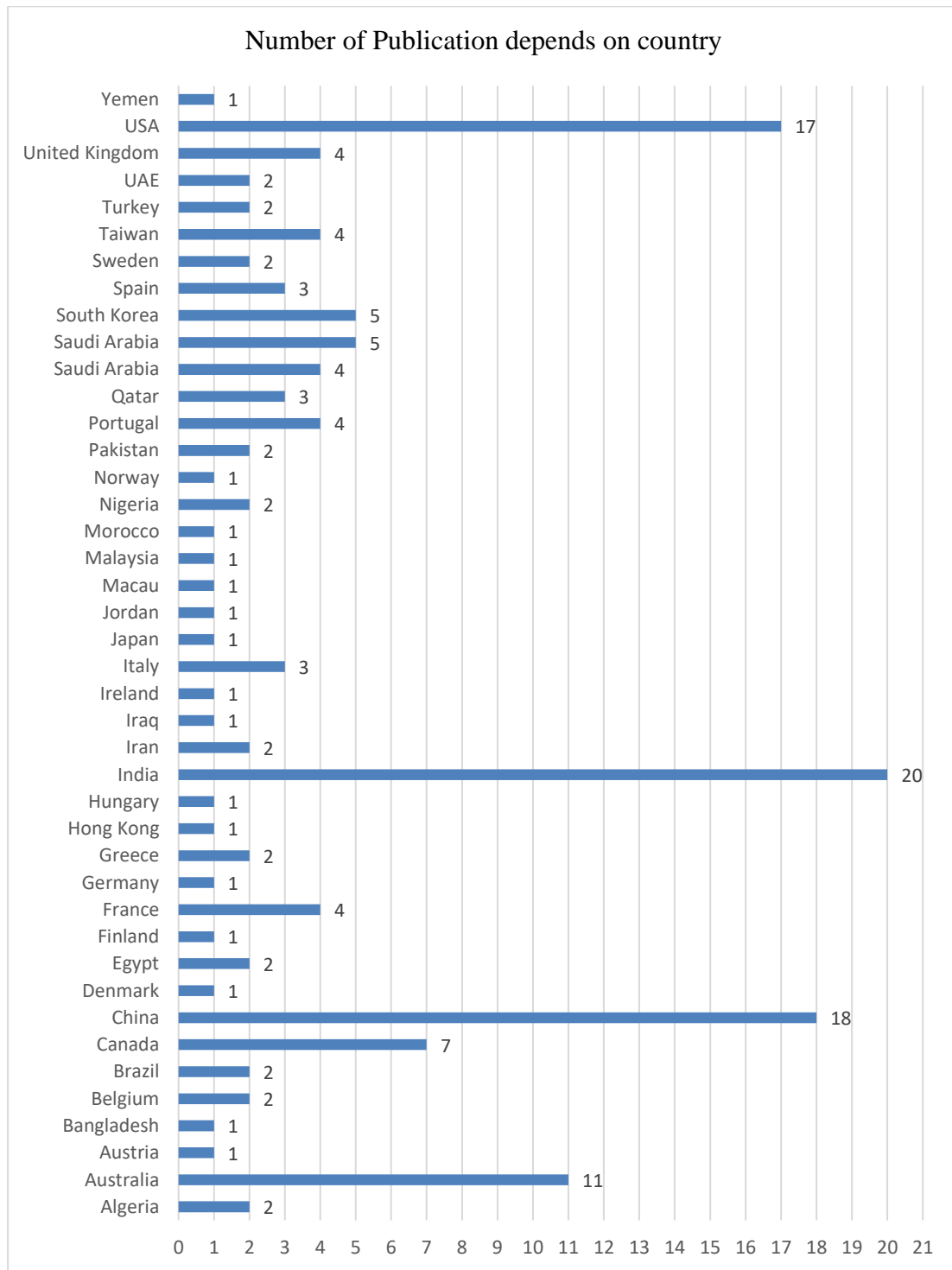


Figure 4. Distribution of studies according to country

**5. Classification schemes**

Systematic reviews are an important tool for synthesizing and summarizing the available evidence on a particular topic[18]. When it comes to classification schemes for systematic reviews of misuses attack detection based on data mining in NFV, there are a few different approaches that could be taken. Here are a few possibilities:

**A. Type of attack:**

One approach to classification could be to focus on the different types of attacks that are being detected using data mining techniques in NFV. This could include things like DDoS attacks, malware infections, phishing attempts, and so on.



Misuse attacks are a type of attack that involves exploiting vulnerabilities or weaknesses in a system by using legitimate functionality in an unauthorized or unintended way[19]. Misuse attacks can take many different forms, and the specific types of attacks that are relevant for misuses attack detection based on data mining in NFV may vary depending on the specific security domains and architectures being considered.

However, here are some common types of misuse attacks that could be relevant for misuses attack detection based on data mining in NFV:

1. **Denial-of-Service (DoS) attacks:** These attacks involve overwhelming a system or network with traffic or requests to make it unavailable to users. DoS attacks can be launched from multiple sources and can be difficult to detect and mitigate[20].
2. **Injection attacks:** These attacks involve injecting malicious code or data into a system or network, such as SQL injection or cross-site scripting (XSS) attacks. Injection attacks can bypass security measures and enable attackers to steal data or take control of systems[21].
3. **Malware attacks:** These attacks involve infecting systems or networks with malware, such as viruses, worms, or trojans. Malware can be used to steal data, disrupt operations, or launch further attacks[22].
4. **Brute-force attacks:** These attacks involve guessing passwords or other authentication credentials through trial and error. Brute-force attacks can be time-consuming but can be successful if passwords are weak or easily guessable[23].
5. **Evasion attacks:** These attacks involve attempting to bypass or evade security measures, such as by exploiting weaknesses in firewalls or intrusion detection systems. Evasion attacks can be difficult to detect and mitigate because they are designed to avoid detection[24].

The other attacks are:

- Unauthorized access. Unauthorized access refers to attackers accessing a network without receiving permission.
- Man in the middle attacks.
- Code and SQL injection attacks.
- Privilege escalation.
- Insider threats.

These are just a few examples of the types of misuse attacks that could be relevant for misuses attack detection based on data mining in NFV. The specific types of attacks will depend on the context and the security domains being considered.

## B. Data mining techniques:

Another approach could be to classify the different data mining techniques that are being used to detect misuses attacks in NFV. For example, one review might focus on studies that use decision trees, while another might focus on those that use neural networks or support vector machines.

There are several data mining techniques that can be used for misuses attack detection based on data mining in NFV. Here are some examples:

1. **Decision Trees:** Decision trees are a popular data mining technique for classification tasks. In the context of misuse attack detection in NFV, decision trees can be used to classify network traffic as either normal or malicious based on various features or attributes, such as packet size, protocol, or source IP address[25].
2. **Neural Networks:** Neural networks are another popular data mining technique that can be used for classification and prediction tasks. In the context of misuse attack detection in NFV, neural networks can be trained on historical network traffic data to identify patterns and anomalies that are indicative of malicious activity[26].
3. **Support Vector Machines (SVMs):** SVMs are a type of machine learning algorithm that can be used for classification and regression tasks. In the context of misuse attack detection in NFV, SVMs can be used to classify network traffic as either normal or malicious based on a set of features or attributes[27].
4. **Clustering:** Clustering is a data mining technique that involves grouping similar data points together based on their characteristics. In the context of misuse attack detection in NFV, clustering can be used to identify groups of network traffic that exhibit similar patterns or behaviors, which can then be analyzed further for potential malicious activity[28].
5. **Association Rule Mining:** Association rule mining is a data mining technique that involves identifying relationships or associations between different variables or attributes in a dataset. In the context of misuse attack detection in NFV,

association rule mining can be used to identify patterns or relationships between different network traffic features or attributes that are indicative of malicious activity[29].

Table 1. The intersection between types of misuse attack with data mining techniques.

Types of misuse attack	Denial of Services (DoS) attacks	Injection attacks	Malware attacks	Brute-force attacks	Evasion attacks	Others
Decision Tree	1,2,3,5,6,7,15,17,19,20,21,12,23,2,26,33,35,47,50,51,52,53,56,57,59,60,62,63,65,69,70,71,73,75,78,81,85,87,88,89,90,94,99,100,101,102,103,107,111,112,113,114,115,116	17,25,26,52,57,70,71,85,99	17,19,21,23,30,31,47,50,52,60,62,63,65,69,75,85,89,90,99,105,114	1	3,75	10,6,29,123,148
Neural Networks	1,6,8,14,15,19,20,21,13,14,17,23,25,26,33,34,35,36,41,42,46,47,49,50,51,52,53,55,56,57,58,59,60,62,63,65,67,69,71,73,74,75,76,78,79,81,83,85,87,88,89,90,92,94,96,99,100,101,102,103,107,110,111,112,113,114,115,116	14,25,26,32,42,46,48,49,52,57,71,85,99	8,13,14,19,17,21,23,30,31,35,41,42,43,46,47,50,52,56,60,62,63,65,69,75,79,83,85,89,90,92,96,99,105,110,114	1,56	75,83	,29,10,61,82,116
Support Vector Machine (SVM)	1,2,6,14,15,17,20,21,12,23,26,29,33,34,35,45,46,47,49,50,51,52,53,54,55,56,57,58,59,62,62,63,67,70,71,73,74,77,78,79,85,87,88,89,90,91,92,94,99,101,102,103,104,107,111,112,114,115,116	14,26,32,46,48,49,52,57,70,71,77,85,99	12,14,18,21,23,29,30,35,46,47,50,52,56,60,62,63,77,79,85,89,90,92,99,114	1,56		,10,6
Clustering	1,6,8,7,19,20,21,11,13,23,29,33,35,37,41,50,51,52,55,58,59,60,62,63,65,70,71,73,76,77,78,85,88,89,90,91,99,100,101,104,110,114,115,116	37,32,52,70,71,77,85,99	12,19,13,21,23,29,30,31,35,37,50,52,60,62,63,65,77,79,85,89,90,99,110,114	1		6,5,10,12,29,116
Association Rule Mining	20,33,59,63,71,79,90,104	71	30,63,79			10,12

Table 1 above represents the intersection between Data mining techniques and types of misuse attacks and Figure 5 below represents facet 1 (Types of misuse attacks with data mining techniques).

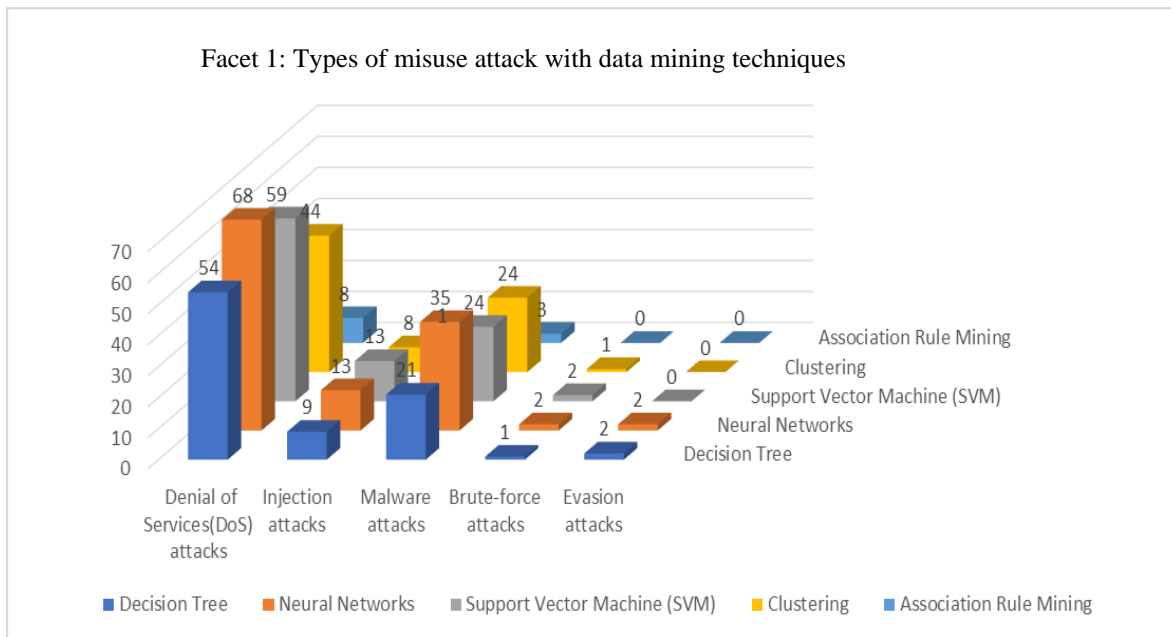


Figure 5. Types of misuse attack with data mining techniques

**6. Conclusion and Comments**

In conclusion, this systematic mapping study focused on the detection of misuses attacks in Network Function Virtualization (NFV) using data mining techniques. Through a comprehensive analysis of the existing literature, we identified and synthesized relevant studies, highlighting the various approaches, methodologies, and tools employed in this domain. The findings reveal that data mining plays a crucial role in the detection of misuses attacks in NFV, enabling the identification of anomalous patterns and the timely mitigation of potential threats. In this study, we apply different approaches like Type of attack and Data mining techniques as a classification schema and we note that most studies were used the Denial of Services (DoS) attacks with Decision Tree, Neural Networks, Support Vector Machine (SVM) and Clustering and at a lower frequency between Malware attacks and Decision Tree, Neural Networks, Support Vector Machine (SVM) and Clustering While attacks of Injection attacks, Brute-force attacks and Evasion attacks with data mining techniques this types have been studied very little compared to other types. The study also emphasizes the need for further research to address existing gaps, such as the development of more robust and efficient algorithms, the consideration of real-time detection, and the exploration of novel data sources. Ultimately, this systematic mapping study provides a valuable foundation for future researchers, practitioners, and stakeholders, serving as a reference point for advancing the field of misuses attack detection in NFV through data mining methodologies.

**References**

- [1] Firoozjaei, M. D., Jeong, J. P., Ko, H., & Kim, H. (2017). Security challenges with network functions virtualization. *Future Generation Computer Systems*, 67, 315-324.
- [2] Alnaim, A. K., Alwakeel, A. M., & Fernandez, E. B. (2022). Towards a security reference architecture for NFV. *Sensors*, 22(10), 3750.
- [3] Guleria, P., & Sood, M. (2014). Data mining in education: A review on the knowledge discovery perspective. *International Journal of Data Mining & Knowledge Management Process*, 4(5), 47.
- [4] Saeed, M. M. (2022). A real-time adaptive network intrusion detection for streaming data: a hybrid approach. *Neural Computing and Applications*, 34(8), 6227-6240.
- [5] Abbas, A. K., Fleh, S. Q., & Safi, H. H. (2015). Systematic Mapping Study On Managing Variability In Software Product Line Engineering: Communication. *Diyala Journal of Engineering Sciences*, 511-520.
- [6] Fleh, S. Q., Abbas, A. K., & Saffer, K. M. (2015, December). A systematic mapping study on runtime monitoring of services. In *The Iraqi Journal For Mechanical And Material Engineering, Special for Babylon First International Engineering Conference*, Issue (A).
- [7] Hameed, S. S., Hassan, W. H., Latiff, L. A., & Ghabban, F. (2021). A systematic review of security and privacy issues in the internet of medical things; the role of machine learning approaches. *PeerJ Computer Science*, 7, e414.
- [8] Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., & Sun, Y. (2019). A survey of networking applications applying the software defined networking concept based on machine learning. *IEEE Access*, 7, 95397-95417.
- [9] Ferrag, M. A., Shu, L., Djallel, H., & Choo, K. K. R. (2021). Deep learning-based intrusion detection for distributed

- denial of service attack in agriculture 4.0. *Electronics*, 10(11), 1257.
- [10] Guizani, N., & Ghafoor, A. (2020). A network function virtualization system for detecting malware in large IoT based networks. *IEEE Journal on Selected Areas in Communications*, 38(6), 1218-1228.
- [11] Sulaiman, N. S., Nasir, A., Othman, W. R. W., Wahab, S. F. A., Aziz, N. S., Yacob, A., & Samsudin, N. (2021, May). Intrusion detection system techniques: a review. In *Journal of Physics: Conference Series* (Vol. 1874, No. 1, p. 012042). IOP Publishing.
- [12] Elsevier, <https://www.elsevier.com>
- [13] Association for Computing Machinery, <https://dl.acm.org/>.
- [14] Proquest, <https://www.proquest.com/>.
- [15] IEEE, <https://ieeexplore.ieee.org/Xplore/home.jsp>.
- [16] Springer, <https://www.springer.com/gp>.
- [17] Lopez-Herrejon, R. E., Linsbauer, L., & Egyed, A. (2015). A systematic mapping study of search-based software engineering for software product lines. *Information and software technology*, 61, 33-51.
- [18] Aromataris, E., Fernandez, R., Godfrey, C. M., Holly, C., Khalil, H., & Tungpunkom, P. (2015). Summarizing systematic reviews: methodological development, conduct and reporting of an umbrella review approach. *JBIM Evidence Implementation*, 13(3), 132-140.
- [19] Shanmugam, B., & Idris, N. B. (2009, December). Improved intrusion detection system using fuzzy logic for detecting anomaly and misuse type of attacks. In *2009 International Conference of Soft Computing and Pattern Recognition* (pp. 212-217). IEEE.
- [20] Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE communications surveys & tutorials*, 18(1), 602-622.
- [21] Sharma, P., Johari, R., & Sarma, S. S. (2012). Integrated approach to prevent SQL injection attack and reflected cross site scripting attack. *International Journal of System Assurance Engineering and Management*, 3, 343-351.
- [22] Kaur, J. (2019). Taxonomy of malware: virus, worms and trojan. *Int. J. Res. Anal. Rev.*, 6(1), 192-196.
- [23] Khan, H. Z. U., & Zahid, H. (2010). Comparative study of authentication techniques. *International Journal of Video & Image Processing and Network Security IJVIPNS*, 10(04), 09-13.
- [24] Corona, I., Giacinto, G., & Roli, F. (2013). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. *Information Sciences*, 239, 201-225.
- [25] Sharma, H., & Kumar, S. (2016). A survey on decision tree algorithms of classification in data mining. *International Journal of Science and Research (IJSR)*, 5(4), 2094-2097.
- [26] Stahl, F., & Jordanov, I. (2012). An overview of the use of neural networks for data mining tasks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(3), 193-208.
- [27] Marir, N., Wang, H., Feng, G., Li, B., & Jia, M. (2018). Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark. *IEEE Access*, 6, 59657-59671.
- [28] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data: Recent advances in clustering* (pp. 25-71). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [29] Treinen, J. J., & Thurimella, R. (2006). A framework for the application of association rule mining in large intrusion detection infrastructures. In *Recent Advances in Intrusion Detection: 9th International Symposium, RAID 2006 Hamburg, Germany, September 20-22, 2006 Proceedings 9* (pp. 1-18). Springer Berlin Heidelberg.
- [30] Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169, 114520.

## Appendix A

- Gulzar, B., & Gupta, A. (2021). DAM: a theoretical framework for SensorSecurity in IoT applications. *International Journal of Next-Generation Computing*, 12(3), 10-47164.
- Zhao, S., Chandrashekar, M., Lee, Y., & Medhi, D. (2015, March). Real-time network anomaly detection system using machine learning. In *2015 11th international conference on the design of reliable communication networks (drcn)* (pp. 267-270). IEEE.
- Barradas, D., Santos, N., Rodrigues, L., Signorello, S., Ramos, F. M., & Madeira, A. (2021, February). FlowLens: Enabling Efficient Flow Classification for ML-based Network Security Applications. In *NDSS*.
- Baktayan, A., & Albaltah, I. A. (2022). A blockchain-based trust management system for 5G network slicing enabled C-RAN. *Sustainable Engineering and Innovation*, 4(1), 8.
- Lee, S., Kim, J., Shin, S., Porras, P., & Yegneswaran, V. (2017, June). Athena: A framework for scalable anomaly detection in software-defined networks. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 249-260). IEEE.
- Li, J., Zhao, Z., & Li, R. (2017). A machine learning based intrusion detection system for software defined 5G network. *arXiv preprint arXiv:1708.04571*.
- Li, J., Zhao, Z., Li, R., & Zhang, H. (2018). Ai-based two-stage intrusion detection for software defined iot

- networks. *IEEE Internet of Things Journal*, 6(2), 2093-2102.
8. Wu, Y., Dai, H. N., & Wang, H. (2020). Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet of Things Journal*, 8(4), 2300-2317.
  9. Jauro, F., Chiroma, H., Gital, A. Y., Almutairi, M., Shafi'i, M. A., & Abawajy, J. H. (2020). Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend. *Applied Soft Computing*, 96, 106582.
  10. Zou, D., Lu, Y., Yuan, B., Chen, H., & Jin, H. (2018). A fine-grained multi-tenant permission management framework for SDN and NFV. *IEEE Access*, 6, 25562-25572.
  11. Darwish, T. S., & Bakar, K. A. (2018). Fog based intelligent transportation big data analytics in the internet of vehicles environment: motivations, architecture, challenges, and critical issues. *IEEE Access*, 6, 15679-15701.
  12. Corrêa, J. H., Ciarelli, P. M., Ribeiro, M. R., & Villaça, R. S. (2021). MI-based ddos detection and identification using native cloud telemetry macroscopic monitoring. *Journal of Network and Systems Management*, 29, 1-28.
  13. Alharbi, T., Aljuhani, A., & Taylor, B. (2019). A collaborative SYN flooding detection ApproachA collaborative SYN. *International Journal of Computer Engineering and Information Technology*, 11(9), 186-196.
  14. Zhou, C., Hu, B., Shi, Y., Tian, Y. C., Li, X., & Zhao, Y. (2020). A unified architectural approach for cyberattack-resilient industrial control systems. *Proceedings of the IEEE*, 109(4), 517-541.
  15. Ferrag, M. A., Shu, L., Djallel, H., & Choo, K. K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics*, 10(11), 1257.
  16. Zhang, H., Wang, Y., Chen, H., Zhao, Y., & Zhang, J. (2017). Exploring machine-learning-based control plane intrusion detection techniques in software defined optical networks. *Optical Fiber Technology*, 39, 37-42.
  17. Salahdine, F., Han, T., & Zhang, N. (2023). Security in 5G and beyond recent advances and future challenges. *Security and Privacy*, 6(1), e271.
  18. DEORE, M., MANE, D., UPADHYE, G., & KITTAD, N. (2022). THE SECURITY CONCERNS AND SOLUTIONS FOR CLOUD-BASED IOT SYSTEM. *Journal of Theoretical and Applied Information Technology*, 100(18).
  19. Kim, H., Kim, J., Kim, Y., Kim, I., & Kim, K. J. (2019). Design of network threat detection and classification based on machine learning on cloud computing. *Cluster Computing*, 22, 2341-2350.
  20. D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2020). Inter-dataset generalization strength of supervised machine learning methods for intrusion detection. *Journal of Information Security and Applications*, 54, 102564.
  21. Ahmad, I., Shahabuddin, S., Malik, H., Harjula, E., Leppänen, T., Loven, L., ... & Riekkki, J. (2020). Machine learning meets communication networks: Current trends and future challenges. *IEEE Access*, 8, 223418-223460.
  22. Kulkarni, P., & Cauvery, N. K. (2021). Personally Identifiable Information (PII) Detection in the Unstructured Large Text Corpus using Natural Language Processing and Unsupervised Learning Technique. *International Journal of Advanced Computer Science and Applications*, 12(9).
  23. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, 6, 35365-35381.
  24. Yang, H. (2020, October). Research on Classification Algorithm for Civil Aviation Internal Network Intrusion Detection Based on Machine Learning. In *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 1-4). IEEE.
  25. Riera, T. S., Higuera, J. R. B., Higuera, J. B., Herraiz, J. J. M., & Montalvo, J. A. S. (2022). A new multi-label dataset for Web attacks CAPEC classification using machine learning techniques. *Computers & Security*, 120, 102788.
  26. Derhab, A., Guerroumi, M., Gumaei, A., Maglaras, L., Ferrag, M. A., Mukherjee, M., & Khan, F. A. (2019). Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security. *Sensors*, 19(14), 3119.
  27. Adhikari, N., & Ramkumar, M. (2023). IoT and Blockchain Integration: Applications, Opportunities, and Challenges. *Network*, 3(1), 115-141.
  28. Overmars, A., & Venkatraman, S. (2020). Towards a secure and scalable iot infrastructure: A pilot deployment for a smart water monitoring system. *Technologies*, 8(4), 50.
  29. Al Makdi, K., Sheldon, F. T., & Hussein, A. A. (2020, November). Trusted Security Model for IDS Using Deep Learning. In *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)* (pp. 1-4). IEEE.
  30. Liu, Y., Yu, F. R., Li, X., Ji, H., & Leung, V. C. (2020). Blockchain and machine learning for communications and networking systems. *IEEE Communications Surveys & Tutorials*, 22(2), 1392-1431.
  31. Zago, M., Gil Pérez, M., & Martínez Pérez, G. (2021). Early DGA-based botnet identification: pushing detection to the edges. *Cluster Computing*, 1-16.
  32. Bertero, C., Roy, M., Sauvanaud, C., & Trédan, G. (2017, October). Experience report: Log mining using natural language processing and application to anomaly detection. In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 351-360). IEEE.
  33. D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2019). In-depth comparative evaluation of supervised machine learning approaches for detection of cybersecurity threats. In *4th International Conference on Internet of Things, Big Data and Security (IoTBDS)* (pp. 125-136).
  34. Phan, T. V., & Park, M. (2019). Efficient distributed denial-of-service attack defense in SDN-based cloud. *IEEE Access*, 7, 18701-18714.
  35. Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2017). HAST-IDS: Learning hierarchical

- spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE access*, 6, 1792-1806.
36. Soldani, D. (2020). On Australia's cyber and critical technology international engagement strategy towards 6G: How Australia May become a leader in cyberspace. *Journal of Telecommunications and the Digital Economy*, 8(4), 127-158.
  37. Iqbal, W., Abbas, H., Daneshmand, M., Rauf, B., & Bangash, Y. A. (2020). An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. *IEEE Internet of Things Journal*, 7(10), 10250-10276.
  38. Padhi, P. K., & Charrua-Santos, F. (2021). 6G enabled tactile internet and cognitive internet of healthcare everything: Towards a theoretical framework. *Applied System Innovation*, 4(3), 66.
  39. Aazam, M., Zeadally, S., & Harras, K. A. (2018). Deploying fog computing in industrial internet of things and industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(10), 4674-4682.
  40. Zunino, C., Valenzano, A., Obermaisser, R., & Petersen, S. (2020). Factory communications at the dawn of the fourth industrial revolution. *Computer Standards & Interfaces*, 71, 103433.
  41. Varga, P., Peto, J., Franko, A., Balla, D., Haja, D., Janky, F., ... & Toka, L. (2020). 5g support for industrial iot applications—challenges, solutions, and research gaps. *Sensors*, 20(3), 828.
  42. Ferrag, M. A., & Shu, L. (2021). The performance evaluation of blockchain-based security and privacy systems for the Internet of Things: A tutorial. *IEEE Internet of Things Journal*, 8(24), 17236-17260.
  43. Joshi, K. D., & Kataoka, K. (2020). pSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN. *Computer Networks*, 178, 107295.
  44. Gedeon, J., Brandherm, F., Egert, R., Grube, T., & Mühlhäuser, M. (2019). What the fog? edge computing revisited: Promises, applications and future challenges. *IEEE Access*, 7, 152847-152878.
  45. Wang, Y., Meng, W., Li, W., Liu, Z., Liu, Y., & Xue, H. (2019). Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. *Concurrency and Computation: Practice and Experience*, 31(19), e5101.
  46. Tien, C. W., Huang, T. Y., Tien, C. W., Huang, T. C., & Kuo, S. Y. (2019). Kubanomaly: anomaly detection for the docker orchestration platform with neural network approaches. *Engineering reports*, 1(5), e12080.
  47. Abbasi, H., Ezzati-Jivan, N., Bellaiche, M., Talhi, C., & Dagenais, M. R. (2019). Machine learning-based EDoS attack detection technique using execution trace analysis. *Journal of Hardware and Systems Security*, 3, 164-176.
  48. Tekerek, A. (2021). A novel architecture for web-based attack detection using convolutional neural network. *Computers & Security*, 100, 102096.
  49. Ujjan, R. M. A., Pervez, Z., Dahal, K., Bashir, A. K., Mumtaz, R., & González, J. (2020). Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. *Future Generation Computer Systems*, 111, 763-779.
  50. Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, 7, 100059.
  51. Kushwah, G. S., & Ranga, V. (2020). Voting extreme learning machine based distributed denial of service attack detection in cloud computing. *Journal of Information Security and Applications*, 53, 102532.
  52. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access*, 7, 41525-41550.
  53. Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access*, 8, 73127-73141.
  54. Deepa, V., Sudar, K. M., & Deepalakshmi, P. (2018, December). Detection of DDoS attack on SDN control plane using hybrid machine learning techniques. In *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 299-303). IEEE.
  55. Yang, K., Ren, J., Zhu, Y., & Zhang, W. (2018). Active learning for wireless IoT intrusion detection. *IEEE Wireless Communications*, 25(6), 19-25.
  56. Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management*, 18(2), 1803-1816.
  57. Deepa, V., Sudar, K. M., & Deepalakshmi, P. (2019, March). Design of ensemble learning methods for DDoS detection in SDN environment. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)* (pp. 1-6). IEEE.
  58. Zhu, Y., Gaba, G. S., Almansour, F. M., Alroobaea, R., & Masud, M. (2021). Application of data mining technology in detecting network intrusion and security maintenance. *Journal of Intelligent Systems*, 30(1), 664-676.
  59. Wu, K., & De Soto, B. G. (2022). Current State and Future Opportunities of Data Mining for Construction 4.0. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 39, pp. 78-85)*. IAARC Publications.
  60. Sangwan, U., & Chhillar, R. S. (2022). Comparison of Various Classification Techniques in Cyber Security Using Iot. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3), 334-339.
  61. Nadig, D., Ramamurthy, B., Bockelman, B., & Swanson, D. (2018, March). Identifying anomalies in gridftp transfers for data-intensive science through application-awareness. In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization* (pp. 7-12).
  62. Awotunde, J. B., Chakraborty, C., & Adeniyi, A. E. (2021). Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wireless communications and mobile computing*, 2021,

- 1-17.
63. Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE communications surveys & tutorials*, 21(1), 686-728.
  64. Yousefian, N., Hansen, J. H., & Loizou, P. C. (2014). A hybrid coherence model for noise reduction in reverberant environments. *IEEE Signal Processing Letters*, 22(3), 279-282.
  65. Al-Hawawreh, M. S. (2017, May). SYN flood attack detection in cloud environment based on TCP/IP header statistical features. In *2017 8th International Conference on Information Technology (ICIT)* (pp. 236-243). IEEE.
  66. Ali, A. K., & Bhaya, W. S. (2021, March). Detection of Misuse Attack in NFV Networks Using Machine Learning. In *Journal of Physics: Conference Series* (Vol. 1818, No. 1, p. 012123). IOP Publishing.
  67. Amaizu, G. C., Nwakanma, C. I., Bhardwaj, S., Lee, J. M., & Kim, D. S. (2021). Composite and efficient DDoS attack detection framework for 5G networks. *Computer Networks*, 188, 107871.
  68. Anwer, H. M., Farouk, M., & Abdel-Hamid, A. (2018, April). A framework for efficient network anomaly intrusion detection with features selection. In *2018 9th International Conference on Information and Communication Systems (ICICS)* (pp. 157-162). IEEE.
  69. Mozo, A., Pastor, A., Karamchandani, A., de la Cal, L., Rivera, D., & Moreno, J. I. (2022). Integration of Machine Learning-Based Attack Detectors into Defensive Exercises of a 5G Cyber Range. *Applied Sciences*, 12(20), 10349.
  70. Rezvani, M. (2018). Assessment methodology for anomaly-based intrusion detection in cloud computing. *Journal of AI and Data Mining*, 6(2), 387-397.
  71. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1), 303-336.
  72. Chauhan, S., & Vig, L. (2015, October). Anomaly detection in ECG time signals via deep long short-term memory networks. In *2015 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 1-7). IEEE.
  73. Chou, H. H., & Wang, S. D. (2015, September). An adaptive network intrusion detection approach for the cloud environment. In *2015 international carnaham conference on security technology (iCCST)* (pp. 1-6). IEEE.
  74. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, 5, 21954-21961.
  75. Nazir, A., & Khan, R. A. (2019). Combinatorial optimization based feature selection method: A study on network intrusion detection. *arXiv preprint arXiv:1906.04494*.
  76. Cotroneo, D., Natella, R., & Rosiello, S. (2017, October). A fault correlation approach to detect performance anomalies in virtual network function chains. In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 90-100). IEEE.
  77. Cruz, T., Rosa, L., Proença, J., Maglaras, L., Aubigny, M., Lev, L., ... & Simões, P. (2016). A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics*, 12(6), 2236-2246.
  78. Dimolianis, M., Pavlidis, A., & Maglaris, V. (2021). Signature-based traffic classification and mitigation for ddos attacks using programmable network data planes. *IEEE Access*, 9, 113061-113076.
  79. Kazemi, S., Aghazarian, V., & Hedayati, A. (2015). Improving false negative rate in hypervisor-based intrusion detection in IaaS cloud. *IJCAT Int. J. Comput. Technol.*, 2(9), 348.
  80. Pallotta, G., Vespe, M., & Bryan, K. (2013). Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy*, 15(6), 2218-2245.
  81. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science*, 89, 213-217.
  82. Moustafa, N., Keshky, M., Debiez, E., & Janicke, H. (2020, December). Federated TON\_IoT Windows datasets for evaluating AI-based security applications. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (pp. 848-855). IEEE.
  83. Gamal, M., Abbas, H. M., Moustafa, N., Sitnikova, E., & Sadek, R. A. (2021). Few-shot learning for discovering anomalous behaviors in edge networks. *Computers, Materials and Continua*, 69(2), 1823-1837.
  84. Alnaim, A. K. (2022). Misuse Patterns from the Threat of Modification of Non-Control Data in Network Function Virtualization. *Future Internet*, 14(7), 201.
  85. Bhardwaj, A., Mangat, V., & Vig, R. (2020). Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud. *IEEE Access*, 8, 181916-181929.
  86. Gandhi, K., & Qaddour, J. (n.d.). Implementation Problems Facing Network Function Virtualization and Solutions.
  87. Idhammad, M., Afdel, K., & Belouch, M. (2018). Distributed intrusion detection system for cloud environments based on data mining techniques. *Procedia Computer Science*, 127, 35-41.
  88. Li, J., Zhao, Z., & Li, R. (2018). Machine learning-based IDS for software-defined 5G network. *Iet Networks*, 7(2), 53-60.
  89. Dhaliwal, S. S., Nahid, A. A., & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information*, 9(7), 149.
  90. Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2021). Machine learning towards intelligent systems: applications, challenges, and opportunities. *Artificial Intelligence Review*, 54, 3299-3348.
  91. Aiken, J., & Scott-Hayward, S. (2019, November). Investigating adversarial attacks against network intrusion detection

- systems in sdns. In 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) (pp. 1-7). IEEE.
92. Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In 2017 International conference on information networking (ICOIN) (pp. 712-717). IEEE.
  93. Thang, N. C., & Park, M. (2020). Detecting Malicious Middleboxes In Service Function Chaining. *J. Internet Serv. Inf. Secur.*, 10(2), 82-90.
  94. Koc, L., Mazzuchi, T. A., & Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications*, 39(18), 13492-13500.
  95. Lavin, A., & Ahmad, S. (2015, December). Evaluating real-time anomaly detection algorithms--the Numenta anomaly benchmark. In 2015 IEEE 14th international conference on machine learning and applications (ICMLA) (pp. 38-44). IEEE.
  96. Leu, F. Y., Tsai, K. L., Hsiao, Y. T., & Yang, C. T. (2015). An internal intrusion detection and protection system by using data mining and forensic techniques. *IEEE Systems Journal*, 11(2), 427-438.
  97. Tian, Z., Cui, Y., An, L., Su, S., Yin, X., Yin, L., & Cui, X. (2018). A real-time correlation of host-level events in cyber range service for smart campus. *IEEE Access*, 6, 35355-35364.
  98. Lu, H., Li, Y., Mu, S., Wang, D., Kim, H., & Serikawa, S. (2017). Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE internet of things journal*, 5(4), 2315-2322.
  99. Modi, C. N., & Acha, K. (2017). Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. *the Journal of Supercomputing*, 73(3), 1192-1234.
  100. Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1-3), 18-31.
  101. Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE access*, 6, 48231-48246.
  102. Wang, Z., Liu, Y., He, D., & Chan, S. (2021). Intrusion detection methods based on integrated deep learning model. *Computers & Security*, 103, 102177.
  103. Shareena, J., Ramdas, A., & AP, H. (2021). Intrusion detection system for iot botnet attacks using deep learning. *SN Computer Science*, 2(3), 205.
  104. Youssef, A., & Emam, A. (2011). Network intrusion detection using data mining and network behaviour analysis. *International journal of computer science & information technology*, 3(6), 87.
  105. Peiravian, N., & Zhu, X. (2013, November). Machine learning for android malware detection using permission and api calls. In 2013 IEEE 25th international conference on tools with artificial intelligence (pp. 300-305). IEEE.
  106. Razdan, S., Gupta, H., & Seth, A. (2021, April). Performance analysis of network intrusion detection systems using j48 and naive bayes algorithms. In 2021 6th International Conference for Convergence in Technology (I2CT) (pp. 1-7). IEEE.
  107. de Miranda Rios, V., Inácio, P. R., Magoni, D., & Freire, M. M. (2021). Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. *Computer Networks*, 186, 107792.
  108. Sauvanaud, C., Lazri, K., Kaâniche, M., & Kanoun, K. (2016, October). Anomaly detection and root cause localization in virtual network functions. In 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE) (pp. 196-206). IEEE.
  109. Sauvanaud, C., Lazri, K., Kaâniche, M., & Kanoun, K. (2016, June). Towards black-box anomaly detection in virtual network functions. In 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W) (pp. 254-257). IEEE.
  110. Thamilarasu, G., & Chawla, S. (2019). Towards deep-learning-driven intrusion detection for the internet of things. *Sensors*, 19(9), 1977.
  111. Alsharif, M., & Rawat, D. B. (2021). Study of machine learning for cloud assisted iot security as a service. *Sensors*, 21(4), 1034.
  112. Zhong, M., Zhou, Y., & Chen, G. (2021). Sequential model based intrusion detection system for IoT servers using deep learning methods. *Sensors*, 21(4), 1113.
  113. Song, C., Park, Y., Golani, K., Kim, Y., Bhatt, K., & Goswami, K. (2017, July). Machine-learning based threat-aware system in software defined networks. In 2017 26th international conference on computer communication and networks (ICCCN) (pp. 1-9). IEEE.
  114. Soni, S., & Bhushan, B. (2019, July). Use of Machine Learning algorithms for designing efficient cyber security solutions. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT) (Vol. 1, pp. 1496-1501). IEEE.
  115. Stroeh, K., Mauro Madeira, E. R., & Goldenstein, S. K. (2013). An approach to the correlation of security events based on machine learning techniques. *Journal of Internet Services and Applications*, 4, 1-16.
  116. Aboueata, N., Alrasbi, S., Erbad, A., Kassler, A., & Bhamare, D. (2019, July). Supervised machine learning techniques for efficient network intrusion detection. In 2019 28th International Conference on Computer Communication and Networks (ICCCN) (pp. 1-8). IEEE.



**Conflict of Interest Notice**

The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

**Availability of data and material**

Not applicable.

**Plagiarism Statement**

This article has been scanned by iThenticate™.



# Performance Evaluation of OTFS-NOMA Scheme for High Mobility Users

İnci Umakoğlu<sup>1</sup> , Mustafa Namdar<sup>1</sup> , Arif Başgümüş<sup>2</sup> 

<sup>1</sup> Department of Electrical and Electronics Engineering, Kütahya Dumlupınar University, Kütahya, Türkiye

<sup>2</sup> Department of Electrical and Electronics Engineering, Bursa Uludağ University, Bursa, Türkiye



## Corresponding author:

İnci Umakoğlu, Department of Electrical and Electronics Engineering Kutahya Dumlupınar University

E-mail address:  
[inci.umakoglu@dpu.edu.tr](mailto:inci.umakoglu@dpu.edu.tr)

Submitted: 16 November 2023  
Revision Requested: 29 November 2023  
Last Revision Received: 30 November 2023  
Accepted: 20 December 2023  
Published Online: 27 December 2023

Citation: İ. Umakoğlu, M. Namdar, and A. Başgümüş, Performance Evaluation of OTFS-NOMA Scheme for High Mobility Users. *Sakarya University Journal of Computer and Information Sciences*. 6 (3) <https://doi.org/10.35377/saucis...1391813>

## ABSTRACT

Orthogonal Time Frequency Space (OTFS) is a promising approach which is widely employed in sixth generation (6G) wireless network systems. Because of its superior performance in high-mobility environments, OTFS modulation has received a lot of attention lately. Due to OTFS modulation works in the delay-Doppler (DD) domain rather than the conventional time-frequency (TF) domain, it works effectively in such circumstances. The idea of non-orthogonal multiple access (NOMA) is integrated into OTFS as an important approach to improve the spectral efficiency (SE) to investigate the efficiency potential and performance. In this research, we study OTFS modulated NOMA system for two destination users in the high mobility environment. The message passing detection algorithm is utilized to examine bit error rate (BER) performance for both near and far users in the proposed OTFS modulated NOMA system. The BER simulation results demonstrate that the power allocation (PA) coefficient, delays, and Doppler effects significantly impact the performance of the system. It is observed that the performance of the far user did not drop below a particular BER level. The BER value for the  $U_1$  user is 0.1, while the BER value for the  $U_2$  user is nearly 0.25 at 10 dB SNR, resulting in a 2.5 times better BER performance in the 4-QAM scenario. The BER value is about 0.27 for the  $U_1$  user, while the BER value for the  $U_2$  user is approximately 0.33 at 10 dB SNR in the 16-QAM approach. It is concluded that the Doppler effect causes BER performance degradation for both users.

**Keywords:** Bit Error Rate, Doppler Effect, NOMA, OTFS

## 1. Introduction

Non-orthogonal multiple access (NOMA) is accepted as an important multiplexing approach for new generation wireless networks. The concept of NOMA is to improve the spectral efficiency (SE) of fifth generation (5G) and beyond communication networks. NOMA provides different power levels to multiple users at the same time in the same frequency band. Successive interference cancellation (SIC) is carried out in NOMA to decode incoming message information at the receiver side. As opposed to conventional orthogonal multiple access (OMA) methods, NOMA is a methodology that can offer higher SE performance, lower latency and maintain user fairness. Scenarios with quality of service requirements or varying channel conditions are grouped together for users with low-speed mobility in studies [1]–[3]. Numerous communication networks, including millimeter wave networks, multi-input multiple-output networks, and visible light communication systems are utilized with NOMA techniques.

Wireless communication is the area of the communication industry that is growing most rapidly. The demand for research beyond 5G and sixth generation (6G) is growing daily as scientists and researchers realize that new wireless communication techniques have to be developed for the growth of future infrastructure. Reliable communication is anticipated to be enabled in both time-invariant and time-variant wireless channels. It is difficult to meet the ever-increasing needs of 6G. Traditional orthogonal frequency-division multiplexing (OFDM) modulation, which has been widely utilized in 5G cellular networks, is susceptible to the high Doppler effect. The 5G methodologies and specifications are insufficient to fulfill the demands of future applications. The objective of sixth generation mobile technology is to improve communication in scenarios with high-speed mobility, such as vehicle-to-everything, Internet of Things, unmanned aerial vehicles, and high-speed rail. Orthogonal frequency division multiplexing (OFDM) is a very popular



solution for 4G and 5G wireless networks. OFDM is developed specifically to avoid inter-symbol interference due to the channel's time dispersion. Nonetheless, because of the Doppler effect, subcarriers are no longer orthogonal and intercarrier interference occurs in OFDM. Therefore, this causes severe performance degradations and OFDM is no longer robust in high-speed mobility environments [4]. To enhance the system performance and the SE of OFDM, numerous alternative multicarrier waveforms have been studied, including unified filter multicarrier, filter bank multicarrier, and generalized frequency-division multiplexing. The aforementioned waveforms, however, have been designed for low-mobility channels and would significantly degrade in performance when employed in high-mobility conditions because of the negative impact of a strong Doppler effect. Recently, the orthogonal time-frequency-space (OTFS) waveform has been proposed as a way of avoiding the limitation of OFDM in time-varying channels. OTFS has the benefit of allowing time-invariant channel gains to be used in the delay-Doppler (DD) domain in contrast to conventional modulation techniques such as OFDM. In conditions of high-speed mobility, this makes signal detection and channel estimation simpler. OTFS transmits information symbols in the DD domain as opposed to OFDM, which delivers in the time frequency (TF) domain. By doing this, a time-varying channel becomes time-invariant, ensuring that the frequency and time selectivity have the same effect on all symbols in the DD domain.

The methods that are proposed currently in the literature are usually categorized as either OMA or NOMA. Because of user multiplexing in the DD domain, only one user can have access to a given resource block in OTFS-OMA. However, due to Doppler spread, users suffer from multi-user interference. This interference is eliminated by adding guard bands between the destination users. Nevertheless, this leads to a loss of SE [5]. As an alternative approach, OTFS-NOMA allows users to utilize the same resource block. This is the reason underlying the latest suggestions in the literature for power domain [6], [7], and code domain [8], [9] multiplexed OTFS-NOMA system models. In this study, we concentrate on the power domain OTFS-NOMA. This study investigates its applicability to a communication environment in which users have profiles of high-speed mobility. OTFS modulation is used due to its performance in scenarios with doubly selective channels. OTFS utilizes the DD domain by placing users' signals orthogonally. In this study, a two-user downlink OTFS-NOMA system bit error rate (BER) performance is presented. The message transmission (MP) detection technique, which makes use of the DD channel's sparseness, is used at the receiver to accomplish symbol detection. By employing a sparse factor graph-based Gaussian approximation of the interference terms, this method significantly decreases complexity.

Future vehicular networks, underwater acoustic communications (UACs), non-terrestrial networks (NTN), as well as Millimeter-wave and Terahertz communication, all common wireless communication scenarios especially for the feasibility of high-speed mobility scenarios in 6G wireless networks find OTFS to be a desirable option due to its advantageous robustness to the Doppler effect. In addition to being resistant to time-varying channels, it also has a lower peak-to-average power ratio than its OFDM equivalent. Owing to its strong OFDM compatibility, OTFS is also a strong alternative for 6G communication systems.

Especially, BER performances have not been widely investigated for the power-domain downlink OTFS-NOMA system. Motivated by this, we provide a thorough description of the OTFS-NOMA scheme for high mobility users in this manuscript, including its system architecture, channel model, numerical results with delays, Doppler effect, and modulation types.

The study consists of the following sections. In the second section of this study, the OTFS-NOMA system scheme is demonstrated. In this section, the OTFS-NOMA system model, TF domain, DD domain, channel model, and general principles of OTFS are described. In the third section of this study, the BER performance for the proposed system model and numerical results are presented. Finally, the conclusion is given in Section 4.

## 2. OTFS-NOMA System Model

In the OTFS-based power domain NOMA system model, users with high-speed mobility use the same DD domain source with different transmission power. As seen in Figure 1, an OTFS-based downlink NOMA system model with source  $T$ , two destination users namely near the user and from the user,  $U_i$ ,  $i \in \{1,2\}$ , the number of users,  $K = 2$ , and reflectors are investigated. The received signal is the sum of delayed, attenuated, and Doppler-shifted copies for the destination users. The delay is a function of the length of each propagation path, while the Doppler shift is a result of the relative motion of the receiver and reflectors for the case where the transmitter is considered stationary.

### 2.1 TF Domain and DD Domain

OTFS-NOMA uses both the TF domain and the DD domain efficiently. By sampling with time interval  $T$  and frequency interval  $\Delta f$ , a discrete TF domain is obtained as seen in Equation 1

$$\Lambda_{TF} = \{(nT, m\Delta f), n = 0, \dots, N - 1, m = 0, \dots, M - 1\}, \quad (1)$$

where  $N, M > 0$  [10], [11]. Accordingly, the discrete DD domain is as seen in Equation 2

$$\Lambda_{DD} = \left\{ \left( \frac{k}{NT}, \frac{l}{M\Delta f} \right), k = 0, \dots, N - 1, l = 0, \dots, M - 1 \right\}, \tag{2}$$

where  $N$  and  $M$  are the total number of time intervals and frequency subcarriers.

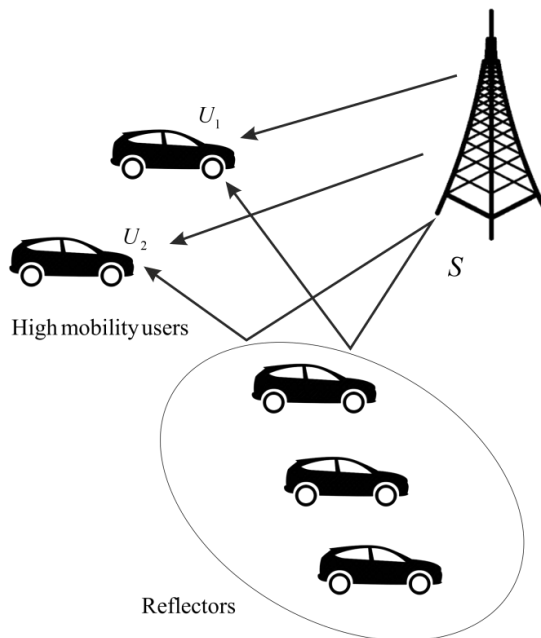


Figure 1 OTFS-based power domain NOMA System Model

### 2.2 Channel Model

In a multi-user mobility communication network where the transmitter communicates with  $K$  users,  $\tau$  delay,  $v$  Doppler shift,  $h_i(\tau, v)$  indicates the channel response for  $1 \leq i \leq K$ , in the DD domain. OTFS facilitates channel estimation and signal detection by using the wireless channel sparsity in the DD domain. As a result, it is considered that there are a few propagation paths between transmitter and receiver. As seen in Equation 3, the channel impulse response is defined in a DD domain as

$$h_i(\tau, v) = \sum_{p=1}^{P_i} h_{i,p} \delta(\tau - \tau_{i,p}) \delta(v - v_{i,p}), \tag{3}$$

where  $P_i$  is the number of propagation paths between the transmitter and user  $i$ ,  $h_{i,p}$  is the independent and uniformly distributed (i.i.d) complex Gaussian channel gain,  $\tau_{i,p}$  is the delay in the propagation path,  $v_{i,p}$  is the Doppler shift in the propagation path and  $\delta$  indicates the Dirac delta function. Each user has a total power of 1, where the channel gain  $h_{i,p} \sim \mathcal{CN}(0, \frac{1}{P_i})$ .  $\frac{1}{M\Delta f}$  and  $\frac{1}{NT}$  are delay and Doppler resolution of OTFS, respectively.

### 2.3 General Principles of OTFS

OTFS general modulation/demodulation block diagram is given as seen in Figure 2.  $x[k, l]$  information bits are transmitted as  $M \times N$  QAM symbols. The inverse symplectic fast Fourier transform (ISFFT) is then applied to convert the DD domain signal  $x[k, l]$  into the TF domain signal  $X[n, m]$ . After applying the Heisenberg transformation to the  $x[n, m]$  matrix, the  $s(t)$  signal is obtained and transmitted to the communication channel. At the receiver, the Wigner transform is first applied to the time domain signal  $r(t)$  to obtain the TF domain signal  $Y[n, m]$ . In the demodulation part, a symplectic fast Fourier transform (SFFT) is employed to obtain the DD domain signal  $y[k, l]$ . Finally, the signals are detected by applying the MP detection method [12]. This algorithm performs detection based on the SIC principle and alternates the decision outputs iteratively [13].

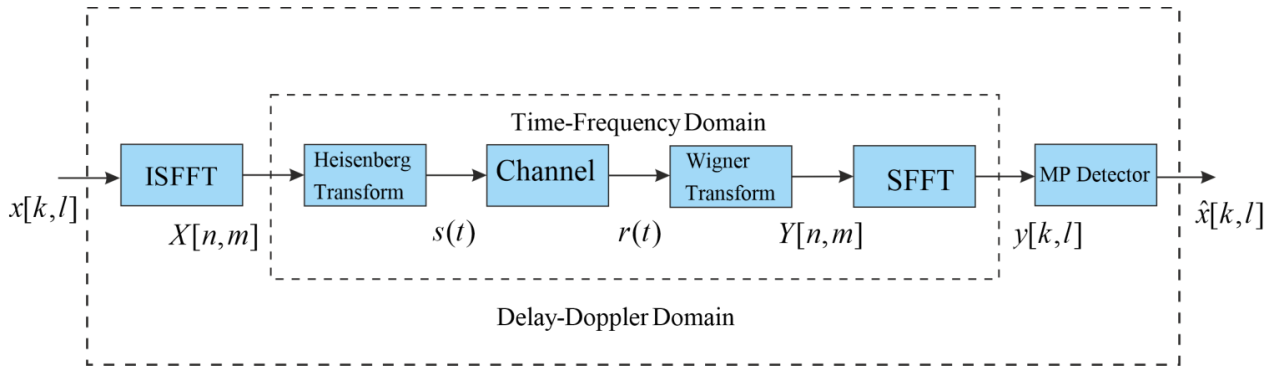


Figure 2 OTFS General Modulation/Demodulation Block Diagram

The symbols  $x_i[k, l]$  in the DD domain for the  $i$ -th mobile user transmitted by the OTFS transmitter can be formulated as the signal  $X_i[n, m]$  in the TF domain as seen in Equation 4

$$X_i[n, m] = \frac{1}{\sqrt{NM}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} x_i[k, l] e^{j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}, \quad (4)$$

where  $x_i[k, l]$  represents the matrix for the  $i$ -th user and  $n = 0, \dots, N - 1, m = 0, \dots, M - 1$  [11]. Then, a continuous time signal is created by applying the Heisenberg transform to the TF signal matrix  $X_i[n, m]$ . The  $i$ -th NOMA user's signal is as seen in Equation 5

$$s_i(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (\sqrt{\xi_i} \alpha_i X_i[n, m] g_{tx}(t - nT) e^{j2\pi m \Delta f (t - nT)}), \quad (5)$$

where  $\xi_i$  is the transmission power of  $i$ -th user,  $T = 1/\Delta f$  is the symbol duration,  $g_{tx}(t)$  is the transmit pulse shaping waveform while  $\alpha_i$  indicates the power allocation coefficient for destination users and  $\sum_{i=0}^K \alpha_i = 1$ . For the far user, high priority is given due to the quality-of-service requirement,  $\alpha_2 > \alpha_1$ . The signal  $s_i(t)$  is transmitted over a channel  $h_i(\tau, \nu)$  with a channel impulse response, resulting in the signal  $r_i(t)$  as seen in Equation 6

$$r_i(t) = \iint h_i(\tau, \nu) s_i(t - \tau) e^{j2\pi \nu (t - \tau)} d\tau d\nu + w_i(t), \quad (6)$$

where  $w_i(t)$  indicates the complex additive white Gaussian noise (AWGN) and  $\sigma_i^2$  is the variance. The received signal  $r_i(t)$  is sampled with period  $T_s$  ( $t = qT_s, q = 0, \dots, NM - 1$ ) and received discrete signal samples are as seen in Equation 7

$$r_i[q] = \sum_{l=0}^{L-1} h_i[q, l] s_i[q - l] + w_i[q], \quad (7)$$

where  $h_i[q, l]$  is the channel impulse response of the  $i$ -th user. Here,  $q$  is the instant time and  $l$  is the delay. The discrete-time signal for the  $i$ -th user is received within a matrix as  $\mathbf{r}_i = \mathbf{H}_i \mathbf{s} + \mathbf{w}_i$  where  $\mathbf{w}_i$  is  $MN \times 1$  complex AWGN vector and  $\mathbf{H}_i$  is the  $MN \times MN$  channel matrix of  $i$ -th user generated from the impulse responses. At the receiver, the following cross-ambiguity function is first calculated in a matched filter as seen in Equation 8

$$Y_i(t, f) = \int g_{rx}^*(t' - t) r_i(t') e^{-j2\pi f (t' - t)} dt', \quad (8)$$

where  $g_{rx}^*(t)$  represents the received waveform. As seen in Equation 9, it is possible to acquire the output of the matched filter by sampling  $Y(t, f)$  as

$$Y_i[n, m] = Y_i(t, f)|_{t=nT, f=m\Delta f}. \quad (9)$$

Equations 8 and 9 denote the Wigner transformation. Then, SFFT is applied to  $Y_i[n, m]$  samples and the symbols  $y_i[k, l]$  are obtained in the DD domain as

$$y_i[k, l] = \frac{1}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} Y_i[n, m] e^{-j2\pi\left(\frac{nk}{N} - \frac{ml}{M}\right)}. \tag{10}$$

Here, we assume the orthogonality between transmitted and received pulses. The received signal is modeled as seen in Equation 11

$$Y_i[n, m] = H_i[n, m]X_i[n, m] + W_i[n, m], \tag{11}$$

where  $W_i[n, m]$  represents AWGN in the TF domain and  $H_i(n, m) = \iint h_i(\tau, \nu) e^{j2\pi\nu nT} e^{-j2\pi(\nu+m\Delta f)\tau} d\tau d\nu$ . Ultimately, the transmitted signal will be recovered as  $\hat{x}[k, l]$  by signal detection and demodulation [14].

### 3. Numerical Results

The BER performance of the power domain downlink OTFS-NOMA simulation results under Rayleigh fading distribution with 4-QAM/16-QAM modulation is presented in this section. The PA coefficient is assumed to be 0.85 for the far user and 0.15 for the near user. The simulation parameters are listed as seen in Table 1. The BER performance is based on the assumption that the destination users move at a maximum speed of 380.7 km/h. Here, the number of propagation paths is taken as  $P_i = 4$  and the DD grid size is  $M = 32, N = 32$ .

Table 1 Simulation Parameters

Parameter	Value
Number of paths ( $P_i$ )	4
Carrier Frequency ( $f_c$ )	4 Ghz
Subcarrier Spacing ( $\Delta f$ )	15 kHz
Modulation Alphabet	4-QAM/ 16-QAM
Max Speed (Kmph)	380.7
Delay-Doppler Grid Size	$M, N = 32, 32$
Symbol duration ( $T = 1/\Delta f$ )	0.0667 ms

The carrier frequency offset is represented as  $f_D = \nu f_c/c$ , where  $f_c$  is the carrier frequency,  $c$  is the speed of the light and  $\nu$  is the speed of the movement between the transceivers [15]. As seen in Table 2, it is considered that the maximum Doppler is set to 1410 Hz, and velocity is set to 380.7 km/h at a carrier frequency of 4 GHz. The maximum multi-path delay is set to 8.4  $\mu s$  [16].

Table 2 Delay and Doppler Models

Path index(p)	1	2	3	4
Delay ( $\tau_i, \mu s$ )	2.1	4.2	6.3	8.4
Doppler ( $\nu_i, Hz$ )	0	470	940	1410

As seen in Figures 3 and 4 demonstrate the BER values derived from the MP detection algorithm based on the signal-to-noise ratio (SNR). Figure 3 demonstrates the BER performance analysis that varies with SNR for both destination users. In this figure, we assume that the modulation type is 4-QAM and OTFS frame size is  $M = 32, N = 32$ . It is observed that the performance of the far user did not drop below a particular BER level as expected. It is assumed that the far user recognizes the interference caused by the near user as noise and is unable to decode. It is seen that the BER value for the  $U_1$  user is 0.1, while the BER value for the  $U_2$  user is nearly 0.25 at 10 dB SNR, resulting in 2.5 times better BER performance. Furthermore, the improvement in BER in  $U_1$  is better than in  $U_2$  as SNR increases.

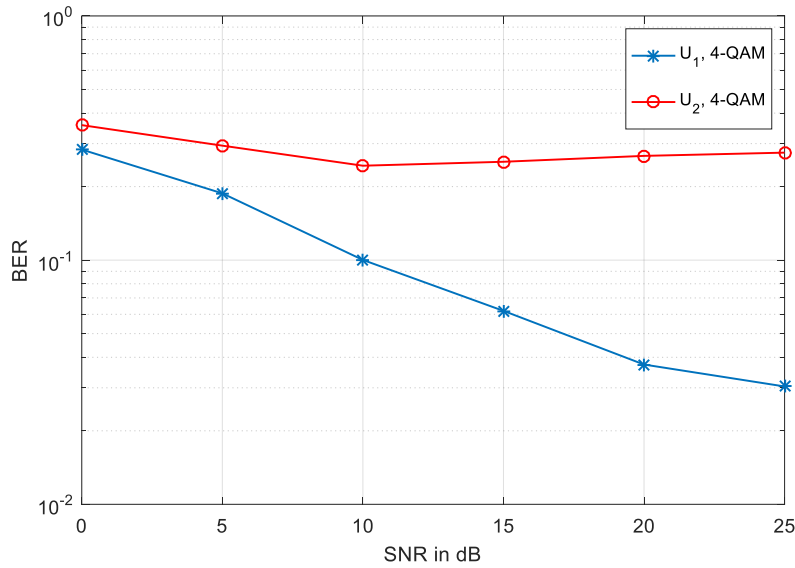


Figure 3 BER Evaluation for the OTFS-NOMA (4-QAM,  $M = 32, N = 32$ )

Figure 4 examines the BER performance for the OTFS-NOMA for both near and far users varying with SNR under 16-QAM modulation and OTFS frame size of  $M = 32, N = 32$ . It is observed that the Doppler effect causes the BER performance of both users to become poorer. It is apparent that the BER value is nearly 0.27 for the  $U_1$  user, while the BER value for the  $U_2$  user is approximately 0.33 at 10 dB SNR, resulting in a better BER performance. Thus, the improvement in BER in  $U_1$  is better than in  $U_2$  as SNR increases. Simulation results with 16-QAM show worse results than the BER performance obtained with 4-QAM. This is a conclusion that can be drawn from both figures. This result has also been observed in a similar study documented in the literature [17].

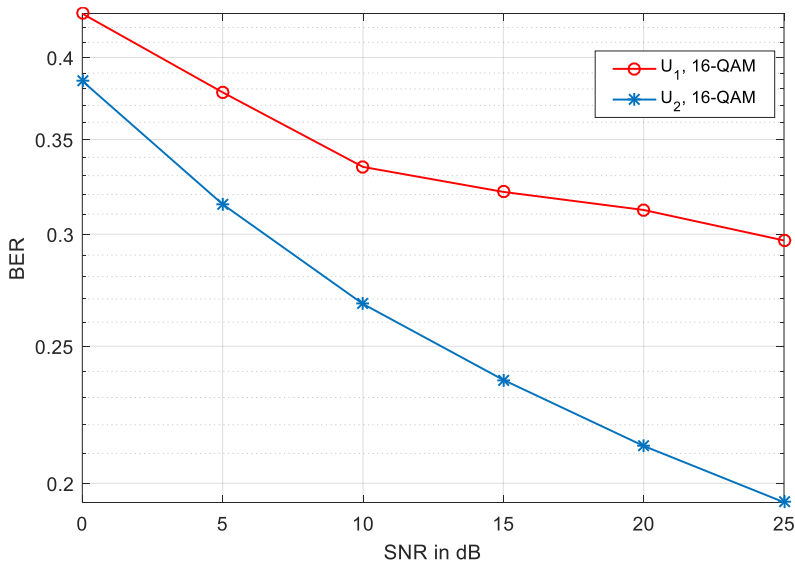


Figure 4 BER Evaluation for the OTFS-NOMA (16-QAM,  $M = 32, N = 32$ )

#### 4. Conclusions

In this work, the BER performance is analyzed for the power-domain downlink OTFS-NOMA system model with high-speed mobility. In the BER simulation results, PA coefficient, delays, and Doppler effects caused by the velocity parameters of the users in motion are taken into consideration. In the proposed OTFS-NOMA system, the message passing detection algorithm is utilized to analyze the BER performance for destination users. It is observed that reflections and delays from users at higher velocities reduce the system performance. It is noted that the far user's performance did not deteriorate below a specific BER threshold. In the 4-QAM situation, the BER performance is 2.5 times higher for the  $U_1$  user (BER value = 0.1) than for the  $U_2$  user (BER value = approximately 0.25 at 10 dB SNR). In the 16-QAM technique,

the BER value for the  $U_1$  user is around 0.27, while the BER value for the  $U_2$  user is roughly 0.33 at 10 dB SNR. It is observed that both users' BER performance degrades due to the Doppler effect.

Although we investigated the BER performance for OTFS modulated NOMA system for two destination users in the high mobility environment, it is possible to extend the provided analysis to more generalized results for the enabling technologies in 6G wireless networks, such as unmanned aerial vehicles (UAVs) [18-21], reflecting intelligent surfaces (IRS), non-terrestrial networks, and integrated sensing and communications (ISAC). Integrating these enabling technologies for OTFS-based NOMA will be a promising way to enhance the functionality of our future study. Besides, cognitive radio [22-23] architecture will be a different solution scenario. Secondary user signal detection algorithms to increase the spectral efficiency [24] for high-speed mobile environments may be another topic of our study. In addition, reliability, secrecy performance analysis, and channel estimation schemes are planned to work on OTFS modulated NOMA systems.

## References

- [1] I. Umakoglu, M. Namdar, A. Basgumus, F. Kara, H. Kaya, and H. Yanikomeroğlu, "BER Performance Comparison of AF and DF Assisted Relay Selection Schemes in Cooperative NOMA Systems," *IEEE 9th International Black Sea Conference on Comm. and Networking*, Bucharest, Romania, 2021.
- [2] Z. Ding, M. Peng, and H.V. Poor, "Cooperative non-orthogonal multiple access in 5G systems," *IEEE Comm. Lett.*, vol. 19, no. 8, pp. 1462–1465, 2015.
- [3] Y. Saito, A. Benjebbour, Y. Kishiyama, and T. Nakamura, "System level performance evaluation of downlink non-orthogonal multiple access (NOMA)," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Comm.*, London, U.K., Sep. 2013, pp. 611–615.
- [4] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the performance of nonorthogonalmultiple access in 5G systems with randomly deployed users," *IEEE Signal Process. Lett.*, vol. 21, no. 12, pp. 1501–1505, Dec. 2014.
- [5] S. McWade, M. F. Flanagan, A. Farhang, "Low-Complexity Equalization and Detection for OTFS-NOMA", arXiv preprint arXiv:2211.07388, 2022.
- [6] G. D. Surabhi, R. M. Augustine, A. Chockalingam, "Multiple access in the delay-Doppler domain using OTFS modulation," arXiv preprint, 2019.
- [7] Z. Ding, R. Schober, P. Fan, and H. Vincent Poor, "OTFS-NOMA: An Efficient Approach for Exploiting Heterogenous User Mobility Profiles," *IEEE Transactions on Comm.*, vol. 67, no. 11, pp. 7950–7965, 2019.
- [8] A. Chatterjee, V. Rangamgari, S. Tiwari, and S. S. Das, "Nonorthogonal Multiple Access With Orthogonal Time–Frequency Space Signal Transmission," *IEEE Systems Journal*, vol. 15, no. 1, pp. 383–394, 2021.
- [9] K. Deka, A. Thomas, and S. Sharma, "OTFS-SCMA: A Code-Domain NOMA Approach for Orthogonal Time Frequency Space Modulation," *IEEE Transactions on Comm.*, vol. 69, no. 8, pp. 5043–5058, 2021.
- [10] H. Wen, W. Yuan, and S. Li, "Downlink OTFS Non-Orthogonal Multiple Access Receiver Design based on Cross-Domain Detection," in *IEEE International Conference on Comm. Workshops*, 2022, pp. 928–933.
- [11] P. Raviteja, K. T. Phan, Y. Hong and E. Viterbo, "Interference Cancellation and Iterative Detection for Orthogonal Time Frequency Space Modulation," in *IEEE Transactions on Wireless Comm.*, vol. 17, no. 10, pp. 6501-6515, 2018, doi: 10.1109/TWC.2018.2860011.
- [12] L. Xiao, S. Li, Y. Qian, D. Chen and T. Jiang, "An Overview of OTFS for Internet of Things: Concepts, Benefits, and Challenges," in *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7596-7618, 2022, doi: 10.1109/JIOT.2021.3132606.
- [13] H. Zhang, K. Niu, J. Xu, J. Dai and J. Zhang, "Iterative SIC-Based Multiuser Detection for Uplink Heterogeneous NOMA System," *2022 IEEE Globecom Workshops*, Rio de Janeiro, Brazil, 2022, pp. 94-99.
- [14] I. Umakoglu, M. Namdar, A. Basgumus, S. Özyurt and S. Kulaç, "BER Performance Analysis for NOMA Systems with OTFS Modulation," *2023 31st Signal Process. and Comm. Appl. Conference*, Istanbul, Turkiye, 2023, pp. 1-4.
- [15] Y. Zhang, S. Zhang, B. Wang, Y. Liu, W. Bai and X. Shen, "Deep Learning-Based Signal Detection for Underwater Acoustic OTFS Communication," *Journal of Marine Science and Engineering*, vol. 10, no. 12, 2022.
- [16] T. Thaj and E. Viterbo, "Low-Complexity Linear Diversity-Combining Detector for MIMO-OTFS," in *IEEE Wireless Comm. Lett.*, vol. 11, no. 2, pp. 288-292, Feb. 2022, doi: 10.1109/LWC.2021.3125986.
- [17] K. Yadav, P. Singh, H.B. Mishra, and R. Budhiraja, "Closed Form BER For ZF OTFS Receivers," *IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications*, 2021.
- [18] I. Umakoglu, M. Namdar, and A. Basgumus, "UAV-Assisted Cooperative NOMA System with the nth Best Relay Selection," *Advances in Electrical and Computer Engineering*, vol. 23, no. 3, pp. 39-46, 2023.
- [19] T. Yılmaz, A.A. Bacanlı, and H. İlhan, "UAV-Assisted NOMA-Based Network with Alamouti Space-Time Block Coding," *Politeknik Dergisi*, vol. 25 no. 3, pp. 967-973, 2022.
- [20] S. Koşu, and S.Ö. Ata, "NOMA-enabled Cooperative V2V Communications with Fixed-Gain AF Relaying," *Balkan Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 1-12, 2023.



- [21] A. Basgumus, F. Kocak, and M. Namdar, "BER performance analysis for downlink NOMA systems over cascaded Nakagami-m fading channels," *Annals of Telecommunications*, 1-7, 2023.
- [22] F.K. Bardak, M. Namdar, and A. Basgumus, "Ergodic Capacity Analysis of the Relay Assisted Downlink NOMA Systems in Cognitive Radio Networks," *Journal of Engineering Sciences and Design*, vol. 9, no. 3, pp. 992-1002, 2021.
- [23] M. Namdar, A. Guney, F.K. Bardak, and A. Basgumus, "Ergodic Capacity Estimation with Artificial Neural Networks in NOMA-Based Cognitive Radio Systems" *Arabian Journal for Science and Engineering*, 1-10, 2023.
- [24] A. Basgumus, M. S. Ardic, and M. Namdar, "Capacity Analysis of the Secondary Users in Spectrum Sharing Model over Nakagami-m and log-normal Fading Channels," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 38, no. 4, pp. 2205-2212, 2023.

**Conflict of Interest Notice**

The authors declare that there is no conflict of interest regarding the publication of this paper.

**Ethical Approval and Informed Consent**

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

**Availability of data and material**

Not applicable.

**Plagiarism Statement**

This article has been scanned by iThenticate™.