IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 2, April, 2023, pp. 442-450
Journal homepage: http://iieta.org/journals/mmep

Securing E-Governance Services Based on Two Level Classification Algorithms

Ibrahim Khaleel Ibrahim*, Shaimaa Ali Elmorsy, Nermeen Mahmoud Kashef, Mahmoud Mohamed Mustafa Al-Borai

Department of Mathematics and Computer Science/Faculty of Science, Alexandria University, Alexandria 21544, Egypt

Corresponding Author Email: Ibrahim.khaleel_PG@alexu.edu.eg

https://doi.org/10.18280/mmep.100208

Received: 30 November 2022
Accepted: 13 March 2023

Keywords:
E-governance services, classification algorithms, SVM, attack types, multi-class

ABSTRACT

Due to the expansion of cybercrime and cyberwarfare, the necessity for cyber security has recently expanded substantially. There are several trends in cyber security, but the most important is e-governance. E-government is regarded as one of the most essential platforms for transmitting data and services over the internet that frequently contain valuable and confidential data, making them subject to threats. The majority of e-governance systems rely on public-use services. The paper proposed a framework to detect threats in internet traffic flows. This paper uses a famous dataset that was collected from internet traffic called the UNSW-NB15 dataset, which consists of 307,099 instances. The framework consists of several steps, including pre-processing, identifying a correlation between features, and selecting the best ones. Finally, different machine learning algorithms are used to distinguish the normal traffic from the malware traffic. The findings uncover that SVM achieved very high accuracy (99.16%). Additionally, in the second part, which is called multi-class and consists of two stages, in the first one the study classified the abnormal flows into nine attacks with a lower accuracy of 77.80%. In the second stage with binary classification, the dataset contained both normal and abnormal, and the accuracy improved significantly to 97.48% for SVM.

1. INTRODUCTION

The ICT boosts the e- governance according to the authors Kim and Layne as the public expects more services to be delivered online with high efficiency [1]. After Internet 's innovation, the administration for the public by the governments is changed from the traditional services that require a person to attend by himself into a digital service that serve a person remotely. The people can access and understand the services provided by a government in a comfort in their own homes. E-governance refers to using the information and communication technology (ICT) to provide services of the governments, information exchange, and build a strong relationship between governments and customer (G2C), government and business (G2B), government and government (G2G) [2]. Citizens can access government services in a convenient and an efficient way by applying e-governance in which it brings governments and citizens closer together. As mentioned in the abstract section, it will be rather easy to follow these rules as long as you just replace the "content" here without modifying the "form".

In 2017, a ransomware programme encrypted databases and information for many sectors, such as banking, electricity, and medical care, forcing these companies to pay more than $8 billion just to decrypt their information and take control of their systems [3]. Firewalls and encryption are the most traditional methods that have been used to manipulate these attacks and hackers. However, a system that is based on machine learning algorithms (MLs) is the key to facing such risks effectively [4].

The creation of security models based on machine learning that analyze numerous cyberattacks or anomalies and ultimately detect or forecast threats can be advantageous for intelligent security services [5]. Detection models are typically used to handle various cyber-attacks, referred to as a multi-class model, or to detect abnormalities, referred to as a binary-class model. Various models for machine learning have been considered to detect and prevent anomalies due to the variation in security features that could be large and include known and unknown attacks. This work presents a practical study that shows different models and their efficiency in dealing with security issues [6]. Two primary models of classification are proposed in this study: the binary classification model and the multi classification model. The former model classifies traffic flows as normal or not, and this is a first step in our work. The latter model, which is a second step, detects threat types (i.e., DoS, botnets, or worms) and classifies them into nine different categories. Various machine learning methods are applied for this purpose, such as support vector machines (SVM), K-nearest neighbours (KNN), random forests (RF), nave Bayes (NB), adaptive boosting, and decision trees. Initial steps should be carried out before applying machine learning methods, including pre-processing and feature selection to remove irrelevant information and increase the accuracy of models. The UNSW-NB15 dataset is used to evaluate these models [7].

The authors of this paper noticed from the literature review that the studies carried out a binary classification or multi-classification. This study classified traffic into two approaches: binary classification and multi-class classification. This motivation of the current study. Therefore, this paper focuses on utilizing machine learning algorithms to

secure E-governance services via the abilities of automated learning according to databases. The contribution of this study is to propose and examine mechanisms to define normal and abnormal behavior within internet traffic. In addition, it investigates the abnormality to classify and identify the types of attacks. The studies are carried out by employing machine learning techniques with a collection of characteristics as input to generate classifiers that will be checked and evaluated based on the UNSW-NB15 dataset.

The results of the first stage revealed that the SVM classifier outperformed the other classifiers, with an accuracy of 99.16%. The second stage performed the accuracy checks for classifying the malware traffic into nine attacks: Fuzzers, Analysis, Backdoors, Exploits, Generic, Reconnaissance, Shellcode, Worms, and DoS. The results showed that deleting the normal cases from the dataset greatly affected the accuracy. Furthermore, when compared to typical traffic, the remaining instances for nine attacks were lower. When the classification procedure was applied to all traffic using the same techniques in the third step, the accuracy of the four algorithms improved, and SVM obtained a higher accuracy of about 97.48 percent.

2. BACKGROUND

Different studies have been published in the area of e-governance; to date, 33 articles have been reviewed by Muzafar et al. [8]. The author aimed to identify emerging cybersecurity vulnerabilities in the context of e-governance and to use machine learning techniques to achieve precise results. Similarly, Shah [9] addressed numerous critical difficulties and challenges confronting e-government development as well as various departments that supply e-services. Practically, Shareef [10] examined and analyzed the security threats in the e-governance system, as well as the important elements that may aid in reducing these threats to the information security of such a system. The author claimed that policies and procedures for security needed to be installed and configured regularly to ensure robust e-governance systems. The study also concluded that a suitable public-key infrastructure is substantial to provide authentication and integrity for the e-governance system. Sharma et al. [11] believed that such a new environment of e-governance needs new legislation and rules that contain electronic signatures, data matching, archiving, data protection, internet crime, and intellectual property rights. The authors emphasized the importance of governments enacting strong legislation to reduce malicious activity and inappropriate use of e-governance applications. According to Froehlich et al. [12], the research community's investigations have revealed that cybercrime laws and regulations are critical for securing e-governance systems.

Technology behemoths like Microsoft, ESET, and the NSA, as well as law enforcement agencies like Interpol and the FBI, are making significant efforts to combat security breaches, and as a result, the number of malicious activities may decrease. Employers, such as network service providers (ISPs), large corporations, and users, must also play a role in enhancing the security of cyberspace within a country [13]. Therefore, Ahmet and KAZDAL [14] discussed the evolution of security and its tools in light of the increasing risks and threats. The authors presented the most recent trends in information security that emerged and were innovated by researchers to address internal and external threats. To ensure appropriate levels of security for electronic public services provided through e-government applications and government agencies, it is important to design and implement special security measures such as firewalls, encryption, and intrusion detection and prevention software. Despite the modern technologies employed by governments and specialists, virus processes and hackers must be taken into account. The study also presented machine learning algorithms (MLs) that have become a weapon that helps both cyber defenders and hackers execute repetitive and intensive tasks. The defenders use MLs for threat identification and suspicious activity tracking, while the hackers exploit this technology to search for vulnerabilities in the network to attack. Employees who have regular access to e-government systems should obtain an education in cyber security and make it a part of their job.

A study from Columbia University's Department of Computer Science which done by Bowen et al. [15] demonstrated how the human aspect influences cyber security policy and how this knowledge can help government employees enhance the security of e-governance. Studies, on the other hand, proposed analyzing web traffic to identify and classify network attacks. For instance, Kachavimath et al. [16] proposed a detection strategy for DDoS attacks that uses machine learning to improve enterprise network security. The machine-learning system collects high-level data from network traffic by selecting the best features that are able to identify attacks. Similarly, Kondeti et al. [17] explored the various states of India's financial status utilizing SVM, Naive Bayes, classification, and regression methods. The confusion matrix is created to estimate performance. Therefore, smart e-government environment-based structures are being built on the foundation of AI execution. The cost and processing time are reduced, while citizen satisfaction is increased [18].

Another group of studies has used deep learning approaches to detect cyber-attacks. For example, Gaur et al. [19] employed this technique to classify different threats in different network areas. In this study, deep learning was employed to effectively manage a variety of cyber security issues, including intrusion detection, malware or botnet identification, phishing, forecasting of cyberattacks, denial of service (DoS), fraud detection, and cyber abnormalities. Because deep learning is more exact, especially when learning from huge security datasets, it has an advantage for building security models [20]. Similarly, Wang and Wang [21] created a set of deep learning models with the goal of automating many e-government services. Afterwards, the study offers a smart e-government platform architecture that facilitates the development and implementation of e-government AI applications. However, such a technique requires additional resources and needs more computational processing, in addition to needing a huge dataset to implement such a system. Therefore, this study utilized traditional machine learning algorithms to build a cyber-attack in an e-governance environment, as these algorithms take less computational time.

The purpose of this research is to identify and analyse common gaps in cyber security. The author has determined the main security flaws and their rate of occurrence after thoroughly analyzing the selected research. Studies and syntheses have been conducted using major targeted organizations, apps, and publishing data that are readily available. This study's findings demonstrated that security measures typically only target security and emphasised the importance of the solutions offered in these studies for further

experimental authentication and practical application. The goal of this study is to identify new cybersecurity dangers to e-governance in the modern period, with a focus on applying machine learning approaches for precise findings.

Table 1 summarizes previous research on various machine learning algorithms for detecting and classifying threats; the current work, which classified traffic into two approaches: binary classification and multi-class classification, is presented in the last Table.

Table 1. Comparison of machine learning models for detecting malware and various attacks

Aim	Used method	Type	Reference
Classifying Attacks classification for building intrusion detection system	Feature selection and SVM	Multiclass	Gauthama Raman et al. [22]
Anomaly detection	Feature selection and Ada Boost	Multiclass	Mazini et al. [23]
Classifying anomaly and normal traffic	Feature selection and Decision tree	Binary	Sarker et al. [24]
Creating a method for early detection to improve prevention	Naïve Bayes, logistic model tree, the probabilistic neural network, J48 (C4.5), the classification and regression tree, JRip, and the gradient boosting machine	Binary	Kondeti et al. [17]
Detecting attacks in Smart City	RF learning	Binary	Alrashdi et al. [25]
Detecting threats in IoT system	SVM, LR, RF, DT and ANN	Multiclass	Hasan et al. [26]
Classifying DDoS attacks	K-NN and Naïve Byes	Multiclass	Kachavimath et al. [16]
To detect malware traffic and classifying various attacks	Random Forest, SVM, K-NN, and Naïve Byes	Binary and multiclass	Present work

3. PROPOSED METHODS

The proposed model utilized machine learning algorithms to distinguish normal traffic from malware and classify the latter into different types of attacks. This model is suggested to secure E-governance system that can detect and identify attacks based on utilizing machine learning algorithms. Highlighting the key components of the proposed scheme with a description of principal stages as follows:

(1) Exploring the dataset: The proposed model is evaluated based on the UNSW-NB15 dataset [27], which contains more than 307099 instances.

(2) Pre-processing the dataset: this stage included various steps in order to prepare the dataset for classification by machine learning.

(3) Features selection: this stage involved minimizing features and selecting the best ones to avoid any redundant ones in the dataset.

(4) Machine learning/Classification: This is the final stage of the proposed work that contains the usage of classification algorithms from machine learning to detect and classify the attacks.

Figure 1 describes the block diagram of the model.

3.1 Dataset exploring

The dataset contains a number of features that describe the state of the Internet network at a given time [27]. For example, the IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra synthesized the raw network packets for the UNSW-NB 15 dataset in order to provide a mix of genuine contemporary normal activities and synthetic current attack behaviors. 100 GB of the raw traffic were captured using the tcpdump utility (Pcap files). Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine attack categories in this dataset. To produce a total of 49 characteristics with the class label, the Argus and Bro-IDS tools are utilized, and twelve methods are built. These features can be used to build a machine learning model to simulate real network traffic. Understanding this dataset and its features is critical for detecting attacks or anomalies. Our model is based on using the UNSW-NB15 dataset [7] to analyze and evaluate the proposed method. The UNSW-NB15 dataset consists of 307099 instances with 9 types of attacks, which are described in Table 2. The table shows each type of attack with the associated instances. There are 47 features included in the dataset that are utilized in the proposed system and are described in Table 3.

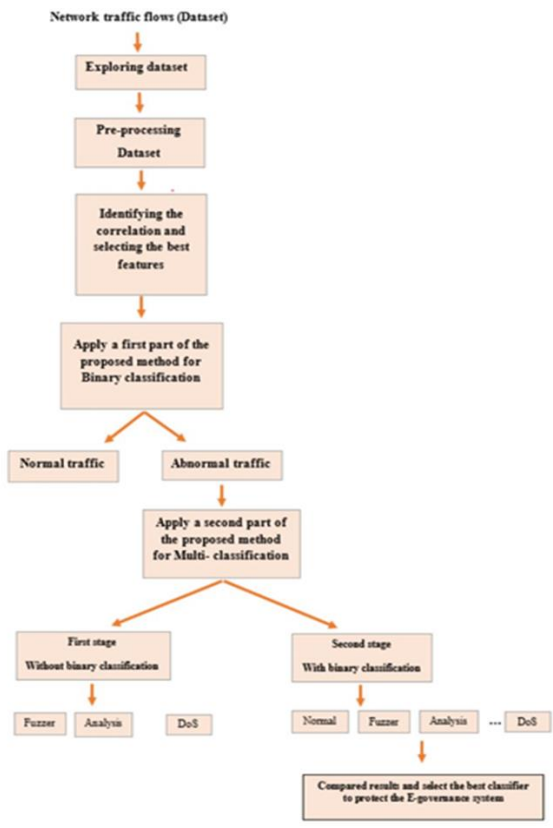

Figure 1. Proposed method of the current study

Table 2. Attack types and their instances

Symbol	Attack type	Instances
0	Normal	56693
1	Fuzzers	5051
2	Analysis	526
3	Backdoors	534
4	Exploits	5409
5	Generic	7522
6	Reconnaissance	1759
7	Shellcode	223
8	Worms	24
9	DoS	1167

Table 3. Features description for the dataset

No.	Name	Details
1	Source IP	The address of the source IP
2	Source port	The number of the source port
3	Destination IP	The address of the Destination IP
4	Destination port	The number of destination port
5	Protocol	The protocol that determines the transaction mechanism
6	State	Indicates to the state and its dependent protocol, e.g., ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	Duration	The total record duration
8	Source bytes	Bytes transferred from Source to Destination in total
9	Destination bytes	The total number of bytes sent from source to destination
10	Source time	from source to destination, time to live value
11	Destination time	Time to live value from source to destination
12	Source loss	Packets loss for the source
13	Destination loss	Packets loss for the destination
14	service	Type of the service such as http, ftp, and smtp
15	Source load	Number of bits per second for the source
16	Destination load	Number of bits per second for the destination
17	Source packets	Number of packet count from Source to destination
18	Destination packets	Number of packet count from destination to source
19	Source Ad	Advertisement value for the source
20	Destination Ad	Advertisement value for the destination
21	Source sequence	The sequence number for the source
22	Destination sequence	The sequence number for the destination
23	Source mean	The mean of packet size for the source
24	Source mean	The mean of packet size for the destination
25	Transaction depth	The pipelined depth of an http request/response transaction's connection.
26	res_bdy_len	Actual uncompressed data size transferred from the server's http service.
27	Source jitter	Source jitter in the (mSec)
28	Destination jitter	Destination jitter in the (mSec)
29	Start time	The record of the start time
30	Last time	The record of the last time
31	Source inter-arrival time	Source of the inter-arrival packet time in the (mSec)
32	Destination inter-arrival time	Destination of the inter-arrival packet time in the (mSec)
33	Tcp round trip time	The sum of TCP connection setup round-trip time
34	Syn ack	TCP connection establishment time, the interval between SYN and SYN ACK packets
35	Ack dat	the interval between the SYN ACK and ACK packets, and the TCP connection setup time
36	IPs-ports	If the port numbers (2), (4), source (1), and destination (3) are all equal, this variable has the value 1. If not, it uses the value 0.
37	state_ttl	Based on a given range of source/destination time to live (10) there are six numbers for each state.
38	flw_http_mthd	The quantity of http service flows that contain methods like Get and Post.
39	ftp_login	If the ftp session is accessed using the user and password, then 1; else, 0.
40	ftp_cmd	The number of flows in an ftp session that have a command
41	srv_src	In 100 connections, the last time there were 14 connections with the same service and 1 connection with the same source address (26)
42	srv_dst	In 100 connections, there are now 14 connections with the same service and 3 connections with the same destination address (26)
43	dst_ltm	In 100 connections, there were three connections with the same destination address, based on the most recent data (26)
44	src_ ltm	Based on the most recent time, the proportion of connections with the same source address (one) in 100 connections.
45	src_dport_ltm	Number of connections in 100 connections based on the most recent time that share the same source address (1) and destination port (4) (26).

46	dst_sport_ltm	In the last 100 connections, there have been three connections with the same destination address and source port (26).
47	dst_src_ltm	According to the most recent time, there are 100 connections with the same source (1) and destination (3) addresses (26).
48	attack_cat	The various assault types' names. This data set has nine categories, including Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode, and Worms.
49	Label	0 for normal and 1 for attack records

3.2 Data pre-processing

This stage includes three additional steps: feature encoding, feature scaling, and finally splitting the dataset into training and testing to prepare for applying machine learning algorithms.

3.2.1 Features encoding

The dataset contains different types of variables, such as integer, nominal, and timestamp. The nominal feature type must be converted into vectors in order to be processed by machine learning models. Table 2 depicts these characteristics with the numbers 1, 3, 5, 6, and 14. Label encoding was utilized for this purpose, which converts the value of the features into a number type that can be used by the classification model.

3.2.2 Features scaling

For many machine learning algorithms, the preprocessing step of feature scaling through standardization, also known as "Z-score normalization," might be crucial. Rescaling the features to give them the characteristics of a typical normal distribution with a mean of zero and a standard deviation of one is known as standardization. When machine learning algorithms measure distances between data points, the magnitude (scale) of the features may dominate the results rather than their values. This issue can be resolved by scaling the features to a fixed range.

3.2.3 Dataset splitting

The dataset was split into two parts: The first part was used for training the models and contained 80% of instances. The second part, which contained 20% of the instances, was used to test the model.

3.3 Feature selection

This technique is used to minimize the computational processing time and increase the performance of the proposed system. The method is based on removing features that are irrelevant or repetitive. In this work, the Boruta selection scheme was utilized to reduce the features from 47 to 43, which are the inputs to the models of machine learning. The Boruta algorithm for feature selection strategy works as follows:

- The algorithm adds unpredictability to a dataset by generating shuffled copies of the features known as "shadow features."
- The next step is to train a random forest classifier on the expanded dataset and compute the mean decrease accuracy.
- Compared to other features, those with higher means are more significant for the study.
- Finally, the algorithm stops when the required number of random forest iterations has been reached, or when all features have been either accepted or rejected.

3.4 Machine learning algorithms

Different machine learning methods are used in this work to identify the attacks and classify them into various categories. Random forest classifier [28], support vector machine (SVM) [29], K-nearest neighbour (KNN) [30], and Naive Bayes (NB) [31] are examples of these methods. Firstly, the selected model identified whether the traffic flow contains attacks or not; in other words, it specifies if the traffic is normal or not. Secondly, the model tries to classify the abnormal traffic into nine attacks, which are explained in Table 2. Thirdly, the model classifies all traffic (i.e., normal and abnormal) into ten classes to increase accuracy.

4. RESULTS AND DISCUSSION

The results of the first stage computed the accuracy for four machine learning methods, as shown in Table 4. Afterwards, precision, recall, and f1-score measurement metrics are calculated for each method as shown in Table 5. The SVM classifier outperformed the other classifiers in terms of accuracy, with a score of 99.16%.

In the second stage, the malware traffic is classified into nine attacks using the four machine learning algorithms, and the outcomes are shown in Table 6. The findings showed that the accuracy reduced significantly after removing the normal instances from the dataset. In addition, the remaining instances for nine attacks are fewer compared with normal traffic. In addition, precision, recall, and f1-score are also calculated for each method as shown in Table 7.

In the third stage, the classification process is applied to all traffic using the same methods, and the results revealed improved accuracy in the four algorithms as presented in Table 8. In addition, precision, recall, and f1-score measurement metrics are calculated for each method as shown in Table 9. According to the findings, the SVM classifier had a higher accuracy of 97.48 percent.

The accuracy is just the proportion of correctly classified instances to all instances. Tables 10 and 11 display the confusion matrices to describe the performance of the four classifiers for each class for further examination across all attacks. The row displays instances from the predicted class, whereas the column displays instances from the actual class. The matrix's diagonal reflects the number of samples successfully classified as an interest class and referred to as "true positives" (TP). The remaining values in each application's row are misclassified False Positives (FP), while the remaining values in each application's column are misclassified False Negatives (FN). The overall performance of the classifiers is very high for recognizing normal traffic from attacks, and the model detects some attacks (i.e., fuzzers, exploits, generics, and reconnaissance) efficiently. However, the attack analysis, backdoors, shellcode, worms, and DoS were not detected in any of the tested samples.

Table 4. Comparison of accuracy results for binary classification

Random forest classier	SVM	K-NN	Naive Bayes
98.81%	99.16%	99.12%	97.91%

Table 5. Four machine leaning algorithms for other metrics

Algorithm	RF		SVM		K-NN		Naïve Bayes	
Class	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
Precision	100%	86%	100%	92%	100%	94%	100%	78%
Recall	99%	100%	99%	97%	100%	94%	98%	100%
F1-score	99%	93%	100%	94%	100%	94%	99%	87%
Support	56928	4492	56928	4492	56928	4492	56928	4492

Table 6. The results of accuracies for abnormal traffic

Random forest classier	SVM	K-NN	Naive Bayes
77.80%	76.32%	74.16%	48.39%

Table 7. The results of precision, recall, and f1-score

	RF classifier			SVM classifier			
Class	**Precision**	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**	**Support**
1	60%	99%	74%	65%	87%	74%	1027
2	0%	0%	0%	67%	2%	3%	112
3	0%	0%	0%	0%	0%	0%	96
4	77%	75%	76%	68%	78%	72%	1080
5	100%	92%	96%	100%	93%	96%	1508
6	80%	71%	75%	62%	71%	66%	374
7	0%	0%	0%	0%	0%	0%	43
8	0%	0%	0%	0%	0%	0%	2
9	11%	0%	1%	38%	6%	10%	250
	K-NN classifier			Naive Bayes classifier			
Class	Precision	Recall	f1-score	Precision	Recall	f1-score	Support
1	64%	86%	74%	74%	26%	39%	1027
2	21%	8%	11%	16%	97%	27%	112
3	0%	0%	0%	0%	0%	0%	96
4	69%	71%	70%	84%	56%	67%	1080
5	98%	94%	96%	100%	70%	83%	1508
6	64%	60%	62%	16%	15%	16%	374
7	0%	0%	0%	4%	97%	8%	43
8	0%	0%	0%	2%	29%	3%	2
9	21%	7%	11%	30%	3%	6%	250

Table 8. Comparison of accuracy results for multi-class classification

Random forest classier	SVM	K-NN	Naive Bayes
97.32%	97.48%	97.21%	94.18%

Table 9. Precision, recall and f1-score for RF, SVM, K-NN, and Naïve Bayes classifiers

	RF classifier			SVM classifier			
Class	**Precision**	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**	**Support**
0	99%	100%	99%	100%	99%	99%	56928
1	46%	68%	55%	46%	78%	58%	1027
2	0%	0%	0%	0%	0%	0%	112
3	0%	0%	0%	0%	0%	0%	96
4	71%	71%	71%	75%	69%	72%	1080
5	100%	92%	96%	96%	91%	94%	1508
6	83%	63%	72%	63%	75%	69%	374
7	0%	0%	0%	33%	2%	4%	43
8	0%	0%	0%	0%	0%	0%	2
9	55%	2%	5%	67%	4%	8%	250
	K-NN classifier			Naive Bayes classifier			
Class	Precision	Recall	f1-score	Precision	Recall	f1-score	Support
0	99%	100%	99%	100%	92%	96%	56928
1	47%	61%	53%	48%	28%	35%	1027
2	3%	2%	2%	3%	93%	6%	112
3	0%	0%	0%	0%	1%	0%	96

4	67%	67%	67%	36%	53%	43%	1080
5	98%	92%	95%	0%	0%	0%	1508
6	63%	55%	59%	1%	7%	8%	374
7	30%	7%	11%	3%	100%	6%	43
8	0%	0%	0%	0%	0%	0%	2
9	2.5%	5%	9%	1%	2%	1%	250

Table 10. The confusion matrices for random forest and SVM classifiers

Attack type	Normal	Fuzzers	Analysis	Backdoors	Exploits	Generic	Reconnaissance	Shellcode	Worms	DoS
Random Forest Classifier										
Normal	56720	183	0	0	22	0	3	0	0	0
Fuzzers	312	668	0	0	42	0	5	0	0	0
Analysis	11	101	0	0	0	0	0	0	0	0
Backdoors	2	91	0	0	3	0	0	0	0	0
Exploits	58	241	0	0	774	0	5	0	0	2
Generic	10	16	0	0	90	1386	5	0	0	1
Reconnaissance	73	20	0	0	56	0	225	0	0	0
Shellcode	6	13	0	0	11	0	13	0	0	0
Worms	0	1	0	0	1	0	0	0	0	0
DoS	28	117	0	0	100	0	4	0	0	1
SVM Classifier										
Normal	56666	224	0	0	13	0	25	0	0	0
Fuzzers	205	769	0	0	28	0	24	0	0	1
Analysis	10	95	0	0	7	0	0	0	0	0
Backdoors	0	81	0	0	10	0	4	0	0	1
Exploits	28	237	0	0	766	0	41	1	0	7
Generic	7	19	0	0	79	1386	15	0	0	2
Reconnaissance	21	36	0	0	44	3	270	0	0	0
Shellcode	4	9	0	0	2	0	27	1	0	0
Worms	0	1	0	0	1	0	0	0	0	0
DoS	7	109	0	0	101	0	17	1	0	15

Table 11. The confusion matrices for K-NN and naïve Bayes classifiers

Attack type	Normal	Fuzzers	Analysis	Backdoors	Exploits	Generic	Reconnaissance	Shellcode	Worms	DoS
K-NN classifier										
Normal	56720	1	0	10	2	5	0	0	0	0
Fuzzers	259	625	15	27	69	5	11	1	0	15
Analysis	10	52	1	12	28	0	0	0	0	9
Backdoors	1	57	1	0	31	0	2	0	0	4
Exploits	62	165	9	24	734	9	38	3	0	36
Generic	12	16	0	2	59	1407	11	0	0	1
Reconnaissance	43	56	0	3	62	0	204	2	0	4
Shellcode	6	7	0	0	2	0	24	3	0	1
Worms	1	0	0	0	0	0	0	0	0	1
DoS	17	74	8	13	102	3	19	0	0	14
Naive Bayes classifier										
Normal	55465	257	74	32	227	0	85	431	21	336
Fuzzers	0	292	201	23	31	1	104	334	41	0
Analysis	0	0	105	0	7	0	0	0	0	0
Backdoors	0	3	86	1	0	0	0	6	0	0
Exploits	0	50	218	10	577	0	46	104	71	4
Generic	1	16	2	31	44	1325	31	35	17	6
Reconnaissance	0	9	1	0	0	0	34	318	12	0
Shellcode	0	0	0	0	0	0	0	43	0	0
Worms	0	0	0	0	0	0	1	1	0	0
DoS		19	110	7	66	0	5	24	14	5

5. CONCLUSIONS

This paper presented a block diagram for the proposed system using a set of features for attack identification based on the UNSW-NB15 dataset. The dataset is prepared first by applying various steps such as feature encoding and scaling, and later these features are filtered by deploying the Boruta selection scheme to select the most important ones. The last stage utilises four machine learning algorithms that contain two parts. The first part detects the malware traffic from the normal traffic, which is a binary classification. This part specifies only the existing attacks in the network traffic and applies a policy to a server that contains the e-governance services. The second part includes more advanced steps to identify the attack types and classifies them into nine groups (i.e., fuzzers, analysis, backdoors, exploits, generics, reconnaissance, shellcode, worms, and DoS). In addition, all traffic that contains normal and abnormal content is classified

into ten classes overall to be suitable for real time detection. Therefore, the literature study revealed to the writers of this paper that studies either used binary classification or multiple classifications. In this study, there were two methods used to categorize traffic: binary classification and multi-class classification.

REFERENCES

[1] Babaoglu, C., Akilli, H.S., Demircioglu, M.A. (2012). E-government education at the public administration departments in Turkey. In Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance, pp. 71-74. https://doi.org/10.1145/2463728.2463745

[2] Moustafa, N., Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, pp. 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[3] Qu, X., Yang, L., Guo, K., Ma, L., Sun, M., Ke, M., Li, M. (2021). A survey on the development of self-organizing maps for unsupervised intrusion detection. Mobile Networks and Applications, 26(2): 808-829. https://doi.org/10.1007/s11036-019-01353-0

[4] Vujović, Z. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6): 599-606.

[5] Sarker, I.H., Furhad, M.H., Nowrozy, R. (2021). Ai-driven cybersecurity: an overview, security intelligence modeling and research directions. SN Computer Science, 2(3): 1-18. https://doi.org/10.1007/s42979-021-00557-0

[6] Sarker, I.H., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P., Ng, A. (2020). Cybersecurity data science: an overview from machine learning perspective. Journal of Big data, 7(1): 1-29. https://doi.org/10.1186/s40537-020-00318-5

[7] 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. https://doi.org/10.1080/19393555.2015.1125974

[8] Muzafar, S., Humayun, M., Hussain, S.J. (2022). Emerging cybersecurity threats in the eye of E-governance in the current era. In Cybersecurity Measures for E-Government Frameworks, pp. 43-60. https://doi.org/10.4018/978-1-7998-9624-1.ch003

[9] Shah, I.A. (2022). Cybersecurity issues and challenges for E-government during COVID-19: A review. Cybersecurity Measures for E-Government Frameworks, 187-222. https://doi.org/10.4018/978-1-7998-9624-1.ch012

[10] Shareef, S.M. (2017). Security of E-government; Risks, Threats, and Success Factors. Journal of Raparin University-Vol, 4(10): 61.

[11] Sharma, S., Kumar Kar, A., Gupta, M.P. (2021). Unpacking digital accountability: Ensuring efficient and answerable e-governance service delivery. In 14th International Conference on Theory and Practice of Electronic Governance, pp. 260-269. https://doi.org/10.1145/3494193.3494229

[12] Froehlich, A., Ringas, N., Wilson, J. (2022). How space can support African civil societies: Security, peace, and development through Efficient Governance Supported by space applications. Acta Astronautica, 195: 532-539. https://doi.org/10.1016/j.actaastro.2021.06.006

[13] Muradov, İ. (2022). Problems of E-governance in government agencies and their solutions. https://essuir.sumdu.edu.ua/handle/123456789/87509

[14] Ahmet, E.F.E., Kazdal, H. (2019). It security trends for e-government threats. International Journal of Multidisciplinary Studies and Innovative Technologies, 3(2): 105-110.

[15] Bowen, B.M., Devarajan, R., Stolfo, S. (2011). Measuring the human factor of cyber security. In 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 230-235. https://doi.org/10.1109/THS.2011.6107876

[16] Kachavimath, A.V., Nazare, S.V., Akki, S.S. (2020). Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics. In 2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA), Bangalore, India, pp. 711-717. https://doi.org/10.1109/ICIMIA48430.2020.9074929

[17] Kondeti, P.K., Ravi, K., Mutheneni, S.R., Kadiri, M.R., Kumaraswamy, S., Vadlamani, R., Upadhyayula, S.M. (2019). Applications of machine learning techniques to predict filariasis using socio-economic factors. Epidemiology & Infection, 147: e260. https://doi.org/10.1017/S0950268819001481

[18] Al-Mushayt, O.S. (2019). Automating E-government services with artificial intelligence. IEEE Access, 7: 146821-146829. https://doi.org/10.1109/ACCESS.2019.2946204

[19] Gaur, L., Ujjan, R.M.A., Hussain, M. (2022). The influence of deep learning in detecting cyber attacks on E-government applications. In Cybersecurity Measures for E-Government Frameworks, pp. 107-122. https://doi.org/10.4018/978-1-7998-9624-1.ch007

[20] Alagumuthukrishnan, S., Nirmalkumar, A., Naga Rama Devi, G. (2021). Analyze and develop a model for sentimental reviews of e-government services using deep learning algorithms with CNN framework. In AIP Conference Proceedings, 2358(1): 050024. https://doi.org/10.1063/5.0057936

[21] Wang, K., Wang, Z. (2022). Deep learning models and social governance guided by fair policies. Scientific Programming, 2022: 8376325. https://doi.org/10.1155/2022/8376325

[22] Gauthama Raman, M.R., Somu, N., Jagarapu, S., Manghnani, T., Selvam, T., Krithivasan, K., Shankar Sriram, V.S. (2020). An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm. Artificial Intelligence Review, 53(5): 3255-3286. https://doi.org/10.1007/s10462-019-09762-z

[23] Mazini, M., Shirazi, B., Mahdavi, I. (2019). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. Journal of King Saud University-Computer and Information Sciences, 31(4): 541-553. https://doi.org/10.1016/j.jksuci.2018.03.011

[24] Sarker, I.H., Abushark, Y.B., Alsolami, F., Khan, A.I. (2020). Intrudtree: A machine learning based cyber

security intrusion detection model. Symmetry, 12(5): 754. https://doi.org/10.3390/sym12050754

[25] Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., Ming, H. (2019). Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0305-0310. https://doi.org/10.1109/CCWC.2019.8666450

[26] 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. https://doi.org/10.1016/j.iot.2019.100059

[27] Cordero, C.G., Vasilomanolakis, E., Wainakh, A., Mühlhäuser, M., Nadjm-Tehrani, S. (2021). On generating network traffic datasets with synthetic attacks for intrusion detection. ACM Transactions on Privacy and Security (TOPS), 24(2): 1-39. https://doi.org/10.1145/3424155

[28] Ho, T.K. (1998). The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832-844. https://doi.org/10.1109/34.709601

[29] Sagheer, M.W., He, C.L., Nobile, N., Suen, C.Y. (2010). Holistic Urdu handwritten word recognition using support vector machine. In 2010 20th International Conference on Pattern Recognition, pp. 1900-1903. https://doi.org/10.1109/ICPR.2010.468

[30] Li, L., Zhang, Y., Zhao, Y. (2008). k-Nearest neighbors for automated classification of celestial objects. Science in China Series G: Physics, Mechanics and Astronomy, 51(7): 916-922. https://doi.org/10.1007/s11433-008-0088-4

[31] Huang, Y., Li, L. (2011). Naive Bayes classification algorithm based on small sample set. In 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, pp. 34-39. https://doi.org/10.1109/CCIS.2011.6045027