An Intelligent Intrusion Detection System for Internet of Things Attack Detection and Identification Using Machine Learning

Trifa S. Othman and Saman M. Abdullah

Department of Software Engineering, Faculty of Engineering, Koya University,
Koya KOY45, Kurdistan Region - F.R. Iraq

Abstract—**The usability and scalability of Internet of things (IoT) technology are expanding in such a way that they facilitate human living standards. However, they increase the vulnerabilities and attack vectors over IoT networks as well. Thus, more security challenges could be expected and encountered, and more security services and solutions should be provided. Although many security techniques propose and promise good solutions for that intrusion detection systems (IDS)s still considered the best. Many research works proposed machine learning (ML)-based IDSs for IoT attack detection and classification. Nevertheless, they suffer from two main gaps. First, few of the works utilized or could analyze an up-to-date version of IoT-based attack behaviors. Second, few of the works can work as multi-class attack detection and classification. Therefore, this work proposes an intelligent IDS (IIDS) by exploiting the ability of ML algorithms to classify and identify malicious from benign behaviors among IoT network packets. The methodology of this work investigates the efficiency of three ML classifier algorithms, which are K-Nearest Neighbor, support vector machine, and artificial neural network. The developed models have been trained and tested as binary and multi-class classifiers against 15 types of attacks and benign. This work employs an up-to-date dataset known as IoT23, which covers millions of malicious and benign behaviors of IoT-connected devices. The process of developing the proposed IIDSs goes under different preprocessing phases and methods, such as null value solving, SMOTE method for the imbalanced datasets, data normalization, and feature selections. The results present IIDSs as good binary and multi-class classifiers even for zero-day attacks.**

***Index Terms*—Internet of things networks, Intrusion detection system, Machine learning, Intelligent attack classification, and identification.**

I. Introduction

Network technology has mostly oriented toward a new trend called the Internet of Things (IoT). Based on this technology, connected devices can communicate with each other independently with or without human permission (Nagisetty and Gupta, 2019). Although the scalability indicator of networks has been improved with IoT, new challenges have been encountered and measured especially in the ones that are related to networks' security or connected devices' security. Some of the security challenges are related to energy consumption (Malik and Dutta, 2022) and others are related to the system environment of the IoT applications (Ho, 2022).

There are many reasons that make the devices connected over IoT networks be considered vulnerable more to attacks and intruders. Most devices are having resource limitations, such as power and memory limitations. With such limitations, the security tools could not work efficiently as they depend on complex algorithms. Another reason for vulnerability is the inability to build some security standards for connected objects among IoT vendors. IoT devices have been manufactured by many vendors and companies and each of them follow specific security standards. All these reasons increase the number of attacks and threats over IoT networks and expand the vulnerabilities and attack surfaces (Nawir, et al., 2016). Due to that most researchers are focusing on and addressing these open problems, and they are investigating machine learning (ML) techniques and tools for building classifier models to distinguish benign from malicious behaviors of packets that flow over IoT networks (Chen, et al., 2018; Saharkhizan, et al., 2020).

ML techniques have great abilities for detecting and classifying objects. They depend on analyzing some predefined behaviors or attributes, numerically, then mapping them to a class among some available classes (Radivilova, et al., 2019). To build any classifier models, all ML techniques should follow two phases, which are the training and testing phases. Although both phases are important for getting a perfect classifier model, the training phase needs more work and must be more focused. This is because the training phase teaches the ML model through a use of a training dataset, and when collected, such datasets need much preprocessing work that, if not done, it influences negatively on the accuracy rate of the ML classifiers (Sanmorino, 2019). Therefore, one of the questions that this work wants to investigate is about

ARO-The Scientific Journal of Koya University
Vol. XI, No. 1 (2023), Article ID: ARO.11124. 12 pages
DOI: 10.14500/aro.11124
Received: 26 December 2022; Accepted: 02 May 2023
Regular research paper: Published: 22 May 2023
Corresponding author's email: saman.mirza@koyauniversity.org

the possibility of improving the accuracy rate of classifier models through preprocessing steps. This question has been investigated in the security field of IoT networks, especially, using a new training dataset called (IoT-23) (Parmisano, Garcia and Erquiaga, 2020). Based on best knowledge of this work, the concept of the attack classification over IoT has been mentioned for the first time in the book (Giusto, et al., 2010). Since then, many works have been conducted and many investigations have been published (Kareem and Jasim 2022; Kumari and Mrunalini, 2022; Li, Rios and Trajković, 2021). Although the methodology that followed by those works and many other works depended on employing one of the ML techniques for training and evaluating the classifier model and then comparing the obtained results with results of some other works, many influenced parameters on the accuracy rate have not been investigated yet. Therefore, research projects in this field still not saturated. Besides that, most of the conducted research projects utilized some training datasets that were already collected though monitoring non-IoT networks (Tabassum, et al., 2021; Tabassum et al., 2022). Therefore, among the aims that this work wants to focus is, firstly, evolving a most recent collected dataset for attackers over IoT networks. Secondly, to investigate many preprocessing techniques against three ML algorithms to find out the best and more efficient classifier models that could be used for attack and malicious detection over IoT networks.

II. Work Contributions

As mentioned in Section 1, the main objective of this work is to build intelligent intrusion detection systems (IIDSs)-based classifier model that can detect and identify attacks by analyzing the packet behaviors of IoT-based networks. The main contributions of the work could be summarized as below:

1) The work focuses on the most recent dataset (IoT23) that is purely related to IoT-based attacks and benign behaviors excluding behaviors of the traditional networks.
2) The work focuses on analyzing fifteen types of attacks through training the proposed IIDS model on the dataset that mentioned in point (1). The focused type of attacks is up-to-date attacks and mostly related to behavior flow of the IoT-based network packets.
3) Few of works were conducted research projects on analyzing IoT-based behaviors using ML-based classifier models as IIDS binary and multiclass calcification.

III. Related Works

During the review process, it has been found that classifying attacks over IoT-networks depends on a variety of orientations, such as the type of the utilized ML techniques, dataset types and versions, types of the preprocessing techniques, and the performance indicators that used for evaluating the efficiency of the exploited ML algorithms. The orientation that covers the type of ML presents the most important ML techniques that have been proposed by authors of the previous works as detection and classification models. The review presents, as well, the advantages and disadvantages of each technique in the viewpoint of the authors. Another focus of this study could be on the types of datasets used for training and testing ML models. More orientations are available, such as Feature selection, Data normalization, and/or data encoding. Finally, several studies could be categorized based on the utilized performance indicators to measure the efficiency of ML algorithm. In the subsequence sections, many articles have been reviewed based on these orientations.

Despite of diversity in investigating the ability of ML techniques in classifying IoT-based attacks, most researchers agree that expanding the scalability of networks makes connection of new devices to the internet or IoT-based networks becomes more vulnerable than before. It is true that such expansion makes networks be important for our daily life and increases the capability of connecting more devices. Nevertheless, the expansion increases the number of the cyber-attacks over the networks as well, especially over the IoT-based networks as they have limited resources and capabilities. The most important problem is detecting zero-day attacks, which means detecting new patterns or policies of attacks. To overcome this problem, most researchers investigated ML algorithms to build intelligent detection models that can classify new patterns of attacks through learning from known similar patterns. However, there is a disparity over the ability of the ML algorithms as each previous work has proposed a specific algorithm and has justified its ability. Therefore, reviewing those works is necessary.

In general, there are two types of ML algorithms. The first type of ML algorithm is known as classical or conventional algorithms; however, there are some other techniques known as deep learning algorithms (Picon Ruiz, et al., 2020; Sewak, Sahay and Rathore, 2018). The classical ML algorithms are less complex than the deep learning ones. Although, algorithms in both categories are utilized in different works as attacks classification or identification, this work focuses more on the classical ML algorithms as the second type of algorithms needs less resources than deep learning algorithms.

A. Artificial Neural Network (ANN)

The first ML algorithm that could be considered a most distinguished technique is ANN. The ANN algorithm has been utilized by (Soe, et al., 2019) to build an IDS. The author of that work argued that building an ANN model to detect different type of attacks is not sufficient. Instead, the work proposed sequential ANNs in which for each type of attacks an ANN has been designed and developed. Although the paper showed good and high accuracy rate, the structure of such model needs to be updated and a new ANN mast be added to the sequence when a new type of attack or a zero-day attack comes to the live. ANN is considered as a supervised learning algorithm that could be utilized as classifier model. This fact has been used in (Hanif, Ilyas and Zeeshan, 2019) to build an attack detection over IoT-based networks. The work showed that results of a 10-fold cross-validation reach to 84%, which somehow is not good enough. Moreover, the versions of the datasets that have been utilized for training the proposed ANN are going back to 1999 and 2015 which, somehow, are not up to date enough. Another work that focused on ANN to classify attacks over IoT-based networks has been proposed by (Fatayer and Azara, 2019). The work argued that IoT-based networks needs more security as different types of attacks can easily penetrated them. The work built an ANN model to detect many attack types and the work obtained a very good accuracy (97%). However, the work also utilized an old version of dataset (KDD CUP 99), in which, the behaviors of traditional networks have been analyzed and the ANN model cannot be evaluated with recent behaviors of IoT-based attacks. There are many recent works that focused on the ANN based attack classification (Gopi, et al., 2021; Churcher, et al., 2021; Mehmood, Khan and Elhadef, 2022) to classify attacks over IoT-based networks. However, a part of them focused only on one type of attack, other works focused on many types of attack through utilizing some outdated version datasets. Therefore, it is very necessary to investigate the efficiency of the ANN against classifying the most recent behaviors of the devices that connected to the IoT-based networks as binary or/and mutli-class based classifier models.

B. K-Nearest Neighbor (KNN)

Another type of the supervised learning algorithm is called KNN. This type is somehow considered as a lazy learner supervised algorithm as the training phase of this algorithm takes place while the prediction phase is stared (Churcher, et al., 2021). Many recent works utilized KNN as classifier model for detecting attacks over IoT-based networks. However, based on the best of our knowledge, the work (Li, et al., 2014) was the first that utilized the KNN algorithm for attacks and penetrating detection over IoT-based networks. The work proposed the KNN to distinguish intruder sensors over the sensor networks through keeping the authorization of connected objects. One of the most recent works that utilized KNN for IoT-based network attacker is (Iman, 2022). The work proposed the KNN algorithm and argued detecting DDoS attacks over IoT-based networks with minimum consuming of energy. Although the work presented 99% as accuracy rate, the test of the work simulated in SDN environment and it focused only on one type of attack over IoT-based networks. Another recent work that utilized KNN for classifying IoT attacks has been trained with Bot-IoT dataset (Alfarshouti and Almutairi, 2022). The work also presented a taxonomy on the IoT-based attacks. The taxonomy work categorized the available attacks based on their relationship with each layer. Another recent work (Islam, et al., 2022) focused on the IoT-based attacks considering banking systems as an environment case. The work showed that KNN can detect malicious activities up to 98.7%. The work only focused on DDoS attack. Another work that utilized KNN (Aslam, et al., 2022) was depended on adjusting some ML algorithms in the SDN environment and focused on the real-time sniffing packets. The work showed 99% of accuracy and concluded that using SDN controller could be more studied in the future for detection models. However, the work proposed a model to detect phishing attack as a future work. This means that single detection attack always needs to be updated when a new type of attacks comes to the live. Therefore, one of the objectives that addressed by this work is to propose a KNN that could be trained over classifying and detecting most recent type of attacks that penetrating IoT-based networks, not only one attack type.

C. Support Vector Machine (SVM)

SVM is another supervised ML algorithm that could be used for classification, regression, and outlier detection. In the field of IoT-based attack classification, SVM has more frequently used as a common ML algorithm. A recent work that utilized SVM to build an attack detection system over IoT-based networks has depended on Bot-IoT dataset (Alfarshouti and Almutairi, 2022). The work made a comparison between the results that have been obtained from their proposed SVM model with another type of detection model that designed using KNN classifier algorithm. More recent works have utilized SVM as detection method (Islam, et al., 2022). The work investigated one type of the IoT-based attacks, which is DDoS. The work argued that banking system is one of the important environments that should be kept more securable against IoT attacks, especially, DDoS attack which denies bank servers to serve authorized users. The work (Islam, et al., 2022) utilized banking dataset for training the suggested algorithms. Results of the work showed that 99% of the accuracy could be obtained with SVM. Most datasets that used for training the attack detection have complex dimensionality. Therefore, most works employed a process called feature selection for reducing the dimensionality size of the training dataset. A most recent work (Majeed Alhammadi, 2022) utilized principle component analysis (PCA) as a feature selection method to reduce the dimensionality of the training dataset to build a SVM-based attack detection model. The work depended on the outdated version of the intrusion behavior dataset, which known as NLS-KDD and contains 41 attributes. Another work compared the performance of the SVM with decision tree on two types of attacks (DDoS and Code Injection). The work proposed an intrusion detection system for attacks over IoT-based networks in smart city applications. They focused also on a comparison between two types of feature selection (constant removal and recursive feature elimination). The performance of SVM that obtained in that work was 98%. The summary of the research works that have been reviewed throughout sub-sections 3.1, 3.2, and 3.3 could be illustrated in Table I, which somehow summarizes the differences between the most reviewed works with this work.

Table I shows some differences that distinguish methodology of this work with methodology that followed

TABLE I
The Work Review Summary

Reference	ML tools	Datasets	Binary or multiclass	Number of attacks
Soe, et al., 2019	ANN	N-BaIoT	Multiclass	2
Hanif, Ilyas and Zeeshan, 2019	ANN	UNSW-15	Binary	1
Fatayer and Azara, 2019	ANN	KDD CUP 99	Binary	1
Iman, 2022	KNN	SDN simulation	Binary and DDoS	1
Islam et al., 2022	SVM	Bot-IoT	Binary and DDoS	1
Majeed Alhammadi, 2022	STV and DT	NLS-KDD	Multiclass	2
This work	ANN, KNN, and SVM	IoT-23	Binary and multiclass	8

ML: Machine learning, ANN: Artificial neural network, SVM: Support vector machine, DDoS: Distributed denial of service, KNN: K-nearest neighbor

by some previous works. The distinction of the work methodology of this work could be summarized as below:

This work investigates three common ML algorithms, which are ANN, KNN, and SVM. There are other works already employed these ML algorithms; however, based on the best knowledge of us, they have been utilized in different fields or for different topics.

1. Most of the reviewed works have utilized the ML algorithms that mentioned in (1) with outdated detests or with non-IoT based datasets. However, this work focused on the IoT23 dataset that could be considered as a most recent dataset for analyzing behaviors of IoT-connected devices.
2. Most reviewed works have been developed as binary classification, which means distinguishing behaviors of one type of attack with begin behaviors. However, this work develops binary classification and mutli-class classification.
3. With the present work, more than 15 types of attacks have been included for developing a multi-class classification model.
4. One of the major differences between this work and reviewed works is classifying zero-day-attacks as malicious behavior based on the behaviors of some known attacks.

It is important to consider all above-mentioned points together to highlight the differences between this work and most reviewed previous works. Because considering each difference individually decreases the difference and the gap between this work and previous works.

IV. Materials and Methods

In this section and the subsequence sub-sections, the methodology and the materials that have been unitized by this work will be explained. Fig. 1 shows the framework of this project.

A. Dataset

This work utilizes the IoT23 dataset for training and testing the proposed IIDS. The work imports the dataset from (Garcia, Parmisano and Erquiaga, 2020), in which, records in this dataset represent benign and malicious behaviors of packets that flow over IoT-based networks. In the dataset, there are three groups or scenarios of benign behaviors and 20 groups or scenarios of malicious behaviors. Whether a behavior is benign or malicious, it consists of 21 features or attributes. The last feature is the label which represents the class of the correspondence behavior.

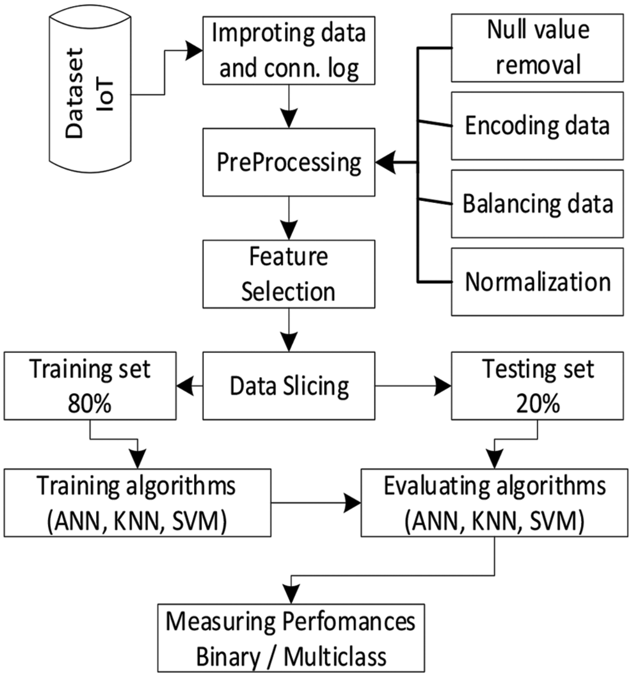

Dataset
IoT
Improting data and conn. log
Null value removal
Encoding data
Balancing data
Normalization
PreProcessing
Feature Selection
Training set 80%
Data Slicing
Testing set 20%
Training algorithms (ANN, KNN, SVM)
Evaluating algorithms (ANN, KNN, SVM)
Measuring Perfomances Binary / Multiclass

Fig. 1. The framework of the Internet of things attack classification and identification.

Behaviors in the IoT23 dataset could be labeled as benign or malicious when the goal is developing a binary classification. However, the dataset has been prepared for developing a multi-class classification as well, because a behavior in the dataset may has different attack classes. As an example, there are two different labels of attack (C&C and PartOfAHorizontalPortScan); however, a class of attack comes with label (C&C -PartOfAHorizontalPortScan), which means that this class is belong to a flow contains malicious activities from both type of attacks. Below are the description of each attack type and Table II shows the name of the available classes and the number of each class's observations.

1. Attack: This type of attack could be encountered when an infected device attacks another host, and it tries to take an advantage of a vulnerability.
2. Benign: Is a device which no suspicious or malicious activities detected from its flow over network
3. C&C: Is a command and controlled server that an infected device can connect to and control it. The infected device was connected to a CC server.

TABLE II
Number of Flows for Each Attack Class in the Dataset

Serial number	Label	Flows
1	Benign	30,864,692
2	Attack	9,398
3	DDoS	19,538,713
4	Part of a horizontal port scan	213,852,924
5	Part of a horizontal port scan-attack	5
6	Okiru	60,990,708
7	Okiru-attack	3
8	File download	18
9	C and C	21,995
10	C and C-heart beat	33,673
11	C and C-file download	53
12	C and C-heart beat-attack	834
13	C and C-heart beat-file download	11
14	C and C-part of a horizontal port scan	888
15	C and C-Torii	30
16	C and C-Mirai	2
	Total	325,313,947

DDoS: Distributed denial of service

4. DDoS: Is a Distributed Denial of Service attack that an infected device lunch a malicious activity to penetrate another device.
5. FileDownload: Is encountered when a file is being downloaded to an infected device.
6. HeartBeat: with this attack the track of the infected host by the C&C server will be sent through a packet.
7. Mirai: The connections have characteristics of a Mirai botnet.
8. Okiru: The connections have characteristics of a Okiru botnet.
9. PartOfAHorizontalPortScan: A horizontal port scan has been lunched by infected device to gather information for performing further attacks.
10. Torii: The connections have characteristics of a Torii botnet.

The imported dataset has a size of 20 GB. The dataset has been distributed over 23 folders; three of them are representing the benign datasets and the rest of 20 folders are representing the malicious activates over IoT-based networks. Inside each folder, there is a conn.log file (this is the Zeek conn.log file obtained by running the Zeek network analyzer using the original pcap file), and the file is containing the flow activities. Files in all folders focus on the same number of features, which are 23 features, including the target label feature, as shown in the Table III.

The flow activities in each file have not specified for a single type of attack, in the contrast, each file contains different malicious activities of IoT malwares.

B. Data Preprocessing

As shown in the Fig. 1, four main preprocessing activities have been utilized by this work and have been applied on the imported dataset. The pre-processes are:

1. Removing the null values and features with zero impact
2. Coding and encoding
3. Data balancing
4. Normalization.

For the first preprocessing, null values in the employed dataset have been handled. Three features in the imported dataset are empty and without any records, which means all cells in these three attributes have null value records. Those three features are local_orig, local_resp of connection types, and tunnel parents (No. 13, 14, and 21 in the Table III). These three features have been removed from the selected feature list, because null value methods cannot be applied on a feature that totally empty.

The process of feature selection will exclude other features as well. There are many features having zero impact on the classification process, which are feature no. 1 and feature no. 2 in the Table III. History is another feature which this work decided to delete it, as it describes only the history of conn_state. This work removes all these features as shown in Table IV.

IoT 23 dataset includes three numerical features that include missing value which are Duration, Origin Bytes and Respond Bytes (No. 9, 10 and 11). Although some categorical features also include missing value, they have not been dealt as missing value. For example, in Service variable the symbol (-) means no service is available and it has been replaced as (Nos) value as an indicator that this value shows that there is no service rather than considering it as a null or missing value. Class-based mean method is used to handle the null values in this method. The mean value of a variable is used to replace missing values, and missing values for benign and malicious observations within the same variable are computed separately (Lee and Zeng, 2008). Finally, the removing process also covered the duplicated observations. The output of this process reduces the dimensionality of the dataset. The number of features that remains in the dataset becomes 15 features.

The dataset needs Feature Encoding as it has six categorical features after applying the null value cleansing process on the dataset. Those categorical values should be changed to numerical variables. The process of encoding includes three steps (Label Encoding, Encoding categorical features, and IP Address Encoding). The labels of IoT 23 dataset are categorical values and must be encoded to numerical values for ML algorithms. As this study implements three classifiers (KNN, SVM, and ANN), the work requires two forms of Label-Encoding for identification. Ordinal encoding is used for (KNN and SVM) classifiers (as indicated in Table IV), while One Hot Encoding is used for (ANN) classifiers. Since in binary classification, the same label encoding is used for all classifiers, with 0 being assigned to benign label values and 1 to malicious values.

In Encoding Categorical Features, three categorical features of IoT 23 dataset (Protocol, Service and conn-state) encoded using frequency encoding, which according to this method, each value in a categorical feature must be modified with the total count or frequency of the value.

The two variables (id. orig_h Address, id. resp_h Address) of IoT 23 dataset are IP Address format, and they have been encoded to numerical format using IP Splitting method. According to this method, an IP address will be divided into four distinct octets number, in which each octet number will

TABLE III
Name and Description of Features

#	Feature	Description
1	Time	Time for flow starting
2	uid	Unique ID
3	id.orig-h	Source IP address
4	id.orig-p	Source port
5	id.resp-h	Destination IP address
6	id.resp-p	Destination port
7	Protocols	Transaction protocol: icmp, udp, tcp,
8	Service	dhcp, dns, http, irc, ssh, ssl
9	Duration	Total duration of flow
10	orig_bytes	Number of payload bytes the originator sent
11	resp_bytes	Number of payload bytes the responder sent
12	conn_state	Connection state. Possible values are found in Table III
13	local_orig	T if the connection originated locally and F if it originated remotely
14	local_resp	T if the connection is responded locally and F if it is responded remotely
15	missed_bytes	Number of bytes missed in content gaps, which is representative of packet loss
16	History	State history of connections as a string of letters. The letter is uppercase if it comes from the responder and lowercase if it comes from the originator. Possible letters can be seen in Table IV
17	orig_pkts	Number of packets that the originator sent
18	orig_ip_bytes	Number of IP level bytes that the originator sent
19	resp_pkts	Number of packets that the responder sent
20	resp_ip_bytes	Number of IP level bytes that the responder sent
21	Tunnel parents	The connection's ID, if it was tunneled
22	Label	Whether the capture was normal or malicious
23	Detailed_label	Identify the malicious capture type

Table IV
Ordinal Encoding of the Labels of Internet of Things 23 Dataset for K-Nearest Neighbor and Support Vector Machine Models

Labels	Encoded label
C and C	1
C and C-heart beat-attack	2
C and C-Part of a horizontal port scan	3
Attack	4
C and C-heart beat	5
DDoS	6
Okiru	7
Part of a horizontal port scan	8
Part of a horizontal port scan-attack	9
Okiru-attack	10
File download	11
C and C-file download	12
C and C-heart beat-file download	13
C and C-Torri	14
C and C-Mirai	15

DDoS: Distributed denial of service

be assigned to a distinct variable. It means, the attribute of IP address will be converted to four distinct variables. In this way, this work encoded both the source and destination IP addresses, and as a result eight new variables have been added to the list of features. The two IP address variables in 32-bit address format were then removed.as shown in Fig. 2.

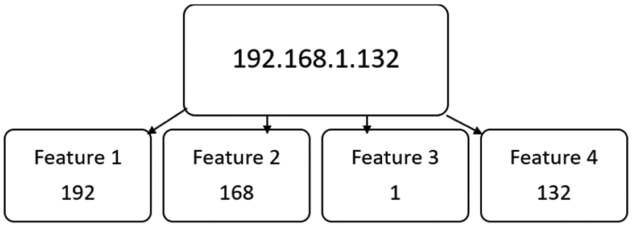


Fig. 2. IP splitting example.

The third preprocessing step is balancing dataset. The IoT23 dataset is imbalanced in the number of the observations, it has in reference to each class. Based on the number of observations per classes, classes could be categorized into three main groups. The first group covers those classes that having millions of observations (number 1, 3, 4, 6 in the Table I). The second group is those classes that having thousands of observations (2, 9, and 10 in the Table I), and third group is those classes that having <1000 observations (5, 7, 8, 11, 12, 13, 14, 15, and 16 in the Table I). Fig. 3 clearly shows the imbalanced status of the dataset. This work achieves the process of balancing the IoT-23 dataset through two phases. At the first phase, the work reduces the gap that exists among the attack's classes in the number of observations that they have. To achieve that this work randomly picks 2000 samples from the first and second groups of attack's class in the IoT-23 dataset.

The second phase for illuminating the imbalances in the dataset is applying the SMOTE algorithm to the third group of attack's class to rise the number of the observations in all attack's classes up to 2000. The SMOTE method is a statistical technique, and it stands for Synthetic Minority Oversampling Techniques. According to recent research work (Wongvorachan, He, and Bulut, 2023), SMOTE uses for increasing the number of cases in a dataset for balancing purposes. It uses for increasing the number of cases in a dataset for balancing purposes. The technique works through generating new cases from the existing minority instances. The main condition that

SMOTE has is the implementation of the technique should not change number of the majority cases. The technique should inset just copies of the existing cases. Instead, the technique takes samples from the feature space of each targeted class and its nearest neighbors. Finally, the SMOTE will be applied to entire dataset; however, it only increases the percentage of minority cases.

Focusing on the IoT23 dataset, Fig. 3 shows the imbalanced status of the dataset before processing the dataset under the SMOTE algorithm, and Fig. 4 shows the output of the SOMT method.

Finally, this work applied the normalizing method on the dataset to put records in all remain features in the same range. This work uses the min-max normalization method. This process can speed up the training and testing phases of the classifier models.

C. Feature Selection

Feature selection is a method of reducing the number of attributes that utilized by the proposed model through selecting only relevant feature(s) and getting minimizing of noise in a dataset (Abdulla, Al-Dabagh and Zakaria, 2010). There are six features already have been excluded before feature selection process. Table IV presents these features and the reason of excluding each of them. For the rest of attributes, this work depends on computing the correlations coefficient among the attributes, first, then to compute the correlation coefficient between each attribute and the target attribute. According to this method, the selection of the attributes (or features) in the dataset depends on the condition that states "Attributes should never have correlations among them. If any two correlated attributed found, the one that has less correlation with the target attribute will be excluded"(Weller-Fahy, Borghetti and Sodemann, 2014) (Abdulla, Al-Dabagh and Zakaria, 2010). After checking the correlations, the remaining features are (id. resp_h Address, id. resp_p port, Protocol, Service, Duration, Origin Bytes, Respond Bytes, conn_state, missed_bytes). Fig. 4 shows the correlation status among the attributes or features. Fig. 5 shows a sample of the dataset after the preprocessing steps, excluding the normalization process.

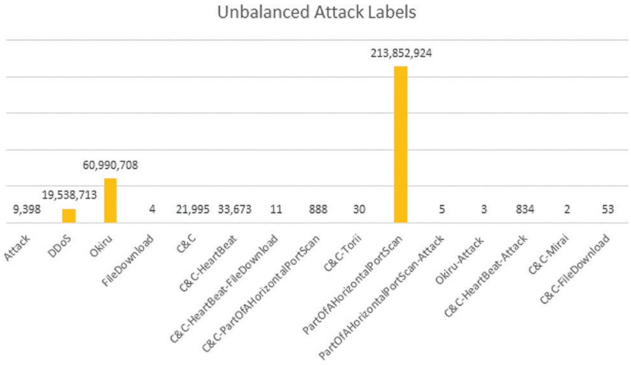


Fig. 3. The unbalanced Internet of things-23 structure.

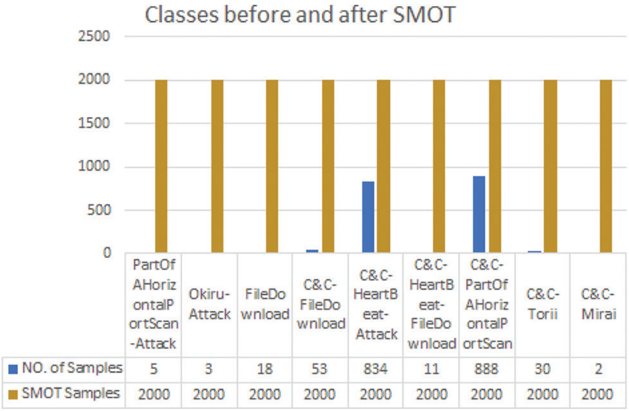


Fig. 4. Phase two output of unbalanced Internet of things-23 structure.

D. Data Slicing

This process is about splitting the dataset into two subsets, the training and the testing. Although the obtaining subsets will be directly used and fed to the ML classifier models, this process still be considered as a step of preprocessing activities. This work allocates 20% of the dataset as a testing subset and 80% of the dataset assigns for training phase. The process of extracting samples from the dataset for training and testing has been achieved randomly.

This work takes from the benign class 20% of records randomly, and the remain 80% will be used for training. However, taking the samples from the attack classes is slightly different for keeping the balance of the dataset in the viewpoint of attack participating. The work allocated from each attack class 20% for testing and 80% for training. Then, all 20% parts will be collected to form on testing subset and same is true for the training subsets.

E. Performance Indicators

Fig. 7 shows details of a typical confusion matrix (Bhandari, 2020). From the confusion matrix, all necessary accuracy indicators could be obtained. Although every index in the figure means something useful, rate of accuracy is most common that utilized to check the performance of detection and classification models.

V. Experimental Evaluation

This work utilized three major ML algorithms named ANN, SVM, and KNN. The aim of this work is building an intelligent binary and multi-class classification. The experimental evaluation in this work depends on k-fold method, by which, the dataset will be divided into five partitions, each time, a part will be used for testing and the remain nine parts used for training.

A. ANN Based Classification

ANN is a common ML-based model that functions based on how the human brain operates. It is a supervised learning algorithm that its structure consists of neurons or nodes. Those nodes are distributed over three main layers, namely, input layer, hidden layer, and output layer. Nodes at each layer have different functionalities. At the input layer, nodes take

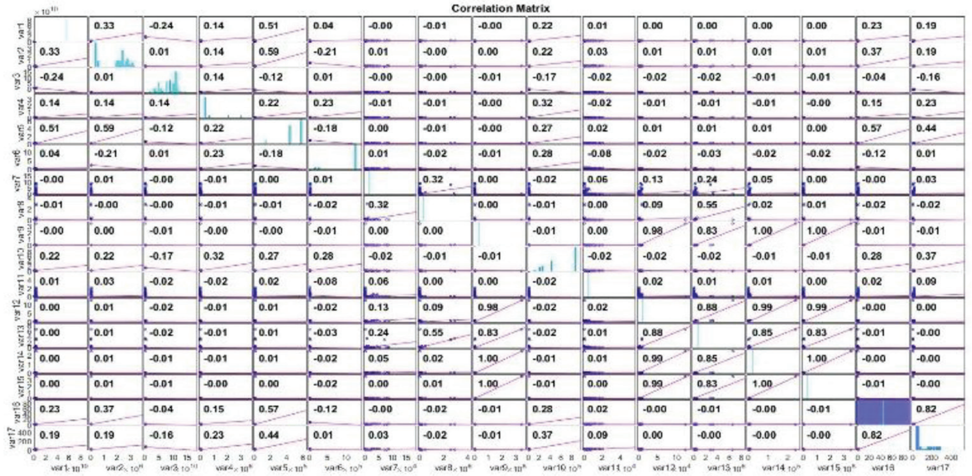
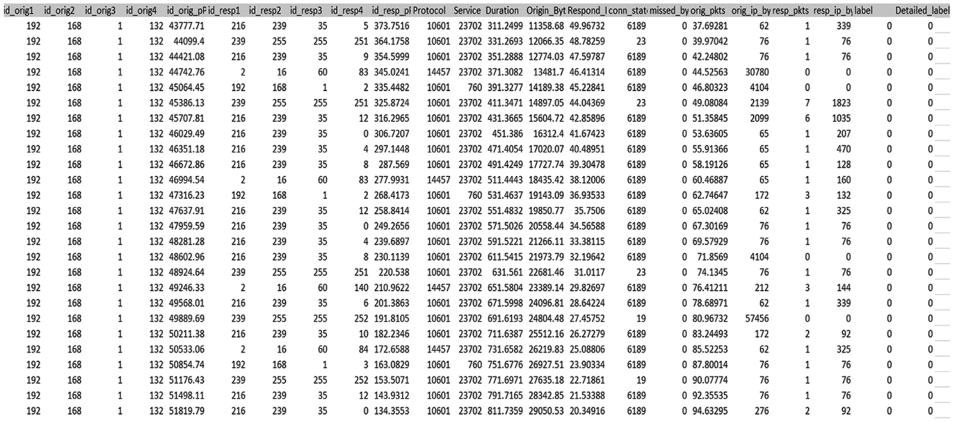
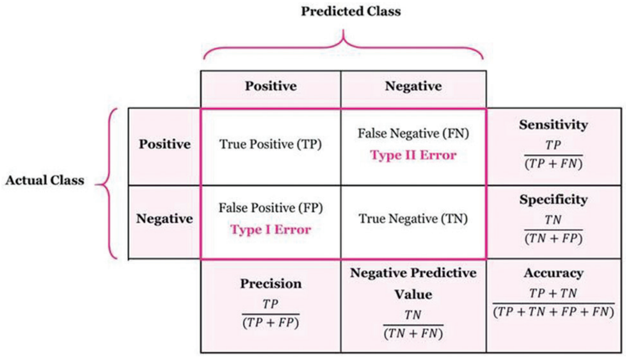


Fig. 5. Correlation coefficient graph of Internet of things 23 dataset features.



id_orig1	id_orig2	id_orig3	id_orig4	id_orig_pF	id_resp1	id_resp2	id_resp3	id_resp4	id_resp_pl	Protocol	Service	Duration	Origin_Byt	Respond_I	conn_stat	missed_by	orig_pkts	orig_ip_by	resp_pkts	resp_ip_by	label	Detailed_label
192	168	1	132	43777.71	216	239	35	5	373.7516	10601	23702	311.2499	11358.68	49.96732	6189	0	37.69281	62	1	339	0	0
192	168	1	132	44099.4	239	255	255	251	364.1758	10601	23702	331.2693	12066.35	48.78259	23	0	39.97042	76	1	76	0	0
192	168	1	132	44421.08	216	239	35	9	354.5999	10601	23702	351.2888	12774.03	47.59787	6189	0	42.24802	76	1	76	0	0
192	168	1	132	44742.76	2	16	60	83	345.0241	14457	23702	371.3082	13481.7	46.41314	6189	0	44.52563	30780	0	0	0	0
192	168	1	132	45064.45	192	168	1	2	335.4482	10601	760	391.3277	14189.38	45.22841	6189	0	46.80323	4104	0	0	0	0
192	168	1	132	45386.13	239	255	255	251	325.8724	10601	23702	411.3471	14897.05	44.04369	23	0	49.08084	2139	7	1823	0	0
192	168	1	132	45707.81	216	239	35	12	316.2965	10601	23702	431.3665	15604.72	42.85896	6189	0	51.35845	2099	6	1035	0	0
192	168	1	132	46029.49	216	239	35	0	306.7207	10601	23702	451.386	16312.4	41.67423	6189	0	53.63605	65	1	207	0	0
192	168	1	132	46351.18	216	239	35	4	297.1448	10601	23702	471.4054	17020.07	40.48951	6189	0	55.91366	65	1	470	0	0
192	168	1	132	46672.86	216	239	35	8	287.569	10601	23702	491.4249	17727.74	39.30478	6189	0	58.19126	65	1	128	0	0
192	168	1	132	46994.54	2	16	60	83	277.9931	14457	23702	511.4443	18435.42	38.12006	6189	0	60.46887	65	1	160	0	0
192	168	1	132	47316.23	192	168	1	2	268.4173	10601	760	531.4637	19143.09	36.93533	6189	0	62.74647	172	3	132	0	0
192	168	1	132	47637.91	216	239	35	12	258.8414	10601	23702	551.4832	19850.77	35.7506	6189	0	65.02408	62	1	325	0	0
192	168	1	132	47959.59	216	239	35	0	249.2656	10601	23702	571.5026	20558.44	34.56588	6189	0	67.30169	76	1	76	0	0
192	168	1	132	48281.28	216	239	35	4	239.6897	10601	23702	591.5221	21266.11	33.38115	6189	0	69.57929	76	1	76	0	0
192	168	1	132	48602.96	216	239	35	8	230.1139	10601	23702	611.5415	21973.79	32.19642	6189	0	71.8569	4104	0	0	0	0
192	168	1	132	48924.64	239	255	255	251	220.538	10601	23702	631.561	22681.46	31.0117	23	0	74.1345	76	1	76	0	0
192	168	1	132	49246.33	2	16	60	140	210.9622	14457	23702	651.5804	23389.14	29.82697	6189	0	76.41211	212	3	144	0	0
192	168	1	132	49568.01	216	239	35	6	201.3863	10601	23702	671.5998	24096.81	28.64224	6189	0	78.68971	62	1	339	0	0
192	168	1	132	49889.69	239	255	255	252	191.8105	10601	23702	691.6193	24804.48	27.45752	19	0	80.96732	57456	0	0	0	0
192	168	1	132	50211.38	216	239	35	10	182.2346	10601	23702	711.6387	25512.16	26.27279	6189	0	83.24493	172	2	92	0	0
192	168	1	132	50533.06	2	16	60	84	172.6588	14457	23702	731.6582	26219.83	25.08806	6189	0	85.52253	62	1	325	0	0
192	168	1	132	50854.74	192	168	1	3	163.0829	10601	760	751.6776	26927.51	23.90334	6189	0	87.80014	76	1	76	0	0
192	168	1	132	51176.43	239	255	255	252	153.5071	10601	23702	771.6971	27635.18	22.71861	19	0	90.07774	76	1	76	0	0
192	168	1	132	51498.11	216	239	35	12	143.9312	10601	23702	791.7165	28342.85	21.53388	6189	0	92.35535	76	1	76	0	0
192	168	1	132	51819.79	216	239	35	0	134.3553	10601	23702	811.7359	29050.53	20.34916	6189	0	94.63295	276	2	92	0	0

Fig. 6. A sample of the obtained dataset through preprocessing.



		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	Negative Predictive Value $\frac{TN}{(TN+FN)}$	Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Fig. 7. Typical Confusion Matrix with performance indicators.

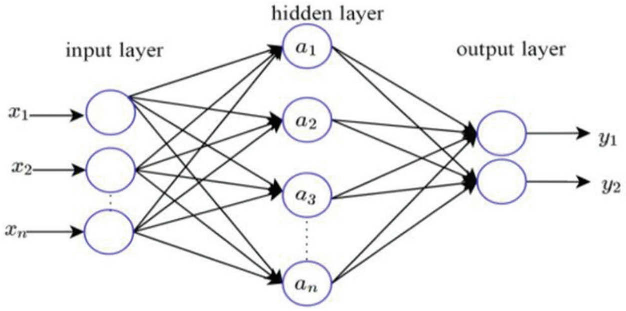


Fig. 8. Typical structure of artificial neural network.

the input information and pass them to hidden layer. The core computation of the ANN is occurred in the hidden layer, which in some cases, there are more than one layer. The results from the hidden layer(s) will be passed to the output layer. For the supervised ANN, the expected output and desired output will be involved in error computation which shows the accuracy rate of the training phase. When the obtained error is more the goal, ANN will start to modify the value of wights that exist between each two nodes in two different layers. This process will be repeated until minimum error will be obtained. The typical structure of an ANN is shown in the Fig. 5.

For this work, the model has been designed and codded using Matlab-R2021a. It has been installed on a PC with intel CORE i7 (11th generation). The ANN that utilized by this work is called "Pattern Recognition Neural Network". According to the dataset sample that shown in the Fig. 8, the number of the input feature in this work is (15). Therefore, the number of the input node of the proposed ANN for this work is 15. The work has tested the ANN to find out the best or the more efficient structure (number of hidden layer). The work set the number of hidden layers on one and the nodes in this layer on 10 nodes. Fig. 9 shows the ANN structure that designed by this wok for binary classification of attacks.

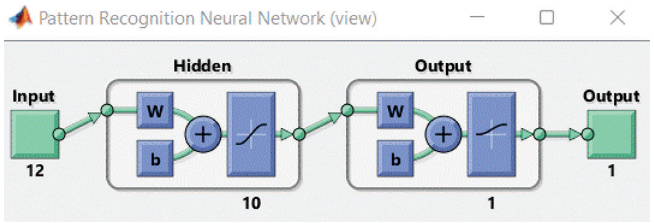
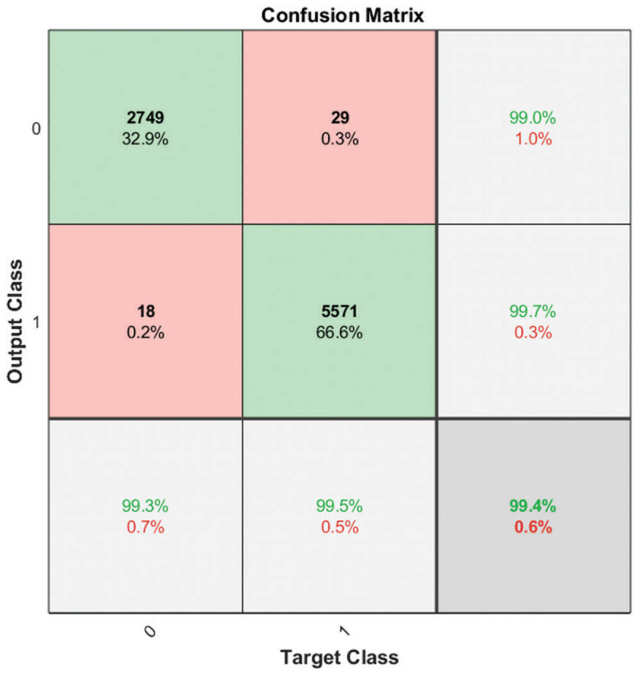


Fig. 9. The artificial neural network-based binary classification.



	0	1	
0	2749 32.9%	29 0.3%	99.0% 1.0%
1	18 0.2%	5571 66.6%	99.7% 0.3%
	99.3% 0.7%	99.5% 0.5%	99.4% 0.6%

Fig. 10. Confusion Matrix for artificial neural network-based binary classification.

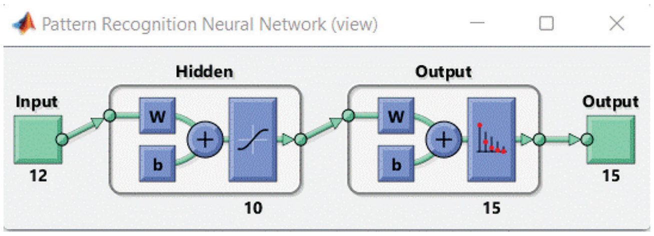
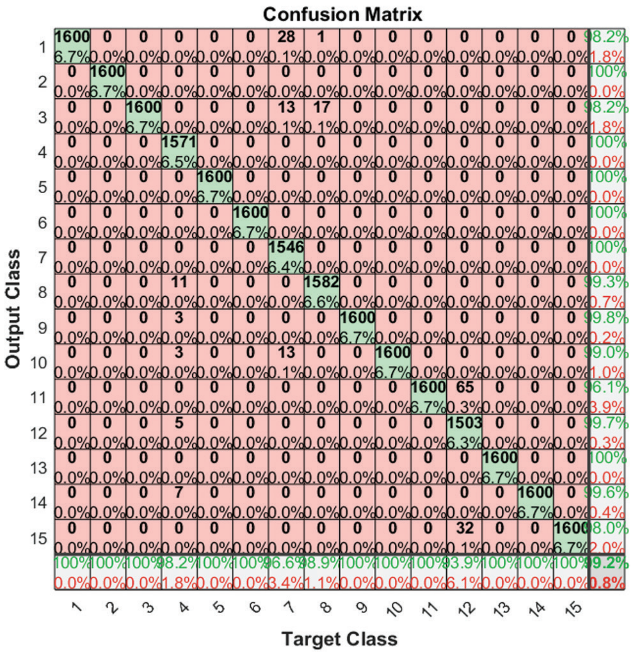
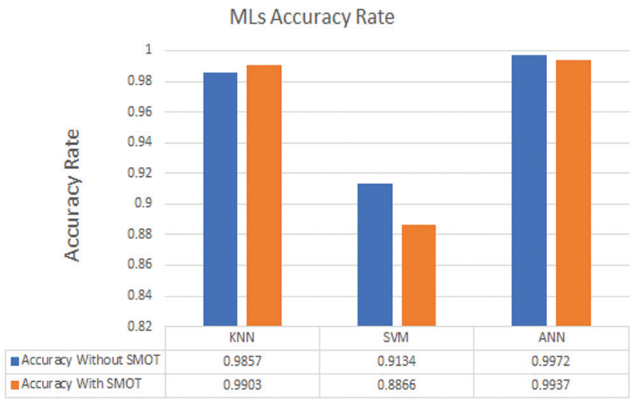


Fig. 11. The ANN based multi-class classification.



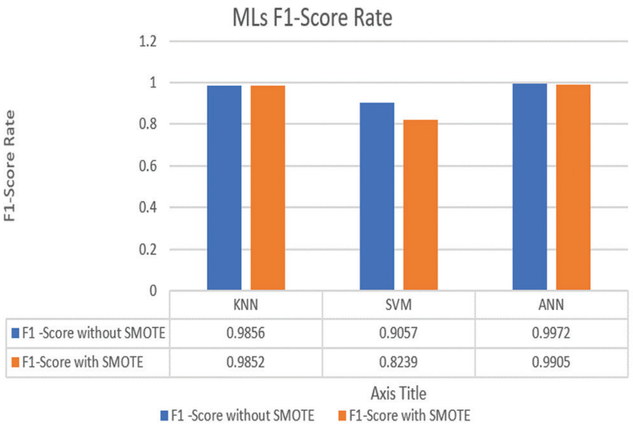
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	28 0.1%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.2% 1.8%
2	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	13 0.1%	17 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.2% 1.8%
4	0 0.0%	0 0.0%	0 0.0%	1571 6.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1546 6.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	11 0.0%	0 0.0%	0 0.0%	0 0.0%	1582 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.3% 0.7%
9	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
10	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	13 0.1%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.0% 1.0%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	65 0.3%	0 0.0%	0 0.0%	0 0.0%	96.1% 3.9%
12	0 0.0%	0 0.0%	0 0.0%	5 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1503 6.3%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	0 0.0%	100% 0.0%
14	0 0.0%	0 0.0%	0 0.0%	7 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 6.7%	0 0.0%	99.6% 0.4%
15	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	32 0.1%	0 0.0%	0 0.0%	1600 6.7%	98.0% 2.0%
	100% 0.0%	100% 0.0%	100% 0.0%	98.2% 1.8%	100% 0.0%	100% 0.0%	96.6% 3.4%	98.9% 1.1%	100% 0.0%	100% 0.0%	100% 0.0%	93.9% 6.1%	100% 0.0%	100% 0.0%	100% 0.0%	99.2% 0.8%

Fig. 12. The confusion Matrix of artificial neural network-based multi-class classifier (SMOTE).



	KNN	SVM	ANN
🟦 Accuracy Without SMOT	0.9857	0.9134	0.9972
🟧 Accuracy With SMOT	0.9903	0.8866	0.9937

Fig. 13. Compression of the MLs accuracy.



	KNN	SVM	ANN
🟦 F1 -Score without SMOTE	0.9856	0.9057	0.9972
🟧 F1-Score with SMOTE	0.9852	0.8239	0.9905

Fig. 14. Compression of the machine languages F1-score.

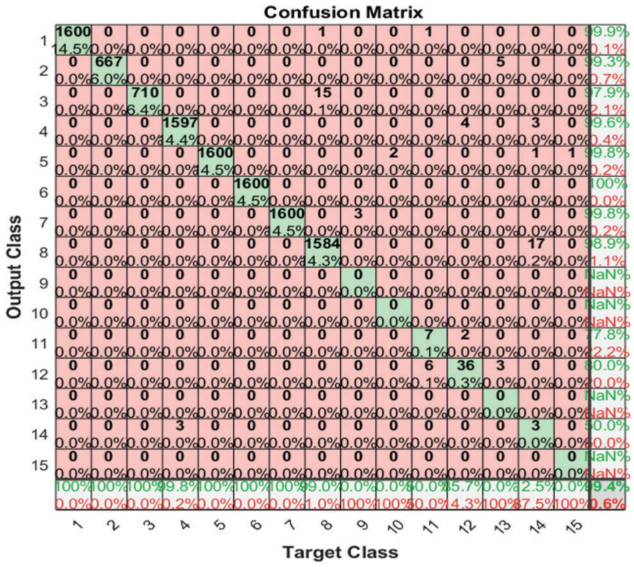
The proposed ANN just required 45 epochs for getting perfect training with error around 1.5%. To test the ANN, this work used 20% of the dataset and the result of testing is shown as confusion matrix for the binary in the Fig. 10. The result of the testing is 99%.

The next step of with ANN is to identify the type attacks after identifying a flow as attack. For this step, the name of the ANN is still "Pattern Recognition Neural Network". However, the structure of ANN has not been changed as shown in Fig. 11 and with the same number of epochs.

The accuracy that obtained through the multi-class classification, as shown in the Fig. 12, is about 99.2%. Through both classifiers, it becomes clear that classifying benign from attacks and identifying the type of attacks with ANN pattern recognition can reach up to 99% as an average.

VI. ML Comparison Results

This work utilizes another two major ML techniques to evaluate their accuracy with the ANN based on attack classification and identification. These techniques are KNN and SVM. This work compared the ANN based model with both KNN and SVM when they work as binary classifiers and work as multi-class classifiers. The comparisons result of the binary classification is shown in Figs. 13 and 14, and the results of multiclass classification are shown in the Tables VI and VII. The results also show the impact of the SMOTE technique on the accuracy and F1-score rates of the ML techniques. Moreover, the work compared the three ML techniques as the multi-classifiers. In general, the accuracy rate for ML techniques as binary classifiers ranged between 88.66% and 99.72%. SMOTE has a greater impact on multiclass classification than it has on binary classification.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	1600 14.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.9% 0.1%
2	0 0.0%	667 6.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 0.0%	0 0.0%	0 0.0%	99.3% 0.7%
3	0 0.0%	0 0.0%	710 6.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.9% 2.1%
4	0 0.0%	0 0.0%	0 0.0%	1597 14.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.0%	0 0.0%	3 0.0%	0 0.0%	99.6% 0.4%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 14.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	99.8% 0.2%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 14.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1600 14.5%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1584 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	17 0.2%	0 0.0%	98.9% 1.1%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.1%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	77.8% 22.2%
12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 0.1%	36 0.3%	3 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
14	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.0%	0 0.0%	50.0% 50.0%
15	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	100% 0.0%	100% 0.0%	100% 0.0%	99.8% 0.2%	100% 0.0%	100% 0.0%	100% 0.0%	99.0% 1.0%	0.0% 100%	0.0% 100%	50.0% 50.0%	85.7% 14.3%	0.0% 100%	12.5% 87.5%	0.0% 100%	99.4% 0.6%

Fig. 15. The confusion Matrix of artificial neural network-based multi-class classifier (without SMOTE).

TABLE V
Excluded Features

Features	Reason of exclusion
Time	Not relevant to attack classification and identification
Uid	Not relevant to attack classification and identification
local_orig	All records are empty
local_resp	All records are empty
History	It is a description of another feature (conn state)
Tunnel parents	All records are empty

There is one fact that should be presented at the beginning of this discussion, which is "The accuracy of any classification model that trained with unbalance dataset is useless even it has a very good rate". This is because unbalanced dataset usually makes the training process to bias to a class that has more observations than other classes. Consequently, we tested the models using F1-Score as well, and the influence of SMOTE appeared significantly, as shown in Table VII. The results indicated that the labels (9, 10, 13, and 15) had 0% F1-score rate, which is due to the small number of observations in these labels, as shown in Fig. 15. The results of the SMOTE dataset then solved the problem, as illustrated in Fig. 12.

There are many arguments about SMOTE applying to the dataset. Much research works perusing applying the SMOTE only over the training part of the dataset. Others are focused on applying the SMOTE over all dataset. Therefore, this work investigates whether SMOTE overfits the model if applied on test part of the dataset or not. This work conducted another experiment to check that as shown in Table VIII. In this experiment SMOTE, only applied to Train set. The experimental results show that the models (KNN, SVM, and ANN) did not identify several attack classes, such as (10, 11, and 15). This is because the number of records in the test set data is quite low. In the test set, such attack classes (10 and 15) have only one record.

The classes (12, 13, and 14) have the same issue; however, in these classes, some methods produced at least some outcomes.

TABLE VI
The Identification Accuracy Rate of Machine Learning Techniques for Each Attack

Labels	KNN		SVM		ANN	
	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE
1	1	1	0.9999	1	0.9998	0.9988
2	0.9999	0.9999	0.9999	1	0.9995	1
3	0.9994	0.9998	0.9993	0.9994	0.9986	0.9988
4	0.9984	0.9993	0.9948	0.9809	0.9991	0.9988
5	1	1	1	1	0.9996	1
6	1	1	1	1	1	1
7	1	0.9998	0.9735	0.9853	0.9997	0.9977
8	0.9990	0.9998	0.9712	0.9872	0.9970	0.9988
9	0.9997	1	0.9997	1	0.9997	0.9999
10	0.9998	0.9998	0.9998	0.9975	0.9998	0.9993
11	0.9999	0.9991	0.9988	0.9837	0.9992	0.9973
12	0.9993	0.9989	0.9973	0.9946	0.9986	0.9957
13	1	1	0.9993	1	0.9993	1
14	0.9996	0.9998	0.9984	0.9999	0.9978	0.9997
15	0.9999	1	0.9997	1	0.9999	0.9987

ANN: Artificial neural network, SVM: Support vector machine, KNN: K-nearest neighbor

TABLE VII
The Identification F1-score Rate of Machine Learning Techniques for Each Attack

Labels	KNN		SVM		ANN	
	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE
1	1	1	0.9997	1	0.9994	0.9910
2	0.9993	0.9997	0.9993	1	0.9962	1
3	0.9951	0.9981	0.9944	0.9956	0.9895	0.9907
4	0.9944	0.9943	0.9822	0.8447	0.9968	0.9909
5	1	1	1	1	0.9987	1
6	1	1	1	1	1	1
7	1	0.9984	0.9161	0.8969	0.9990	0.9829
8	0.9966	0.9981	0.8905	0.8939	0.9897	0.9909
9	0	1	0	1	0	0.9990
10	0	0.9984	0	0.9813	0	0.9950
11	0.9630	0.9935	0	0.8881	0.6087	0.9801
12	0.8947	0.9922	0.4444	0.9587	0.8276	0.9672
13	1	1	0	1	0	1
14	0.8750	0.9978	0	0.9994	0.2000	0.9978
15	0	1	0.4000	1	0	0.9901

ANN: Artificial neural network, SVM: Support vector machine, KNN: K-nearest neighbor

TABLE VIII
The Identification F1-score and Accuracy Rate of Machine Learning Techniques, Which SMOTE Has Not Been Applied to Test Set Data

Labels	KNN		SVM		ANN	
	Accuracy	F score	Accuracy	F score	Accuracy	F score
1	1	1	1	1	0.9996	0.9989
2	0.9996	0.9970	1	1	0.9996	0.9969
3	1	1	0.9986	0.9889	0.9986	0.9889
4	0.9968	0.9886	0.9856	0.9476	0.9693	0.8803
5	1	1	1	1	1	1
6	1	1	1	1	1	1
7	0.9996	0.9988	0.9755	0.9217	0.9996	0.9989
8	0.9993	0.9975	0.9744	0.9026	0.9978	0.9925
9	1	1	1	1	1	1
10	0.9996	0	0.9996	0	0.9996	0
11	0.9986	0.3333	0.9859	0	0.9870	0
12	0.9982	0.8000	0.9986	0.8333	0.9913	0.4545
13	1	1	1	1	0.9935	0.2502
14	0.9982	0.7059	0.9986	0.7500	0.9960	0.5218
15	0.9993	0	0.9996	0	0.9996	0.6667

ANN: Artificial neural network, SVM: Support vector machine, KNN: K-nearest neighbor

So, their percentage within the dataset is the issue with the identification of smaller types of attacks. All algorithms were able to predict the major categories with at least 99% accuracy.

VII. Conclusion

This work proposes three major ML techniques as binary classifier and multi-class classifiers. The work utilizes these ML techniques as Intrusion Detection System for IoT based attacks detection and the attack's class identification. The work develops an IIDS through utilizing an up-to-date dataset, known as IoT 23. Through a systematic review of recent works, this work finds some gaps that sill not addressed by previous works such as using up to date IoT-based dataset and building a multi-class classification for detecting zero-day attacks. Accordingly, this work addresses those gaps and addresses new objectives. To achieve the targeted objectives, this work proposes a distinguished methodology that starts from data collection, preprocessing steps, training and testing phases, until evaluation of results.

Through implementing the work's methodology, it has been found that most classical ML algorithms could work perfectly as binary and multi-class classification for distinguishing malicious behaviors among IoT-based network packets. Therefore, no need to employ deep learning algorithms for developing IIDS as the structure of the deep learning algorithms is more complex than the classical ML algorithms. Consequently, the time complexity and the space complexity of the developed IIDS with deep learning algorithm expected be increased.

Another conclusion that has been ended through the work implementation is the impact of some preprocessing methods such as SMOTE on the accuracy rate of the developed IIDS. SMOTE method is usually applied on an imbalanced dataset to avoid under and/over fitting of the developed model, and maximizing the accuracy rate of the classifier models. The strange results that have been obtained in this work are the ineffectiveness of the SOMTE method in improving the accuracy rate of the proposed IIDS model as the security rate without SMOTE reached to an excellent level. Although the accuracy of the proposed IIDS showed extraordinary rates, this work suggests investigating more statistical and non-statistical properties of the IoT23 dataset to get more explanations on the ineffectiveness of the SMOTE method for improving the accuracy rate over imbalanced datasets.

References

Abdulla, S.M., Al-Dabagh, N.B., and Zakaria, O., 2010. Identify features and parameters to devise an accurate intrusion detection system using artificial neural network. *International Journal of Computer and Information Engineering*, 4, pp.1553-1557.

Alfarshouti, A.M., and Almutairi, S.M., 2022. An intrusion detection system in IoT environment using KNN and SVM classifiers. *Webology*, 19, pp.130-143.

Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S.A., Elaziz, M.A., Al-Qaness, M.A.A., and Jilani, S.F., 2022. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT. *Sensors*(*Basel*), 22, p.2697.

Bhandari, A., 2020. *Everything you Should Know about Confusion Matrix for Machine Learning', Analytics Vidhya*. Available from: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model [Last accessed on 2022 Aug 26].

Chen, K., Zhang, S., Li, Z., Zhang, Y., Deng, Q., Ray, S., and Jin, Y., 2018. Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice. *Journal of Hardware and Systems Security*, 2, pp.97-110.

Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B., and Buchanan, W.J., 2021. An experimental analysis of attack classification using machine learning in IoT networks. *Sensors* (*Basel*), 21, p.446.

Fatayer, T.S., and Azara, M.N., 2019. IoT secure communication using ANN classification algorithms. In: *2019 International Conference on Promising Electronic Technologies (ICPET)*. IEEE, New Jersey, pp.142-146.

Garcia, S., Parmisano, A., and Erquiaga, M.J., 2020. I*oT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic*. Stratosphere Lab., TechRep, Praha, Czech Republic.

Giusto, D., Iera, A., Morabito, G., and Atzori, L., 2010. *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*. Springer Science and Business Media, Berlin.

Gopi, R., Sathiyamoorthi, V., Selvakumar, S., Manikandan, R., Chatterjee, P., Jhanjhi, N.Z., and Luhach, A.K., 2021. Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things. *Multimedia Tools and Applications*, pp.1-19.

Hanif, S., Ilyas, T., and Zeeshan, M., 2019. Intrusion Detection in IoT using Artificial Neural Networks on UNSW-15 Dataset. In: *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life using ICT & IoT and AI (HONET-ICT)*. IEEE, New Jerssey, pp.152-156.

Ho, E.S.L., 2022. Data security challenges in deep neural network for healthcare IoT systems. In: *Security and Privacy Preserving for IoT and 5G Networks*. Springer, Berlin.

Iman, A.I.N., 2022. *Low Rate DDOS attack Detection using KNN on SD-IOT*. Universitas Muhammadiyah Malang, Indonesia.

Islam, U., Muhammad, A., Mansoor, R., Hossain, M.S., Ahmad, I., Eldin, E.T., Khan, J.A., Ur Rehman, A., and Shafiq, M., 2022. Detection of Distributed Denial of Service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models. *Sustainability*, 14, p.8374.

Kareem, M.I., and Jasim, M.N., 2022. Fast and accurate classifying model for denial-of-service attacks by using machine learning. *Bulletin of Electrical Engineering and Informatics*, 11, pp.1742-1751.

Kumari, K., and Mrunalini, M., 2022. Detecting denial of service attacks using machine learning algorithms. *Journal of Big Data*, 9, p.56.

Lee, S.J., and Zeng, X., 2008. A Modular Method for Estimating Null Values in Relational Database Systems. In: *2008 Eighth International Conference on Intelligent Systems Design and Applications*. IEEE, New Jerssey, pp.415-419.

Li, W., Yi, P., Wu, Y., Pan, L., and Li, J., 2014. A new intrusion detection system based on KNN classification algorithm in wireless sensor network. *Journal of Electrical and Computer Engineering*, 2014, p.240217.

Li, Z., Rios, A.L.G., and Trajković, L., 2021. Classifying Denial of Service Attacks Using Fast Machine Learning Algorithms. In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, New Jerssey, pp.1221-1226.

Majeed Alhammadi, N.A., 2022. Comparative study between (SVM) and (KNN) classifiers by using (PCA) to improve of intrusion detection system. *Iraqi Journal of Intelligent Computing and Informatics* (*IJICI*), 1, pp.22-33.

Malik, M., and Dutta, M., 2022. Security Challenges in Internet of Things (IoT) integrated power and energy (PaE) systems. In: *Intelligent Data Analytics for Power and Energy Systems*. Springer Nature, Berlin, pp.555-566.

Mehmood, A., Khan, A.N., and Elhadef, M., 2022. HeuCrip: A malware detection approach for internet of battlefield things. *Cluster Computing*, 26, pp.977-992.

Nagisetty, A., and Gupta, G.P., 2019. Framework for Detection of Malicious Activities in IoT Networks using Keras Deep Learning Library. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, United States, pp.633-637.

Nawir, M., Amir, A., Yaakob, N., and Bi Lynn, O., 2016. Internet of Things (IoT): Taxonomy of Security Attacks. In: *2016 3rd International Conference on Electronic Design (ICED)*. IEEE, United States, pp.321-326.

Garcia, S., Parmisano, A. and Equiaga, M.J. (2020). *IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic* (Version 1.0.0) [Data Set]. Europe: Zenodo. http://doi.org/10.5281/zenodo.4743746

Picon Ruiz, A., Gila, A.A., Irusta, U., and Huguet, J.E., 2020. Why deep learning performs better than classical machine learning? *Dyna Ingenieria E Industria*, 95, pp.119-122.

Radivilova, T., Kirichenko, L., Ageiev, D., and Bulakh, V., 2019. Classification Methods of Machine Learning to Detect DDoS Attacks. In: *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* (*IDAACS*). IEEE, United States, pp. 207-210.

Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K.K.R., and Parizi, R.M., 2020. An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic. *IEEE Internet of Things Journal*, 7, pp.8852-8859.

Sanmorino, A., 2019. A study for DDOS attack classification method. *Journal of Physics: Conference Series*, 2019, p.012025.

Sewak, M., Sahay, S.K., and Rathore, H., 2018. Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection. In: *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (*SNPD*). IEEE, United States, pp.293-296.

Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R., and Sakurai, K., 2019. A Sequential Scheme for Detecting Cyber Attacks in IoT Environment. In: *2019 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conferenced on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, United States, pp.238-244.

Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., and Guizani, M., 2022. FEDGAN-IDS: Privacy-preserving IDS using GAN and federated learning. *Computer Communications*, 192, pp.299-310.

Tabassum, A., Erbad, A., Mohamed, A., and Guizani, M., 2021. Privacy-preserving distributed IDS using incremental learning for IoT health systems. *IEEE Access*, 9, pp.14271-14283.

Weller-Fahy, D.J., Borghetti, B.J., and Sodemann, A.A., 2014. A survey of distance and similarity measures used within network intrusion anomaly detection. *IEEE Communications Surveys and Tutorials*, 17, pp.70-91.

Wongvorachan, T., He, S., and Bulut, O., 2023. A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information*, 14, p.54.