International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218517 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

Intrusion Detection Systems for IoT Attack Detection and Identification Using Intelligent Techniques

Trifa Sherko Othman
Department of Software Engineering, Koya University, University Park, Danielle Mitterrand Boulevard, Koya KOY45, Kurdistan Region, Iraq.
trifa.sherko@koyauniversity.org

Saman Mirza Abdullah
Department of Software Engineering, Koya University, University Park, Danielle Mitterrand Boulevard, Koya KOY45, Kurdistan Region, Iraq.
saman.mirza@koyauniversity.org

Received: 28 December 2022 / Revised: 02 February 2023 / Accepted: 10 February 2023 / Published: 26 February 2023

Abstract – The Internet of Things (IoT) and its connected objects have resource limitations, which lead to weak security concerns over the IoT infrastructures. Therefore, the IoT networks should always be attached with security solutions. One of the promising security solutions is intrusion detection system (IDS). Machine Learning (ML) algorithms become one of the most significant techniques for building an intelligent IDS based model for attack classification and/or identification. To keep the validation of the ML based IDS, it is essential to train the utilized ML algorithms with a dataset that cover most recent behaviors of IoT based attacks. This work employed an up-to-date dataset known as IoT23, which contains most recent network flows of the IoT objects as benign and other flows as attacks. This work utilized different data preprocessing theories such data cleansing, data coding, and SMOT theory for imbalanced data, and investigating their impact on the accuracy rate. The study's findings show that the intelligent IDS can effectively detect attacks using binary classification and identify attacks using multiclass classification.

Index Terms – IoT Networks, Intrusion Detection, IDS, IoT Attack, Machine Learning, Attack Detection.

1. INTRODUCTION

Internet of Things (IoT) is a recent development that has dominated network technology. With this technology, connected devices and objects can independently communicate with one another, with or without human consent. [1]. Although the IoT has improved the scalability indicator of networks, new challenges have been observed and measured, particularly in those that relate to the security of networks or connected devices, some of which are related to energy consumption [2] and others to the system environment of the IoT applications [3]. There are various reasons to make IoT-connected devices more susceptible to attacks and intrusions. Most devices are constrained by resources like memory and energy. The security tools were unable to function effectively due to these restrictions. Inability to create a security standard for connected objects since they are produced by several manufacturers and companies is another cause of vulnerability. These factors all lead to a growth in IoT network attacks and threats, as well as an expansion of attack surfaces and vulnerabilities. [4]. Due to the above, most of the researchers are focusing on and addressing these unsolved issues, and they are investigating machine learning (ML) tools and approaches for developing intelligent classifier models to classify normal from malicious behaviors of packets that flow through the IoT networks. [5, 6].

Machine learning algorithms are very sufficient at identifying and classifying behaviors. They numerically depend on some predefined characteristics or behaviors, then transfer those characteristics to a class among the available classes. [7]. To develop any classifier models, ML techniques should follow two phases, which are training and testing. Although both phases are important for getting a perfect classifier model, the training phase needs more work and has to 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 affects negatively on the accuracy rate of the ML classifiers [8]. Therefore, one of the questions that this work wants to answer is about the possibility of improving the accuracy rate of classifier models through preprocessing steps. This question has been investigating in the security field of IoT networks, especially, while a new training dataset such as (IoT-23) will be employed [9].

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 [10]. Since then, many works have been

RESEARCH ARTICLE

conducted and many investigations have been published [11-13]. Although the methodology that followed by these works and many other works depends on employing one of the ML techniques for building the classifier model then testing and comparing their accuracy rates, many influenced parameters on the accuracy rate have not been investigated yet. Therefore, research projects in this field could be considered as not saturated. Besides that, most of the conducted research projects depended on some training datasets that were already collected though monitoring non-IoT networks [14] [15]. Therefore, among the aims that this work want to focus is evolving a most recent collected dataset for attackers over IoT networks and investigating many preprocessing techniques with comparing many ML algorithms to find out the best and more efficient classifier models that could be used as attack and malicious detection over IoT networks.

1.1. Work Contributions

As mentioned in the section 1, the main objective of this work is building an ML based classifier model that can detect and identify attacks through analyzing the IoT based networks packets. The main contributions of the work could be summarized as below:

- The work focuses on the most recent dataset that purely related to the IoT based attacks' behaviors excluding behaviors of the traditional networks.
- The work focuses on analyzing as much attack as possible through training the proposed ML model. The focused type of attacks are up-to-date attacks and mostly related to IoT based networks.
- Few of works were conducted research projects on analyzing the IoT based behaviors using ML based classifier models as binary and multiclass calcification.

1.2. Problem Statements

The main problems that addressed by this work could be summarized as below:

- The work focuses on specific dataset on IoT packet activities. Most previous works used general datasets that include packets for IoT networks and normal networks.
- The work focuses on attack classification and identification. Most works utilized ML algorithms for attack classification.
- The work investigated detecting zero-day attacks, which has not been mentioned in most previous works.

The rest of the paper has been organized as section 2 which covers the most relevant works in the field of ML as IoT based attack classification. Section 3 is work methodology, which presents most steps that this work use them from building ML models. The experimental part in section 4 covers the training and testing the proposed ML models and evaluated the obtained results. Finally, the conclusion of the entire work has been presented in section 5.

2. RELATED WORKS

During the review process, it has been found that classifying attackers over IoT-Networks depends on a variety of orientations, such as machine learning techniques, dataset types and versions, some preprocessing techniques, and performance indicators. The orientation that covers the type of machine learning 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 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 classed according to the performance index measures. In the subsequence section, many articles have been reviewed based on these study orientations. Expanding the scalability of networks makes connection of new devices to the Internet or to IoT based networks becomes easier than before. Such expansion makes networks very important for daily life and increases the capability of connecting more devices. However, this expansion increases the number of the cyber-attacks over the networks as well, especially over the IoT-based networks as such networks 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 machine learning algorithms to build intelligent detection models that can classify new patterns of attacks after learning some 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 algorithms is known as classical or conventional algorithms, however there are some other techniques known as deep learning algorithms [16, 17]. Both are utilized in different works as attacks' classification or identification.

2.1. Artificial Neural Network (ANN)

The first ML algorithm, which could be considered as a most famous, is known as Artificial neural network (ANN). The ANN algorithm has been utilized by [18] to build an IDS. The author of this work argued that building an ANN model to detect different type of attacks are not sufficient. Instead, the work proposed a sequential ANNs in which for each type of attack an ANN will be responsible for detecting. Although the paper showed good and high accuracy rate, such model needs

RESEARCH ARTICLE

to be updated when a new type of attacks or zero-day attacks will be detected. ANN is considered as a supervised learning algorithm that could be utilized as classifier model. This fact has been used in [19] to build an attack detection over IoT networks. The work showed that results of a 10-fold cross validation reaches to 84%, which somehow is not good enough. Moreover, the version of the datasets that have been utilized for training the 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 networks has been proposed by [20]. The work argued that IoT 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). This means behaviors of traditional networks have been analyzed and the ANN model cannot be tested with recent behaviors of IoT based attacks. There are many recent works that focused on the ANN based attack classification [21-23] to classify attacks over IoT based networks. However, a prat of them focused only on one type of attack, other works focused on many types of attack, but they utilized some old version datasets.

2.2. 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 [22]. Many recent works utilized KNN as classifier model for detecting attacks over IoT networks, however, based on the best knowledge we have the work [24] was the first to use the KNN for attacks that penetrating IoT 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 network attacker is [25]. The work proposed the KNN algorithm and argued detecting DDoS attacks over IoT network with minimum consuming of energy. Although the work presented 99% as accuracy rate, the test of the work simulated in SDN environment and focused only on one type of attacks over IoT networks. Another recent work that utilized KNN for classifying IoT attacks has been trained with Bot-IoT dataset [26]. The work also presented some taxonomy on the IoT based attacks based on the expected layers that an IoT networks using them during packet communication among connected devices. Another recent work [27] focused on the IoT based attacks considering banking systems as an environment case. The work showed that KNN can detection malicious activities up to 98.7%. The work only focused on DDoS attack detection. Another work that utilizing KNN [28] was depended on adaptive 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 future for detection models. However, they work showed their future work is making the proposed model to detect phishing attack. This means that single detection attack always needs to be updated when new attacks have been countered. Therefore, one of the objectives that addressed by this proposed work is to include most recent type of attacks that penetrating IoT networks.

2.3. 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 the SVM to build an attack detection system over IoT network has depended on Bot-IoT dataset [26]. The work made a comparison between the results that have been obtained from the proposed SVM model with another type of detection model that designed using KNN classifier algorithm. More recent works have utilized SVM as detection method [27]. 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 makes bank servers out of services. The work [27] utilized banking dataset for training the suggested algorithms. Results of the work showed that accuracy with 99.8% could be obtained with SVM. Most datasets that used for training the attack detection have complex dimensionality. Therefore, most works depend on a process called feature selection for reducing the dimensionality size of the training dataset. A most recent work [29] 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 old version of the intrusion behavior dataset, which knows as NLS-KDD dataset and contains 41 attributes. Another work compared the performance of the SVM with Decision tree on two types of attacks (DDoS and Injection). The work proposed an intrusion detection system for attacks over IoT networks in smart city applications. The 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 is 98%. The summary of the reviewed works in the sub-sections 3.1, 3.2 and 3.3 could be illustrated in Table-1, which somehow summarizes the difference between most the reviewed works and this work.

For the Table-1, it could be easily seen that this work investigates three common ML algorithms (ANN, KNN, and SVM) and utilized the most recent dataset that specified for IoT communication and networks. From another side, this work tests all utilized algorithms with binary and multiclass classification.

RESEARCH ARTICLE

Table 1 The Work Review Summary

Reference	ML Tools	Datasets	Binary or Multiclass	Number of Attacks
[18]	ANN	N-BaIoT	Multi.	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 DDoS	1
(Islam et al. 2022	SVM	Bot-IoT	Binary DDoS	1
majeed Alhammadi 2022	STV DT	NLS-KDD	Multi.	2
This work	ANN KNN SVM	IoT-23	Binary and Multi	8

3. MATERILAS 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. Figure-1 shows the framework of this project.

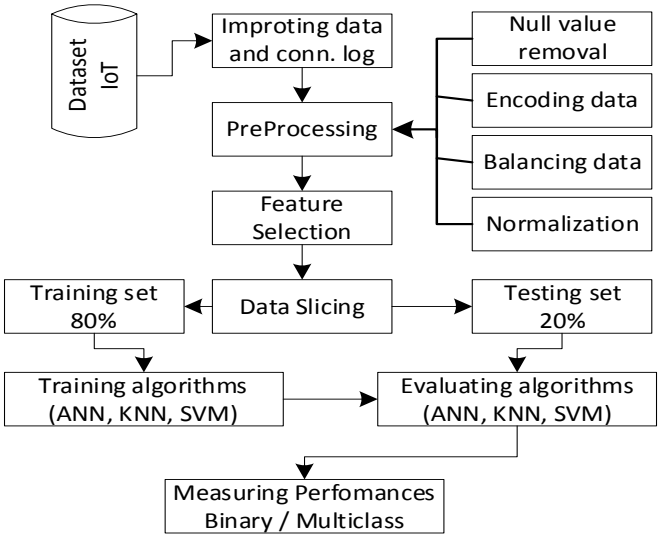

Figure 1 The Framework of the IoT Attack Classification

3.1. Dataset

The IoT23 dataset has been utilized by this work for training and testing, and it has been imported form [30]. The dataset is collected from an IoT network traffic that represents communication behaviors of connecting of three scenarios for benign IoT device traffic and twenty malware scenarios run in IoT devices. The dataset has 21 features / attributes. The last feature is the label, and the dataset is considered as multi-labels and each label may has different attack class. As an example, there are two different labels of attack (C&C and PartOfAHorizontalPortScan), however a class of attack may come with lable (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-2 shows the name of the available classes and the number of each class's observations.

- 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.
- Benign: is a device which no suspicious or malicious activities detected from it flow over network
- 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.
- DDoS is a Distributed Denial of Service attack that an infected device lunch a malicious activity to penetrate another device.
- FileDownload: is encountered when a file is being downloaded to an infected device.
- HeartBeat: with this attack the track of the infected host by the C&C server will be sent through a packet.
- Mirai: the connections have characteristics of a Mirai botnet.
- Okiru: the connections have characteristics of a Okiru botnet.
- PartOfAHorizontalPortScan: a horizontal port scan has been lunched by infected device to gather information for performing further attacks.
- Torii: the connections have characteristics of a Torii botnet.

The imported dataset has a size of 8.8 GB. The dataset has been distributed over 23 folders; three of them were representing the benign datasets and the rest of 20 folders were representing the malicious activates over IoT 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) containing the flow activities.

RESEARCH ARTICLE

Files in all folders focused on same number of features, which are 22 features, including the target label feature, as shown in the table 3.

Table 2 Number of Flows for Each Attack Class in the Dataset

N.	Label	Flows
1	Benign	30,864,692
2	Attack	9,398
3	DDoS	19,538,713
4	PartOfAHorizontalPortScan	213,852,924
5	PartOfAHorizontalPortScan-Attack	5
6	Okiru	60,990,708
7	Okiru-Attack	3
8	FileDownload	18
9	C&C	21,995
10	C&C-HeartBeat	33,673
11	C&C-FileDownload	53
12	C&C-HeartBeat-Attack	834
13	C&C-HeartBeat-FileDownload	11
14	C&C-PartOfAHorizontalPortScan	888
15	C&C-Torii	30
16	C&C-Mirai	2
Total		325,313,947

Table 3 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

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

3.2. Data Preprocessing

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

- Removing the null values and features with zero impact.
- Coding and encoding

RESEARCH ARTICLE

- Data balancing.
- Normalization.

There are two features in the imported dataset having null value inside. Those two features are local_orig and local_resp of connection types (No. 13 and 14 in the table 3). Both features have been removed as all cells of these two-feature containing null value. The process of the removing covers some other features as well. There are many features having zero impact on the classification process, which are (No. 1, 2). tunnel parents is another feature in the dataset that is empty for all records such as (No. 21) and history is another feature which we decided to delete because it is just the history of conn_state. This work removed all these features as shown in Table 5. 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 but we did not consider them as missing value we put them as a special character to be encoded in encoding phase, for example in Service variable we replaced all (-) values to (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 [31]. Finally, the removing process also covered the duplicated observations. The output of this process reduced 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 6 categorical features after dataset cleaning that must 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 machine learning algorithms.as this study implements three classifiers (KNN, SVM, and ANN), we require two forms of Label-Encoding for identification. Ordinal encoding is used for (KNN and SVM) classifiers (as indicated in Table 4), whereas 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 3 categorical features of IoT 23 dataset (Protocol, Service and conn-state) encoded by using frequency encoding, in this method, each value in a categorical feature is 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, they encoded to numerical format by using IP Splitting method, this method divides the octets of an IP address into four distinct numbers, which are then assigned to four distinct variables. We encoded both the source and destination IP addresses, resulting in the creation of eight new variables. The two IP address variables in 32-bit address format were then removed.as shown in Figure 2.

Table 4 Ordinal Encoding of the Labels of IoT 23 Dataset for KNN and SVM Models

Labels	Encoded Label
C&C	1
C&C-HeartBeat-Attack	2
C&C-PartOfAHorizontalPortScan	3
Attack	4
C&C-HeartBeat	5
DDoS	6
Okiru	7
PartOfAHorizontalPortScan	8
PartOfAHorizontalPortScan-Attack	9
Okiru-Attack	10
FileDownload	11
C&C-FileDownload	12
C&C-Heartbeat-FileDownload	13
C&C-Torri	14
C&C-Mirai	15

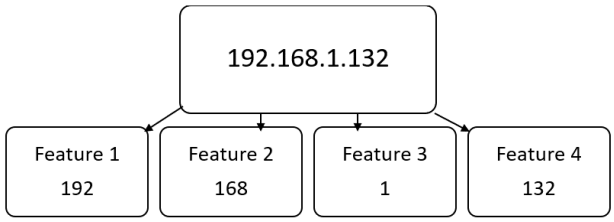

Figure 2 IP Splitting Example

The process of balancing IoT-23 dataset in this work achieved through two phases. At the first phase, this work reduces the gap that exist among the attack's classes in the number of observations they have. There are classes having millions of observations (number 1, 3, 4, 6 in the Table 1), some classes having thousands of observations (2, 9, and 10 in the Table 1),

RESEARCH ARTICLE

and others are having less than 1000 observations (5,7,8, 11,12, 13,14, 15 and 16 in the Table 1). Figure 3 clearly As the first phase of balancing the dataset, this study randomly picks 2000 samples from the first and second groups of attack's class in the IoT-23 dataset.

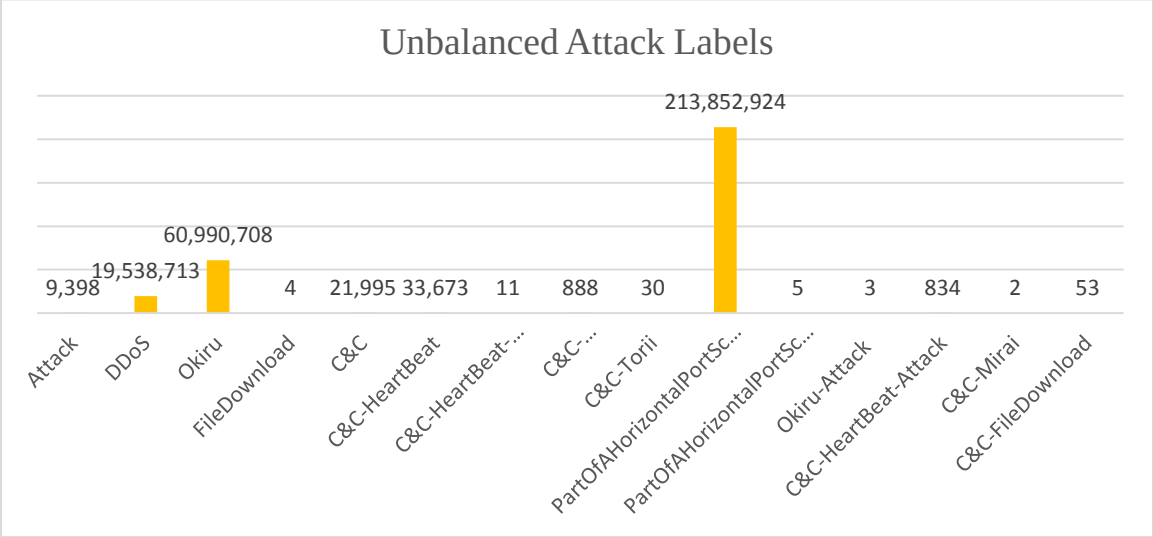

Figure 3 The Unbalanced IoT-23 Structure

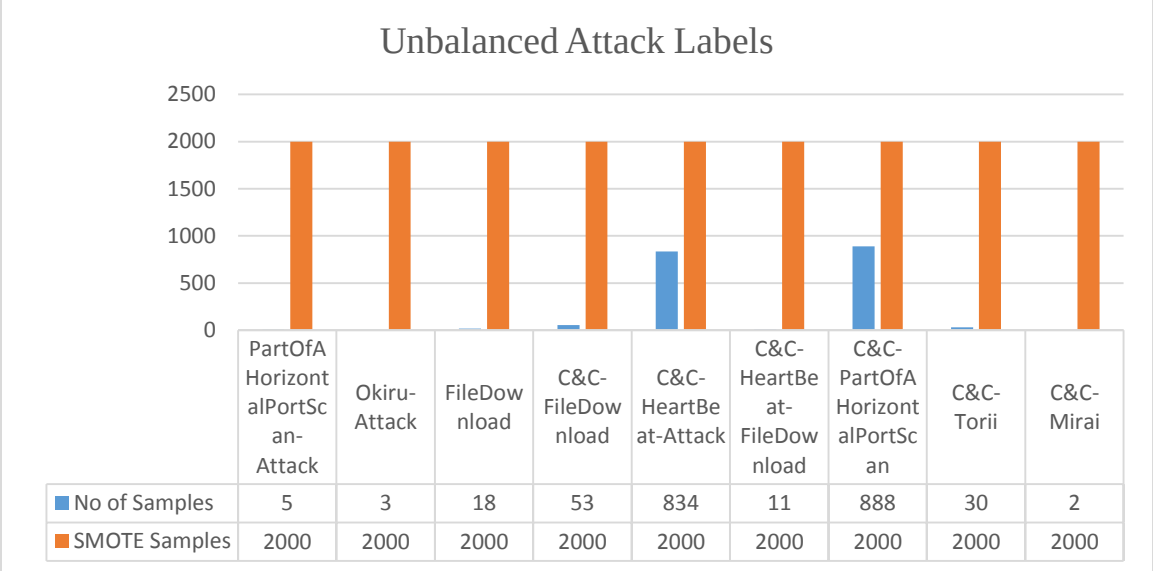

Figure 4 Phase Two Output of Unbalanced IoT-23 Structure

The second phase is applying the SMOTE algorithm to the third group of attack's class so that observations in all attack's classes reaches 2000. Figure 4 shows the imbalanced classes of the dataset before and after processing the dataset under the SMOTE algorithm.

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

3.3. Feature Selection

Feature Selection could be defined as a method that can reduce the number of attributes that utilized by the proposed model through selecting only relevant feature(s) and getting

RESEARCH ARTICLE

minimizing of noise in dataset [32]. There are six features already have been excluded before feature selection process. Table 4 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.

Table 5 Excluded Features

Feature	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.

Figure 5 Correlation Coefficient Graph of IoT 23 Dataset Features

id_orig1	id_orig2	id_orig3	id_orig4	id_orig_pF	id_resp1	id_resp2	id_resp3	id_resp4	id_resp_pF	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

Figure 6 A sample of the Obtained Dataset Through Preprocessing

RESEARCH ARTICLE

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 canceled"[33] [32]. 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). Figure 5 shows the correlation status among the attributes or features, and Figure 6 shows a sample of the dataset after the preprocessing steps.

3.4. Data Slicing

This process is about splitting the dataset into two groups, training part of the dataset and testing part of the dataset. Although the obtaining parts will be directly used and fed to the ML classifier model, this process still be considered as a step of preprocessing activities. This work allocates 20% of the dataset and keeps it for testing phase 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 remaining 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 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 set and same is true for the training sets.

3.5. Performance Indicators

Figure 7 shows details of a typical confusion matrix [34]. 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.

		Predicted Class: Positive	Predicted Class: 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** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure 7 Typical Confusion Matrix with Performance Indicators

4. EXPERIMENTAL EVALUATION

This work utilized three major ML algorithms named ANN, SVM, and KNN. The aim of this work is building a binary and multi-class classification using those ML algorithms. The experimental evaluation in this work depends on *k-fold* method, by which, the dataset will be divided into 5 partitions, each time, a part will be used for testing and the remaining nine prats used for training.

4.1. ANN Based Classification

Artificial Neural Network (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 arranged to form three types of layer input, hidden, and output layers. Nodes at each layer have different functionalities. At input layer, takes information provided and passes it onto hidden layer. The core computation of the ANN is occurred in the hidden layer. The results will be passed to the output layer. For supervised ANN, the expected output and desired output will be considered for compute the accuracy of the training. 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 obtained. The typical structure of an ANN is shown in the Figure 8.

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 dataset sample that shown in the Figure 6, 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 sets the number of hidden layers on one and the nodes in the layer are 10 nodes. Figure 9 shows the ANN structure that designed by this wok for binary classification of attacks.

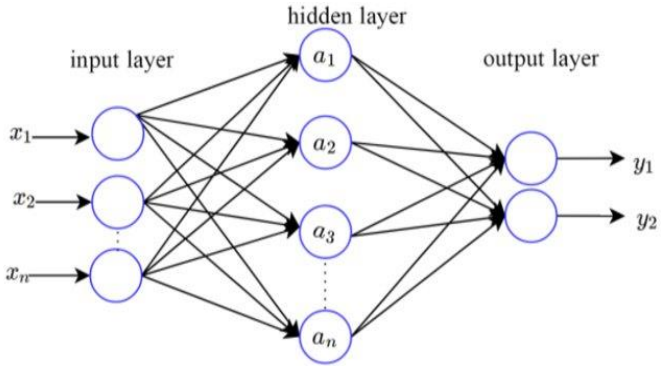

Figure 8 Typical Structure of ANN

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218517 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

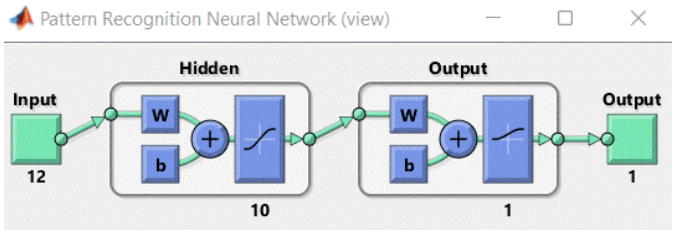

Figure 9 The ANN Based Binary Classification

The proposed ANN just required 45 epochs for getting perfect training. 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 Figure 10. The result of the testing is 99 %.

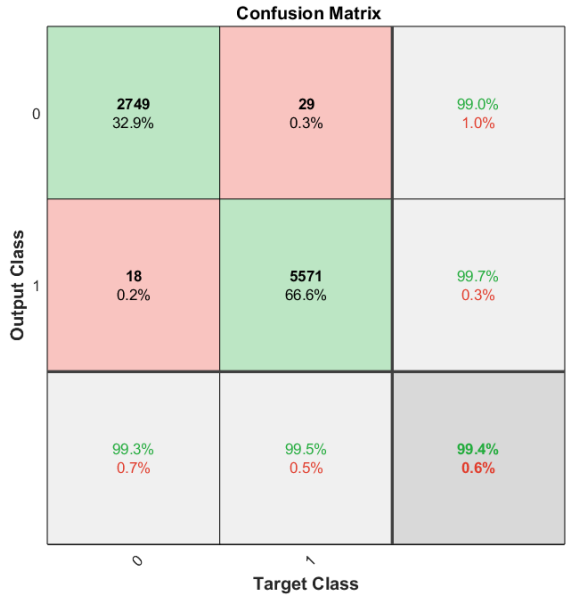

Figure 10 Confusion Matrix for ANN Based Binary Classification (SMOTE)

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 the Figure 11and with the same number of epochs.

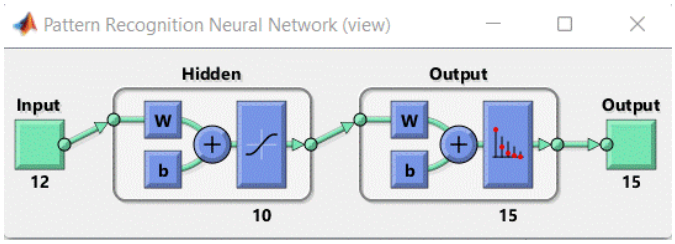

Figure 11 The ANN Based Multi-Class Classification

The accuracy that obtained through the multi-class classification, as shown in the Figure 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.

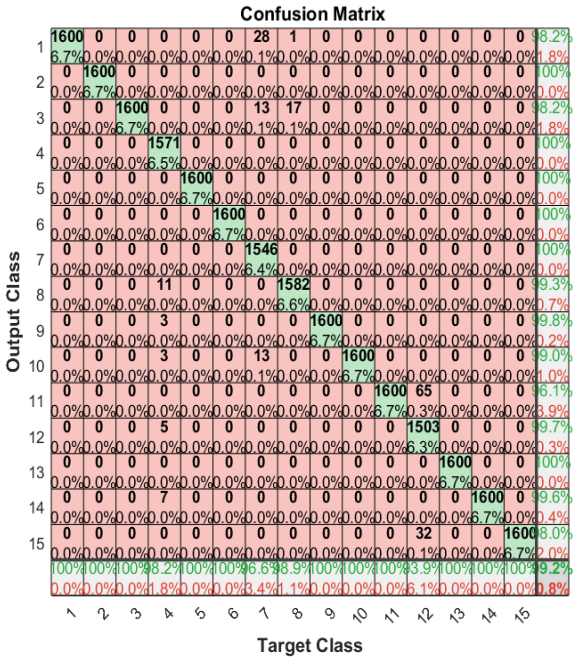

Figure 12 The Confusion Matrix of ANN Based Multi-Class Classifier (SMOTE)

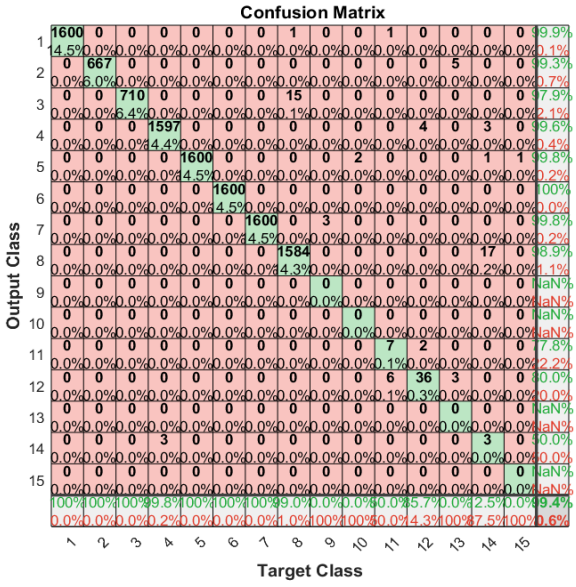

Figure 13 The Confusion Matrix of ANN Based Multi-Class Classifier (without SMOTE)

RESEARCH ARTICLE

4.2. ML Comparison Results

This work utilizes another two major ML techniques to compare the performance of the ANN based attack classification and identification. These techniques are KNN and SVM. This work compared the ANN based model when it works as binary classifier and when it works as multi-class classifiers. The comparisons result of the binary classification is shown in the Figure 14 and Figure 15, and the results of multiclass classification is shown in the Table 6 and Table 7. 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 % to 99.72 %. SMOTE has a greater impact on multiclass classification than it has on binary classification.

There is one fact 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. As a consequence, we tested the models using F1-Score as well, and the influence of SMOTE appeared significantly, as shown in Table 7. 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 Figure 13. The results of the SMOTE dataset then solved the problem, as illustrated in Figure 12.

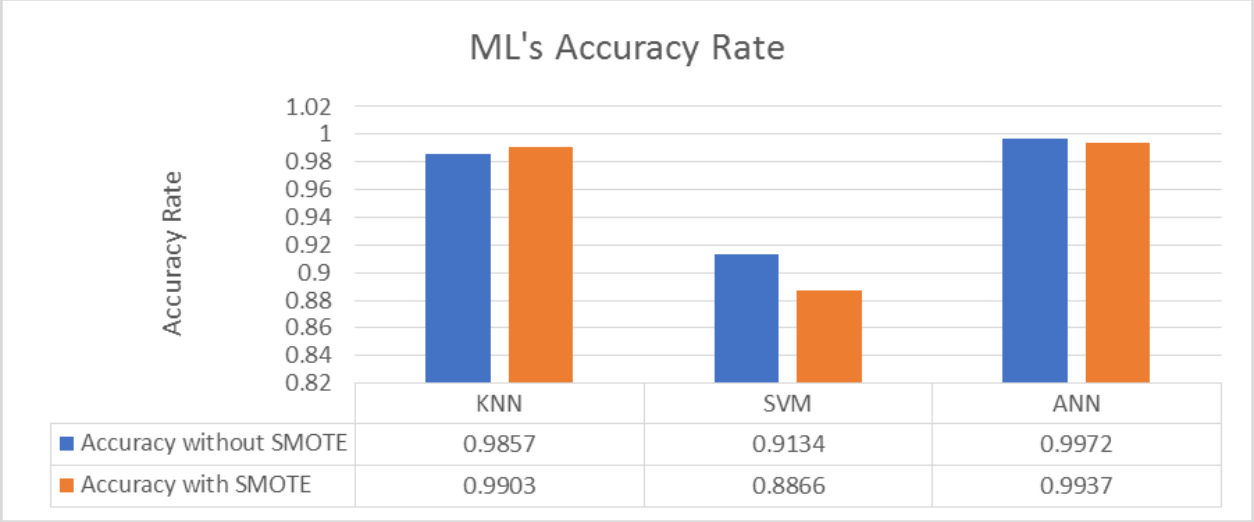

Figure 14 Compression of the ML's Accuracy

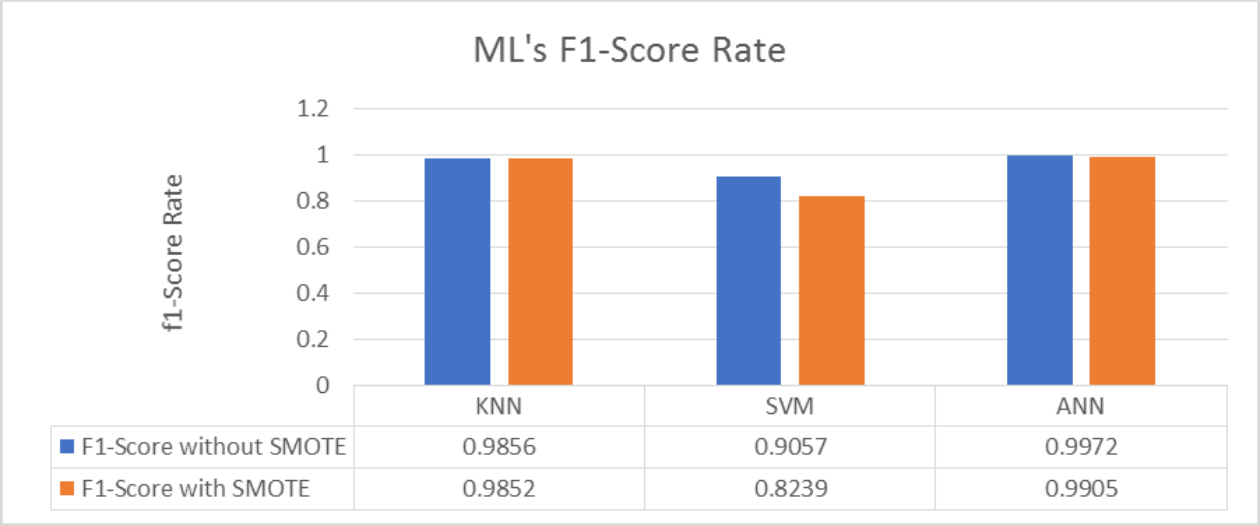

Figure 15 Compression of the MLs F1-Score

RESEARCH ARTICLE

Table 6 The Identification Accuracy Rate of ML Techniques for Each Attack

	KNN		SVM		ANN	
Labels	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

Table 7 The Identification F1-Score Rate of ML Techniques for Each Attack

	KNN		SVM		ANN	
Labels	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

RESEARCH ARTICLE

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

5. CONCLUSION

This work tests three major ML techniques as binary classifier and multi-class classifiers for detecting IoT based attacks. The major points that have been concluded by this work could be summarized as:

1. Throughout the review process, it has been concluded that many gaps still need investigation, such as designing IDS for IoT infrastructure based on analyzing packets that sniffing from IoT networks. This work utilized an up-to-date dataset, known as IoT23, throughout the training and testing processes.
2. Moreover, the impact of some prepressing steps agist the accuracy rate has not been investigating such as SMOT theory.
3. Regarding the respond of ML techniques against some preprocessing methods, it has been concluded that each ML technique has unique respond, and the way of respond changes based on the mathematical concept of each ML technique.
4. The proposed ML techniques can actively detect known attacks, and to good extend they can detect zero-day attacks. However, the utilized techniques cannot identify such kind of attacks, as they are not introduced to the detection model. However, the proposed models could identify them as one of the known attacks as there is behavior similarity between the unknown attacks and known attacks.

REFERENCES

[1] Nagisetty, A. and G.P. Gupta. 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). 2019. IEEE.
[2] Malik, M. and M. Dutta, Security Challenges in Internet of Things (IoT) Integrated Power and Energy (PaE) Systems. Intelligent Data Analytics for Power and Energy Systems, 2022: p. 555-566.
[3] Ho, E.S., Data Security Challenges in Deep Neural Network for Healthcare IoT Systems, in Security and Privacy Preserving for IoT and 5G Networks. 2022, Springer. p. 19-37.
[4] Nawir, M., et al. Internet of Things (IoT): Taxonomy of security attacks. in 2016 3rd international conference on electronic design (ICED). 2016. IEEE.
[5] Chen, K., et al., Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice. Journal of Hardware and Systems Security, 2018. 2(2): p. 97-110.
[6] Saharkhizan, M., et al., An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic. IEEE Internet of Things Journal, 2020. 7(9): p. 8852-8859.
[7] Radivilova, T., et al. 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). 2019. IEEE.
[8] Sanmorino, A. A study for DDOS attack classification method. in Journal of Physics: Conference Series. 2019. IOP Publishing.
[9] Parmisano, A., S. Garcia, and M. Erquiaga, Aposemat IoT-23: A labeled dataset with malicious and benign IoT network traffic. Accessed: Jul, 2020. 31: p. 2020.
[10] Giusto, D., et al., The internet of things: 20th Tyrrhenian workshop on digital communications. 2010: Springer Science & Business Media.
[11] Kareem, M.I. and M.N. Jasim, Fast and accurate classifying model for denial-of-service attacks by using machine learning. Bulletin of Electrical Engineering and Informatics, 2022. 11(3): p. 1742-1751.
[12] Kumari, K. and M. Mrunalini, Detecting Denial of Service attacks using machine learning algorithms. Journal of Big Data, 2022. 9(1): p. 1-17.
[13] Li, Z., A.L.G. Rios, and L. Trajković. Classifying Denial of Service Attacks Using Fast Machine Learning Algorithms. in 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2021. IEEE.
[14] Tabassum, A., et al., FEDGAN-IDS: Privacy-preserving IDS using GAN and Federated Learning. Computer Communications, 2022. 192: p. 299-310.
[15] Tabassum, A., et al., Privacy-Preserving Distributed IDS Using Incremental Learning for IoT Health Systems. IEEE Access, 2021. 9: p. 14271-14283.
[16] PICON RUIZ, A., et al., Why deep learning performs better than classical machine learning? Dyna Ingenieria E Industria, 2020.
[17] Sewak, M., S.K. Sahay, and H. Rathore. 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). 2018. IEEE.
[18] Soe, Y.N., et al. A sequential scheme for detecting cyber attacks in IoT environment. in 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). 2019. IEEE.
[19] Hanif, S., T. Ilyas, and M. Zeeshan. 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). 2019. IEEE.
[20] Fatayer, T.S. and M.N. Azara. IoT secure communication using ANN classification algorithms. in 2019 International Conference on Promising Electronic Technologies (ICPET). 2019. IEEE.
[21] Gopi, R., et al., Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things. Multimedia Tools and Applications, 2021: p. 1-19.
[22] Churcher, A., et al., An experimental analysis of attack classification using machine learning in IoT networks. Sensors, 2021. 21(2): p. 446.

RESEARCH ARTICLE

[23] Mehmood, A., A.N. Khan, and M. Elhadef, HeuCrip: a malware detection approach for internet of battlefield things. Cluster Computing, 2022: p. 1-16.
[24] Li, W., et al., A new intrusion detection system based on KNN classification algorithm in wireless sensor network. Journal of Electrical and Computer Engineering, 2014. 2014.
[25] Iman, A.I.N., LOW RATE DDOS ATTACK DETECTION USING KNN ON SD-IOT. 2022, Universitas Muhammadiyah Malang.
[26] Alfarshouti, A.M. and S.M. Almutairi, An Intrusion Detection System in IoT Environment Using KNN and SVM Classifiers. Webology, 2022. 19(1).
[27] 27. Islam, U., et al., Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. Sustainability, 2022. 14(14): p. 8374.
[28] Aslam, M., et al., Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. Sensors, 2022. 22(7): p. 2697.
[29] majeed Alhammadi, N.A., Comparative study between (SVM) and (KNN) classifiers by using (PCA) to improve of intrusion detection system. Iraqi Journal of Intelligent Computing and Informatics (IJICI), 2022. 1(1): p. 22-33.
[30] Garcia, S., A. Parmisano, and M.J. Erquiaga, IoT-23: A labeled dataset with malicious and benign IoT network traffic. Stratosphere Lab., Praha, Czech Republic, Tech. Rep, 2020.
[31] Lee, S.-J. and X. Zeng. A modular method for estimating null values in relational database systems. in 2008 Eighth International Conference on Intelligent Systems Design and Applications. 2008. IEEE.
[32] Abdulla, S.M., N.B. Al-Dabagh, and O. Zakaria, Identify features and parameters to devise an accurate intrusion detection system using artificial neural network. International Journal of Computer and Information Engineering, 2010. 4(10): p. 1553-1557.
[33] Weller-Fahy, D.J., B.J. Borghetti, and A.A. Sodemann, A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Communications Surveys & Tutorials, 2014. 17(1): p. 70-91.
[34] Bhandari, A. Everything you Should Know about Confusion Matrix for Machine Learning. April 17, 2020 June 14th, 2022 August 26, 2022]; Available from: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model.

Authors

Trifa Sherko Othman received a B.Sc. in Software Engineering from Koya University (KOU) in Koya, Kurdistan Region, Iraq (2015), and is presently pursuing a master's degree. She formerly worked as a Software Engineer in the Koya University Presidency. Her main research interests are in network security, Machine Learning Techniques, Internet of Things (IoT), and Intrusion Detection Systems.

Saman Mirza Abdullha is working as assitnat prof in Department of Software Enigneering at Koya Universit. He got his PhD in FSKTM univesity of Malaya in Malaysia on 2013 in Computer Security. He is memebr in IEEE and ACM associations. His resaerch field is in security of IoT, intrusion detection system, PE malware detection, Machinel learning, data secince and anlysis, and social media vulnerabilities.

How to cite this article:

Trifa Sherko Othman, Saman Mirza Abdullah, "Intrusion Detection Systems for IoT Attack Detection and Identification Using Intelligent Techniques", International Journal of Computer Networks and Applications (IJCNA), 10(1), PP: 130-143, 2023, DOI: 10.22247/ijcna/2023/218517.