Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.036856
Article

Tech Science Press

Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique

Moody Alhanaya and Khalil Hamdi Ateyeh Al-Shqeerat*

Department of Computer Science, College of Computer, Qassim University, Saudi Arabia
*Corresponding Author: Khalil Hamdi Ateyeh Al-Shqeerat. Email: kh.alshqeerat@qu.edu.sa
Received: 13 October 2022; Accepted: 13 December 2022

Abstract: The increasing number of security holes in the Internet of Things (IoT) networks creates a question about the reliability of existing network intrusion detection systems. This problem has led to the developing of a research area focused on improving network-based intrusion detection system (NIDS) technologies. According to the analysis of different businesses, most researchers focus on improving the classification results of NIDS datasets by combining machine learning and feature reduction techniques. However, these techniques are not suitable for every type of network. In light of this, whether the optimal algorithm and feature reduction techniques can be generalized across various datasets for IoT networks remains. The paper aims to analyze the methods used in this research and whether they can be generalized to other datasets. Six ML models were used in this study, namely, logistic regression (LR), decision trees (DT), Naive Bayes (NB), random forest (RF), K-nearest neighbors (KNN), and linear SVM. The primary detection algorithms used in this study, Principal Component (PCA) and Gini Impurity-Based Weighted Forest (GIWRF) evaluated against three global ToN-IoT datasets, UNSW-NB15, and Bot-IoT datasets. The optimal number of dimensions for each dataset was not studied by applying the PCA algorithm. It is stated in the paper that the selection of datasets affects the performance of the FE techniques and detection algorithms used. Increasing the efficiency of this research area requires a comprehensive standard feature set that can be used to improve quality over time.

Keywords: Machine learning; internet of things; intrusion detection system; feature selection technique

1 Introduction

The rapid emergence and evolution of the Internet of Things (IoT) have increased the number of security attacks and associated risks [1]. More effective and efficient management of the environment is needed due to the increasing number of connected devices Fig. 1 shows a simplified architecture of the Internet of Things environment. The growing number of connected devices and the dangers of unauthorized access to the data led governments and businesses to look for new ways to protect them.

Unfortunately, current security methods for IoT networks cannot prevent unprecedented attacks [2]. More effective approaches to improving intrusion detection systems are needed to resolve this problem. These systems are designed to analyze the traffic flows in an IoT network to detect security threats. They are also designed to protect digital assets and confidentiality [3].

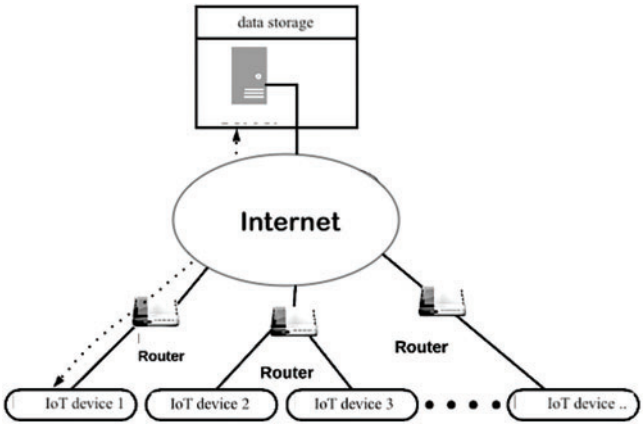

Figure 1: Architecture of IoT environment

An intrusion detection system (IDS) is essential for any network's security, regardless of its configuration [4]. This system provides high levels of security to detect anomalies and prevent threats posed to the network by intrusions or hosts. The primary purpose of IDS is to detect and prevent any potential cyber-attacks and threats and ensure they are adaptable. Anomaly IDS and misuse IDS are the two main types of IDS. The former matches and compares the signatures of incoming traffic against a database of known attacks [5]. Although these tools provide high levels of detection accuracy, they need to identify zero-day and modified threats.

Furthermore, due to the continuously changing techniques and strategies used by attackers, IDSs must be adaptive to detect and prevent threats. Unfortunately, the current method of tuning signatures could be more reliable. Anomaly-based IDSs are designed to overcome the limitations of misuse IDSs by utilizing advanced statistical methods to analyze the behavioral patterns of network traffic. It can be used to detect abnormal events in real-time. Anomaly detection can be performed using various statistical techniques such as machine learning a. Unlike signature IDSs; anomaly IDSs can achieve higher detection rates and accuracy when it comes to zero-day attacks [6]. However, they can also suffer from high false alarm rates due to their ability to identify anomalous traffic. As a result, they can detect attacks that deviate from the safe behavior of the network. Currently, misused IDSs cannot detect the signature of attacks coming through IoT's network [7]. As there is no signature for novel attacks, several techniques have been developed to prevent the recurrence of such attacks. Machine learning (ML) techniques can be used to analyze the data collected by network traffic to achieve this purpose. Therefore, ML can be beneficial in identifying security threats to IoT networks [8].

Various machine learning models with feature reduction algorithms have been developed and implemented to improve the performance of detecting and monitoring the data collected by IoT networks. However, although the results of these models have been obtained, they have yet to be reliable for IoT networks.

Instead of developing an IDS application that relies on machine learning models, researchers focus on improving a specific dataset's performance [9]. This trend has led to a large number of academic

research projects conducted in this field. Although the cost of errors associated with these techniques is higher than in other domains, they still need to be more reliable in a real-world environment [10]. Typically, machine learning models are evaluated on a single dataset with a list of features not collected from an IoT environment. Also, due to their hyper-parameters, they can be improved by implementing them over a specific set of datasets. This paper discovers an overview of the generalizability of the various machine learning models and FE algorithms combined with IDS datasets. It evaluates six ML models: LR, DT, RF, KNN, and Linear SVM. In addition, two FE algorithms, namely, the Principal Component Analysis PCA and GIWRF, have been evaluated and compared to their effects on selected ML using three benchmark datasets: ToN-IoT, UNSW-NB15, and Bot-IoT. The full dataset is also compared and analyzed.

1.1 Machine Learning in IoT Network

ML plays a vital role in extracting complex patterns from data which is a part of artificial intelligence (AI). In particular, it can help identify threats through network traffic. Two steps are included to build a model using this technology: training and testing. In the training stage of model development, the dataset containing malicious network packets includes for training the model. It then tests the model against a set of unlabeled network packets to see if it can identify anomalous traffic and unseen attacks [11]. The most famous supervised learning algorithms are LR, DT, NB, RF, KNN, and SVM.

LR is a linear classification model that is used in predictive analyses. It considers the likelihood that a given output will be an output class between 0 and 1. Despite its ease of implementation, this method may not work well in non-linear situations. The limited broyden–fletcher–goldfarb–shanno (L-BFGS) is an optimization algorithm used to prevent overfitting. Stopping criteria and the regularization strength are taken into account. The maximum number of iterations that can be performed is 100 times.

DT model follows a tree series that represents a high-level feature. End nodes defined each label class, and branches represented Outputs. The labels represent the leaves. This model uses a supervised learning (SL) method to classify and interpret the data. The DT's model is widely used for building complex trees but can also be over-complex compared to the training data. Input features are used to construct binary trees using the CART algorithm. The Gini impurity function is used to measure the quality of the split.

The NB algorithm is a time-efficient method that considers the class-conditional distribution of various features. It follows the "Naive" principle, which assumes that the independence between the inputs is not dependent on the other set. Algorithms set the variance smoothing to 1e-9 as a default value. RF classifier is a DT technique that can classify various data types. It can be used to classify a group of trees that takes thousands of input variables and performs a single vote for the most common class in the data. As opposed to neural networks and support vector machines, the RF requires few parameters to be specified. Its main goal is to create a DT representing the forest. This method is carried out by training a subset of the data composed of around two-thirds of the total. The Out Of Bag (OOB) samples are used for internal cross-validation to evaluate RF's classification accuracy. The RF does not require many resources to perform its task compared to other methods. Also, it is insensitive to parameters and outliers, making it unnecessary to prune the trees manually.

KNN classification algorithm is a widely used machine learning and data mining tool for classification. One of the main reasons it is considered one of the leading methods is its ability to use different distance weighting measures. The K-NN algorithm is used to classify various elements of

a record set, such as distance measures and the number of neighbors. It is a type of lazy classification. For instance, the distance measure considers the distance between two points, while the number of neighbors is considered. The class value of an unknown record is computed by taking into account the training data of its nearest neighbors.

SVM is a supervised learning system that considers the training data and then uses a hyperplane to classify it. The hyperplanes are used to split the training data into two classes. These two classes are represented as lines or planes in a multi-dimensional area. SVM aims to find the maximum space between the support vectors. This model is ideal for detecting intrusions. In addition to being able to classify data, it also handles various other tasks, such as extracting the necessary information from the data. The kernel is a set of mathematical functions that transform the data into a usable form.

1.2 Types of Intrusion Attacks

In an IoT environment, it is essential to identify the different types of attacks encountered and define the different types of attacks. Various research works have been developed to define an advanced intrusion detection system. However, the process is still ongoing to develop an efficient method for identifying a wide range of threats. There are several major attacks associated with the IoT, including scanning attacks, DoS attacks, injection attacks, DDoS attacks, password attacks, XSS attacks, ransomware attacks, backdoor attacks, man-in-the-middle attacks (MITM), generic attacks, exploit attacks, fuzzing attacks, reconnaissance attack, analysis attack, and worms attack.

This paper is structured as follows. Section 2 reviews some research literature. The methodology is described in detail in Section 3. Finally, Section 4 discusses the experimental results and findings.

2 Literature Review

Numerous researchers have explored and analyzed supervised classification techniques in various application areas. In most studies, theory validation or performance evaluation was primarily used to discover a versatile classifier using a dataset [12,13]. Reference [14] conducts a comparison of ML algorithms, and the author applied the most popular classification ML algorithms beside their ensembles on Botnet detection. To strengthen ML algorithms, the author evaluated the ensemble methods known besides testing the three algorithms, neural network, DT, and NB, to see if they provide a developed prediction of Botnet detection. However, one public dataset, CTU-13, was used to evaluate these techniques by measuring the ML running times of each f-measure and its MCC scores. In [15], SVM, RF, ANN, and KNN algorithms were used to detect DDoS attacks in IoT systems. The results of RF were slightly better than those of the other classifiers. Limited feature sets were used in real-time classification to reduce computational overhead and upgrade the system's applicability. Obtained network traffic features were used to train RF to recognize IoT devices correctly. Network traffic data from 17 IoT devices were extracted and manually labeled to evaluate selected algorithms. There is still work to be done on other IDS network data datasets to determine the system's effectiveness. For accurately detecting malicious traffic on computer networks, the authors [16] analyzed two open-source intrusion detection systems, Suricata and Snort. Snort was chosen for further experiments since the authors noted that it had higher detection accuracy. In this study, ML algorithms were used to evaluate such systems. After applying an empirical study with different ML algorithms like SVM, an adaptive plug-in was included to enhance this system's performance. The result shows better detection accuracy was obtained by embedding SVM in the system. (DARPA IDS, NSA Snort IDS Alert Logs, and NSL-KDD IDS datasets) are used to analyze the performance of the detection system. While this study evaluated the proposed approach on a different data set, assessing the scenario using at least one

of the recently collected intrusion detection network datasets is still necessary. In [17], the authors also implement multiple ML algorithms for the detection system for IoT security to suggest a model of a hybrid algorithm from selected an effective ML algorithm using a single data collection. The BoT-IoT dataset was used with 44 practical features selected from some features for the candidate algorithm. The authors have chosen NB, DT, and RF for anomaly traffic detection and malicious identification. In performance evaluation, ML algorithm metrics are used like Accuracy, Precision, Recall, and TP Rate. The experimental results show that the NB ML algorithm is operative for intrusion and anomaly detection in IoT networks. Soe et al. in [18] Orrelated-set thresholding on gain-ratio (CST-GR) algorithm is proposed to select essential features. An ML-based Intrusion detection system is designed and implemented using a new feature selection algorithm. The proposal's performance was examined in detail with Classifiers of Tree-Based to pick up the best option for the system. This study focuses on the feature reduction algorithm by indicating a new technique capable of identifying minimal features for each type of attack. There is no doubt that the results of this study are satisfactory; however, they should be applied to another dataset containing other types of attacks and compared to different feature-reduction algorithms. Jotikabukkana et al. in [19] present a practical, inclusive framework that is easy to follow to protect network data. The authors applied a comparison between using ML and Deep Learning algorithms in terms of detection accuracy. Naïve Bayes, Support Vector Machine (SVM) were used as ML algorithms. The experiment was inducted with feature engineering techniques like Standard Deviation and Principle Component Analysis (PCA). The obtained result shows that each dataset characteristic can fit a specific algorithm that, in turn, reduces the computation time. The study focused on an important aspect, which is the match of the selected algorithm with the selected dataset. However, the researchers in this study implemented the proposed intrusion detection system by using only UNSW-NB15 as the dataset to evaluate its performance. Thus, assessing and developing the system with additional network traffic and attack scenarios is necessary.

The majority of papers found used a single dataset for experiments, which raises questions about the generalizability of their techniques. Furthermore, datasets contain their unique features, resulting in variations in the information presented. As a result, these proposed approaches may have different performances depending on the dataset selected. Consequently, this paper investigates the performance of FE algorithms combined with ML models on various datasets to gain insight into ML-based applications. Furthermore, this will enable us to determine whether the combination of the best datasets can be generalized. In this study, the performance of sex-supervised ML classifiers is analyzed on three benchmark datasets (UNSW-NB15, ToN-IoT, and Bot-IoT datasets) for detecting intrusions along the following dimensions: Feature selection and effect of the Network Design. The classifier analysis is expected to develop a robust and quantitative risk assessment framework for various cyber defense systems.

3 Methodology

Using IoT, physical objects and the Internet are integrated to create solutions related to home automation, industrial processes, health, and the environment. There are many benefits to having Internet-connected devices in our everyday lives, but some security challenges accompany the use of such devices. For example, networks and information systems have been protected against cyberattacks for more than two decades through IDS. Implementing ML-based intrusion detection systems in the IoT is a clear response to increasing cyber-attacks. This paper presents the effects of three FE techniques on six different SL classifiers: DT, RF, KNN, LR, SVM, and NB. The optimal number of features is determined by conducting several experiments on GIWRF and PCA dimensions. Three publicly available NIDS datasets are used to perform the experiments. The overall representation of

the system architecture is shown in Fig. 2. The selected datasets have been processed and analyzed to maximize the efficiency of ML and FE procedures. Then, the classifier predictions and evaluation metrics are determined. The experiments are performed using the Python programming language and the Tensor-Flow outlearn libraries.

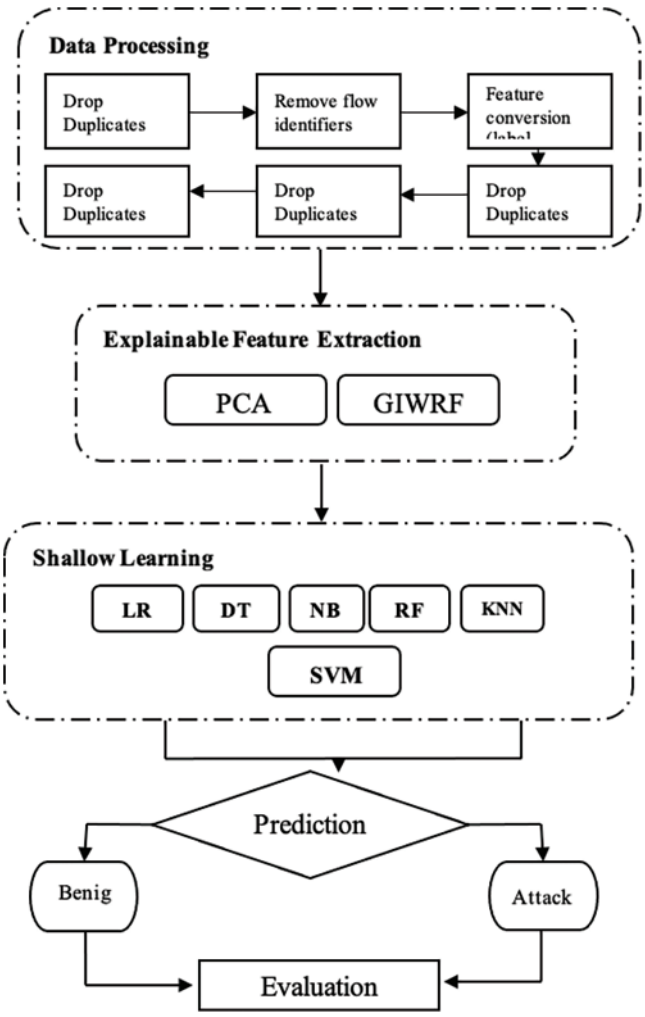

Figure 2: Architecture of the proposed method

3.1 Data Collection

The data selection process is essential to determine the reliability of a machine learning model's evaluation phase. It is also required to ensure that the model's security and privacy are protected. Unfortunately, production networks do not have the necessary infrastructure to generate labeled flows. To ensure that the evaluation phase of a machine learning model is appropriately performed, researchers have created benchmark datasets that are publicly available. The datasets are generated using a virtual network testbed. It mixes synthetic attack traffic with regular network traffic. They are then analyzed using various tools and procedures. A label feature is then added to the datasets to indicate whether a flow is malicious or benign. The type of flow is defined, and its attributes are shared.

The data log between the two end nodes is considered a unidirectional record to classify the packets. Three datasets have been used in this research (ToN-IoT, UNSW-NB15, and Bot-IoT dataset).

A recent release by ACCS in 2019 [20] included a network traffic portion collected over an IoT ecosystem. This portion was mainly used for attack samples. It comprises primarily attack samples with a ratio of 21,542,641 (96.44%) attack flow to 796,380 (3.56%) benign ones, that is, 22,339,021 flows.

In the flow, 44 features are extracted from Bro-IDS. These include backdoor, injection, DoS, distributed DoS (DDoS), and multiple attack settings such as password, ransomware, and Cross-site Scripting (XSS). On-integer features are also included in the flow. These include a service and conn state, a conn state, an SSL version, an SSL cipher, a DNS query, a method, an Orig mime type, and a user agent. The flow's identifiers are ts, DST IP, DST port, and ts. Its Boolean features include adns rejected, adns AA, adns RD, adns RA, SSL resumed, and any weird notice.

A synthetic environment was established in the UNSW cybersecurity lab to create the UNSW-NB15 dataset [21]. In addition to regular traffic in the dataset, 49 features and nine major attacks include analysis, fuzzing, backdoors, exploits, dos, injection, reconnaissance, generic, and worms. Each set of records is pre-split into a training and testing set. The feature identifiers are id, dstip, dport, sport, srcip, ltime, and stime. In addition, non-integer features such as proto, service, and state are included in the dataset. There are 2,218,761 instances of malicious traffic and 321,283 instances of normal traffic in the dataset. It is more balanced (less imbalanced). Compared to the other two datasets, there is a relatively lesser difference in malicious and normal records (normal: malign ratio is approximately 1:0.14).

The BoT-IoT dataset was created by the Cyber Range Lab of the University of NSW's Canberra Cyber. It features a realistic network environment composed of a botnet and normal traffic [22]. Moreover, it includes attacks such as Service Scan, Keylogging, Data exfiltration, DoS, and DDoS attacks. DDoS and DoS attacks can be further organized according to the protocol involved. The dataset is pre-split into two sets of tests and training. There are over 73,360,900 malicious and 9,543 normal traffic instances, resulting in a significant imbalance between malicious and normal traffic that can affect the model's bias.

3.2 Preprocessing Step

The first step in evaluating a machine learning model is to analyze and improve its data preprocessing. This process involves removing duplicate samples and ensuring that the data is stored in a secure and resilient environment. In addition, the flow identifiers are removed to avoid bias in favor of the attacker's end nodes or applications. The non-numeric and string features are then mapped to numerical values using a method known as categorical encoding. These are then used to collect various services and protocols stored in the native string values. ML models that use categorical encoding are designed to perform efficiently with numerical values. One of the main methods used for mapping the features is the label encoding technique. This method takes into account the number of features and adds them to X categories to represent the presence of a certain category. The label encoding technique then maps each feature to an integer. In the following steps, the empty, dash, and infinity attribute values are substituted with 0 to create a numerical-only dataset. The label encoding technique then converts the categorical values in an attribute into an index. For instance, if a protocol has a categorical value of ICMP, then its attribute is converted into a value of TCP. This method improves the performance of the model. When false and true, the values of non-binary features are replaced by one and then 0, respectively. The min-max feature scaling method then brings all the feature values

between 0 and 1, which helps reduce complexity. It also allows the model to assign more weight to certain features due to their nature. Based on the label features, the dataset is divided into two parts, one for training and one for testing.

3.3 Feature Extraction

FE is a process that aims to reduce the number of features or dimensions in a dataset. It does so by extracting relevant and valuable information from the raw data. It then projects these features into a reduced number of features. The two principal algorithms used in this process are PCA and GIWRF.

3.3.1 Principal Component Analysis

The principal component analysis is a process that takes advantage of statistical procedures to extract features from a dataset. It finds the eigenvalues of the various features in the dataset and projects them into a lower dimensional space [23]. The extracted features are referred to as principal components. Although PCA is sensitive to missing or outlier values, it aims to maintain a minimal dimensionality without affecting valuable or essential information. This paper shows a method combining singular value decomposition (SVD) and PCA algorithms. The dimensions of a given input are explored to determine the optimal number of features to be extracted. This process also helps in identifying the effects of altering the input dimensions.

3.3.2 Gini Impurity-Based Weighted Random Forest (GIWRF)

The RF ensemble classifier [24] is built on several tree models and supports various vital measures. One of the most important metrics is to derive the importance score from the training set. It is done by training the classifier against the expected classes. The traditional method of building classification models assumes that all the classes in a training set are related to the same importance, but this does not consider the imbalance in the training data distribution. This technique utilized a weight adjustment method in RF to understand the relevance of the features in the imbalanced data. After analyzing the Gini impurity, the technique revealed how well a split is performed to divide the total samples of a given class in a specific node.

3.4 Evaluation Metrics

FE algorithms and ML models are evaluated by the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) evaluation metrics. TP is the number of attack samples correctly classified, while TN is the benign sample correctly classified. In contrast, FP refers to the number of misclassified attack samples, while FN represents the number of misclassified benign samples. Furthermore, a sample's accuracy (ACC) is calculated as the ratio of the number of correctly classified samples to the total number of samples as in Eq. (1).

$$\mathrm{ACC} = \mathrm{TP} + \mathrm{TN}/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{1}$$

Detection Rate (DR), also called recall, is the proportion of classified and correctly detected attacks to the total number of samples. Eq. (2) displays the formula of DR.

$$\mathrm{DR} = \mathrm{TP}/\mathrm{TP} + \mathrm{FN} \tag{2}$$

Precision is calculated as the ratio of expected positive outcomes to the total expected negative outcomes plus false positive outcomes. Eq. (3) presents the precision formula.

$$\mathrm{Precision} = \mathrm{TP}/\mathrm{TP} + \mathrm{FP} \tag{3}$$

The area under the curve (AUC) is the area under the receiver operating characteristics (ROC) curve that reflects the trade-off between the FAR and DR where Eq. (4) displays the formula of FAR.

$$FAR = FP/(FP + TN) \tag{4}$$

The imbalance of classes in a dataset affects most metrics. For instance, if a model has a high accuracy score and a high DR, it can be considered a model that is not ideal. Furthermore, a single metric cannot differentiate between models.

4 Experimental Results and Findings

The results of the testing programs are analyzed using a stratified folding method. The mean and the sample results are then calculated and discussed. The discussion section also provides a look at the various datasets that are included in the study. The early comparisons between the FE algorithms and models are carried out using AUC. The effects of implementing the different dimensions of the FE algorithms on the other models are presented in the study. A combination of the two is then selected to evaluate the model's performance against each attack type. Experiments were split across these three datasets. Three scenarios were then implemented on each algorithm using the selected dataset.

In the ToN-IoT dataset, the RF model achieves the best results when applied to the entire dataset and using GIWRF_FE. The LR model performs poorly when using the full dataset with ACC at 88.3%, DR at 70%, and AUC at 79%, but performs better when using the GIWRF_FE with 88% in ACC, 80% in DR, and 85% in AUC. It obtains the best results using PCA_FE as it stabilizes after 24 dimensions by getting almost 99% in ACC, DR, and AUC. DT classifier performs nearly the same compared to the entire dataset and GIWRF_EF in ACC, DR, and AUC. However, it performs reasonably well with PCA_FE when it starts to stabilize with three dimensions with almost 99% in all metrics. NB model achieves better results with GIWRF_EF with dimensions greater than 9 with 91%, 73%, and 94% for ACC, DR, and AUC, respectively. However, compared to the PCA_FE, it produces unsatisfactory results, with almost 70% in ACC and DR 76% in AUC. KNN model performs well on this dataset using the various FE algorithms and the entire dataset. It gets approximately 99% in all metrics. However, it achieves the results in PCA_FE only with five dimensions. The SVM technique performs poorly with the entire dataset and has a few dimensions less than 10. It is better with the PCA algorithm when it has more than 20 dimensions, with 92% in ACC and DR and 94% for AUC. Although the PCA algorithm improves the performance of all models, it reduces the NB classifier of the models compared to the entire dataset or GIWRF algorithm. The full metrics of the best-performing models on the ToN-IoT dataset are shown in Table 1.

Table 1: ToN-IoT classification metrics

	FE	DIM	ACC	DR	F1	Precision	AUC
LR	FULL	37	0.883649	0.70315	0.77473	0.76548	0.79315
	GIWRF	5	0.88062	0.80315	0.87473	.86548	0.85396
	PCA	24	0.998303	0.99229	0.995441	0.99887	0.9991437
DT	FULL	37	0.879666	0.79981	0.753076	0.7176801	0.96942
	GIWRF	9	0.874580	0.749326	0.718191	0.71441	0.969612
	PCA	3	0.99332	0.990552	0.991770	0.9918575	0.99505

(Continued)

Table 1: Continued

	FE	DIM	ACC	DR	F1	Precision	AUC
NB	FULL	37	0.886844	0.81099	0.77351	0.77066	0.893456
	GIWRF	9	0.910891	0.73052	0.769927	0.880563	0.943889
	PCA	30	0.72121	0.7059	0.69353	0.693779	0.7668
RF	FULL	37	0.99964	0.99940	0.999412	0.999669	0.99999
	GIWRF	9	0.99996	0.99998	0.99997	0.99998	1.0
	PCA	5	0.996695	0.995528	0.995989	0.996546	0.99988
KNN	FULL	37	0.99832	0.997863	0.99812	0.9984014	0.999493
	GIWRF	9	0.99811	0.997903	0.99815	0.998432	0.99932
	PCA	5	0.99054	0.989928	0.98728	0.98988	0.997983
SVM	FULL	37	0.892128	0.79997	0.734107	0.76764	0.78745
	GIWRF	9	0.728163	0.8	0.8959	0.86410	0.77390
	PCA	28	0.9206599	0.91803	0.92180	0.927021	0.9468090

The performance of various ML models is significantly improved by using FE algorithms, such as PCA_FE, compared to the entire dataset. The RF model performs well compared to the whole dataset and the GIWRF_FE. Its best performance is achieved with the GIWRF_FE with nine dimensions, as its ACC and AUC are almost 1.0 and its DR model is 99.99%. SVM and NB classifiers are the least effective among the six ML models. In all learning models except NB, GIWRF_FE is unreliable compared to PCA_FE. DT, RF, and KNN models obtained their best results for PCA_FE using five dimensions, making that its optimal number of dimensions, while GIWRF_FE required a higher number of 20.

Fig. 3 shows the Confusion Matrix of different attacks detected in the ToN-IoT dataset and their number of samples. ML and FE are combined to produce the best prediction. It is mainly applied to an RF classifier with GIWRF_FE. However, few ransomware and MITM attacks are available, and most of these attacks are almost fully detected. On the other hand, despite sufficient samples, injection and scanning attacks have 97.68% DRs, which may indicate complex patterns. Each number in Fig. 3 indicates the type of attack classified by the classifier, where 0 = normal, 1 = scanning, 2 = DoS, 3 = injection, 4 = DDoS, 5 = password, 6 = XSS, 7 = ransomware, 8 = backdoor, 9 = MITM.

The results of the various ML models used to analyze the UNSW-NB15 dataset are similar. The KNN, SVM, and LR models perform best, while the NB algorithm gets the worst. The PCA_FE of KNN and SVM models increases until dimension 15 with almost 95% for ACC and AUC and 91% in DR for KNN, while SVM gets 93% in ACC, 79%in DR, and 91% in AUC. DT, RF, and LR models require only ten dimensions for PCA_FE [25]. The GIWRF_FE model performs better with fewer dimensions, reducing the PCA_FE of the KNN, DT, and RF models.

On the other hand, the NB classifier performs poorly when using any of the available FE algorithms. The reduction in the number of dimensions affects the performance of the GIWRF model. It performed better with the ten dimensions model than the whole dataset, where the ACC and AUC were almost 86% and 77% for DR. It achieved a higher score using the RF classifier but a lower one using the NB model. The former with GIWRF_FE gets 97% in ACC, 93% in DR, and 99% in AUC, while the latter with PCA_FE obtains 88%, 78%, and 85% for ACC, DR, and AUC, respectively. Table 2 shows the best results of the various ML models used to analyze the data. The RF algorithm

outperforms the other classifiers when applied using the GIWRF technique. It achieves an AUC score of 0.9953, higher than the PCA method. The performance of the other models is also improved by using the GIWRF technique. The performance of the SVM, DT, and LR models is significantly improved when applied to the complete dataset without needing additional software.

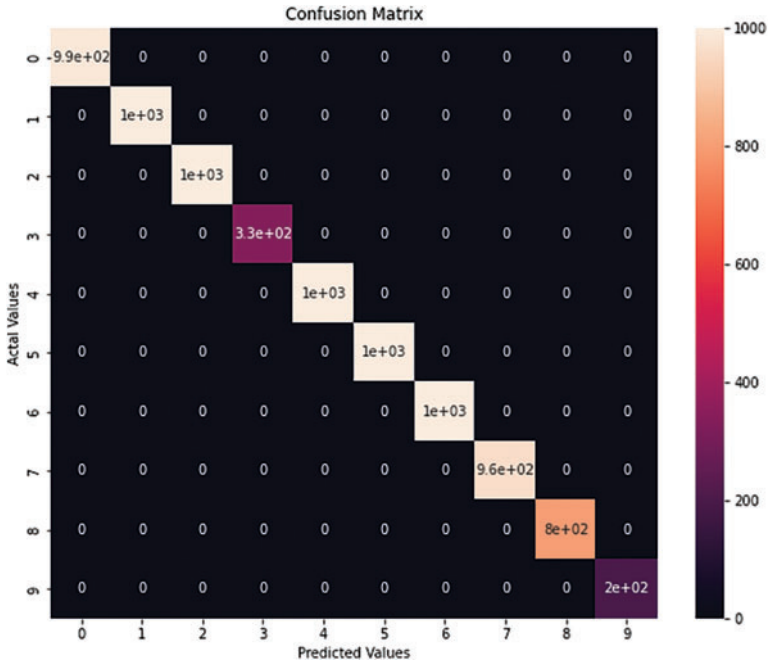

Figure 3: Confusion matrix of RF with GIWRF by ToN-IoT dataset

The PCA technique causes a reduction in the performance of these models. However, it does improve the NB's AUC and DR metrics. In terms of performance, the GIWRF model performs better than the PCA technique in all the ML models except the NB. This suggests that the correlation between the labels and the features of the dataset is extreme. The optimal number of dimensions for the GIWRF and PCA models is 10.

Table 2: UNSW-NB15 classification metrics

	FE	DIM	ACC	DR	F1	Precision	AUC
LR	FULL	40	0.9569	0.83418	0.87789	0.9439104	0.965828
	GIWRF	10	0.950064	0.839211	0.883177	0.95066	0.934703
	PCA	10	0.93501	0.789445	0.93501	0.936017	0.91766
RF	FULL	40	0.96455	0.911593	0.9264285	0.947824	0.994584
	GIWRF	10	0.970324	0.932669	0.939340	0.950515	0.995357
	PCA	10	0.95303	0.8598	0.894693	0.943503	0.974096
NB	FULL	40	0.831926	0.76198	0.731280	0.747199	0.878268
	GIWRF	10	0.86574	0.777796	0.76792	0.787960	0.86627
	PCA	35	0.88045	0.78124	0.77338	0.77953	0.859329

(Continued)

Table 2: Continued

	FE	DIM	ACC	DR	F1	Precision	AUC
DT	FULL	40	0.950371	0.82290	0.878305	0.972720	0.926439
	GIWRF	10	0.95042	0.82309	0.8783	0.97275	0.914813
	PCA	10	0.94566	0.812210	0.86652	0.9603483	0.8736
KNN	FULL	40	0.9468	0.877700	0.891118	0.91519	0.955518
	GIWRF	10	0.951325	0.895776	0.90258	0.91936	0.961084
	PCA	15	0.94625	0.91278	0.88931	0.912785	0.950712
SVM	FULL	40	0.95985	0.838823	0.88447	0.952564	0.9611013
	GIWRF	10	0.956760	0.85340	0.89837	0.9654162	0.92570
	PCA	15	0.936752	0.790013	0.842399	0.938087	0.91387

The best-performing model for analyzing attacks was developed using RF with the GIWRF technique, which measures the DR measure of each attack type. It could detect almost all of the attack types in the dataset. The lowest DRs metric was obtained by DoS and backdoor attacks, as shown in the Confusion Matrix of the model in Fig. 4. Where numbers indicate each type of attack that is classified by the classifier, where 0 = normal, 1 = generic, 2 = exploits, 3 = fuzzers, 4 = dos, 5 = reconnaissance, 6 = analysis, 7 = backdoor, 8 = injection, 9 = worms.

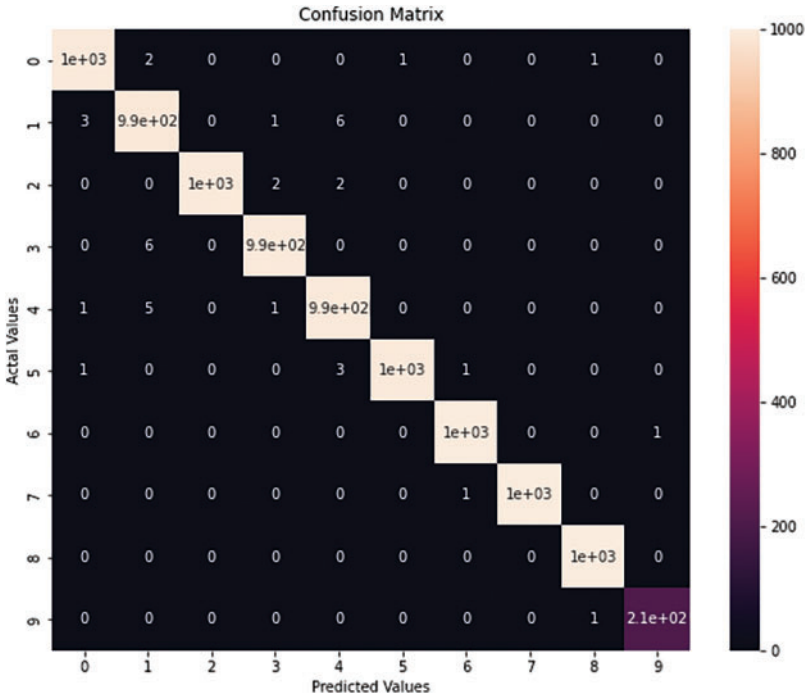

Figure 4: Confusion matrix of RF with GIWRF by UNSW-NB15 dataset

In the BoT_IoT dataset, when using the PCA model, the LR algorithm performed well in detecting the whole dataset; it gets 99% in ACC, 75% in DR, and 87% in AUC. The effects of the two FE techniques on the KNN and RF performance are equal. Both models achieve about 99.99% ACC and

other metrics using full dataset attributes, which match the result of [26] for KNN and [27] for RF. PCA technic is slightly better than that of GIWRF_FE for all models. NB classifier is very similar to the DT model due to its difficulty in classification when dealing with a lower number of dimensions. Compared to the GIWRF method applied to the full dataset, the DT model performs well but is a little poor with the PCA method; it gets 99% for ACC, DR, and AUC. The GIWRF technique requires 13 dimensions to stabilize and reach its maximum AUC value. The SVM model obtains its worst results using the full dataset with 77%, 79%, and 75% for ACC, DR, and AUC, respectively, while the PCA method peaks the highest performance at dimension 20 with almost 99% for all metrics.

Moreover, GIWRF and PCA methods have similar impacts on all ML models except the SVM classifier, for which PCA_FE significantly outperforms GIWRF_FE except the DT model. The results of the experiments indicate that the SVM model performs poorly when applied to the BoT_IoT dataset. Table 3 shows the best performance of the various FE algorithms for each model. The KNN, RF, and DT models perform well when applied to the whole dataset, while the FE algorithms improve the performance of NB and LR classifiers. GIWRF performs worse than the PCA technique for all models except the DT algorithm. On the other side, the results of the studies indicate that the three main algorithms, namely, SVM, LR, and NB, are ineffective in identifying attacks in the BoT_IoT dataset. The optimal numbers for both PCA and GIWRF methods are 20 and 13.

Table 3: BoT_IoT classification metrics

	FE	DIM	ACC	DR	F1	Precision	AUC
LR	FULL	44	0.986480	0.7494	0.756773	0.76785	0.8677
	GIWRF	13	0.96479	0.61795	0.6214	0.700511	0.86196
	PCA	20	0.999219	0.99612	0.997157	0.998351	0.9999896
RF	FULL	44	0.99999	0.999995	0.99882	0.99777	0.9999
	GIWRF	13	0.9999	0.99995	0.99995	0.99985	0.99998
	PCA	25	0.99876	0.98846	0.99242	0.99811	1.0
NB	FULL	44	0.93985	0.79687	0.77872	0.782322	0.96752
	GIWRF	22	0.93011	0.77907	0.76820	0.7589	0.91236
	PCA	20	0.954990	0.88084	0.899036	0.9481908	0.99056
DT	FULL	44	0.99992	0.9949	0.99711	0.99995	0.997901
	GIWRF	13	0.99935	0.9961	0.99746	0.99922	0.99845
	PCA	18	0.982633	0.90877	0.9146028	0.924209	0.97045
KNN	FULL	44	0.99992	0.995407	0.99763	0.99995	0.99939
	GIWRF	13	0.99954	0.9970	0.99833	0.99971	0.99354,
	PCA	30	0.99850	0.98017	0.98670	0.99752	0.99989
SVM	FULL	44	0.77838	0.79838	0.80784	0.78838	0.75828
	GIWRF	13	0.79477	0.764265	0.872416	0.79693	0.77693
	PCA	20	0.998959	0.992752	0.9952737	0.99817	0.9922737

Fig. 5 shows the Confusion Matrix of different attacks detected in this dataset by applying the RF and PCA algorithms for prediction; the RF classifier was applied to 13 extracted dimensions. Each

number in Fig. 5 indicates the type of attack classified by the classifier, where 0 = DDoS, 1 = DoS, 2 = Reconnaissance, 3 = Normal, and 4 = Theft.

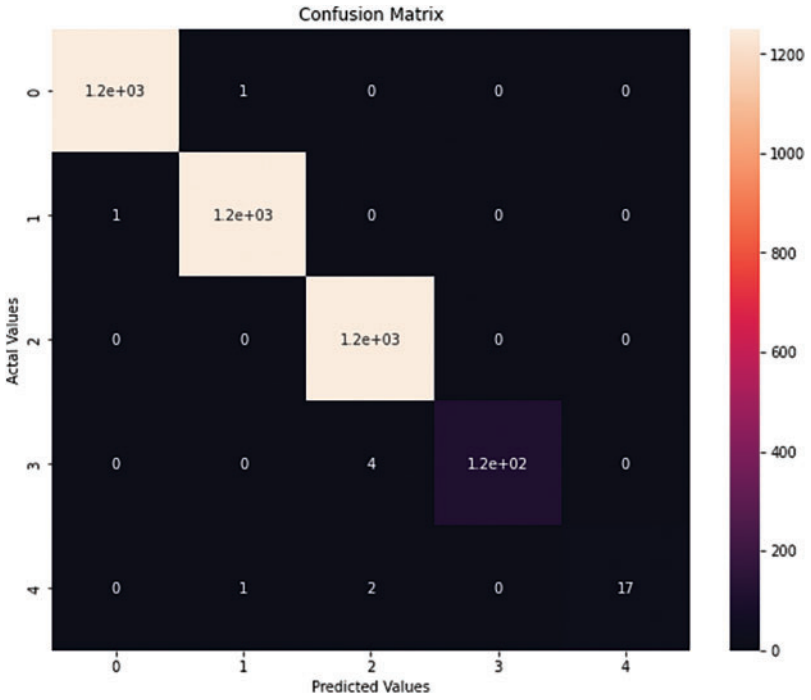

Figure 5: Confusion matrix of RF with GIWRF by BoT_IoT dataset

5 Conclusions

This paper evaluates the performance of the GIWRF and PCA feature extraction methods in combination with various ML models. The evaluation of their performance is carried out using a variety of statistical techniques. In addition, the correlation between the detection accuracy and the number of dimensions is analyzed. This paper presents a framework combining six ML models to comprehensively analyze three NIDS datasets: the DT, RF, KNN, and SVM models. The optimal combination of these models is shown to improve the performance of the classification. The analysis of the variance in the data collected is also performed to identify the optimal number of feature dimensions for each dataset. 18 combinations of the FE algorithm and the ML classifier have been tried, and none performs well across the three NIDS datasets. The combination of the RF classifier and GIWRF gave a high result of up to 97% in the two datasets. In contrast, the combination of the RF classifier and PCA gave the most accurate result in the third dataset, with almost up to 99%. This suggests that finding a suitable combination of the two is not as trivial as it seems. Further investigation is needed to determine which combination performs well in different datasets and which practical applications can be used. Although this research aims to improve the performance of the classification algorithms for a particular feature set, we also believe that a stronger emphasis should be placed on their generalizability. This paper comprehensively evaluates the proposed algorithms' performance in network scenarios. The generic feature sets used for classifying NIDS datasets must be designed to be efficient and applicable to different network settings and applications. This will allow the research community to better understand the performance of other ML models.

Funding Statement: The researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Doda, Y. Wang, M. Kahl, E. J. Hoffmann, H. Taubenböck *et al.,* "So2sat POP–A curated benchmark data set for population estimation from space on a continental scale," *Scientific Data*, vol. 9, pp. 1–14, 2022.
[2] S. Smys, B. Abul and W. Haoxiang, "Hybrid intrusion detection system for internet of things (IoT)," *Journal of ISMAC*, vol. 2, no. 4, pp. 190–199, 2020.
[3] F. Y. Sattarova, "Integrating intrusion detection system and data mining," in *Proc. 2008 Int. Symp. on Ubiquitous Multimedia Computing*, Hobart, TAS, Australia, pp. 256–259, 2008.
[4] S. Subarna, "Modified gray wolf feature selection and machine learning classification for wireless sensor network intrusion detection." *IRO Journal on Sustainable Wireless Systems*, vol. 3, no. 2, pp. 118–127, 2021.
[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems, and challenges," *Computers & Security*, vol. 28, no. 1–2, pp. 18–28, 2009.
[6] P. V. Amoli, T. Hamalainen, G. David, M. Zolotukhin and M. Mirzamohammad, "Unsupervised network intrusion detection systems for zero-day fast-spreading attacks and botnets," *International Journal of Digital Content Technology and its Applications*, vol. 10, no. 2, pp. 1–13, 2016.
[7] M. J. Hashemi, G. Cusack and E. Keller, "Towards evaluation of NIDSs in adversarial setting," in *Proc. the 3rd ACM CoNEXT Workshop*, Orlando FL USA, pp. 14–21, 2019.
[8] C. Sinclair, L. Pierce and S. Matzner, "An application of machine learning to network intrusion detection," in *Proc. 15th Annual Computer Security Applications Conf. (ACSAC'99)*, Phoenix, AZ, USA, pp. 371–377, 1999.
[9] E. Anthi, L. Williams and P. Burnap, "Pulse: An adaptive intrusion detection for the internet of things," in *Proc. Living in the Internet of Things: Cybersecurity of the IoT–2018*, London, pp. 1–4, 2018.
[10] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. 2010 IEEE Symp. on Security and Privacy*, Oakland, CA, USA, pp. 305–316, 2010.
[11] A. Mahfouz, "Comparative analysis of ml classifiers for network intrusion detection," in *Proc. Fourth Int. Congress on Information and Communication Technology*, Singapore, pp. 193–207, 2019.
[12] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang *et al.,* "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2007.
[13] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[14] S. Ryu and B. Yang, "Comparison of machine learning algorithms and their ensembles for botnet detection," *Journal of Computer and Communications*, vol. 6, no. 5, pp. 119–129, 2018.
[15] R. Doshi, N. Apthorpe and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *Proc. 2018 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, pp. 29–35, 2018.
[16] S. Ali, R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.
[17] M. Shafiq, Z. Tian, Y. Sun, X. Du and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.
[18] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto and K. Sakurai, "Towards a lightweight detection system for cyber-attacks in the IoT environment using corresponding features," *Electronics*, vol. 9, no. 1, pp. 144–162, 2020.

[19] P. Jotikabukkana and V. Sornlertlamvanich, "The holistic framework of using machine learning for an effective incoming cyber threats detection," in *Proc. 29th EJC*, Lappeenranta, Finland, pp. 363–380, 2019.
[20] N. Moustafa, "New generations of internet of things datasets for cybersecurity applications based machine learning: Ton_iot datasets," in *Proc. eResearch Australasia Conf.*, Brisbane, Australia, pp. 3–5, 2019.
[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. MilCIS*, Canberra, ACT, Australia, pp. 1–6, 2015.
[22] N. Koroniotis, N. Moustafa, E. Sitnikova and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2018.
[23] Z. Elkhadir, K. Chougdali and M. Benattou, "Intrusion detection system using PCA and kernel PCA methods," in *Proc. Mediterranean Conf. on Information & Communication Technologies (MedICT'15)*, Saïdia, Morocco, pp. 489–497, 2015.
[24] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using gini impurity-based weighted random forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, pp. 1–22, 2022.
[25] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher and M. Portmann, "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Communications and Networks*, 2022.
[26] A. Churcher, R. Ullah, J. Ahmad, S. Rehman, F. Masood *et al.,* "An experimental analysis of attack classification using machine learning in IoT networks," *Sensors*, vol. 21, no. 2, pp. 446–477, 2021.
[27] S. Behal, A. Chopra and V. Sharma, "Evaluating machine learning algorithms to detect and classify DDoS attacks in IoT," in *Proc. 8th Int. Conf. on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 517–521, 2021.