Egyptian Informatics Journal 25 (2024) 100450

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full Length Article

Enhancing DDoS attack detection in IoT using PCA

Sanjit Kumar Dash [a], Sweta Dash [a], Satyajit Mahapatra [a], Sachi Nandan Mohanty [b], M. Ijaz Khan [c,d,*], Mohamed Medani [e], Sherzod Abdullaev [f,g], Manish Gupta [h]

[a] Odisha University of Technology and Research, Bhubaneswar, Odisha, India
[b] School of Computer Science & Engineering (SCOPE), VIT-AP University, AP, India
[c] Department of Mechanical Engineering, Lebanese American University, Beirut, Lebanon
[d] Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44000, Pakistan
[e] Department of Computer Science, College of Science and Art at Mahayil, King Khalid University, Muhayil Aseer 62529, Saudi Arabia
[f] Faculty of Chemical Engineering, New Uzbekistan University, Tashkent, Uzbekistan
[g] Department of Science and Innovation, Tashkent State Pedagogical University named after Nizami, Tashkent, Uzbekistan
[h] Division of Research and Technology, Lovely Professional University, Phagwara, India

ARTICLE INFO

Keywords:
DDoS attack
Feature Selection
Principal component analysis
Random forest
KNN
Naïve Bayes

ABSTRACT

Internet of Things (IoT) security and reliability rely on the capacity to identify distributed denial-of-service (DDoS) assaults in IoT networks. This research presents a comprehensive study on DDoS attack detection using the NSL-KDD dataset. The dataset contains a diverse set of network traffic data. This paper proposes two approaches, one utilizing Principal Component Analysis (PCA) and another without PCA, to compare their performance. Robust scaling and encoding techniques are applied as preprocessing steps. The experiment outcomes demonstrate a noteworthy improvement in the accuracy of DDoS attack detection in IoT devices by integrating PCA and Robust Scaler. Notably, the Random Forest and KNN classifiers demonstrate exceptional performance with an accuracy of 99.87 % and 99.14 %, respectively, while Naïve Bayes shows a lower accuracy of 87.14 %. The findings from this experiment contribute valuable insights into enhancing the security of IoT devices against DDoS attacks. The proposed approach showcases the importance of appropriate preprocessing techniques in achieving robust intrusion detection systems for IoT environments.

1. Introduction

Any online system is vulnerable to distributed denial-of-service (DDoS) assaults, which are common and can cause significant disruption. The need to make sure that IoT devices are secure and can withstand cyber threats is growing as their use spreads across different industries [1,2,3,26]. Due to their heterogeneous transmission technologies, low processing capabilities, hardware restrictions, and lack of built-in security, IoT devices are primarily susceptible to threats. Botnets and the majority of cyberattacks nowadays are reportedly DDoS attacks, according to recent reports. Over the last decade, both the frequency and severity of these attacks have skyrocketed [4,6,27]. By sending a deluge of traffic to the victim's network, these assaults cripple it and prevent it from serving genuine customers [5,28]. Consequently, it is critical for network administrators and security personnel to be able to detect DDoS attacks. Important services, private information, and money can all be lost due to these kinds of attacks. The potential impact is magnified by the networked nature of IoT devices. Once compromised, these devices can be used to conduct large-scale attacks. For IoT installations to be secure, reliable, and available, it is critical to detect these assaults in IoT devices. The NSL-KDD dataset allows for the evaluation of several machine learning methods in both normal and attack settings [7,29].

Machine learning techniques have been useful for detecting distributed denial of service attacks (DDoS) due to their ability to sift through mountains of network traffic data in search of patterns linked to malicious behavior [8]. Machine learning techniques are able to identify distributed denial of service (DDoS) attacks by predicting future network traffic patterns. Internet of Things (IoT) networks often employ algorithms like K-Nearest Neighbor, Decision Tree, and neural networks to identify distributed denial of service (DDoS) assaults. Using labeled datasets to train machine learning models allows for real-time detection and classification of anomalous traffic patterns that may indicate DDoS attacks. In recent years, machine learning methods have become more popular and effective in detecting and preventing these types of attacks

* Corresponding author at: Department of Mechanical Engineering, Lebanese American University, Beirut, Lebanon.
E-mail addresses: sachinandan09@gmail.com (S.N. Mohanty), scientificresearchglobe@gmail.com (M.I. Khan).

https://doi.org/10.1016/j.eij.2024.100450
Received 14 August 2023; Received in revised form 20 January 2024; Accepted 4 February 2024
Available online 13 February 2024

[9]. Botnet and distributed denial of service (DDoS) assaults can target Internet of Things (IoT) devices, and once compromised, these devices can launch a variety of DDoS attacks, according to recent research [10].

Addressing the increasing security issues in IoT devices, particularly in detecting DDoS attacks, is the driving force behind this work. The functioning, availability, and general security of these devices are jeopardized by their susceptibility to distributed denial of service (DDoS) assaults, which are becoming more common as IoT installations spread across many domains. Consequently, creating efficient DDoS detection systems that are customized for IoT environments is an urgent necessity. The objective of this work is to examine and contrast various machine learning algorithms for detecting distributed denial of service attacks in Internet of Things devices. The main objective is to compare and contrast different machine learning methods for identifying malicious DDoS traffic from legitimate Internet of Things (IoT) traffic. We intend to evaluate various ML algorithms' accuracy, precision, recall, and F1 score using the NSL-KDD dataset, which contains realistic data on network traffic from IoT devices. Through analyzing their performance data, we determine which algorithms are the most effective in achieving precise and dependable detection.

2. Related work

DDoS assaults have been a major concern for Internet security for a long time. To address this issue, researchers strengthened network behavior to make it more resistant to these types of attacks. A growing number of experts are relying on different Machine Learning algorithms to identify DDoS attacks.

In their proposal, Hussain et al. trained two ResNet-18 models, which are considered state-of-the-art deep learning algorithms, to use in tandem with machine learning. The goal of training two ResNet-18 models was to recognize distributed denial of service (DDoS) assaults and scanning activity during the early stages of an attack in order to detect and prevent botnet attacks on the internet of things (IoT) [11]. With a 91.01 % success rate, their two-pronged approach can identify and counter 60 distinct DDoS attempts and 33 distinct scanning assaults. Chavan et al. proposed a machine learning-based method for botnet DDoS attack identification and prevention [12]. New Botnet prevention functionality scans URLs for dangerous content and prevents visitors from being directed to such sites. The accuracy of Logistic Regression, support vector machine (SVM), KNN, and decision tree was 90.40 %, 90.36 %, 89.15 %, and 82.28 %, respectively, when tested with the detection process using four separate classifiers. In their work, Chen et al. [13] developed a method to protect 5G-enabled IoT from distributed denial-of-service attacks in real time. For 5G core network packet-level identification, this system compiles data from several sources. In typical attack circumstances, this method has the ability to guarantee a detection rate of 99 % while concurrently reducing the rate of packet inspection to less than 37 %.

In response to IEEE P2668, Liu et al. [14] presented a DRL-MLDS, a multi-layer IoT distributed denial-of-service defense system. They were able to get their model to defend against a single protocol attack with a 97 % success rate and attack that protocol with a 96 % success rate. The proposed model, DRL-MLDS, has an application index of 3.2; with reward metrics that are in line with the IEEE P2668 standard, it can be raised to 4.4. In their study on collaborative IoT packet sampling in response to a DDoS attack, Chen et al. [15] used a Stackelberg game model. They found a lower limit for the packet sampling needed to fight against DDoS attacks by doing an equilibrium analysis. Online and proactive protection against DDoS traffic can be facilitated by their suggested method of packet sampling. The model not only reduces the sampling rate by over 70 %, but it also shows remarkable resilience when faced with changes to the boundary conditions. To identify distributed denial of service (DDoS) assaults on 5G networks, Hussain et al. [16] suggested a deep learning-based approach. They proposed a unified system that can identify distributed denial of service (DDoS) assaults initiated by a botnet of compromised devices in a timely manner. Authentic network data and deep convolutional neural networks (CNNs) are the building blocks of this approach. For silent calls, the CNN model used the DRC model, and it achieved a detection accuracy of over 91 % for both normal and under-attack cells. Furthermore, when applied to a more intricate blended strategy, the framework attained an accuracy of over 97 %.

By considering the inputs from a northbound SDN application, Bousalem et al. developed an OpenAirInterface-based prototype that allows for the formation of network slices on demand and the dynamic management of physical resources based on user behavior [17]. Demonstration of a deep learning prototype for 5G and beyond mobile network threat detection and mitigation. By keeping the false positive rate below 4 %, their model is able to get an accuracy of 97 %. One approach proposed by Alghazzawi et al. [18] is to use benchmark data to train a hybrid deep learning (DL) model, such as a CNN with BiLSTM (bidirectional long/short-term memory). Bidirectional long/short-term memory (BiLSTM) was incorporated into the CNN model. The features that were deemed most pertinent to the issue were selected after they were ranked and their scores were compared to those in the given dataset. Through training, testing, and confirmation on the CIC-DDoS2019 data set, the experimental results demonstrate that the suggested CNN-BI-LSTM achieved an accuracy of up to 94.52 %. To detect intrusions using semi-supervised learning, Duan et al. suggested a DLGNN (dynamic line graph neural network) [19]. The traffic flow via a network is shown by this paradigm using a set of spatiotemporal graphs. The use of a dynamic GNN (DGNN) allows us to record the changing nature of IP pair communication and to derive location data from each snapshot. This approach can detect abnormalities with a high accuracy (98.15–99.8 % utilizing a lesser number of tagged samples), according to experiments on six new datasets. The average DDoS detection accuracy across all six datasets is 95.32 %, demonstrating that we have achieved state-of-the-art levels of multiclass performance. Problems with harmful wireless IoT causing distributed denial of service attacks on Internet of Things servers were addressed by Nagarathna and Mercy [20]. In order to protect IoT servers from distributed denial of service (DDoS) assaults, their security solution incorporates cloud computing and the software-defined networking (SDN) architecture. A semi-supervised machine learning approach, a topology emulator of the network, and a testbed for LEDEM evaluation make up their new learning-driven detection mitigation (LEDEM) solution. A 96.2 % improvement in DDoS attack detection accuracy was found by comparing the findings obtained by LEDEM to those of state-of-the-art systems.

In their evolutionary support vector machine model, Sahoo et al. employed kernel principal component analysis (KPCA) to reduce feature vector dimensions. Contrarily, GA improves the model's accuracy by optimizing several SVM parameters [21]. To lessen the noise caused by the feature gap, they have employed N-RBF. With an accuracy rating of 98.90 %, the model surpasses the typical support vector machine. To detect and counteract LR-DDoS attacks, Perez-Diaz proposed a flexible security architecture based on SDN [22]. Utilizing the CICDoS dataset, they developed and proved that the suggested method successfully detected and prevented numerous types of low-resource distributed denial-of-service attacks (LR-DDoS). The proposed intrusion prevention system can thwart any assault that the intrusion detection system has already detected, and its detection rate is 95 %. In order to identify low-rate DDoS attacks, Zhijun et al. published their findings [23]. This study investigates the potential vulnerabilities of SDN's data layer to low-rate DDoS attacks. In addition to suggesting a defensive strategy based on the dynamic deletion of flow rules, the success rate of forwarding ordinary packets was 97.85 %. The detection rate of LR-DDoS against the SDN data layer is 95.60 % when using this method.

Using data about the asymmetry and rate characteristics of the flows, this program was able to identify potentially harmful traffic [24]. Tan et al. suggested a hybrid machine learning approach based on k-means and KNN to exploit these features and detect suspicious flows signaled

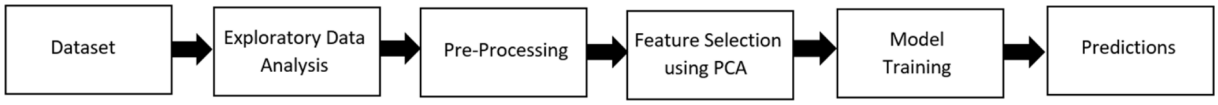

Fig. 1. Proposed Machine Learning Model.

by the detection trigger mechanism. Afterwards, in response to attacks, the player will activate the character's defenses. The authors of this study proposed a new paradigm for control plane and data plane cooperative detection techniques. With any luck, this framework will be able to successfully block DDoS attacks on SDN and increase detection accuracy to 98.85 %. Two feature selection approaches, IG and RF, were proposed by Sayed et al. [25] for the purpose of finding relevant characteristics. The three flow-based benchmark datasets used to assess the suggested feature selection strategy are SDN, CICIDS2017, and CICIDS2018. Using the same set of features, they have also demonstrated a model based on deep learning. The results demonstrate that the proposed method, when combined with certain feature techniques, can effectively decrease model complexity while maintaining accuracy. This model was created using the precise features that were chosen.

3. Proposed model

Several important steps are included in the created model, as illustrated in Fig. 1. It starts with the input dataset and then moves on to exploratory data analysis (EDA) to discover more about the data. To deal with missing data, duplication, and features that have been scaled or normalized, preprocessing procedures are used. We use principal component analysis (PCA) feature selection to find the most useful characteristics. The preprocessed data is used to train multiple machine learning algorithms that can accurately classify DDoS attacks. We measure the models' performance with F1-score, recall, accuracy, and precision. The goal of this all-encompassing strategy was to improve the DDoS attack detection model for IoT devices by using a suitable dataset, efficient preprocessing, feature selection, different classifiers, and rigorous model evaluation.

3.1. Data preprocessing

Preprocessing the NSL-KDD dataset ensured unbiased classifiers and prepared the data for model training and evaluation, making it ideal for DDoS attack detection in IoT devices. Due to the dataset's prior exclusion of duplicate entries, there is no longer any bias towards more frequent records in the training set. The dataset was preprocessed with the help of scikit-learn's RobustScaler. The numerical features were normalized and outliers were efficiently handled using this technique. To apply robust scaling to the numerical properties of the dataframe, it makes use of the 'RobustScaler' from scikit-learn. A dataframe called'std_df' is returned after being scaled. An input dataframe ('dataframe') is passed through a preprocessing function that applies different preparation operations. It begins by removing the 'cat_cols' category columns from the dataframe in order to focus on the 'df_num' numerical characteristics. The numerical feature column names are saved in 'num_cols.' Next, we scale the numerical features of 'df_num' using the 'Scaling' function. Then, we assign the scaled dataframe to'scaled_df.' We remove the mounted numerical columns from the 'dataframe,' and then we add back the original numerical columns. 'Outcome,' the target variable, is encoded as a binary number, with 0 representing normal traffic and 1 representing attacks. The 'protocol_type,''service,' and 'flag' categorical columns are expanded into binary columns representing the different categories using 'pd. get_dummies.' This process is known as one-hot encoding. A dataframe that has been preprocessed is returned. The training dataset, 'data_train,' is fed into the 'preprocess' function to produce the preprocessed dataframe,'scaled_train.' By removing the 'outcome' and 'level' columns from'scaled_train,' we can extract the features ('x') and assign the target variable ('y') to the 'outcome' column. 'Scaling' takes a numerical data frame ('df_num') and a list of column names ('cols') as inputs. The dataset was cleaned, scaled, and encoded suitably throughout these pretreatment processes to ensure reliable DDoS attack detection in IoT devices.

3.2. Feature selection

By reducing the number of dimensions in high-dimensional data, principal component analysis (PCA) becomes an essential statistical tool for feature selection and dimensionality reduction. In order to extract the most important features from the dataset, a dimensionality reduction method called Principal Component Analysis (PCA) was employed. To reduce the dataset's dimensionality, the 'PCA' class from scikit-learn was used. By using the Principal Component Analysis (PCA) algorithm with n_components = 20 on the features ('x'), we can deduce that there will be 20 principal components in the reduced feature space. To produce the reduced feature representation ('x_reduced'), the 'PCA' object is fitted to 'x' using 'pca.fit' and the features are modified using 'pca. transform'. Principal component analysis (PCA) enhances the detection capabilities for distributed denial of service (DDoS) assaults in internet of things (IoT) networks by lowering the number of characteristics from 42 to 20. This allows for more effective analysis while also reducing processing cost. By offering a practical method for improving the identification and mitigation of DDoS attacks, our work helps to advance the area of Internet of Things security, which in turn protects IoT ecosystems and guarantees the dependability and authenticity of linked devices.

3.3. ML classifiers

We have chosen six supervised learning classifiers—Random Forest, KNN, Decision Tree, SVM, Logistic Regression, and Naive Bayes—to create and train our models. Because of its effectiveness in detecting DDoS attacks, Random Forest was selected because of its scalability, robustness, and capacity to handle high-dimensional data. The simplicity and ability of KNN to capture local patterns in the data led to its employment in attack detection. The interpretability and transparency of Decision Tree's rules for attack identification led to their utilization. Because of their capacity to generate ideal hyperplanes, which facilitate class separation, Support Vector Machines (linear) were chosen. To evaluate the likelihood of an attack, the probabilistic model of choice was Logistic Regression. Finally, Naive Bayes was used since it is good at recognizing assaults with certain traits and it assumes that features are independent. To provide a varied range of methods for better security in IoT environments, these classifiers were selected based on their individual capabilities and their capacity to detect distributed denial of service (DDoS) attacks in Internet of Things (IoT) devices.

4. Dataset description and analysis

Building and testing machine learning models was based on the NSL-KDD dataset, which was created for the express purpose of detecting distributed denial of service attacks in Internet of Things devices (https://www.unb.ca/cic/datasets/nsl.html) [30]. There are a total of 125,973 packets and 22 different kinds of attacks recorded in it. There are labels associated with 42 of its features. You can tell the package's irregularity level (attack or not) only by looking at the label. In order to aid in the creation and assessment of intrusion detection systems (IDSs) for network security, this dataset is designed to identify different kinds

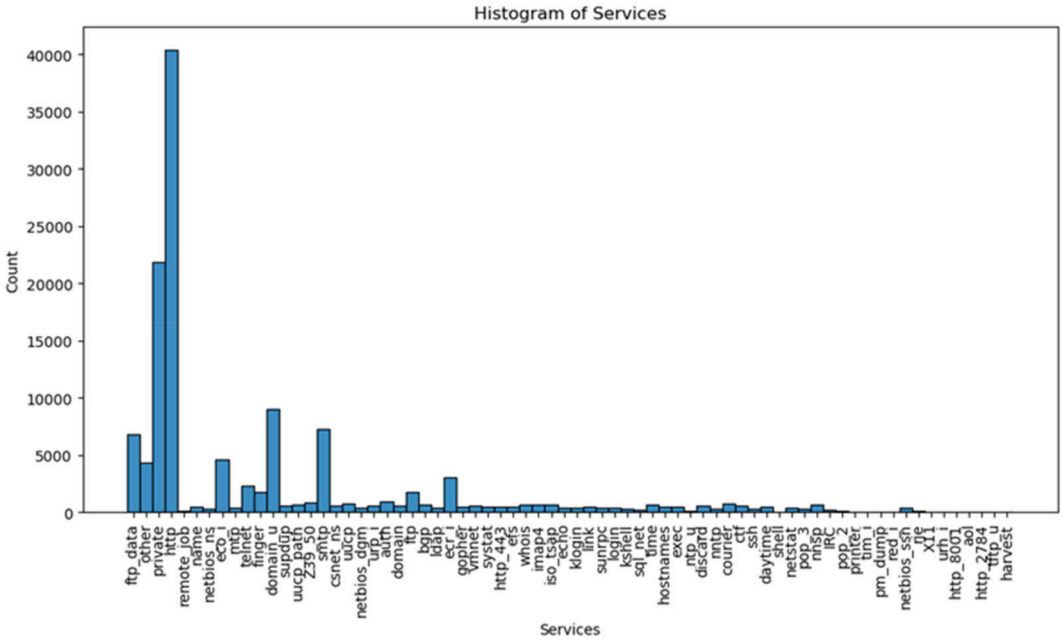

Fig. 2. Histogram of Services.

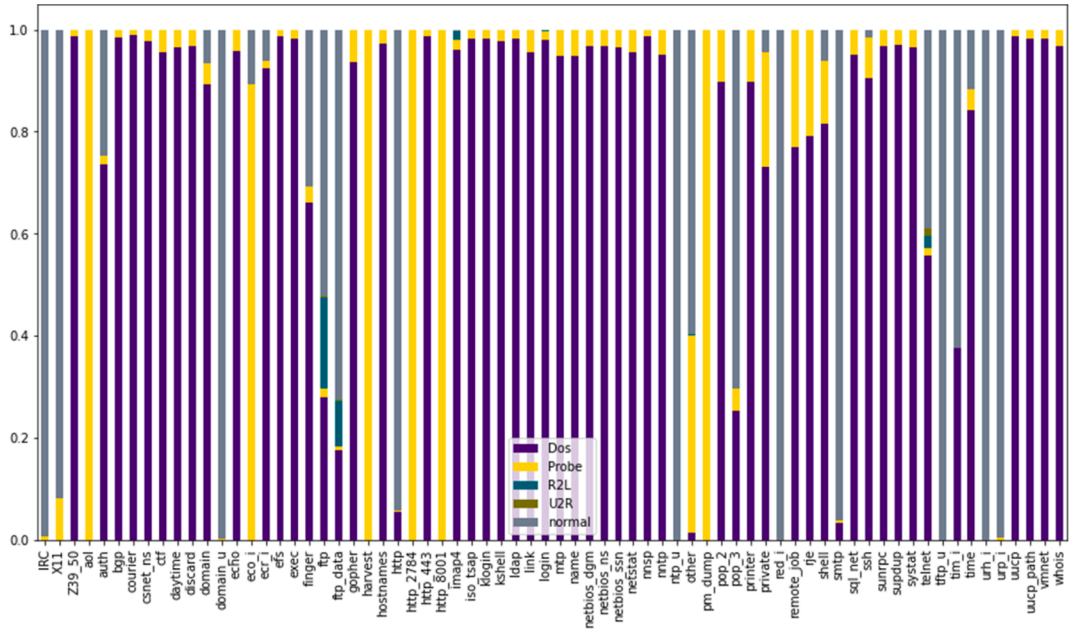

Fig. 3. Different service type dependencies on attack type.

of attacks, such as distributed denial of service (DDoS) attacks, in Internet of Things (IoT) settings. Dataset characteristics, distributions, and interrelationships were all investigated as part of the EDA procedure. A review of the dataset was initiated by calculating descriptive statistics, which included quartiles, standard deviation, median, and mean. This was useful in spotting any data anomalies or outliers.

Fig. 2 displays the dataset's service distribution; each service is represented by a bar, and the y-axis shows the quantity or percentage of services where "http" requests are high relative to "private," which accounts for around half of "http" requests. Between zero and ten percent of the total data falls within the remaining requests. As seen in Fig. 3, the'service' and 'attack_type' columns in the dataset are related, illustrating how various attack types are dependent on various service kinds. The different colors within each bar indicate the many types of attacks, and each bar symbolizes a different service. Each service type's proportion of attacks is shown on the y-axis. The correlation between different kinds of protocols and attacks is graphically shown in Fig. 4. A 'protocol_type' variable is shown on the x-axis, with 'attack' type determining the bar colors. Graph analysis shows that different types of attacks have different impacts on different protocols; for example, "ICMP" is impacted by probe and DoS attacks more than "TCP" while "udp" is impacted by probe attacks more than DoS attacks.

It should be mentioned that the NSL-KDD dataset includes both numerical and categorical variables. It can be more difficult to understand the relationship between categorical variables and the correlation coefficients that are normally computed for numerical variables. Consequently, when estimating the relationship between variables based on categorical features, alternative metrics like Cramer's V or the

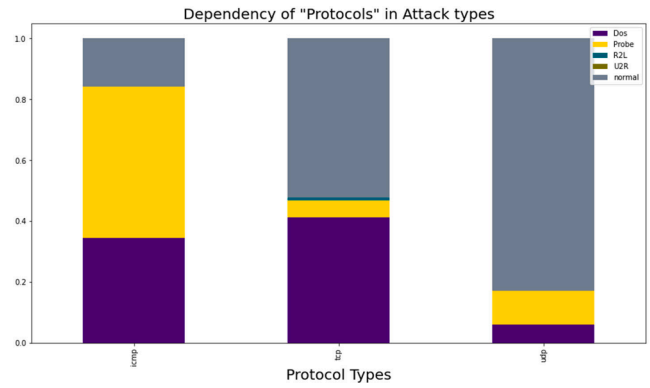

Fig. 4. Different Protocol dependencies on attack type.

correlation ratio can be employed. For feature selection, finding highly correlated variables, or understanding the underlying patterns in the data, the resultant correlation matrix as shown in Fig. 5 would give a numerical representation of the relationships between the numerical features in the dataset.

5. Result and discussion

The experiment was conducted on a Dell Inspiron 5567 laptop with an Intel Core i5 7th generation processor and 16 GB of DDR4 RAM running on the Windows 11 Pro operating system. The research utilized Jupyter Notebook, an interactive coding environment, and the Anaconda distribution, providing a comprehensive Python package suite for data science and machine learning.

Initially, the dataset was preprocessed using a robust scaler as discussed in section 3.1, and then the 'PCA' function was fitted to 'x' using 'pca.fit', and the features were transformed using 'pca.transform' to obtain the reduced feature representation ('x_reduced') resulting the reduction of original 42 features to a more manageable set of 20 features as discussed in section 3.2. The original features ('x') and reduced features ('x_reduced') are split into training and testing sets using 'train_test_split.' The target variable (y) is also divided accordingly. The training set comprises 80 % of the data, while the testing set contains 20 %. The same split is applied to the reduced feature representation ('x_train_reduced,' 'x_test_reduced') and the target variable ('y_train_reduced,' 'y_test_reduced'). The classifiers were trained on the following preprocessed datasets to evaluate their performance on the original and reduced features.

a) x_train and y_train: The training set consisting of original features ('x_train') and the corresponding target variable ('y_train').
b) x_test and y_test: The testing set consisting of original features ('x_test') and the corresponding target variable ('y_test').

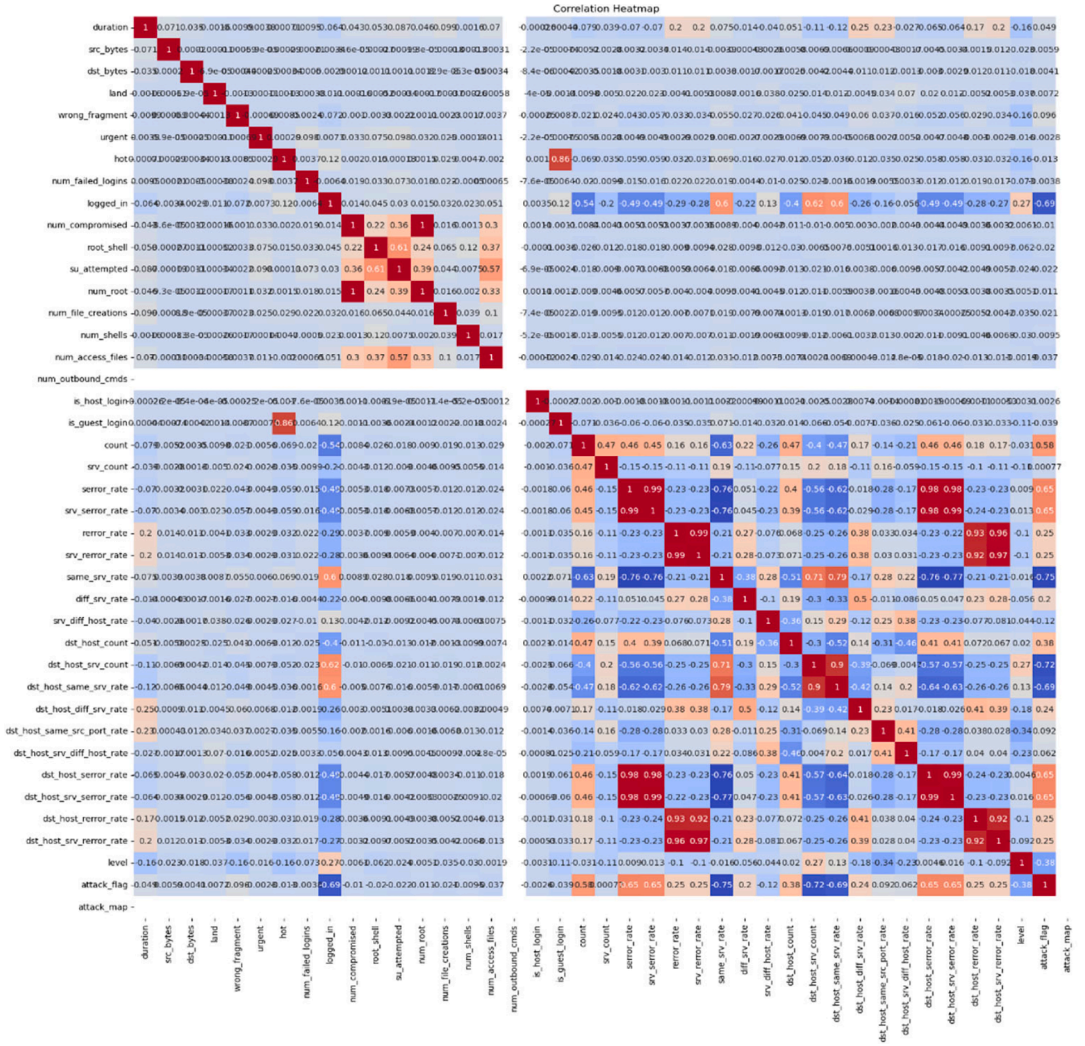

Fig. 5. Correlation Analysis of NSL-KDD Dataset.

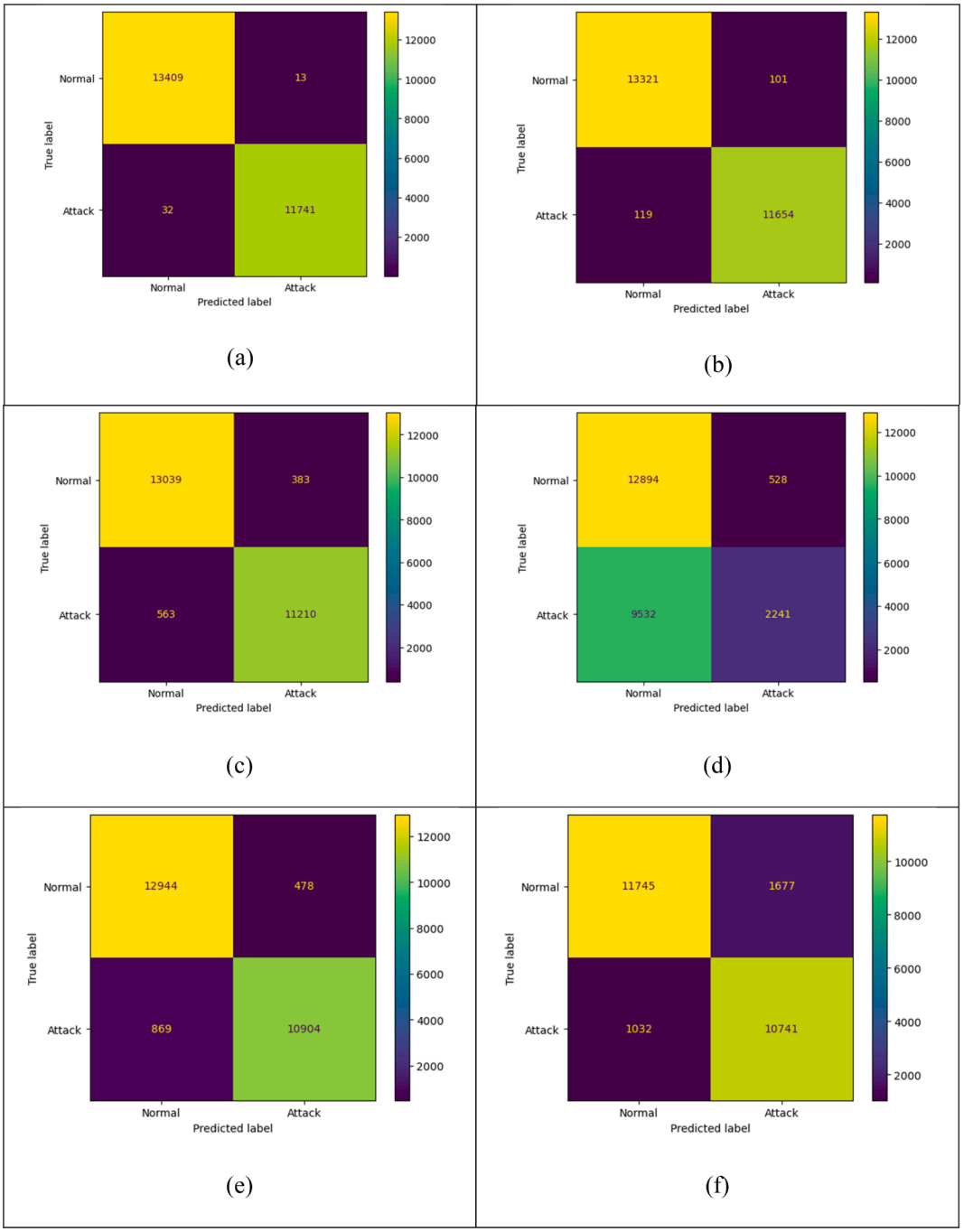

Fig. 6. Confusion matrix without using PCA ((a) Random Forest, (b) KNN, (c) Decision Tree (d) Naïve Baye, (e) SVM, (f) Logistic Regression).

c) x_train_reduced and y_train_reduced: The training set with reduced features obtained from PCA ('x_train_reduced') and the corresponding target variable ('y_train_reduced').
d) x_test_reduced and y_test_reduced: The testing set with reduced features obtained from PCA ('x_test_reduced') and the corresponding target variable ('y_test_reduced').

Each classifier was trained using the training set and then evaluated using the respective testing set to assess their accuracy, precision, recall, and F1 score in detecting DDoS attacks. In detecting a DDoS attack in IoT devices, a confusion matrix can be used to evaluate the performance of a machine learning-based detection system. Fig. 6 and Fig. 7 show the confusion matrix of all the classifiers without using PCA and using PCA, respectively. The confusion matrix has two classes: "attack" and "normal" traffic for each of the classifiers used, where True Positive (TP) represents the number of instances correctly classified as "attack" traffic, False Positive (FP) - Represents the number of instances incorrectly classified as "attack" traffic, but are regular traffic, True Negative (TN) - Represents the number of instances correctly classified as "normal" traffic, False Negative (FN) - represents the number of instances incorrectly classified as "normal" traffic, but are an attack.

Further, we have calculated precision, recall, f1-score, accuracy, and kappa coefficient from the confusion matrix of all the classifiers with and without PCA. These parameters are defined as:

Accuracy: It is the number of times that attack flows were correctly labeled as "attack flows" (i.e., TP) and normal traffic flows were labeled as "normal flows" (i.e., TN). Mathematically, it can be calculated as

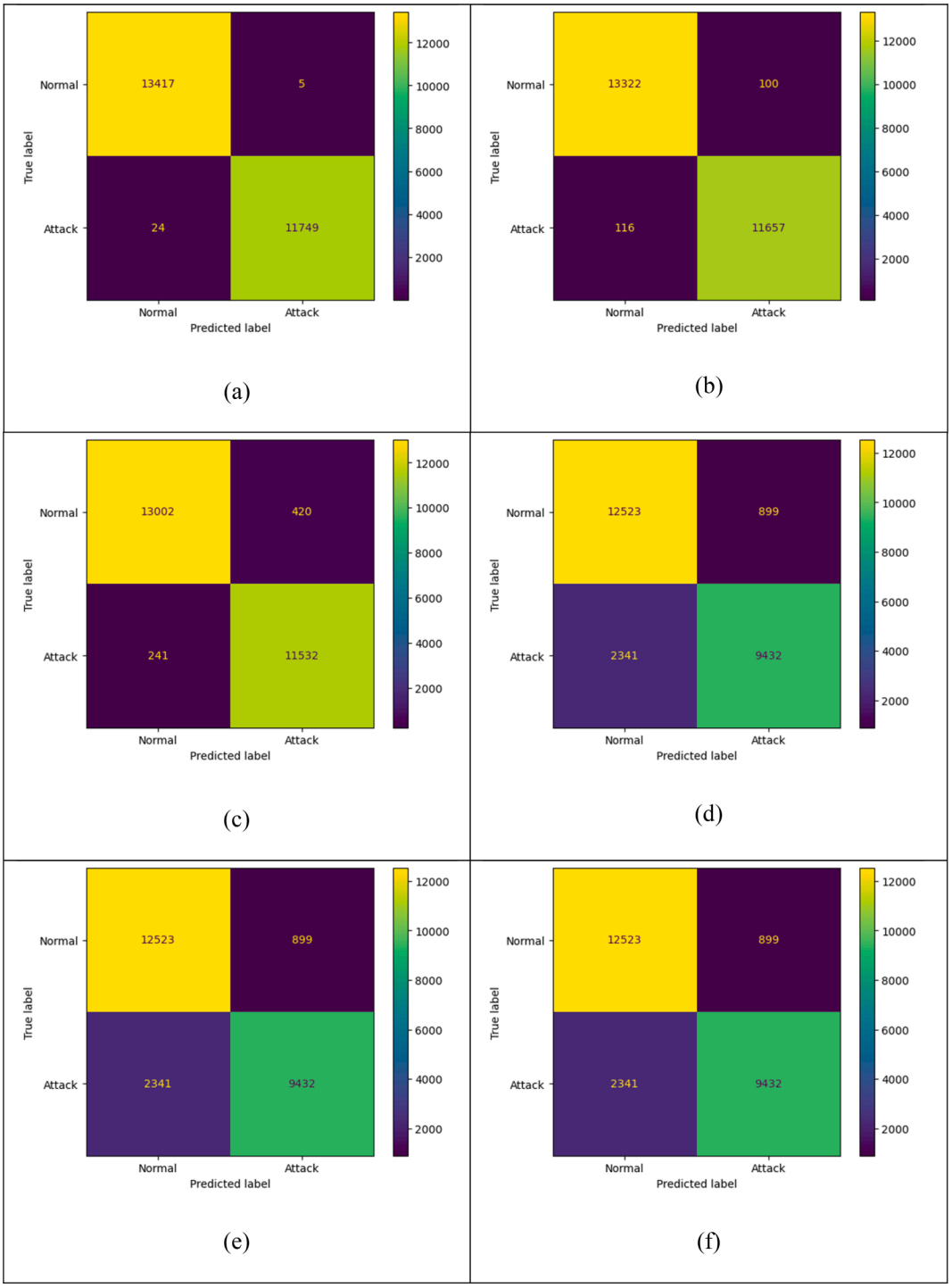

Fig. 7. Confusion matrix using PCA ((a) Random Forest, (b) KNN, (c) Decision Tree (d) Naïve Baye, (e) SVM, (f) Logistic Regression).

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

Precision: It says how many of the predicted attack flows were right. Mathematically, it can be calculated as

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

Recall - It shows how well the system can find the attack when it happens. Mathematically, it can be calculated as

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

F1-Score - It is the weighted harmonic mean of accuracy and recall. Mathematically, it can be calculated as

$$\text{f1}-\text{score} = 2^{*}(\text{precision} * \text{recall})/(\text{precision}+\text{recall}) \quad (4)$$

The kappa coefficient - also known as Cohen's kappa, is a statistical measure that assesses the agreement between two raters or classifiers, considering the agreement that could occur by chance alone. It can be calculated as

$$\kappa = (\text{Po}-\text{Pe})/(1-\text{Pe}) \quad (5)$$

Table 1 and Table 2 show performance comparisons of ML Classifiers without using PCA and using PCA, respectively. Table 1 shows that the Random Forest and K-Nearest Neighbour classifiers demonstrated high

Table 1
Performance Comparision of ML Classifiers without using PCA.

	Precision	Recall	F1-Score	Accuracy	Kappa
Random Forest	0.9970	0.9987	0.9978	0.9980	0.9245
K-Nearest Neighbour	0.9898	0.9914	0.9906	0.9912	0.9964
Decision Tree	0.9521	0.9669	0.9595	0.9624	0.9824
Support Vector Machines Linear	0.9772	0.8753	0.9234	0.9243	0.7847
Logistic Regression	0.9123	0.8649	0.8880	0.8924	0.8923
Naïve Bayes	0.1903	0.8093	0.3082	0.6007	0.1584

Table 2
Performance Comparision of ML Classifiers using PCA.

	Precision	Recall	F1-Score	Accuracy	Kappa
Random Forest	0.9979	0.9994	0.9986	0.9987	0.9473
K-Nearest Neighbour	0.9901	0.9914	0.9908	0.9914	0.9976
Decision Tree	0.9795	0.9648	0.9721	0.9737	0.9827
Support Vector Machines Linear	0.9446	0.9865	0.9651	0.9680	0.8013
Logistic Regression	0.9041	0.8861	0.8951	0.9009	0.9371
Naïve Bayes	0.8011	0.9129	0.8534	0.8714	0.7397

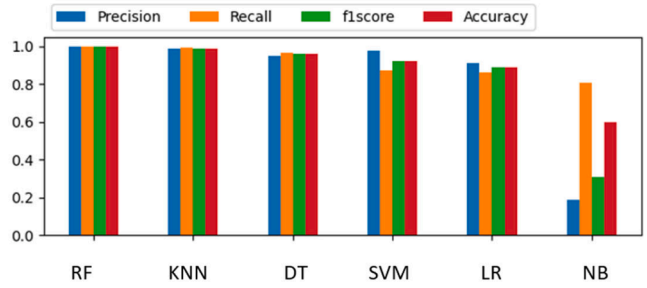

Fig. 8. Bar graph of different ML Classifiers without using PCA.

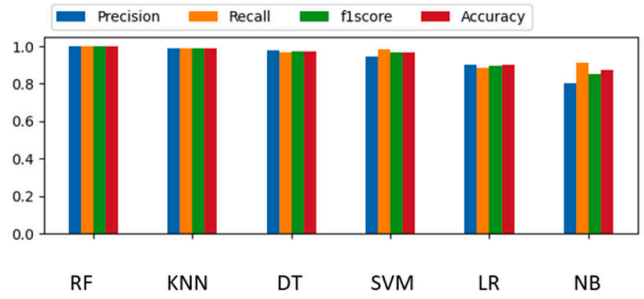

Fig. 9. Bar graph of different ML Classifiers using PCA.

precision, recall, F1-score, accuracy, and kappa coefficient. The Decision Tree classifier also performed reasonably well but showed slightly lower performance than the top-performing algorithms. The Support Vector Machines (Linear) and Logistic Regression models exhibited moderate performance, while the Naïve Bayes classifier showed notably lower performance. On the other hand, when PCA is applied, the overall performance of the classifiers improves. Table 2 shows that the Random Forest classifier exhibits the highest precision, recall, F1-score, and accuracy values. The K-Nearest Neighbour and Decision Tree classifiers show good performance. However, the Support Vector Machines Linear, Logistic Regression, and Naïve Bayes classifiers demonstrate relatively lower performance. These findings highlight the effectiveness of PCA in enhancing the performance and agreement of classifiers for DDoS attack detection, with the Random Forest classifier consistently demonstrating

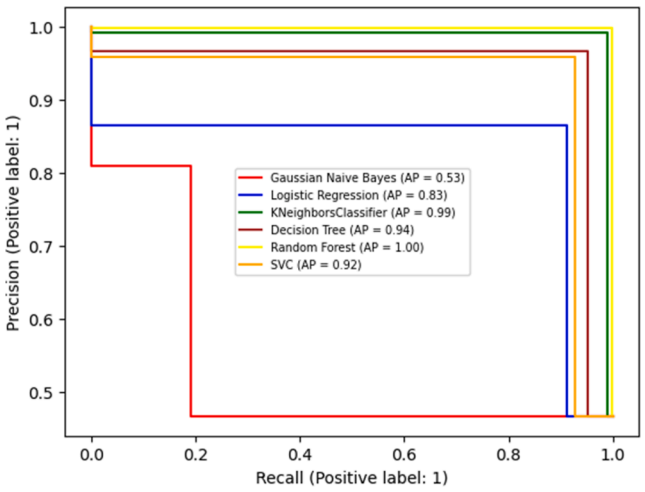

Fig. 10. Precision-Recall Curve without using PCA.

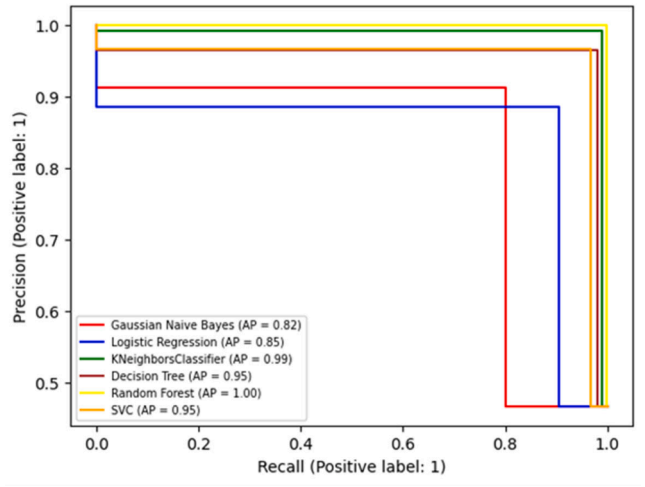

Fig. 11. Precision-Recall Curve using PCA.

strong performance and high agreement across both scenarios. The visual representation of Table 1 and Table 2 are shown in Fig. 8 and Fig. 9, respectively. The x-axis represents the classifiers, while the y-axis represents the metric scores. Each classifier is represented by a bar, and the height of the bar corresponds to the value of the specific metric. This bar graph provides a comparative overview of the performance of each classifier.

The precision-recall curves provide valuable insights into the performance of the classifiers with and without PCA, as shown in Fig. 10 and Fig. 11, respectively. The average precision (AP) values for each classifier indicate the overall quality of the precision-recall trade-off. The higher average precision values generally indicate better classifier performance regarding precision and recall. The results show that all classifiers perform excellently, with Random Forest achieving an average precision of 1.00 in both scenarios. Comparing the results between the two scenarios, it can be observed that using PCA generally improves the performance of the classifiers in terms of average precision. Gaussian Naive Bayes, Logistic Regression, and Support Vector Machines exhibit notable improvements with the inclusion of PCA, while K-Nearest Neighbours and Decision Tree maintain similar performance levels.

Fig. 12 and Fig. 13 represent the ROC curve, without PCA and with PCA, and give information about how well the classifiers work. Most of the time, a better trade-off between precision and recall is shown by a

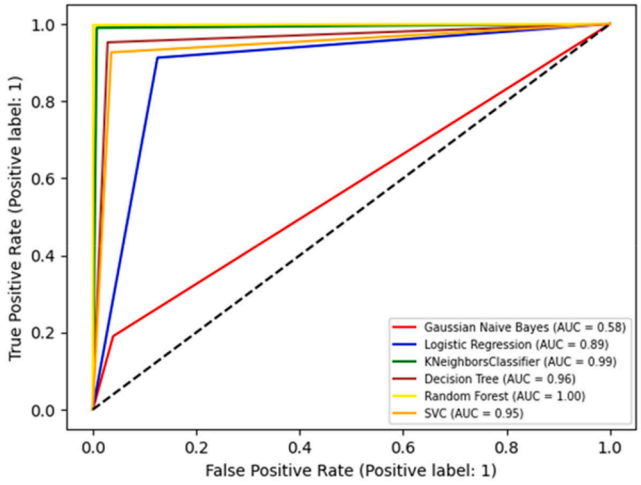

Fig. 12. ROC Curve without using PCA.

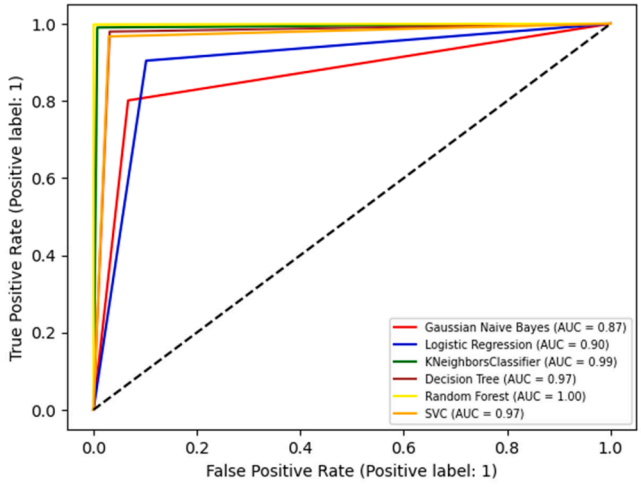

Fig. 13. ROC Curve using PCA.

higher AUC-PR number. Based on the results, it's clear that all models do a great job, with Random Forest always getting an AUC of 1.00 in both cases. When the results of the two cases are compared, it can be seen that using PCA improves the AUC-PR values of the models in general. Gaussian Naive Bayes, Logistic Regression, and Support Vector Machines improve when PCA is added, but K-Nearest Neighbours and Decision Tree don't change much.

Overall, the results suggest that incorporating PCA as a preprocessing step enhances the precision-recall trade-off for DDoS attack detection in IoT. It provides more effective feature representation, leading to improved classifier performance, particularly for algorithms that rely on linear separation or probabilistic modeling.

6. Conclusion

In this research, we examine the NSL-KDD dataset and how machine learning algorithms can detect distributed denial of service (DDoS) assaults on Internet of Things (IoT) devices. The research looked at how well six different ML classifiers could identify DDoS attacks. We tested the classifiers both with and without principal component analysis (PCA) applied first. When it came to correctly identifying DDoS attacks, the results demonstrated that the Random Forest classifier routinely attained the greatest values for precision, recall, F1-score, accuracy, and kappa coefficients. While Naïve Bayes showed relatively poor performance, the K-Nearest Neighbour and Decision Tree classifiers both showed strong results. In most cases, PCA enhanced the classifiers' performance, which in turn increased their accuracy, recall, F1-score, precision, and kappa coefficient values. Our research shows that machine learning classifiers are effective at identifying distributed denial of service (DDoS) assaults on internet of things (IoT) devices. Specifically, we found that Random Forest classifier and principal component analysis (PCA) as a feature selection method performed quite well in this regard.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number RGP2/393/44.

References

[1] Ali I, Ahmed AIA, Almogren A, Raza MA, Shah SA, Khan A, et al. Systematic literature review on IoT-based botnet attack. IEEE Access 2020;8:212220–32.
[2] Hussain F, Abbas SG, Shah GA, Pires IM, Fayyaz UU, Shahzad F, et al. A framework for malicious traffic detection in IoT healthcare environment. Sensors 2021;21(9):3025.
[3] Ghazanfar S, Hussain F, Rehman AU, Fayyaz UU, Shahzad F, Shah GA. Iot-flock: An open-source framework for iot traffic generation. In: 2020 International Conference on Emerging Trends in Smart Technologies (ICETST). IEEE; 2020. p. 1–6.
[4] Hussain F, Abbas SG, Husnain M, Fayyaz UU, Shahzad F, Shah GA. IoT DoS and DDoS attack detection using ResNet. In: 2020 IEEE 23rd International Multitopic Conference (INMIC). IEEE; 2020. p. 1–6.
[5] Sangodoyin AO, Akinsolu MO, Pillai P, Grout V. Detection and classification of DDoS flooding attacks on software-defined networks: a case study for the application of machine learning. IEEE Access 2021;9:122495–508.
[6] Soe YN, Feng Y, Santosa PI, Hartanto R, Sakurai K. Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors 2020;20(16):4372.
[7] Aljuhani A. Machine learning approaches for combating distributed denial of service attacks in modern networking environments. IEEE Access 2021;9:42236–64.
[8] Ali F, Sarwar S, Shafi QM, Iqbal M, Safyan M, Qayyum ZU. Securing IoT Based Maritime Transportation System Through Entropy-Based Dual-Stack Machine Learning Framework. IEEE Trans Intell Transp Syst 2022.
[9] Malik M, Dutta M. Feature engineering and machine learning framework for DDoS attack detection in the standardized internet of things. IEEE Internet Things J 2023.
[10] Sattari F, Farooqi AH, Qadir Z, Raza B, Nazari H, Almutiry M. A hybrid deep learning approach for bottleneck detection in IoT. IEEE Access 2022;10:77039–53.
[11] Hussain F, Abbas SG, Pires IM, Tanveer S, Fayyaz UU, Garcia NM, et al. A two-fold machine learning approach to prevent and detect IoT botnet attacks. IEEE Access 2021;9:163412–30.
[12] Chavan N, Kukreja M, Jagwani G, Nishad N, Deb N. DDoS Attack Detection and Botnet Prevention using Machine Learning. In: 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE; 2022. p. 1159–63.
[13] Chen X, Chen Y, Feng W, Xiao L, Li X, Zhang J, et al. Real-time DDoS defense in 5G-enabled IoT: A multidomain collaboration perspective. IEEE Internet Things J 2022.
[14] Liu Y, Tsang KF, Wu CK, Wei Y, Wang H, Zhu H. IEEE P2668-compliant multi-layer IoT-DDoS defense system using deep reinforcement learning. IEEE Trans Consum Electron 2022.
[15] Chen X, Xiao L, Feng W, Ge N, Wang X. DDoS defense for IoT: A Stackelberg game model-enabled collaborative framework. IEEE Internet Things J 2021;9(12):9659–74.
[16] Hussain B, Du Q, Sun B, Han Z. Deep learning-based DDoS-attack detection for cyber–physical system over 5G network. IEEE Trans Ind Inf 2020;17(2):860–70.
[17] Bousalem B, Silva VF, Langar R, Cherrier S. Deep learning-based approach for DDoS attacks detection and mitigation in 5G and beyond mobile networks. In: *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. IEEE; 2022. p. 228–30.
[18] Alghazzawi D, Bamasag O, Ullah H, Asghar MZ. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. Appl Sci 2021;11(24):11634.
[19] Duan G, Lv H, Wang H, Feng G. Application of a dynamic line graph neural network for intrusion detection with semisupervised learning. IEEE Trans Inf Forensics Secur 2022;18:699–714.

[20] Ravi N, Shalinie SM. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. IEEE Internet Things J 2020;7(4):3559–70.
[21] Sahoo KS, Tripathy BK, Naik K, Ramasubbareddy S, Balusamy B, Khari M, et al. An evolutionary SVM model for DDOS attack detection in software defined networks. IEEE Access 2020;8:132502–13.
[22] Perez-Diaz JA, Valdovinos IA, Choo KKR, Zhu D. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access 2020;8:155859–72.
[23] Zhijun W, Qing X, Jingjie W, Meng Y, Liang L. Low-rate DDoS attack detection based on factorization machine in software defined network. IEEE Access 2020;8:17404–18.
[24] Tan L, Pan Y, Wu J, Zhou J, Jiang H, Deng Y. A new framework for DDoS attack detection and defense in SDN environment. IEEE Access 2020;8:161908–19.
[25] El Sayed MS, Le-Khac NA, Azer MA, Jurcut AD. A flow-based anomaly detection approach with feature selection method against DDoS attacks in SDNs. IEEE Trans Cognit Commun Networking 2022;8(4):1862–80.
[26] Tushir B, Dalal Y, Dezfouli B, Liu Y. A quantitative study of ddos and e-ddos attacks on wifi smart home devices. IEEE Internet Things J 2020;8(8):6282–92.
[27] Vlajic N, Zhou D. IoT as a land of opportunity for DDoS hackers. Computer 2018;51(7):26–34.
[28] Doshi K, Yilmaz Y, Uludag S. Timely detection and mitigation of stealthy DDoS attacks via IoT networks. IEEE Trans Dependable Secure Comput 2021;18(5):2164–76.
[29] Li Z, Kong Y, Wang C, Jiang C. DDoS mitigation based on space-time flow regularities in IoV: A feature adaption reinforcement learning approach. IEEE Trans Intell Transp Syst 2021;23(3):2262–78.
[30] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE; 2009. p. 1–6.