International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 15 Issue 03 Year 2023

Detection of Distributed Denial of Service Attacks in Software Defined Networks by Using Machine Learning

Musmuharam*
Master, Computer Science Department ,BINUS Graduate Program, Computer Science, Bina Nusantara University, Jakarta 10480, Indonesia
musmuharam@binus.ac.id

Suharjito
Doctor, Industrial Engineering Department, BINUS Graduate Program, Bina Nusantara University, Jakarta 11480, Indonesia
suharjito@binus.edu

Article History

Received: 23 July 2023
Revised: 14 October 2023
Accepted: 15 November 2023

Abstract

Within the sphere of Software-Defined Networking (SDN) — an innovative architectural paradigm that segregates the control plane from the data plane — a paramount concern is the defense against Distributed Denial of Service (DDoS) assaults. These attacks pose a significant threat to the integrity and operational sustainability of SDN infrastructures, potentially leading to extensive system disruptions and financial losses.To address this challenge, our study introduces an innovative approach utilizing machine learning strategies to enhance the detection of DDoS threats. We employed a trio of classification algorithms: Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), applied to a publicly available SDN dataset specific to DDoS attacks. Our methodology integrates a blend of feature selection techniques, including Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and t-Distributed Stochastic Neighbor Embedding (t-SNE), with the aim of refining the accuracy of our classifications.In a comparative analysis with existing models, our innovative application of KNN in conjunction with RFE demonstrated exceptional performance, achieving an accuracy of 99.97%, a precision of 99.98%, a recall of 99.96%, and an F1-score of 99.97%. This breakthrough indicates a significant advancement in the field of SDN security.

Keywords: SDN, DDoS, Machine Learning, Feature Selection

1. Introduction

Software-defined networking (SDN) is the arrangement of functions between the control plan and data plan; this is what distinguishes it from traditional networks. In software-defined network (SDN) devices such as switches, routers are only used to carry out packet forwarding, where decision makers and control logic capabilities are in the SDN controller software. SDN technology has great advantages over traditional networks, and the development of Software-Defined Networking (SDN) has made significant progress in meeting organizational needs for operational efficiency. However, SDN also faces potential problems and significant challenges. Security challenges, such as distributed denial of service attacks, are the main difficulty facing SDN [1]. A new network architecture that divides the control and data layers is present in software-defined networks. SDN provides centralized and programmable network control facilities. However, the centralized control feature in SDN also has drawbacks, such as a single point of failure. Additionally, during DDoS attacks, the processing and traffic capacity of SDN controllers can become overloaded [2]. In the first quarter of 2020, Amazon Web Services (AWS) reported that there were 2.3 Tbps of DDoS attacks. Types of DDoS attacks detected by AWS Shield on the network and Web application layer, such as UDP reflection vectors, DNS reflection, TCP attacks, and SYN floods. The attack was the most massive DDoS attack of 2020 against the Amazon website [3].

There are four categories of security problems in software-defined networks: forwarding device attacks, where network traffic is disrupted due to DDoS attacks, which can result in process failure. Threat in the control plane: the use of centralized control can result in failure of the control process. Furthermore, there is a vulnerability in communication channels and Fake Traffic flows where attackers can launch DDoS attacks to eliminate resources on forwarding devices or controls. DDoS attacks have become the preferred tool for hackers due to their consistent threat to users, organizations, and internet infrastructure [4]. Consequently, safeguarding controllers from disruptive attacks like DDoS, capable of service disruptions, is both crucial and time-sensitive. Despite the implementation of various methods to detect DDoS attacks, such as traditional firewall defenses, network analysis, and protocol scrutiny aimed at recognizing unusual traffic patterns, these approaches encounter several challenges and limitations. One major drawback is the constantly evolving nature of DDoS attacks, as attackers continually employ new techniques and methods. Examples of these attacks include botnets, which are used to execute DDoS attacks [5].

Machine learning (ML) refers to a subfield within the domain of Artificial Intelligence (AI) that is devoted to enabling computers to operate without explicit programming for every conceivable scenario. The fundamental essence of ML lies in the creation of algorithms capable of autonomous learning through exposure to extensive datasets or inputs [6]. Processing the dataset is crucial as it involves selecting relevant features. The goal is to improve the efficiency and accuracy of the classification model by using only the best features. Feature selection becomes very important in the development of classification models because it can minimize overfitting, improve model interpretation, and save time and resources in data processing. After the best feature subset is achieved, the training dataset will be reduced using only the relevant features so as to increase the efficiency and accuracy of the classification model by using only the important features [7].

In this research, we concentrate on the utilization of the "DDoS attack SDN Dataset," which consists of 104,345 records categorized into benign and attack traffic, featuring 23 distinct attributes [8]. Our methodology revolves around the implementation of classification algorithms, specifically RF, SVM, and KNN. We have chosen these algorithms due to their demonstrated effectiveness in prior studies, as referenced in [2],[9], particularly in classification scenarios. The primary goal of our study is to determine the most efficient model for detecting DDoS attacks within SDN environments. To accomplish this objective, we conduct a series of experiments that incorporate various feature selection techniques, namely RFE, PCA, and t-SNE. These techniques are deployed to enhance our model's performance by identifying the most crucial features within the dataset. Additionally, we use the SDN dataset as a benchmark to compare our proposed model against existing methodologies. Our results reveal a notable improvement in the reliability of DDoS attack classification detection.

2. Related Works

A number of studies have already been done to assess how well machine learning works for detecting DDoS attacks. For example, In [2] , they conducted experiments to compare the use of feature selection methods with and without feature selection in detecting DDoS attacks. They focused on three variants of DDoS attacks: TCP Flood Attack, UDP Flood Attack, and ICMP Flood Attack. From their research findings, it was discovered that using the Sequential Forward Floating Selection (SFFS) feature selection technique resulted in the highest accuracy of 98.30% using the K-Nearest Neighbors algorithm model.

In the study conducted by [10] , a model that is suggested that uses two machine learning techniques, Polynomial SVM and linear SVM, DDoS assaults in Software-Defined Networks (SDN) were categorized using this. The proposed system generated using the Python code and Scapy packet creation tool in an SDN simulation, UDP Flooding attack traffic and regular traffic are produced. The results of the proposed system were 95% accurate in classifying flood DDoS assaults using the Polynomial SVM algorithm.

In the study cited as reference [11], scholars investigated the application of machine learning methods, namely Decision Trees (DT) and Support Vector Machines (SVM), for the detection of DDoS attacks within SDN. The research involved creating datasets tailored for the SDN environment using Mininet, supplemented by the KDD99 dataset for training and testing purposes. These datasets were bifurcated into two categories: 'attack' (labeled as 1) and 'non-attack' (labeled as 0). Based on the experimental outcomes, it was observed that the SVM algorithm outperformed with an accuracy rate of 85%, whereas the DT algorithm recorded an accuracy of 78%.

In the study referenced as [12], this research using utilization of dimensionality reduction methods, PCA and t-SNE, in the context of Software-Defined Network (SDN) environments.The primary objective was to streamline the data dimensionality to enhance the detection of DDoS attacks. For this purpose, the "DDoS attack SDN dataset" was employed. The study aimed to identify an optimal combination of feature representation methods with machine learning algorithms to boost the efficiency of DDoS attack detection in SDN frameworks. The findings according to the results of the studies, the highest accuracy was attained with the GB algorithm at 99.56% and with XDBoots at 98.25%.

In the study referenced as [8], the dataset used in the experiments is the result of a simulation using the SDN DDoS dataset, with the controller implemented through a Python application created with the help of the RYU API. The total number of records in the dataset is 104,345, and each record consists of 23 features. The dataset has two classes, where 0 represents the label for normal or benign traffic and 1 represents the label for abnormal or malicious traffic. During the SDN network simulation process, test data includes both legitimate TCP, UDP, and ICMP communication as well as malicious TCP sync, UDP Flood, and ICMP traffic. The study's results highlighted the effectiveness of a hybrid model combining Support SVM and RF, which exhibited superior performance with an impressive accuracy rate of 98.8%.

In the research outlined in [13], an innovative architecture was introduced, integrating Intrusion Prevention System (IPS) and Intrusion Detection System (IDS) modules directly into the SDN controller. This setup was further enhanced with the application of various machine learning algorithms, including J48, Random Tree, REP Tree, Random Forest, SVM, and Multilayer Perceptron (MLP). The effectiveness of this approach was evaluated using the CIC DoS 2019 dataset, where the MLP algorithm notably achieved an accuracy rate of up to 95%.

In the study documented as [14], this researchers focused on enhancing the security of Software-Defined Networks (SDN) against DDoS attacks through the development of machine learning methodologies, specifically leveraging a decision tree classification model. The primary aim was to bolster the resilience of SDN systems against intrusions by employing this model to discern attack traffic and categorize SDN traffic into 'attack' or 'normal' classes. Additionally, the research incorporated a genetic algorithm to refine the classification accuracy further.This process included dataset preparation and preprocessing, followed by the optimization of hyperparameters for the decision tree model using the genetic algorithm (GA). The proposed model, an evolutionary decision tree (EDT), was deployed for the delineation of network traffic into normative and assaultive categories. The outcomes of the experimental evaluations indicated that this model exhibited a notable classification precision rate of 99.46%.

The investigation referenced in [15] unfolds in a bifurcated approach, leveraging machine learning algorithms. Initially, the k-means algorithm is applied to distill the most salient features during the data preprocessing phase. Subsequently, the k-Nearest Neighbors (kNN) algorithm is employed to discern attack patterns utilizing the curated feature set in the detection phase.The model's remarkable accuracy rate of 98.85% and recall rate of 98.47% demonstrate its ability in precisely and consistently identifying attack flows.

In the research study referred to as [16], Experiments were carried out to compare the detection of DDoS assaults using six machine learning models: Logistic Regression (LR), Naive Bayes (NB), SVM, K-NN, Decision Tree (DT), and Random Forest (RF). The public dataset NSL KDD was utilized. Model evaluation criteria included Accuracy, Precision, Recall, F1 Score, and computation time. The final results revealed that the K-NN, DT, and RF models outperformed the competition in the most important performance indicator (F1 Score), with 0.98.

In the research detailed in [17], the focus was on identify unusual data traffic activity in the SDN controller. They suggest an ensemble approach that makes use of KNN, NB, SVM, and Self-Organizing Map (SOM) among other machine learning methods. The suggested method seeks to increase the precision and efficacy of anomaly detection in SDN systems by integrating the advantages of these methods. The experimental results showed that the suggested model used SVM-SOM to get a high classification accuracy of 98.12.

In the study presented as [18], researchers introduced a a machine learning-based and proxy-based TCP Flooding Attack Detection (ML-TFAD) method is suggested. The TFAD approach uses SYN and ACK are two proxies. While ACK defends against TCP ACK flood attacks, SYN defends against TCP SYN flood attacks. Before they reach the intended server, SYN flood attacks are detected by the ML-TFAD module using the C4.5 decision tree algorithm. The training of the suggested model uses the CAIDA 2007 DDoS dataset. The ML-TFAD aids in early attack detection before it reaches the server. The accuracy of the KNN model's outputs was 97.15%, whereas the accuracy of the C4.5 decision tree model was 97.43%.

In the research documented as [19], a methodology combining feature extraction methods and machine learning classification on SDN is used to detect DDoS attacks of SDN. The best features are extracted in this procedure and used for both training and testing classifications. Several common classification techniques are used, including Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbor, eXtreme Gradient Boosting (XGBoost), and Naive Bayes (NB). The evaluate of performance is the confusion matrix. According to the experiment's findings, the SVM classification had the highest accuracy, coming in at 99.38%.

In the research study referred to as [20], this research proposed filter-based, Fisher score-based, wrapper-based, and f-test analysis of variance (ANOVA) feature selection techniques to identify DDoS attacks on SDN controllers and to carry out optimization utilizing the entropy algorithm with the Renyi joint entropy algorithm. The dataset used in this study comprises 104,345 traffic flows and 23 attributes. The normal and attack traffic class labels are used to display the TCP, UDP, and ICMP traffic datasets. The machine learning classifiers employed in this analysis included ANN, XGBoost (XGB), SVM, and KNN. The experimental findings indicated that the ANN model outperformed the other classifiers, achieving an accuracy rate of 99.35%.

3. Methodology

This section describes the research procedures used by researchers to classify DDoS attacks on SDN using machine learning RF, SVM, and KNN approaches. In Figure 1, the steps of the method followed in developing a machine learning model. We conducted experiments without using any feature selection techniques, and then we repeated the experiments with feature selection techniques such as RFE, PCA, and t-SNE to choose the pertinent characteristics or minimize the data's dimensionality. The performance of the trained models was compared and examined using evaluation metrics like accuracy, precision, recall, and F1 score.

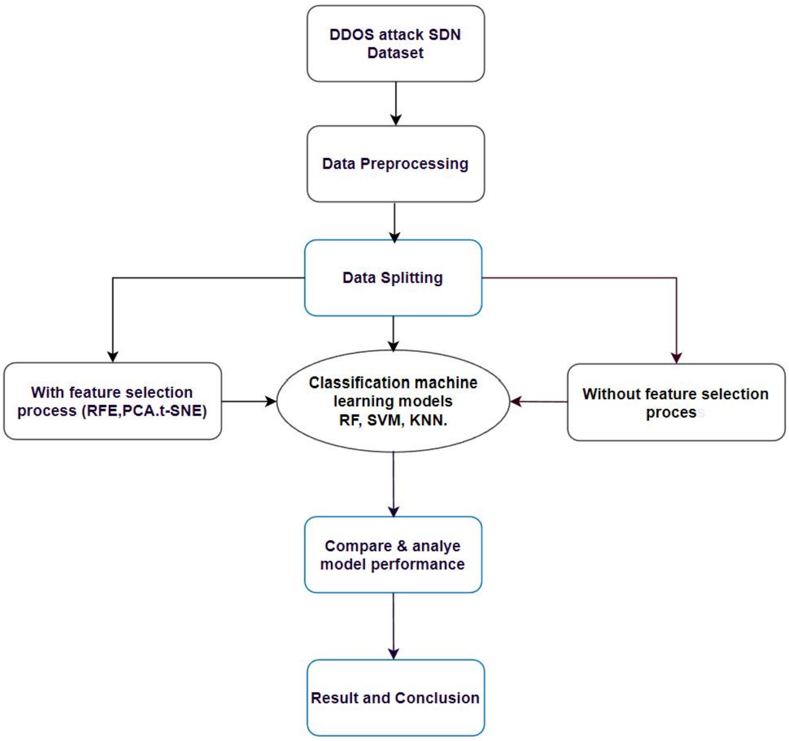
DDOS attack SDN Dataset
Data Preprocessing
Data Splitting
With feature selection process (RFE,PCA.t-SNE)
Classification machine learning models RF, SVM, KNN.
Without feature selection proces
Compare & analye model performance
Result and Conclusion

Figure 1. Research Framework

3.1 Dataset

Dataset is generated using emulator and RYU controller for generating DDoS Attack[10]. The dataset has total 1,04,345 record and consists of 23 features which divided into two classes. Class 0 represents normal or benign traffic, while class 1 represents abnormal or dangerous traffic. The typical traffic of the protocols TCP, UDP, and ICMP, as well as traffic related to TCP Syn attacks, UDP Flood attacks, and ICMP. The DDoS attack SDN Dataset characteristic is shown in Table 1. Table 2 shows the number of benign and malicious data based on traffic protocols.

Table 1. DDOS Attack SDN Dataset

Attribute Name	**Description**
dt	The timestamp of the data was captured
switch	datapath-id for the topology
src	The source address
dst	The destination address
pktcount	The packets contained in the flow
bytecount	The bytes transmitted in the flow
dur	The duration of the flow in seconds
dur_nsec	The duration of flow in nanoseconds
tot_dur	The total duration of the related flow
flows	The recorded number of flows
packetins	The total number of received packets
pktperflow	The typical flow's packet count
byteperflow	The typical amount of bytes per flow

pktrate	The packet rate per second
Pairflow	The associated pair flow
Protocol	The protocol used in the packet
port_no	The port number in the flow
tx_bytes	The volume of data that was sent
rx_bytes	The quantity of bytes obtained
tx_kbps	The transmission speed in kbps
rx_kbps	The reception speed in kilobits per second
tot_kbps	The total speed in kilobits per second
label	The label or classification of data

Table 2. The Number of Benign and Malicious Data Based on Traffic Protocols

Traffic Protocols	**Benign**	**Malicious**
ICMP	31902	9419
UDP	15570	13866
TCP	16089	17499

The dataset used to train machine learning models can include many features. Some features have a large influence on the classification results, while others have little or no effect. Low-impact features in the classification process can impact overall model performance. The model becomes less accurate and less efficient at classifying data. Our goal is to select the most relevant of dataset and influential subset of features in the classification. Because of this, we can optimize model performance by focusing on relevant features.

3.2 Data Preprocessing

Prior to beginning the training of a machine learning model, the data must be preprocessed. The dataset was prepared using a variety of methods, such as preprocessing procedures to clean, handle missing values, transform data, normalize data, handle outliers to ensure it is ready for modeling, and other pre-processing techniques. The purpose of this data pre-processing is to optimally prepare the data to align with the analysis requirements and to quickly running program, maximize the performance of the model that will be developed.

3.3 Recursive Feature Elimination (RFE)

RFE is a method employed in feature selection to identify the most vital subset of features, thereby reducing data dimensionality. This technique relies on the outcomes of a specific algorithm to determine the optimal number of features to be selected. The process of RFE involves iteratively removing less significant features to focus on the most impactful ones. Using RFE can improve model performance by focusing on the most informative features. The detailed procedure of RFE for feature selection is illustrated in Figure 2.

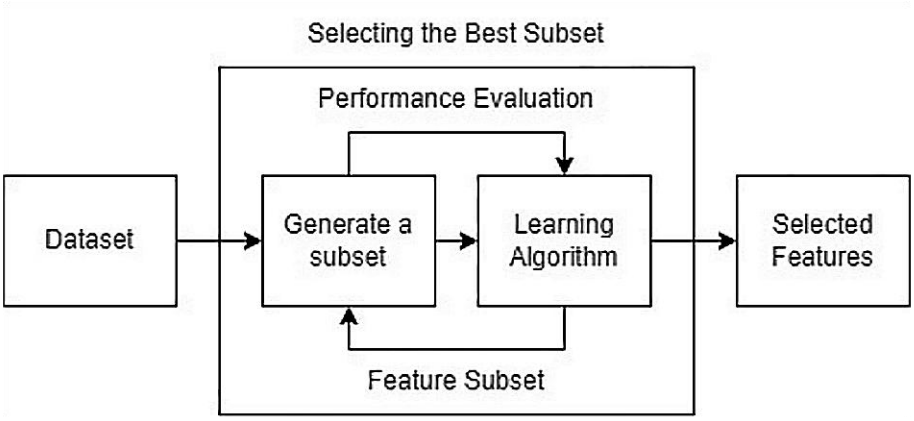

Figure 2. Recursive Feature Elimination Algorithm

The RFE approach is used to eliminate irrelevant features from the training dataset based on feature selection, remove the least important features, and rebuild the model while recalculating the importance level of features. The RFE approach is used to eliminate irrelevant features from the training dataset, remove the least important features, rebuild the model, and recalculate the feature importance levels [21].

3.4 Principal Component Analysis (PCA)

PCA is a linear machine learning approach, is frequently used to decrease the dimensionality of datasets. While reducing dimensions can have some impact on the model's accuracy, it also helps reduce computational complexity, resulting in faster execution of machine learning algorithms. The PCA technique will form a new set of dimensions, which are then ranked based on the variance of the data The covariance matrix is analyzed to get the corresponding eigenvalues, which are then used to calculate the eigenvectors. The k eigenvectors with the highest eigenvalues are selected, determining the dimensions of the new dataset [22].

3.5 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a machine learning technique used for visualizing and reducing the dimensionality of complex data. Model interpretation and visualization can be done with ease and effectiveness using t-SNE [23] . Transforming high-dimensional data into a more condensed and intelligible representation is the core objective of t-SNE. t-SNE works by transforming the distances between data points in the original space into probabilities of similarity distributions between data pairs. Subsequently, t-SNE maps the data into a lower-dimensional space while aiming to preserve the similarity distribution probabilities as closely as possible.

3.6 Random Forest (RF)

RF is an ensemble method that utilizes decision trees, with each tree in the ensemble relying on a random subset of the chosen variables. During classification, predictions are generated by aggregating the majority of results from these trees. The classification process using Random Forest involves combining trees, and the more trees used, the better the accuracy achieved. The classification process begins by randomly dividing the sample data into Decision Trees. To determine the classification results, voting is conducted based on the outcomes of each tree. The constructed trees vote in a majority-only fashion to determine the classification outcome. The outcome is then generated using a random forest, which incorporates the outcomes from each decision tree [24].

3.7 Support Vector Machine

A popular classification algorithm in machine learning is SVM. It aims to find the optimal linear hyperplane that maximizes the margin between two classes. By identifying the largest margin between the classes, SVM effectively separates the data points and makes accurate predictios [25]. The versatility of SVM in handling diverse types of data and delivering high performance has established it as one of the most populer algorithms in the realm of data modeling and analysis. SVM finds extensive usage in classification modeling, primarily due to its computational efficiency and high accuracy [26].

3.8 k-Nearest Neighbors (KNN)

The KNN algorithm is an easy-to-use supervised machine learning method that can be used to resolve classification and regression issues. When new data is received, it decides its class by examining its nearest K neighbors [27] . The space is divided into partitions that represent the learning data criteria. Each piece of learning data is represented as k points in a high-dimensional space. By utilizing the K-NN technique, new data can be rapidly classified into the most relevant category. The space is divided into partitions that represent the learning data criteria. Each piece of learning data is represented as k points in a high-dimensional space. The K-NN technique can be used to quickly classify new data into the most pertinent category.

3.9 Model Evaluation

To gauge the performance of our classification model, we focused on measuring its classification accuracy. A common tool for this assessment is the confusion matrix, which provides insights into the effectiveness of the classifier. This matrix helps in distinguishing between the correctly classified instances and the misclassified ones, thereby revealing the count of true positive and false positive predictions. Utilizing these counts, we calculated various metrics such as accuracy, error rate, among others. These calculations were based on standard formulas, similar to those mentioned in [28], and are detailed in Equation 1, 2, 3, and 4:

$$\text{Accuracy:} \quad \mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{Total}} \tag{1}$$

$$\text{Precision:} \quad \mathrm{Prec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$\text{Recall:} \quad \mathrm{Rec} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$\text{F1-score:} \quad \mathrm{F1} = 2 * \frac{\mathrm{Prec}\,.\,\mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \tag{4}$$

True Positives (TP) are instances wherein the model correctly identifies positive cases. False Positives (FP) refer to the count of negative instances erroneously classified as positive by the model. Conversely, False Negatives (FN) denote the number of positive cases that the model wrongly labels as negative. True Negatives (TN) represent the correct model predictions of negative instances, as explicated in [29].

4. Results and Discussion

4.1 Experiment Setup

For the experimental setup of this study, we utilized computer resources to handle data processing efficiently. The computer system employed for this purpose was equipped with an AMD Ryzen 7 5800U processor, featuring 8 CPUs, and 16 GB of memory operating at up to 3200MHz DDR4 and using the Anaconda navigator application, Jupyter Notebook, and various Python libraries such as NumPy, pandas, and scikit-learn for data processing and creating machine learning models. The SDN DDoS attack dataset is divided into two parts, with an 80% allocation for training and a 20% allocation for testing, as mentioned in [30].

4.2 Result Analysis

In this research, we are two experiments will be conducted. The first experiment involves model classification without the use of feature selection techniques. In the second experiment, RFE, PCA, and t-SNE feature selection techniques will be employed. The results of both processes will be evaluated to determine the algorithm combination that yields the best performance. Table 3 presents the result of experiments conducted using machine learning techniques utilizing RF, SVM, and KNN algorithms without feature selection in the testing model.

Table 3. Classification Result without Feature Selection in the Testing Model

Method	*Accuracy*	*Precision*	*Recall*	*F1-Score*
RF	95.59%	93.14%	95.57%	94.34%
SVM	94.59%	91.33%	94.93%	93.10%

KNN	97.78%	97.55%	96.57%	97.55%

We employ a RandomizedSearchCV for hyperparameter tuning to explore various hyperparameter settings and identify the optimal combination that maximizes the classifier's performance, as illustrated in Table 4. In the second experiment, RFE, PCA, and t-SNE feature selection techniques will be employed. The results of the classifier's performance, as illustrated in Table 5.

Table 4. Analysis of Hyperparameters in Selected Machine Learning Algorithms Classifiers

Method	Hyperparameters	Accuracy	Precision	Recall	F1-Score
RF	max_depth 5	97.11%	93.36%	99.74%	96.36%
RF	max_depth 6	97.73%	94.76%	99.66%	97.15%
SVM	C 0.01 kernel rfb	93.20%	91.43%	90.81%	91.12%
SVM	C 0.1 keber rfb	97.28%	95.96%	97.02%	96.49%
KNN	Neighbors 5	99.97%	99.98%	99.96%	99.97%
KNN	Neighbors 8	99.86%	99.86%	99.78%	99.82%

Table 5. Classification Result with Feature Selection

Method	Accuracy	Precision	Recall	F1-Score
RF+RFE	97.73%	94.76%	99.66%	97.15%
RF+PCA	87.60%	80.76%	88.91%	84.64%
RF+t-SNE	97.78%	97.23%	97.07%	97.15%
SVM+RFE	97.28%	95.96%	97.02%	96.49%
SVM+PCA	92.52%	88.53%	92.53%	90.49%
SMV+t-SNE	97.92%	97.46%	97.18%	97.32%
KNN+RFE	99.97%	99.98%	99.96%	99.97%
KNN+PCA	96.93%	96.13%	96.75%	96.70%
KNN+t-SNE	97.50%	97.14%	96.42%	96.78%

This study's results significantly contribute to the creation of an effective classification model focused on maximizing accuracy. The analysis of the performance, illustrated through confusion matrices in Figure 3 and 4, visually demonstrates the efficacy of the classification models used, particularly emphasizing the K-Nearest Neighbors (KNN) model. In this model, the number of neighbors was fixed at 5. The application of Recursive Feature Elimination (RFE) enabled us to pinpoint the top 5 most influential features: flows, bytecount, dur_nsec, dur, and pktcount, as depicted in Figure 5. These features were chosen for their pivotal role in boosting the accuracy and efficiency of our model, specifically in the detection and classification of DDoS attacks.

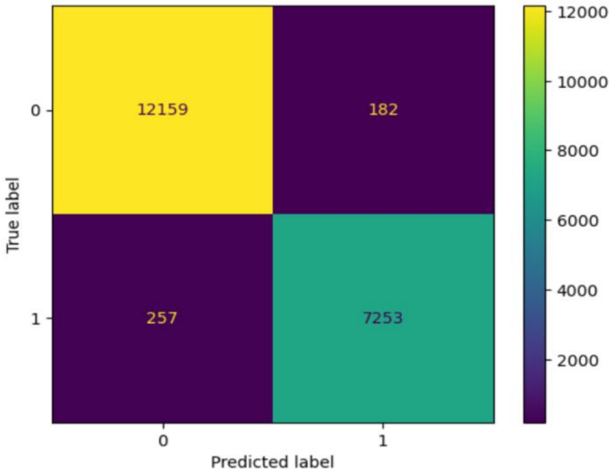

Figure 3. KNN Confusion Matrix without Selection Feature

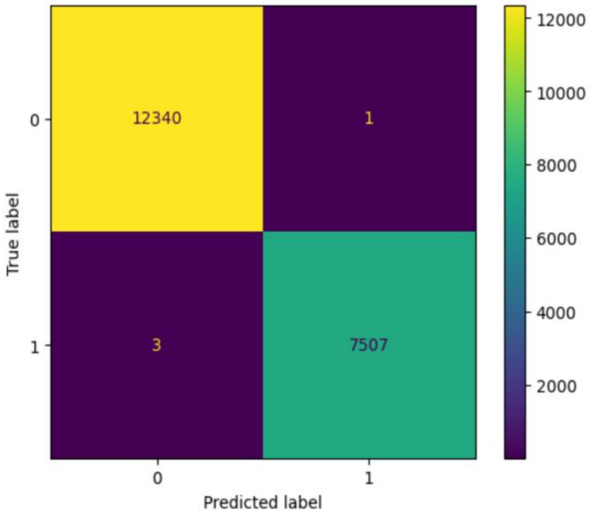

Figure 4. KNN Confusion Matrix with RFE Feature Selection

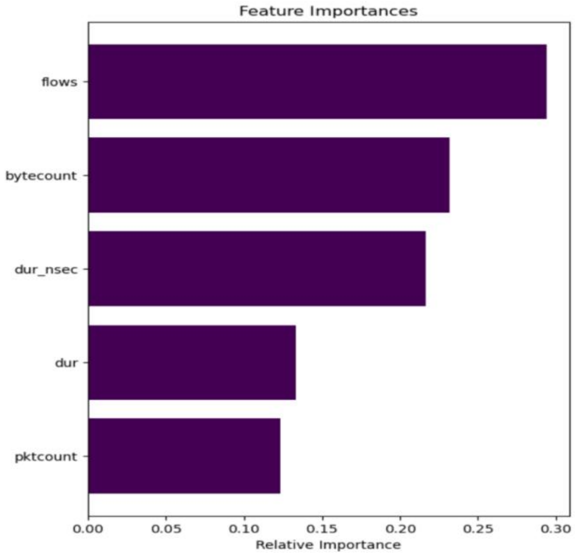

Figure 5. RFE Feature Importance Result

Following the implementation of Recursive Feature Elimination (RFE), we identified the five most critical features for our analysis: Flows, Bytecount, Dur_nsec, Dur, and Pktcount. The results of this experiment show that the use of the RFE technique in feature selection has succeeded in identifying features that have a significant contribution in improving classification performance. By combining the KNN and RFE methods, the resulting evaluation metrics show high levels of accuracy, precision, recall and f1-score, achieving accuracy results of 99.97%, precision 99.98%, recall 99.96% and f1-score 99.97%.

4.3 Comparative Analysis with Existing Results

In evaluating the effectiveness of the approach proposed in this study, we conducted a comprehensive comparative analysis, as detailed in Table 6. This analysis involved comparing our proposed model with previous research in the domain of DDoS attack detection using simulated datasets. The benchmark results from existing studies showed a maximum accuracy of 99.46%. However, our recent advancements, particularly in accuracy, precision, and F1-score, mark a significant improvement. Notably, our model, which combines the K-Nearest Neighbors (KNN) algorithm with Recursive Feature Elimination (RFE) for feature selection, surpasses all previous models, achieving an unparalleled accuracy rate of 99.97%.

Table 6. The Result of Testing Model Comparison

Ref.& Year	Accuracy	Precision	Recall	F1-Score
[8] 2021	98.8%	98.27%	97.91%	97.65%
[14] 2022	99.46%	99.19%	NA	99.42%
[20] 2023	99.35%	98.32%	NA	97.85%
Proposed	99.97%	99.98%	99.96%	99.97%

Table 6 presents a comparative analysis of state-of-the-art results in this field of research. In the study by Ahuja et al. [8], 23 features were extracted from simulated SDN environments, encompassing UDP, TCP, and ICMP attacks, as well as regular traffic. From these, eight key features were identified. Their model achieved 98.8% accuracy, 98.27% precision, 97.91% recall, and 97.65% f1-score. Comparatively, recent advancements in this area have shown improvements in accuracy, precision, recall, and f1-score by 1.17%, 1.71%, 2.05%, and 1.32%, respectively.

In their inquiry, Kamel et al. [14] utilized a genetic algorithm (GA) to finetune the hyperparameters of a decision tree classifier to enhance its efficacy. The GA was configured with 15 generations, a population of 10 individuals, mutation and crossover probabilities set at 0.10 and 0.50 respectively, and a tournament selection size of 3. The data partitioning involved allocating 70% for training and 30% for validation purposes. The refined model demonstrated a 99.46% accuracy, a 99.19% precision, and an F1-score of 99.42%. Recent progress in the field has, however, led to improvements on these metrics, with marginal yet notable increments of 0.51% in accuracy, 0.79% in precision, and 0.55% in F1-score.

In their investigation, Wang et al. [20] harnessed a suite of feature selection techniques to identify DDoS incursions targeting SDN controllers. These techniques included filter, Fisher-score, wrapper methods, and f-test ANOVA, all refined through optimization via the Renyi entropy algorithm. They conducted their trials using an ANN classifier with a configuration of 10 hidden neurons and 14 optimally chosen features. The model culminated in a 99.35% accuracy, a precision of 98.32%, and an F1-score of 97.85%. Notwithstanding, contemporary enhancements have yielded incremental increases in these metrics, with accuracy, precision, and F1-score experiencing elevations of 0.62%, 1.66%, and 2.12%, respectively, over these results.

5. Conclusion

In this study, we conducted an analysis of an SDN-based DDoS detection system by employing machine learning techniques, RF, SVM, and KNN.We did two experiments. In the first proposed approach, we did not employ feature selection techniques, resulting in an accuracy of 97.78%. Subsequently, we proceeded with the second approach, which involved using the Recursive Feature Elimination (RFE) technique and achieved an accuracy of 99.97% using the K-Nearest Neighbors (KNN) algorithm with a number of neighbors of 5 based on table 5, The result of testing the comparability of Recent work has improved in terms of Accuracy, precision, recall, and F1-value. In the future we need to test our proposed method using real-world data. It is also recommended to consider the utilization of an alternative SDN dataset by executing real-time traffic generation specifically designed to emulate DDoS attacks. Furthermore, an exploration of alternative models, such as deep learning, is highly advisable. The profound advantage of deep learning lies in its innate ability to autonomously extract salient features and adeptly handle the intricacies associated with high-dimensional and complex datasets. By incorporating deep learning techniques into the analysis, the improved feature extraction capabilities can potentially yield substantial enhancements in detecting and mitigating DDoS attacks.

References

[1] R. N. Carvalho, L. R. Costa, J. L. Bordim, and E. A. P. Alchieri, "Enhancing an SDN architecture with DoS attack detection mechanisms," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 2, pp. 215-224, 2020, doi: 10.25046/aj050228.

[2] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustainability*, vol. 12, no. 3, p. 1035, Feb. 2020, doi: 10.3390/su12031035.

[3] B. Nugraha and R. N. Murthy, “Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks,” *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2020.

[4] S. Ejaz, Z. Iqbal, P. Azmat Shah, B. H. Bukhari, A. Ali, and F. Aadil, “ Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function,” *IEEE Access,* vol. 7, pp. 46646-46658, 2019.

[5] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, “Performance evaluation of Botnet DDoS attack detection using machine learning,” *Evol Intell,* vol. 13, no. 2, pp. 283-294, Jun. 2020.

[6] A. Al-Nusirat, F. Hanandeh, M. Kharabsheh, M. Al-Ayyoub, and N. Al-Dhufairi, “Dynamic Detection of Software Defects Using Supervised Learning Techniques,” *nternational journal of communication networks and information security,* vol. 11, no. 1, Apr. 2022.

[7] M. T. Kurniawan, S. Yazid, and Y. G. Sucahyo, “Comparison of Feature Selection Methods for DDoS Attacks on Software Defined Networks using Filter-Based, Wrapper-Based and Embedded-Based.” [Online]. Available: www.joiv.org/index.php/joiv

[8] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, “Automated DDOS attack detection in software defined networking,” *Journal of Network and Computer Applications*, vol. 187, Aug. 2021, doi: 10.1016/j.jnca.2021.103108.

[9] H. Polat, O. Polat, and A. Cetin, “Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models,” *Sustainability*, vol. 12, no. 3, Feb. 2020, doi: 10.3390/su12031035.

[10]Z. O. and,Khin. C. of C. Kyaw, C. Electrical Engineering/Electronics, *IEEE Thailand Section, and Institute of Electrical and Electronics Engineers, The 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* . 2020, doi : 10.1109/ECTI-CON49241.2020.9158230.

[11]K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, “Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques,” *in 2021 International Conference on Computer Communication and Informatics, ICCCI 2021, Institute of Electrical and Electronics Engineers Inc.*, Jan. 2021. doi: 10.1109/ICCCI50826.2021.9402517.

[12]M. A. Setitra, I. Benkhaddra, Z. El Abidine Bensalem, and M. Fan, “Feature Modeling and Dimensionality Reduction to Improve ML-Based DDOS Detection Systems in SDN Environment,” *2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP.* 2022. doi: 10.1109/ICCWAMTIP56608.2022.10016507.

[13]J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, and D. Zhu, “A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning, ” *IEEE Access,* vol. 8, pp. 155859-155872, 2020, doi: 10.1109/ACCESS.2020.3019330.

[14]H. Kamel and M. Z. Abdullah, “Distributed denial of service attacks detection for software defined networks based on evolutionary decision tree model,” *Bulletin of Electrical Engineering and Informatics,* vol. 11, no. 4, pp. 2322-2330, Aug. 2022, doi: 10.11591/eei.v11i4.3835.

[15]L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, “A New Framework for DDoS Attack Detection and Defense in SDN Environment,” *IEEE Access*, vol. 8, pp. 161908-161919, 2020, doi: 10.1109/ACCESS.2020.3021435.

[16]B. Mondal, C. Koner, M. Chakraborty, and S. Gupta, “Detection and Investigation of DDoS Attacks in Network Traffic using Machine Learning Algorithms,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 11, no. 6, pp. 1-6, May 2022, doi: 10.35940/ijitee.F9862.0511622.

[17]V. Deepa, K. M. Sudar, and P. Deepalakshmi, “Design of Ensemble Learning Methods for DDoS Detection in SDN Environment,” *2019 International Conference on Vision Towards*

Emerging Trends in Communication and Networking (ViTECoN), 2019, doi: https://doi.org/10.1109/ViTECoN.2019.8899682.

[18]K. M. Sudar, P. Deepalakshmi, A. Singh, and P. N. Srinivasu, "TFAD: TCP flooding attack detection in software-defined networking using proxy-based and machine learning-based mechanisms,"*Cluster Comput,* vol. 26, no. 2, pp. 1461-1477, Apr. 2023, doi: 10.1007/s10586-022-03666-4.

[19]R. K. Chouhan, M. Atulkar, and N. K. Nagwani, "A framework to detect DDoS attack in Ryu controller based software defined networks using feature extraction and classification, " *Applied Intelligence*, vol. 53, no. 4, pp. 4268-4288, Feb. 2023, doi: 10.1007/s10489-022-03565-6.

[20]Y. Wang, X. Wang, M. M. Ariffin, M. Abolfathi, A. Alqhatani, and L. Almutairi, "Attack detection analysis in software-defined networks using various machine learning method," *Computers and Electrical Engineering,* vol. 108, May 2023, doi: 10.1016/j.compeleceng.2023.108655.

[21]P. Misra and A. Singh Yadav, " Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation," International Journal on Emerging Technologies, vol. 11, no. 3, pp. 659-665, 2020, [Online]. Available: www.researchtrend.net

[22]T. N. Varunram, M. B. Shivaprasad, K. H. Aishwarya, A. Balraj, S. V. Savish, and S. Ullas, " Analysis of Different Dimensionality Reduction Techniques and Machine Learning Algorithms for an Intrusion Detection System," *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, Dec. 2021, doi: https://doi.org/10.1109/iccca52192.2021.9666265.

[23]V. Ravi, R. Chaganti, and M. Alazab, " Deep Learning Feature Fusion Approach for an Intrusion Detection System in SDN-Based IoT Networks," *IEEE Internet of Things Magazine*, vol. 5, no. 2, pp. 24-29, Sep. 2022, doi: 10.1109/iotm.003.2200001.

[24]K. B. Dasari and N. Devarakonda, " Detection of different DDoS attacks using machine learning classification Algorithms," *Ingenierie des Systemes d'Information*, vol. 26, no. 5, pp. 461-468, Oct. 2021, doi: 10.18280/isi.260505.

[25]C. Ioannou, V. Vassiliou, and by Ieee, "Classifying Security Attacks in IoT Networks Using Supervised Learning, " *Intelligent Systems for the Internet of Things (ISIoT)* 2021, doi.org/10.3390/jsan10030058.

[26]B. Goparaju and B. Srinivasa Rao, "International Journal of Communication Networks and Information Security A DDoS Attack Detection using PCA Dimensionality Reduction and Support Vector Machine", [Online]. Available: https://ijcnis.org

[27]M. Y. Su, "Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers," *Expert Syst Appl*, vol. 38, no. 4, pp. 3492-3498, Apr. 2011, doi: 10.1016/j.eswa.2010.08.137.

[28]T. K. Luong, T. D. Tran, and G. T. Le, "DDoS attack detection and defense in SDN based on machine learning," *in Proceedings - 2020 7th NAFOSTED Conference on Information and Computer Science, NICS* , Nov. 2020, pp. 31–35. doi: 10.1109/NICS51282.2020.9335867.

[29]S. Walling and S. Lodh, "Performance Evaluation of Supervised Machine Learning Based Intrusion Detection with Univariate Feature Selection on NSL KDD Dataset,"Feb. 2023, doi: 10.21203/rs.3.rs-2537820/v1.

[30]A. Makuvaza, D. S. Jat, and A. M. Gamundani, "Deep Neural Network (DNN) Solution for Real-time Detection of Distributed Denial of Service (DDoS) Attacks in Software Defined Networks (SDNs), " *SN Comput Sci*, vol. 2, no. 2, Apr. 2021, doi: 10.1007/s42979-021-00467-1.

[31]N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, Aug. 2021, doi: 10.1016/j.jnca.2021.103108.