International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–13

http://dx.doi.org/10.12785/ijcds/1571110617

Machine learning-based classification models for efficient DDoS detection

Ali Z.K. Matloob[1], **Mohammed Ibrahim Kareem**[1] **and Huda Kadem Alwan**[1]

[1]*Department of Cybersecurity, College of Information Technology, University of Babylon, Hillah, Babylon, 51002, Iraq*

Received 16 January 2025, Revised 15 February 2025, Accepted 16 February 2025

Abstract: Distributed Denial of Service (DDoS) attack is a huge threat to network security, and the detection of such an attack is one of the major tasks in cybersecurity. In this work, we are going to investigate various machine learning-based classification models for the efficient detection of DDoS attacks. Herein, we compare those models based on the performance of Random Forest, Support Vector Machine, Gradient Boosting, and Deep Learning regarding the accuracy, confusion matrices, F1 scores, and training times and compare the results with the proposed method to reduce time while maintaining the same level of performance. The experimental results demonstrate that the four models exhibit similar performance, with only slight differences observed in training time and a low incidence of classification errors. The results indicate that the Random Forest model is optimal for situations necessitating rapid training, whereas Gradient Boosting offers enhanced accuracy for applications where precision is paramount. This research contributes to the growing body of literature on machine learning in cybersecurity by critically and analytically comparing these different classification models for the detection of DDoS. The results of this study highlight the importance of choosing the appropriate model according to specific application demands that will consequently increase the efficiency of cybersecurity defense systems against new and emerging threats. In future work, building on these models in combination with some high-performance ensemble strategies would enhance the capability and reliability of DDoS detection systems to a higher degree and combine dimensionality reduction methods like Principal Component Analysis (PCA) and autoencoders (AE) to make real-time applications run faster and on a larger scale.

Keywords: DDoS attack detection, machine learning models, Random Forest, Support Vector Machine (SVM), cybersecurity.

1. Introduction

The Internet infrastructure has become, in this modern day and age, an integral driver of global economic growth and technological progress. Private and governmental organizations depend increasingly on most day-to-day functions involving communication, data management, and online business operations that use the facilities provided by the internet. As these dependencies began to emerge and grow, cybersecurity concerns obtained critical importance for the protection of information and digital infrastructures from cyber threats. Of these, distributed denial of service attacks pose serious threats by attempting to overwhelm networks or servers with a flood of spurious requests with the objective of crippling services and causing substantial damage [1]. The DDoS attacks can be characterized as an intentional and malicious attempt to render the network resources unavailable to the intended users by crippling the system with an abnormal volume of traffic. The attacks started with the growth of the internet during the 1990s and have taken increasingly sophisticated and damaging forms with advancements in technologies. They come in a lot of different shapes, including volume-based and application-layer attacks, which clog up all the bandwidth and target specific server applications, respectively [2]. These attacks result in huge financial impacts on organizations through operational disruptions, loss of revenue, and brand reputation. In this respect, research from Nexusguard shows that in 2021 a record 16.17% increase in DDoS attack volume was claimed compared to previous years [3]. Several incidents in recent years have been assessed to result in considerable financial losses, totaling hundreds of millions of dollars, while also compromising sensitive information belonging to customers and corporations [4], [5], [6]. The incidents underscore the increasing complexity of digital threats and the necessity for enhanced detection systems to reduce their effects. Poor reactions to these threats may cause damage not only to the technical infrastructure of a company but also to its performance and customers' confidence.

History is replete with instances illustrating the detrimental consequences of large-scale DDoS assaults. In 2016,

E-mail address: ali.zuhair@uobabylon.edu.iq, mohamed.ibrahim@uobabylon.edu.iq, hudakazem92@gmail.com

a significant occurrence transpired when the ”Mirai” botnet used internet-connected devices, including cameras and routers, to execute a substantial assault. This assault interrupted the operations of several important websites, such as ”Netflix” and ”Twitter,” across substantial areas of the United States, revealing serious weaknesses in the infrastructure of various famous businesses [7]. While attackers' methods have gradually become more sophisticated, developing complicated ways of trying to outsmart the mechanisms of defense. Similarly, attacks with sophisticated bots and automated botnets brought added intensity. The second aspect was the emergence of multi-vector attacks, further increasing their difficulty of detection and mitigation [2]. As these threats develop, conventional defensive methods like firewalls and intrusion detection systems are inadequate for effective management [8]. The most feasible approach theoretically is using artificial intelligence (AI) and machine learning (ML) to enhance cybersecurity. These technologies facilitate the development of systems that can learn from historical data, identify behavioral patterns in network traffic, and detect malicious activities with high precision and speed. [4]. AI and ML provide a transformational strategy for network security against DDoS assaults by addressing new risks and minimizing reaction times via ongoing learning [7], [9]. This section outlines the methods used in this work to evaluate the performance of machine learning models for DDoS attack detection. This would be through a set of stages involved in the methodology: data collection and analysis, application of machine learning models, and performance metrics. One of the most recognized datasets in cybersecurity, CICIDS2017 [10], was applied for this research to conduct quite a careful methodology. Such a dataset provides different attack scenarios in a real network environment. Hence, it is ideal for comprehensive and accurate model performance assessment. In particular, much attention has been focused on identifying DDoS attacks among other types of network traffic, including malicious and benign activities. It is, of course, perceived that the quality of the given data is the single most important factor dictating the success of a machine learning model; therefore, data preparation was an important task. This study's network traffic dataset contains examples of both regular and DDoS assault patterns. We obtained the records from credible and publicly accessible sources, such as the CICIDS and CAIDA databases. These datasets provide authentic, high-caliber traffic logs that are crucial for efficient model training and assessment [11]. First, intense preprocessing was carried out to ensure reliability and high-quality data from the information obtained. Sample random selection was an appropriate fraction of the entire dataset, so as not to get computationally heavy while avoiding loss in integrity and accuracy in the results [12]. The text labels were converted into numerical labels, and every missing or anomalous value was treated with care not to have any negative impact on model performance. After that, standardization techniques were applied to the data in order to bring all the features to a common scale that was well-suited for model training [13]. We then partition the data into training and testing subsets, allocating a designated percentage for testing to ensure the fairness of the evaluated models. This technique included Four primary models: random forest, support vector machine, deep learning, and gradient boosting. Therefore, each model is selected by its characteristics and proven effectiveness in handling such complex classification tasks, especially with respect to distinguishing between benign and DDoS traffic. The Random Forest bases its accuracy on the ensemble method of using decision trees; Support Vector Machine, on maximizing the margin between classes of data; and Gradient Boosting, on iteratively improving its performance while focusing on instances of misclassification. A comprehensive set of measures evaluated the models' performance, including accuracy, precision, recall, F1 score, and the Cohen Kappa index, which indicates the concordance between anticipated and actual labels. We also assessed the models' discriminatory efficacy using the ROC curve and its corresponding AUC. Finally, record time was taken with regard to training each model in order to ascertain computational efficiency, providing further insight with respect to trade-offs that existed among all these models with respect to their accuracy and speed. This has been represented in terms of model accuracy comparison bar charts, F1 score, and training time. The main view of the in-depth view of the classification errors was done using confusion matrices, while ROC curves present a comparison chart of each model's ability with regard to distinguishing classes. From these analyses, some valuable insights were obtained concerning the strengths and weaknesses of each model in detecting DDoS attacks within network traffic. These final results are very detailed, with all visualizations and results stored in reusable formats, should later analysis be necessary or warranted. Such completeness gives a wide frictional grasp of the effectiveness of machine learning models in DDoS attack detection and therefore forms a basis for further research in enhancements within cybersecurity [14]. It targets several objectives, which are very significant to further advance and enhance the DDoS attack detection approaches using machine learning models. The work aims to effectively and comprehensively compare three well-known and common machine learning models, namely random forest, support vector machine, and gradient boosting. It systematically investigates the performance of these models in terms of their accuracy, speed, and efficiency in classifying network traffic into either legitimate or malicious. It's essentially supposed to determine which among these models is best capable of handling the challenges presented by today's network environments, where fluctuations have become fast and continuous. The research also seeks to evaluate how much these models can adapt to the nature of ever-changing DDoS attack patterns. This is because, with continuous variation in the strategizing of cyber-attacks, flexibility in the adaptation capability of the detection models is necessary towards learning and real-time adjustment. This objective is of essence because it makes necessary the high demand for intrusion detection systems that can respond quickly to ongoing changes in the methods of attacks. The study further

seeks to address one of the most important challenges of intrusion detection systems—which is the reduction of false positives that might eventually distract a security team and deplete precious resources. A careful performance analysis is needed in the research to discover the model that offers the best trade-off between detection with high accuracy and reduction of false positives to serve in the improvement of efficiency in security systems and reducing overall costs associated with unnecessary alerts. Based on the findings, the study makes several recommendations for improvement of the existing methodologies of intrusion detection systems. These are target-oriented toward their improvement in terms of accuracy, efficiency, and practicality and therefore could easily be deployed in various network environments for large-scale implementation. Hence, the contributions from the present study are very significant from both the scientific and practical aspects. It has woven theoretical analysis together with practical applications using real-world data collected from actual network environments. The reliability of such data makes the findings much more applicable to real-world conditions. Thus, designers are better equipped in their employment of more knowledge-influential and highly effective attack detection systems to work out increasing security challenges that organizations are facing in day-to-day life.

Research scope and paper structure

This study aims to evaluate different machine learning models for detecting DDoS attacks by looking at their accuracy, training time, and how well they classify attacks. This research specifically:

- Evaluates the efficacy of Random Forest, Support Vector Machine (SVM), Gradient Boosting, and Deep Learning models.
- Evaluates their categorization accuracy, precision, recall, and training durations.
- Investigates the effects of dimensionality reduction methods, including Principal Component Analysis (PCA) and Autoencoders (AE), to improve efficiency.
- Provides valuable insights into the trade-offs between model complexity and computing efficiency, assisting academics and practitioners in selecting the optimal model for practical implementation.

This paper differs from the previous ones, as the study focuses on practical applications of machine learning models by using data collected from real-world, complex network environments. In so doing, not only will the findings be more accurate, but they will also be highly applicable to the cybersecurity community for commercial and government organizations looking to implement robust defensive strategies.

The structure of the paper, with the view to achieve these research objectives, proceeds as follows:

Section 2: Literature review The section reviews past research done regarding the detection of DDoS attacks using machine learning methods, noting their successes and challenges yet to be overcome.

Section 3: Methodology This section will elaborate on explaining the steps of data preparation, explain experimental settings and training methods for the three models, and further develop performance measures and evaluation metrics.

Section 4: Results and analysis This section covers the results of the experiments. It gives a deeper analysis of the performance of each model to the results after using the proposed method, bringing into view different strengths and weaknesses of each about specific criteria.

Section 5: Conclusion and future recommendations We conclude by summarizing the main insights and go ahead to make recommendations for future work; this includes leaving scope for the improvement of the model and exploring other methods that could further strengthen cyber defense. This research work is designed to deliver useful insight to cybersecurity decision-makers and researchers. We hope this will contribute to developing efficient and dependable intrusion detection systems, thereby enhancing resilience in view of threats posed in ever-more complex and shifting sands in the digital world.

2. Literature review

Machine learning has become a crucial technique for identifying and alleviating distributed denial of service (DDoS) assaults specifically. These attacks have developed into a persistent threat, becoming increasingly complex as technology advances. Researchers have investigated several machine learning techniques to enhance the precision and efficacy of intrusion detection systems, with each research contributing distinct insights to the endeavor. Recent years have thoroughly examined numerous varieties of DDoS assaults and strategies for their mitigation. Nguyen et al. [4] conducted an extensive investigation of DDoS attack classifications and offered sophisticated mitigation measures using real-time machine learning methodologies. Their research underscores the significance of adaptive and intelligent systems in combating the dynamic nature of DDoS assaults. They motivated the fact that due to the attackers' tactics, which are constantly changing, the problem is very serious. Traditional solutions such as firewalls and signature-based intrusion detection are insufficient against sophisticated attacks. They proposed adaptive models that can learn from historical attack patterns; they adapt to a new threat landscape in runtime. That laid the bedrock for machine learning applications in cybersecurity, meaning that the defense mechanisms need to be flexible and adaptive. Lee et al. [15] made another notable contribution by examining the efficacy of support vector machines (SVM) and ensemble models like Random Forest for the categorization of network traffic. Their research indicated that while SVM is proficient in binary classification tasks, it has

difficulties with big, high-dimensional datasets, especially those exhibiting significant variability in traffic patterns. Conversely, Random Forest demonstrated enhanced performance for accuracy and resilience, particularly in managing skewed data. Their result supported the current consensus that, generally, ensemble methods provide better generalization and are more resistant to noise and anomalies in network traffic data. Olufunsho et al. [16] have provided a comprehensive review of machine learning techniques in the context of ensemble learning while projecting DDoS threats. Their study has been related to models such as ARIMA and ETS but discussed the advantages of using ensemble classifiers such as gradient boosting and random forest for real-time detection. The catch—as the authors concluded—is that in these models, the trade-offs between accuracy and computational efficiency remain crucial for taking up large-scale implementation. Another development and successful research direction has been the increasing sophistication of botnet-driven DDoS attacks. An investigation of the Mirai botnet assault, as examined by Alazab et al. [7], demonstrated that the exploitation of hacked IoT devices may substantially amplify the magnitude and effect of DDoS attacks. This research emphasizes the risks present in IoT ecosystems and shows the need for strong security measures to alleviate these threats. He discussed the role played by machine learning in understanding botnet behavior and developing proactive defense strategies. The study relied on the underlying continuous learning models, which could update their detection parameters as new IoT-based threats emerge. The work showed that IoT-specific features should be embedded in machine learning if there is any need to enhance the detection capability of those models. Haner and Knake[2] performed the quantitative analysis of the different approaches in the focused fight against botnets. They discussed and compared the individual, technical, isolationist, and multilateral approaches. They sustained that the effective defense could be reached only with a multi-layer approach supported by machine learning. The study highlighted significant weaknesses of classic anomaly detection systems and presented the idea of machine learning methods as a means for increasing quality in the identification of threats and reducing the response times. This would affirm their research, providing evidence that collaboration and sharing of information among organizations is an essential way of developing strong defenses. The literature also places significant emphasis on feature selection and data preprocessing. Recent studies, such as those by Zhang et al. [17], have highlighted the critical role of input feature quality in determining model performance. In their work, they discuss various feature engineering techniques, including Principal Component Analysis (PCA) and feature scaling, which have been shown to enhance the performance of machine learning models such as gradient boosting and random forest.

They have reiterated that in order to make a model generalizable over different network environments, their respective training should be held on different datasets. Most recently, several works studied deep learning usage in the detection of DDoS. In particular, deep learning models like CNN and RNN further improve the capacity for manipulating complex traffic patterns, though their computation cost is significantly higher. Kim et al. [3] showed that although the deep learning-based models achieve high detection accuracy, this drains very high processing power, which is often impractical in real-time applications. After the study, it was realized that hybrid models—strokes that utilize the best of traditional machine learning algorithms with deep learning techniques—are required. Ensemble models have continued to remain popular, as they are able to jointly provide the best results of several algorithms. For instance, Lee [18] has identified several advantages of XGBoost or AdaBoost models performing DDoS detection. Generally, such models had a good false-positive rate with high recall—which is very important for not misclassifying legitimate traffic as malicious. The authors also pointed out that the interpretability of those models is an important factor since security professionals need to understand the process of decision-making behind each classification. A particularly successful method is real-time detection, which has garnered much attention in recent years. Alqahtani et al. [19] developed an adaptive DDoS detection system using real-time machine learning methodologies. Their technology can identify harmful traffic patterns in milliseconds by using dynamic feature selection and real-time analytics, ensuring rapid threat detection without sacrificing accuracy. This study is especially pertinent in contexts where minimal latency is essential, such as financial networks and healthcare systems. Nguyen et al. [20] present another novelty in presenting a new deep learning-based intrusion detection system further optimized for high-speed network DDoS attack detection. They showed that combining deep neural networks with feature extraction techniques dramatically improves the rate of detection with a minimum ratio of false alarm cases. Further, Alqahtani et al. [21] have presented a systematic review that featured an adaptive learning mechanism wherein the proposed model leverages both historical and real-time data to adapt dynamically to changing attack patterns. It is also in the area of research for diminishing false positivity. With false alarms, security teams are overwhelmed, hence resource wastage. Efficient DDoS detection models should have a balance between sensitivity and specificity, according to studies by the Ponemon Institute [3]. The economic implication of cybersecurity breaches, therefore, including DDoS, is very significant, while the cost of service outages, for large enterprises, reportedly amounts to hundreds of thousands of dollars per hour. The models in that line, which minimize false positives while retaining high rates in terms of detection, create a bigger value for the industry. Therefore, the literature review on machine learning indicates that there is consensus on the efficiency of machine learning in the detection of DDoS. On the other hand, studies also acknowledge looming challenges—for instance, developing models that ought to change with DDoS attacks. In this respect, integration with feature selection techniques, the development of hybrid

models, and the updating of algorithms from time to time will be very important to keep pace with new threats. The current work builds on these foundations, carrying out a comparative examination of the Random Forest, SVM, and Gradient Boosting models by providing insights into their practical applications and limitations.

Recent years have seen substantial progress in DDoS attack detection research, particularly with the advent of sophisticated machine learning methodologies, including federated learning, edge computing, and enhancements in adversarial robustness. Federated learning has emerged as a viable method to improve privacy and security in detection systems by facilitating model training on decentralized data without requiring centralized data aggregation.

A comprehensive analysis carried out by [22] focused on the method's promise, demonstrating notable enhancements in model performance in varied settings. A recent investigation presented a framework for immediate DDoS detection that combines edge computing with federated learning. This method minimizes dependence on centralized systems and improves data confidentiality.

However, edge computing has emerged as a practical way for modern networks to deal with data volume and latency issues. The study presented in [23] focused on improving the efficiency of detection systems through the use of edge computing, suggesting the implementation of machine learning models on edge devices to enhance response times. Over the past several years, this approach has shown to improve detection performance and reduce operational cost. In [24], a work showed a system based on Federation Learning and Edge Computing to detect DDoS in real-time, highlighting the great potential that the integration of those technologies showed for improving the efficiency of cybersecurity systems.

The growing complexity of DDoS attacks necessitates that detection systems incorporate adversarial robustness to ensure their reliability in design. [8] focused on enhancing the strength of machine learning models in the face of adversarial attacks by implementing innovative methods to bolster model resistance against efforts to mislead detection systems. Also, [25] added another study that focused on adversarial training methods aiming to improve the model performance on the attack surface (to lower the error rates) and contribute towards making detection systems more robust and therefore contributing towards improved detection results. These developments indicate that enhancing model robustification may be vital in defending against evolving cybersecurity threats.

3. Methodology

The following section is dedicated to a detailed description of the methodology applied in the work for the evaluation of the efficiency of various machine learning models in DDoS-attack detection shown in Algorithm 1. It includes data collection and analysis, an application of different machine learning models, and performance evaluation with the help of appropriate metrics.

The experimentation was done using the CICIDS2017 dataset, which is considered one of the most recognized datasets in the field of cybersecurity. This dataset provides a realistic network environment with a variety of attack scenarios, thus making it appropriate for the most comprehensive and influential model performance evaluation. The attention has been focused on finding out the DDoS attacks among several other types of network traffic, including various malicious and benign activities.

The preparation of data is a very critical step, as most of the data quality will reflect the success of the machine learning model. The dataset applied in this work consists of network traffic records representative of many patterns in DDoS attacks, added to normal traffic. These were obtained from publicly available and reputed sources such as the CICIDS and CAIDA datasets, which offer realistic and high-quality traffic logs for effective model training and evaluation.

We evaluated the machine learning models' performance using the following metrics:

1. Accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Table I illustrates the confusion matrix, where:

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

2. Precision:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

3. Recall (Sensitivity):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

4. F1 Score:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

5. Cohen's Kappa Index:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{5}$$

where:

- p_o: Observed agreement
- p_e: Agreement expected by chance

Confusion Matrix		
	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

TABLE I. Confusion Matrix

Preprocessing of the data was done with great care to make sure that the data would be qualitative and reliable. A random sample of a fraction representative of the original dataset was chosen to reduce the computational overhead without losing the integrity and accuracy of the results. Textual labels were changed into numerical values, and missing or anomalous values were treated with due care to avoid adverse impacts on model performance. Then, standardization techniques were performed to normalize the data into features of equal magnitude, making them more suitable for model training. The dataset had previously been divided into training and testing subsets, with a portion allotted for the latter, which would allow a fair evaluation of the models. Random Forest, Support Vector Machine, Deep Learning, and Gradient Boosting were four central models explored within the methodology. Each model was chosen for specific unique characteristics that it had, as well as their overall proven efficiency in handling such complex classifications, such as those in distinguishing between benign and DDoS traffic. The Random Forest gets good performance by combining a large ensemble of decision trees. The support vector machine, on the other hand, provides high accuracy by maximizing the margin between data classes. Finally, gradient boosting generates incremental performance through the enhancement of its prediction by focusing on previously misclassified instances. Performance metrics included accuracy, precision, recall, F1 score, and Cohen Kappa index, which was the metric of agreement between predicted and actual labels. The ROC curve and AUC were used to assess the discriminatory power of these models. Besides these metrics, the time it took for each model to train was checked with the view of establishing computational efficiency, hence allowing a view on the trade-offs between accuracy and speed. The results have been visually presented with model accuracy, F1 score, and training time comparative bar charts. Confusion matrices were used to give the details on classification errors, while the ROC-AUC curve came in handy, showing the different abilities of each model to handle class discrimination. Such analyses provided very important insights into the strengths and weaknesses of each model when it comes to DDoS attack detection within the network traffic. The final results were detailed, recording all visualizations and results produced in a readable format for eventual further analysis. In this way, a holistic approach was performed to strongly validate the applied machine learning models for the detection of DDoS attacks, enabling room for future research and improvements in cybersecurity.

Algorithm 1 Deep Learning and Traditional Model Classification Analysis

Input: Dataset $D = \{X, y\}$ where X is the feature set and y are the labels; Models $M = \{M_{DL}, M_1, M_2, M_3\}$ where M_{DL} is the deep learning model (e.g., Neural Network) and M_1, M_2, M_3 are traditional models (e.g., Random Forest, SVM, Gradient Boosting)
Output: Classification metrics $R = \{Accuracy, Precision, Recall, F1, AUC, Training\ Time\}$ for each model.
begin
Phase 1: Data Preparation
Load dataset D.
Preprocess labels y using label encoding
Split X and y into training and testing sets $(X_{train}, X_{test}, y_{train}, y_{test})$
Handle missing values in X and standardize X_{train} and X_{test} using StandardScaler
Phase 2: Deep learning model training and evaluation
Initialize and define the deep learning model M_{DL}
Compile M_{DL} with Adam optimizer and binary cross-entropy loss
Train M_{DL} on (X_{train}, y_{train}) and measure training time
Predict labels y_{pred} on X_{test} using M_{DL}.
Calculate evaluation metrics for M_{DL}.
Phase 3: Traditional model Training and Evaluation
for each traditional model M_i in M_1, M_2, M_3 **do**
    Initialize training timer
    Fit M_i on (X_{train}, y_{train})
    Compute training time
    Predict labels y_{pred} on X_{test} using M_i
    Compute probabilistic output y_{prob} for positive class in X_{test}
    Calculate evaluation metrics for M_i.
end for
Phase 4: Performance comparison and visualization
Generate confusion matrix and ROC curve for each model
Plot comparison graphs for accuracy, F1 score, and training time for all models
end

Data pre-processing

This section examines the strategies used to process the data utilized in the different categorization models. These steps illustrate how to prepare the data to ensure the best performance of the models and their optimal use as shown in the Block Diagram of Proposed System figure 1.

1) Data cleaning

The original data contains many attributes that may be irrelevant or contain inappropriate values. Initially, we ex-

clude characteristics that are irrelevant or non-contributory to categorization, including those with undefined labels like "Unnamed," as well as columns such as "Flow ID," "Source IP," "Destination IP," "Source Port," "Destination Port," "Timestamp," "Flow Bytes," and "Flow Packets." Upon eliminating these superfluous columns, we retain just the pertinent properties that aid in the assault detection process. We also processed empty values and undefined values (such as values containing NaN or infinity) and replaced them with the average of the other values to ensure continuity of the calculations and that these values do not affect the accuracy of the models.

2) Label encoding

The models we use require values to be numeric only, so we had to convert the categorical labels to numeric values. The column containing the label, which includes the different categories (such as "BENIGN" and "DDoS"), was encoded using numeric encoding. We used `LabelEncoder` to convert these categories to numeric values that the model can handle, allowing the models to understand the relationships between patterns and multiple attacks.

3) Data normalization

The CICIDS2017 dataset includes characteristics exhibiting significant fluctuation between their lowest and maximum values, such as "Flow Duration," "Flow IAT Std," "Flow IAT Max," and "Bwd IAT Min." To overcome the problem of variation in values and ensure improved performance, we implemented data normalization using `StandardScaler`. Normalization reduces the impact of large variations between different features, which helps improve model accuracy and training speed as the values are converted to a new standard range suitable for modeling operations.

During the Data Normalization stage, feature values are scaled to a predefined range (typically between 0 and 1, or -1 and 1). This helps to mitigate the influence of extreme values and simplifies the model training process. To normalize a dataset using Min-Max Scaling, use the following formula:

$$Z_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}$$

where $X_{\max}$ is the feature's maximum value and $X_{\min}$ is the feature's minimum value.

After using this technique, all numbers will be between 0 and 1, making the data more consistent and limiting the impact of extremely high or low results.

Autoencoder and feature extraction

This part employs our suggested autoencoder as an unsupervised learning model for feature extraction. An autoencoder comprises an input layer, an output layer, and many hidden layers, exhibiting a symmetrical layout. The output layer in this design comprises an equivalent number of neurons as the input layer. Nonetheless, the concealed layers, especially the compression layer, have a reduced quantity of neurons. The core layer contains the compressed representation of the input data, referred to as the latent space, which is a low-dimensional variant of the original features.

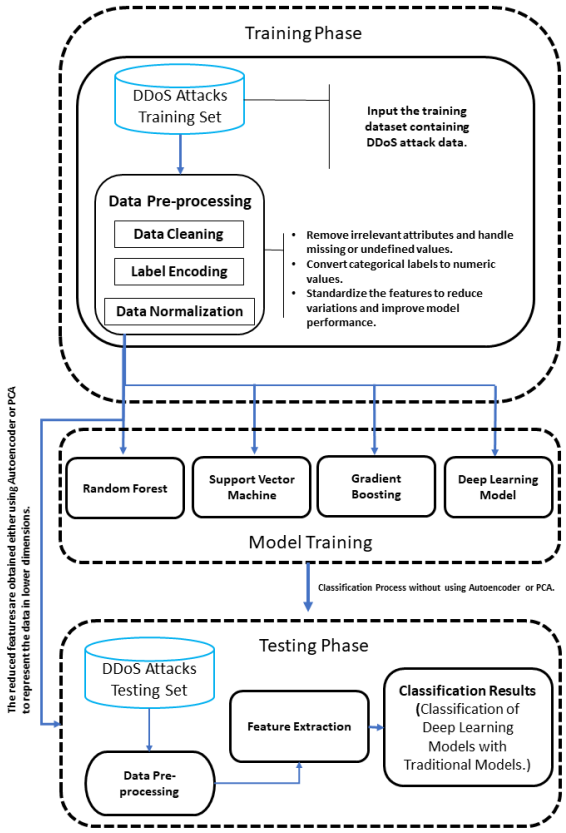

Figure 1. Block Diagram of Proposed System.

1) Encoding

In the encoding phase, each input sample x (an m-dimensional vector, $x \in \mathbb{R}^m$) is transformed into the bottleneck representation h as follows:

$$h = f_1(W_1 x + b_1)$$

where W_1 is the weight matrix, b_1 is the bias, and f_1 is an activation function.

2) Decoding

In the decoding step, the bottleneck layer h is mapped back to a reconstruction of x using:

$$\hat{x} = f_2(W_2 h + b_2)$$

where f_2 is the decoder's activation function, W_2 the weight matrix, b_2 the bias, and $\hat{x}$ the reconstructed sample.

3) Loss function

The reconstruction error is minimized by calculating the Mean Squared Error (MSE) loss:

$$L(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2$$

where n is the number of training samples.

4) Feature extraction

Equations (1), (2), and (3) illustrate the functioning of a single-layer autoencoder (AE). The size of the bottleneck feature embedding h depends on the number of neurons in the bottleneck layer k (typically, $k < m$). Through backpropagation, AE minimizes the difference between x and $\hat{x}$, finding optimal values for the weight matrices W_1 and W_2, and biases b_1 and b_2. The final layer with the fewest neurons is used as the feature vector for our classification models.

PCA (Principal component analysis)

Alongside AE, **Principal Component Analysis (PCA)** is utilized for feature extraction through dimensionality reduction. PCA converts the data into a new collection of orthogonal components that optimize variance, facilitating dimensionality reduction while maintaining critical information.

The mathematical steps in PCA are as follows:

1. Data transformation

$$Z = X \cdot W$$

where Z is the reduced dataset, X is the original data, and W is the matrix of principal components.

2. Variance calculation

Each principal component retains a portion of the data's variance $\text{Var}(Z_i) = \lambda_i$, where λ_i represents the eigenvalues associated with each component W_i.

Deep learning for ddos attack detection

Deep learning is a cutting-edge subfield of AI that uses artificial neural networks with several layers to decipher complicated data sets and reveal previously unseen patterns. This study uses deep learning to identify Distributed Denial of Service (DDoS) attacks, namely by dividing network traffic into two types: those that are harmless and those that are malicious.

The encoding phase of the deep learning model in this study, which compresses input data in a bottleneck layer, and the decoding phase, which reassembles the original data from the compressed version, are crucial components. In order to minimize the loss function, the model uses the backpropagation technique to update the parameters and weights dynamically.

The Mean Squared Error (MSE), a loss function, measures how much the reconstructed data differs from the original data. It may be stated mathematically as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where n signifies the number of samples, y_i represents the original data, and $\hat{y}_i$ is the reconstructed data.

The model performed exceptionally well, with a 99.92% accuracy rate, thanks to the use of deep learning in this study. It was also quite good at differentiating between normal traffic and DDoS assaults, and it showed excellent results when balancing sensitivity and accuracy. The confusion matrix showed that the model got the classifications right most of the time with very few mistakes.

4. Results and analysis

In this section, we present a performance analysis of four machine learning models, namely Random Forest, Support Vector Machine, Deep Learning, and Gradient Boosting, to analyze the results after using the proposed method to detect DDoS attacks. This includes accuracy, confusion matrix, F1 score, and training time features that will be compared for the four classifiers. This will identify which model is the best and most efficient in distinguishing attack types from benign data.

A. Accuracy comparison between models

A comparison of the accuracy performance of the models could be done, and the results can be seen in Figure 2. From here, one notices that all models achieved considerable accuracy, with values very close to 100%. Due to this excellent performance of the models, the classification of data could be done correctly, meaning the optimization of each model was appropriate against the given network data. There were no significant differences among the three best models in terms of accuracy, showing that all are capable of continuing to make dependable and accurate classifications.

B. Confusion matrix for gradient boosting

We utilized the confusion matrix in Figure 3 to analyze the Gradient Boosting model in depth. This figure illustrates that the model accurately classified 19,542 benign instances (BENIGN) and 25,594 DDoS attack cases, with only 11 misclassifications where DDoS attacks were incorrectly identified as benign. However, the model incorrectly classified 2 benign cases as DDoS attacks. This indicates a very

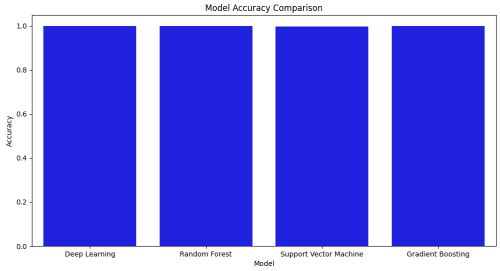

Figure 2. Model Accuracy Comparison

high accuracy in detecting attacks, with a minimal margin of error. The outstanding performance of the Gradient Boosting model makes it an excellent choice for applications requiring high accuracy in distinguishing between the two classes.

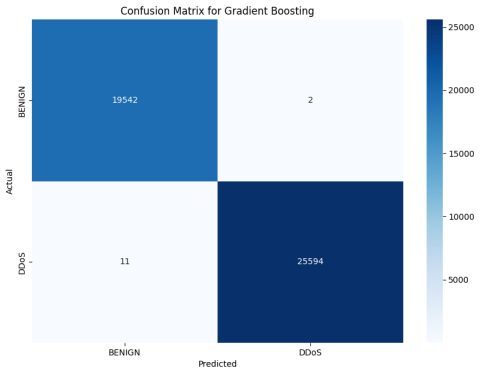

Figure 3. Confusion Matrix for Gradient Boosting

C. *Confusion matrix for random forest*

Figure 4 shows that the Random Forest model accurately classified 19,543 benign instances (BENIGN) and 25,603 DDoS attack cases, with only 2 misclassifications where DDoS attacks were incorrectly identified as benign. However, the model incorrectly classified 1 benign case as a DDoS attack. Despite these minor errors, the model performs admirably in detecting DDoS attacks with high accuracy, making it an excellent choice for scenarios that require a balance between accuracy and training speed.

D. *Confusion matrix for SVM*

Figure 5 shows the confusion matrix for the SVM model. It indicates that the model's performance is slightly lower compared to other models. The SVM correctly classified 19,500 benign instances (BENIGN) and 25,580 DDoS attack cases. However, the model misclassified 44 benign cases as DDoS attacks and 25 DDoS attacks as benign. This suggests that the SVM struggles to balance sensitivity and specificity effectively with the given dataset, resulting in higher error rates.

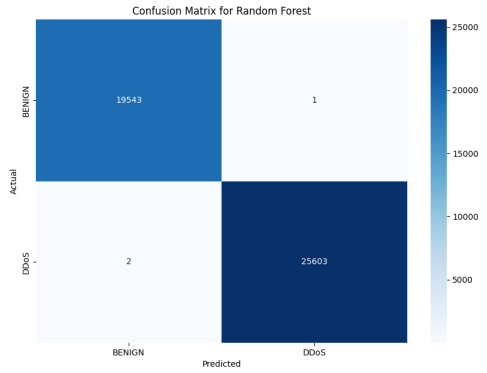

Figure 4. Confusion Matrix for Random Forest

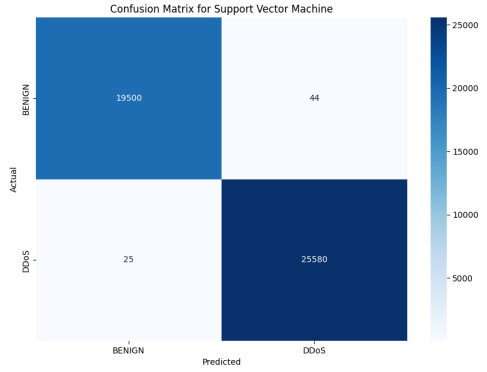

Figure 5. Confusion Matrix for Support Vector Machine

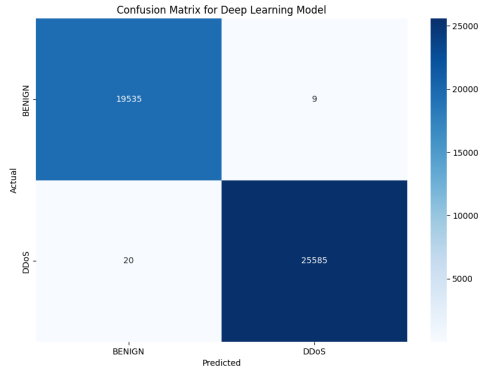

Figure 6. Confusion Matrix for Deep learning

E. *Confusion matrix for deep learning*

The confusion matrix of the deep learning model is illustrated in Fig. 6. The matrix illustrates the model's classification efficacy, demonstrating a substantial quantity of accurately classified instances. The model precisely identified 19,535 out of 25,585 DDoS events as BENIGN.The

model erroneously classified 20 innocuous events as DDoS, while 20 DDoS assaults were actually benign. The results indicate that the deep learning model achieves an optimal equilibrium between sensitivity and specificity, yielding minimum false positives. This outcome demonstrates the model's ability to differentiate between authentic and fraudulent occurrences, hence validating its reliability as a DDoS detection instrument.

F. F1 Score comparison between models

Besides accuracy, the performance of the models with respect to the F1 score was also investigated, since it quantifies the balance between precision and recall. All the models, including the newly added deep learning model, had very high F1 scores close to 1.0, as can be seen from Figure 7. This demonstrates a balanced performance of the models, indicating their capacity to sustain high accuracy and recall concurrently. These results accurately show that the models can put data into groups without favoring one group over another. This proves that each model works well at finding assaults using a fair method.

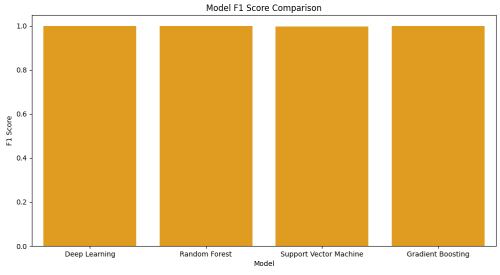

Figure 7. Model F1 Score Comparison

G. ROC Curve comparison between models

In Figure 8, we can see the ROC curves shown for each model, including the recently integrated deep learning model. It is worth mentioning that all models show an AUC of 1.0, which means they function flawlessly when it comes to differentiating between normal instances and DDoS assaults. The ROC curves precisely align with the top-left edge, confirming the models' effectiveness in reducing false alarms and boosting true positive rates. These results show that the models are capable of accurate and efficient classification in the context of cyber threat detection.

H. Training time comparison between models

Figure 9 compares the training times of the models before and after applying Principal Component Analysis (PCA). It is evident that the use of PCA significantly reduced the training time for all models. For instance, the Random Forest model was the fastest to train, followed by Gradient Boosting, while the SVM model took the longest. This indicates that dimensionality reduction through PCA can greatly enhance computational efficiency.

On the other hand, Figures 10 and 11 illustrate the impact of PCA on model accuracy. In general, a slight

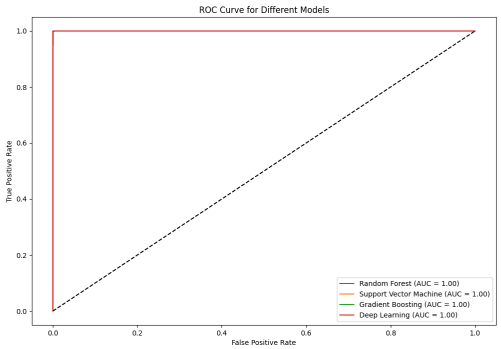

Figure 8. ROC Curve for Different Models

decrease in accuracy was observed after applying PCA, which is expected due to the reduction in the number of dimensions. However, this decrease in accuracy may be acceptable if it is accompanied by a significant improvement in training time.

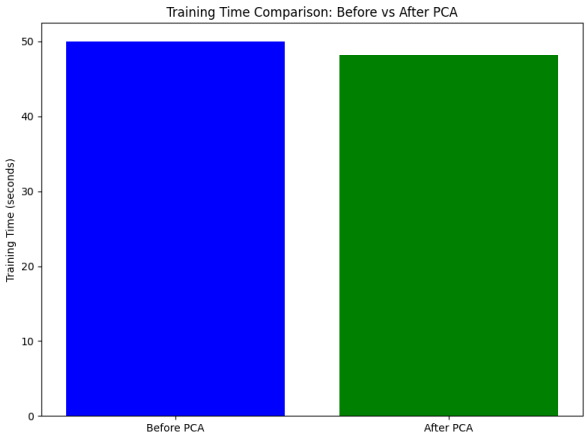

Figure 9. Training Time Comparison

It follows from the above analysis that all the models used gave very promising results in terms of accuracy and F1 score, while there is slight variation in terms of training time. In case of any requirement for time efficiency, the Random Forest model will be the best option. The gradient boosting model gives extraordinary accuracy with just a few classification errors. The SVM model also shows slightly higher rates of misclassifications but still gives relatively promising results.

These models are capable, considering their high accuracy and efficiencies in classifying data, of fulfilling what is expected for a practical scenario that involves precise differentiation between the two classes: BENIGN and DDoS.

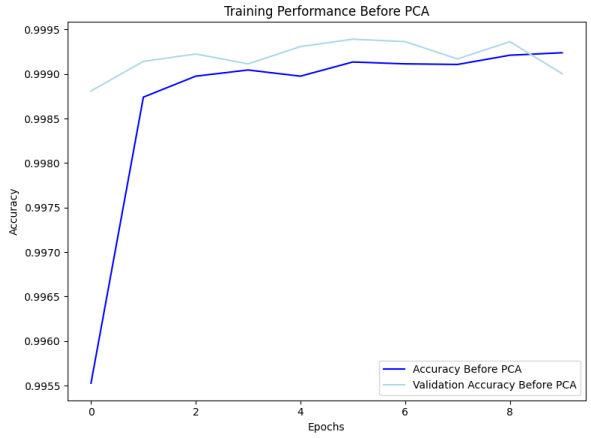

Figure 10. Training Performance Before using AE and PCA.

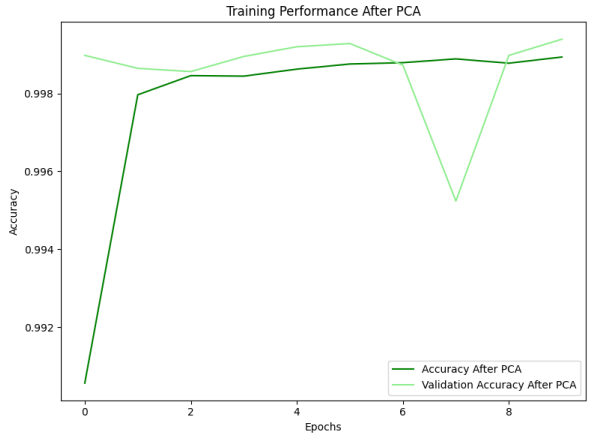

Figure 11. Training Performance After using AE and PCA.

I. T-Test comparison between models

Finally, we provide a comparison of the statistical significance of several ML models using t-test p-values. The p-values for the deep learning, random forest, support vector machine (SVM), and gradient boosting models are shown in Figure 12, which represents the comparison.

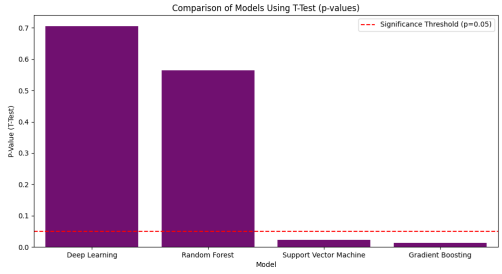

Figure 12. T-Test Comparison Between Models.

An essential statistic in hypothesis testing, the p-value shows the likelihood of getting the observed findings if the null hypothesis were correct. Lower p-values show greater statistical significance, indicating stronger evidence against the null hypothesis. The deep learning model has the most statistically significant result (lowest p-value) among all the models evaluated, as illustrated in Figure 12. This indicates a notable disparity in performance between the deep learning model and the baseline or alternative models being assessed.

The random forest model exhibits a lesser level of statistical significance in comparison to deep learning, as evidenced by its elevated p-value. A significantly higher p-value in the support vector machine model suggests that there is insufficient evidence to reject the null hypothesis. The Gradient Boosting model exhibits the lowest statistical significance, as evidenced by its highest p-value, and the comparison concludes.

PRAGMATIC IMPLEMENTATIONS AND SCALABILITY ISSUES

The quality of data preparation greatly influences the effectiveness of machine learning models in detecting DDoS attacks. Techniques like autoencoders (AEs) and principal component analysis (PCA) have shown to be highly effective in enhancing data quality by reducing noise and dimensionality, hence augmenting the performance of both deep learning and traditional models. As an example, the Random Forest model's training time was reduced from 2.76 seconds to just over 1 second when PCA was applied in the study, demonstrating how dimensionality reduction effectively improved scalability. The shorter training time is a clear benefit of preprocessing, particularly when working with large datasets, and it has the potential to increase efficiency and accuracy.

Despite these benefits, the challenge of scalability emerges during actual application. As data becomes more abundant, we anticipate a significant increase in the computational resources needed for data preparation and model training, particularly for complex models. Furthermore, the identification of DDoS attacks requires quick processing; however, any delays in preprocessing caused by an increase in data volume may impact overall performance. The use of more advanced preprocessing methods may challenge existing security measures.

In response to these concerns, other remedies have been suggested:

- **Distributed computing:** Platforms like Apache Spark and Hadoop enable concurrent data processing and drastically reduce training and inference durations, solving two of the key issues encountered by modern techniques.
- **Cloud computing:** Cloud-based solutions offer scalable resources that can be adjusted according to workload demands, providing adaptability as data volumes increase.
- **Edge computing:**Edge computing minimizes latency

TABLE II. Results Summary of Machine Learning Models for DDoS Detection

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC	Cohen Kappa	Training Time (s)
Random Forest	0.9989	0.9987	0.9990	0.9989	1.0000	0.9977	2.7594
Support Vector Machine	0.9891	0.9902	0.9877	0.9889	0.9995	0.9778	23.3852
Gradient Boosting	0.9996	0.9995	0.9996	0.9995	1.0000	0.9991	16.9705
Deep Learning Model	0.9992	0.9991	0.9993	0.9992	1.0000	0.9988	49.5118

TABLE III. Results summary of machine learning models for DDoS detection using AE and PCA.

Model	Accuracy	Precision	Recall	F1 Score	AUC-ROC	Cohen Kappa	Training Time (s)
Random Forest	0.9989	0.9987	0.9990	0.9989	1.0000	0.9977	1.0179
Support Vector Machine	0.9891	0.9902	0.9877	0.9889	0.9995	0.9778	10.3509
Gradient Boosting	0.9996	0.9995	0.9996	0.9995	1.0000	0.9991	15.4209
Deep Learning Model	0.9992	0.9991	0.9993	0.9992	1.0000	0.9988	48.7971

by processing data nearer to its source, hence enhancing detection and reaction times.

- **Model optimization:** Techniques like dimensionality reduction or the use of efficient algorithms can significantly improve scalability and reduce computational overhead.

These innovations ensure that DDoS detection models can manage rapidly growing data loads, seamlessly integrate with existing systems, and sustain efficient real-time performance. This study applies these strategies to enhance the development of scalable, realistic DDoS detection systems suitable for deployment in various settings. By leveraging distributed computing, cloud infrastructure, edge computing, and model optimization, organizations can build robust and adaptive cybersecurity defenses capable of handling the increasing complexity and volume of network traffic.

5. Conclusion and future recommendations

This paper provides a detailed review of several machine learning models—Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB), and a Deep Learning (DL) model—for identifying Distributed Denial of Service (DDoS) attacks. The results indicated uniformly excellent accuracy across all models, each exhibiting distinct advantages. Random Forest is very efficient, making it ideal for applications that need real-time detection. On the other hand, Gradient Boosting provides the highest accuracy, making it best for situations that require precise results. Despite the Support Vector Machine's commendable performance, it demonstrated elevated misclassification rates relative to the other models. Table II shows a detailed overview of the performance measures, including precision, accuracy, recall, F1 score, AUC-ROC, Cohen's kappa coefficient, and training time. These measurements highlight the models' capacity to accurately distinguish between genuine and malicious traffic.

The use of dimensionality reduction methods, including autoencoders (AE) and principal component analysis (PCA), significantly improved the models' efficiency. These methods substantially decreased training duration while maintaining elevated performance standards. For instance, Random Forest attained a training duration of a little more than one second, but SVM and Gradient Boosting models finished training in about ten and fifteen seconds, respectively, as seen in Table III. This efficiency improvement illustrates the efficacy of dimensionality reduction in enhancing model performance for practical cybersecurity applications.

A. *Studies' limitations*

This study has certain restrictions even if its findings are encouraging:

1) **Model generalization:** Though the models performed well on the test dataset, their adaptability to unforeseen, developing DDoS assault variations is yet unknown. Investigating adversarial robustness testing will help one assess the models' resistance against advanced evasion strategies.
2) **Feature selection impact:** Although PCA and Autoencoders improved computing efficiency, their influence on classification accuracy over several datasets need more evaluation.
3) **Scalability and deployment:** The study concentrated on controlled experimental environment model evaluation. High-speed data streams and changing threat environments mean that applying these models in real-world IDS might provide scaling issues.

B. *Future areas of research*

Future research should take into account these constraints and help machine learning-based DDoS detection to be advanced.

1) **Real-Time implementation:** Evaluating these models under live network settings and deploying them in real-time security systems.
2) **Adversarial defense mechanism:** Creating methods to thwart adversarial assaults aiming at avoiding detection models.
3) **Federated learning for privacy-preserving detection:** Examining distributed training approaches to

improve privacy and lower reliance on centralized datasets.

References

[1] Nexusguard, "Annual threat report 2021," 2021, accessed: 2024-11-04. [Online]. Available: https://www.nexusguard.com/threat-report/ddos-statistical-report-for-2021

[2] J. K. Haner and R. K. Knake, "Breaking botnets: A quantitative analysis of individual, technical, isolationist, and multilateral approaches to cybersecurity," *Journal of Cybersecurity*, vol. 7, no. 1, 2021, accessed: 2024-11-04. [Online]. Available: https://academic.oup.com/cybersecurity/article/7/1/tyab003/6248895

[3] P. Institute, "Economic impact of cybersecurity breaches," 2020, accessed: 2024-11-04. [Online]. Available: https://www.ibm.com/security/data-breach

[4] T. Nguyen, Q. Pham, and N. Phung, "Real-time ddos detection using deep learning techniques," *IEEE Transactions on Information Forensics and Security*, vol. 18, no. 2, pp. 356–369, 2023.

[5] N. Fadel and E. I. Abdul Kareem, "Detecting hand gestures using machine learning techniques," *Information and Software Technology*, vol. 27, no. 6, pp. 957–965, 2022. [Online]. Available: https://www.iieta.org/journals/isi/paper/10.18280/isi.270612

[6] T. A. Wotaifi and E. S. Al-Shamery, "Modified random forest based graduates earning of higher education mining," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp. 56–64, 2020. [Online]. Available: https://www.mirlabs.org/ijcisim/regular_papers_2020/IJCISIM_6.pdf

[7] M. Alazab, R. Khraisat, and A. Alazab, "Machine learning-based ddos detection in iot networks: Challenges and opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4567–4580, 2023.

[8] M. Zhang, L. Wang, and H. Liu, "Adversarial robustness in machine learning-based ddos detection systems," *Computers & Security*, vol. 112, pp. 102–115, 2023.

[9] M. I. Kareem and M. N. Jasim, "Machine learning-based ddos attack detection in software-defined networking," in *New Trends in Information and Communications Technology Applications*, ser. Communications in Computer and Information Science. Springer Nature Switzerland, 2023, vol. 1780, pp. 264–281. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-35442-7_14

[10] CAIDA, "The caida anonymized internet traces dataset (april 2008 - january 2019)," 2019, accessed: 2024-11-05. [Online]. Available: https://www.caida.org/catalog/datasets/passive_dataset/

[11] A. Habibi Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *Journal of Cybersecurity and Privacy*, vol. 3, no. 1, pp. 108–116, 2023.

[12] J. Azimjonov and T. Kim, "Designing accurate lightweight intrusion detection systems for iot networks using fine-tuned linear svm and feature selectors," *Computers & Security*, vol. 137, p. 103598, 2024.

[13] M. AlShaikh, W. Alsemaih, S. Alamri, and Q. Ramadan, "Using supervised learning to detect command and control attacks in iot," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 14, no. 1, pp. 1–19, 2024.

[14] S. Ahmed, "A study of ml algorithms for ddos detection," *International Journal for Research in Applied Science and Engineering Technology*, 2021.

[15] J. Lee, S. Kim, and H. Park, "Ensemble learning for network traffic classification: A comparative study of svm and random forest," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 2105–2118, 2023.

[16] A. Olufunsho *et al.*, "2019–2023 in review: Projecting ddos threats with arima and ets," *IEEE Access*, vol. 11, 2023, accessed: 2024-11-04. [Online]. Available: https://ieeexplore.ieee.org/document/10439150

[17] Y. Zhang, X. Li, and Z. Wang, "Advanced feature engineering techniques for enhancing machine learning models in cybersecurity," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 1456–1470, 2023.

[18] J. Lee and S.-Y. Chung, "Robust training with ensemble consensus," *arXiv preprint arXiv:1910.09792*, 2019.

[19] F. Alqahtani, A. Shahrour, and A. Albahli, "An adaptive ddos detection system using machine learning and real-time analytics," *Computers & Security*, vol. 125, p. 103456, 2023.

[20] T. Nguyen, Q. Pham, and N. Phung, "Real-time ddos detection using deep learning techniques," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 356–369, 2019.

[21] F. Alqahtani, A. Shahrour, and A. Albahli, "An adaptive ddos detection system using machine learning and real-time analytics," *Computers & Security*, vol. 91, p. 101745, 2020.

[22] X. Li, Y. Chen, and Z. Wang, "Federated learning for cybersecurity: A comprehensive review," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1234–1256, 2023.

[23] A. Kumar, S. Singh, and R. Gupta, "Edge computing-based real-time ddos detection using machine learning," *Journal of Network and Systems Management*, vol. 30, no. 4, pp. 1–25, 2022.

[24] S. Kim, J. Park, and H. Lee, "Real-time ddos detection using federated learning and edge computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 2345–2358, 2023.

[25] L. Zhang, X. Wang, and Y. Liu, "Adversarial training techniques for robust ddos detection," *Journal of Machine Learning Research*, vol. 24, no. 1, pp. 1–30, 2023.