RESEARCH PAPER

ENHANCED DDOS DETECTION USING MACHINE LEARNING

[1]Ganapa Vyshnavi, Dr. [2]Alaparthi Sudhir Babu (Professor)
[1,2]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
[1] vyshnaviganapa2002@gmail.com , [2] asudhirbabu@kluniversity.in

Abstract: DDoS attacks remain a major threat to networked systems that lead to a huge financial loss and disruption of services. The requirements include intelligent data-driven detection mechanisms on the basis of which a traditional signature-based intrusion detection framework cannot effectively respond to emerging attack patterns. This paper recommends the use of ML and ensemble learning algorithms to increase the strength of the detection and classification task in a DDoS detection system. The models are trained and tested on the NSL-KDD dataset in binary classification (normal vs. DDoS traffic), and the train and test have an 80:20 ratio. In order to increase the model in generalizability, the following data preprocessing has been carried out: normalization, cleansing and refining the features. The metrics used to compare the performance of different classifiers are accuracy, precision, recall, and F1-score. Some of the classifiers that are used are KNN, LR, RF, Voting Classifier, and Stacking Classifier. Experimental results indicated that the ensemble Voting and Stacking classifiers obtained an accuracy of 100 percent, at a perfect precision, recall, and F1-score quite higher than standalone models, whereas at 99.8 percent, the model under the category RF was doing. The proposed ensemble based framework has been demonstrated to be more reliable and can be used in deployment within intelligent network intrusion detection systems.

"Index terms - *DDoS; Deep Learning; Random Forest; Logistic Regression; KNN; NSL KDD Dataset".*
How to cite this article: Ganapa Vyshnavi Dr. Alaparthi Sudhir Babu (Professor). Enhanced Ddos Detection Using Machine Learning. Int J Drug Deliv Technol. 2026;16(76s):1693-1699. DOI: 10.25258/ijddt.16.76s.160.

1. INTRODUCTION

The fast development of cloud computing, IoT, and massive distributed systems have significantly widened the attack surface of modern networks. One of the most costly, as well as disruptive types of cyberattacks among modern cyber threats, is still DDoS attacks. Reports on cybersecurity in the recent days argue that, DDoS attacks are becoming sophisticated, frequent, and massive and often affect cloud-hosted applications, financial services, and enterprise infrastructure [1]. These attacks aim to overload network bandwidth, computational or application-layer services and, as such, prevent legitimate services to be provided to users.

DDoS attacks are mostly divided into three basic categories according to the way they operate namely the volumetric attacks, protocol-based attacks, and the application-layer attacks. The protocol-based attacks take advantage of the vulnerability in the communication protocols, the volumetric attacks flood network bandwidth, and the application-layer attacks attack individual service endpoints. Conventional signature-based intrusion detection systems have since been found to have diminished effectiveness due to creation of adaptive botnets and low rate covert attacks [2]. Such traditional systems can not detect zero-day threats and innovative traffic patterns because they use predefined attack signatures.

To avoid such limitations, ML methods have been widely used in network intrusion detection. ML-based systems can understand the patterns of traffic behavior and the distinction between normal network flows and malicious work. Recent studies have also revealed that supervised classifiers and ensemble learning methods can significantly improve the detection accuracy and reduce the false positive rates compared to the traditional rule-based systems [3]. However, there are many challenges, which still remain, in spite of the encouraging results. Many complex models of deep learning cannot be deployed in environments with limited resources or that are resource-constrained in nature because of the large amounts of computational resources they use. Moreover, it is true that controlled experimental settings still limit comparative tests of classical machine learning algorithms and ensemble strategies.

Although modern datasets have been introduced in intrusion detection research, NSL-KDD dataset is considered a benchmark because it has reduced redundancy properties and equal representation of classes [4]. However, there have not been systematic studies on ensemble learning methods on this data.

As a result, this paper discusses the effectiveness of classical ML structures and ensemble-structures in detecting DDoS. The study attempts to determine the ability of ensemble learning techniques to enhance the robustness of classification and detection ability and yet remains computationally efficient to enable its practical implementation by introducing systematic comparative study.

2. LITERATURE SURVEY

The ability of ML and DL methods to adjust to changing attack methods and be able to learn difficult traffic patterns has rendered them to be indispensable

in the detection of contemporary DDoS attacks. Decision Trees, RF, and SVM have been successfully used as classical ML classifiers in research on intrusion detection because they are simple to train using labeled network traffic data sets and have a light computational footprint. However, the current studies have shown that ML models often face challenges when they are confronted by high-dimensional traffic features and unevenness in the proportions of classes, especially when faced with covert or low-rate attacks [5].

In a study by Tran et al. to undertake an empirical assessment of the various classical machine learning models on DDoS detection, they conclusively found that ensemble based models tend to be more accurate and recalls more as compared to individual classifiers [6]. This is agreeable with the fact that model diversity is useful in the representation of many aspects of the traffic behavior that could be ignored by single models. Similarly, Najar and Naik performed a comparative study on supervised machine learning algorithms and revealed that algorithms like the RF and the Gradient Boosting achieve higher detection accuracy than the base models like the Logistic Regression [7]. Their study highlights the importance of the role of feature pre-processing and selection in making the classifier more robust.

DDoS detection has also been becoming popular in deep learning approaches, especially when the data is sequential (traffic) and is high-dimensional. Rusyaidi et al. applied deep neural networks in classifying the attacks and achieved significant improvement in the sensitivity of the detection compared to the traditional ML models [8]. They report that it is possible to learn latent features of raw network flows without explicit feature engineering using deep architectures. Nevertheless, it is possible that deep models are applicable in real-time only in resource-heavy environments, due to requirements on large amounts of training data and computation power.

To avoid these limitations, it has been proposed to use hybrid and ensemble approaches incorporating a combination of classical machine learning and deep learning. Ortet Lopes et al. explored a deep learning-powered DDoS determination framework that employs numerous degrees of feature extraction and afterwards categorization, thus showing improved generalization on complicated attack designs [9]. Alasmari et al. proposed a hybrid CNN-LSTM model where convolutional layers are used to extract spatial information, and recurrent layers are used to consider the time features. They were particularly successful in application-layer attacks which had sequence dependencies [10]. Such mixed approaches often deliver a high-quality performance measure, yet also create a high complexity of design.

Current studies have focused on optimization of ML algorithms so as to balance computational performance and detection performance. Al-Eryani et al. have done an evaluation of numerous effective machine learning algorithms to detect DDoS. They found that lightweight models such as optimized RF and XGBoost can be used to have high accuracy at low processing overhead [11].. The importance of algorithmic optimization and hyperparameter tuning in the practical implementation of IDS is based on their study.

Other areas of ensemble unsupervised learning include detection of anomalous traffic in unlabeled environment, performance analyses, and others. In their study, Das et al. presented an ensemble unsupervised ML method that uses a set of clustering and anomaly detection models to detect an anomalous traffic, hence it does not require trained data [12]. Although unsupervised techniques usually have lower accuracy of classification as compared to supervised models, they are beneficial in classification of an unknown type of attack that has not been previously seen.

The contributions made in recent years all point to the fact that classical ML models are still relevant, as they are more interpretable and require fewer resources, even though the accuracy that DL and hybrid methods promise is high. The compromise between the practicality, generalization and performance appears to be at its optimum with many algorithms included in an ensemble strategy, which may be homogeneous or heterogeneous. However, the focus of the current research on a systematic comparison of models, as well as ensemble performance analysis, is predetermined by the lack of systematic comparative assessments, in particular, on benchmark datasets, e.g. NSL-KDD.

Although recent studies have indicated that deep learning and hybrid techniques are very useful in the detection of DDoS, most of the publications do not involve systematic comparisons of conventional machine learning models with the ensemble techniques on a controlled experimental basis. Besides, benchmark datasets, e.g., NSL-KDD, have not been exploited in the context of ensemble robustness analysis and comparative assessment of generalization, even though more advanced benchmark datasets have been widely used. The practicality and interpretability of the models are often not considered in the present literature, which often focuses on performance indicators.

To overcome these limitations, this paper performs an ordered comparative study of the classical classifiers and ensemble methods on the NSL-KDD dataset. This is aimed at determining the practical feasibility of

deployment, the reliability of performance, and the strength of detectiveness.

3. METHODOLOGY

i) Proposed Work:

To enhance the identification of DDoS attacks, the proposed system uses ML algorithms, including the use of Logistic Regression, KNN, and the use of RF. It consists of three major processes: classification, feature extraction and data collection. This approach improves cybersecurity since it can detect DDoS attacks accurately in the network traffic [1, 2, 3]. The DDoS detect project has used a strong ensemble strategy and has a Voting Classifier that consists of the Ran Forest and the AdaBoost and Stacking Classifier which consists of the Ran Forest, the MLP and the LightGBM. These models are aimed at improving the performance of the system through the complementary advantages of diverse ML algorithms [13]. In addition, a user-friendly Flask application that is integrated with SQLite has been developed, which encompasses user signup and signin capabilities to be able to effectively test the user. This improve the level of access and usability of the system in real-world cybersecurity applications.

The data is gleaned and ready during the beginning of the system. The NSL KDD dataset [8] is commonly used in this case, because it consists of network traffic data that has labeled examples of normal traffic and DDoS attack traffic. Data preprocessing is a critical process and involves cleansing, transformation, and preparation of raw data to ease the analysis. It includes the handling of missing values, removing the duplicates and normalization or scaling of values to ensure consistency. The process of feature selection involves the process of selecting the most significant attributes or features of a dataset. This step aims at improving the efficiency of the detection process, to reduce dimensionality. The typical methodologies include feature ranker of importance, mutual information or correlation analysis. It is the base of the system as in this step various machine learning classification algorithms will be deployed to detect DDoS attacks. The implemented models in this scenario are KNN, LR, RF and Voting Classifier, as well as Stacking Classifier. Each algorithm processes the preprocessed data to differentiate network traffic into normal and DDoS attack. Performance metrics are put in place to determine the effectiveness of the DDoS detection system. Such criteria include the measures of precision, recall, accuracy, and F1-score. These measures measure how the system can distinguish legitimate and malicious traffic.

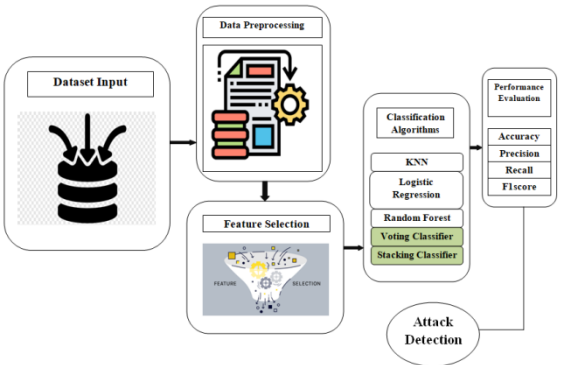

Fig.1 Proposed architecture

iii) Dataset collection:

NSL-KDD DATASET

In this task, we have utilized the NSL-KDD dataset [8] to train ml models. It provides a diverse range of network traffic information, including improved labeling and various types of attacks. The data is an asset in the test of machine learning models and evaluation of intrusion detection systems in a well-balanced and controlled environment.

Thus, these are the five most successful rows of the NSL-KDD dataset. Thus it contains 43 columns; here we are giving a sample of them.

```
train_data.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot
0	0	tcp	ftp_data	SF	491	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0

5 rows × 43 columns

Fig.2 NSL-KDD dataset

iv) Data Processing:

The data processing involves transformation of raw data into useful information to businesses. Overall, the processing of data is the responsibility of data scientists and it includes collecting data, structuring it, cleaning it, verifying and analyzing it and transforming information into an understandable format like a diagram or document. Data process can be done in three ways namely manual, mechanical and electronic. We are aiming to add information value and advance the decision-making procedure. This enables the businesses to improve their operations and make strategic decisions within a timely manner. This is highly affected by automated data processing systems such as the computer software programming. It can convert large volumes of data including big data into insights that can be translated into valuable information to be used to make decisions and quality management.

v) Feature selection:

The process of identifying the most relevant, consistent and non redundant features that should be used in building a model is known as feature selection. With the increasing quantities and types of datasets, there is a need to slim the datasets down systematically. The main aim of features selection is to increase the predictive model and reduce the cost of modelling.

Selecting the most important features to be incorporated into ML algorithms is referred to as feature selection, and it is one of the main elements of feature engineering. To ensure that the number of input variables is reduced, feature selection methods are applied to remove redundant or irrelevant features and also ensure reduced set of features which are most relevant to the ML model itself. The main benefits of feature selection before, as compared to letting the machine learning model identify the most important features [16].

vi) Training and Testing:

To ensure the unbiased model assessment, a holdout validation model was used to divide the dataset. All the data was divided into 80 training and 20 testing. The parameter estimation and model learning was done using the training subset and the testing subset was only used to assess the performance, to determine the ability of generalization. The implementation of cross-validation was not done, and there was no separate validation set. This allocation technique provides an easy estimate of the activity of the classifier on data that has not been seen.

vii) Algorithms:

Logistic Regression: LR is a probabilistic linear classifier that takes the form of a sigmoid and is used to model the relationship between input features and binary class results [13]. It becomes the baseline model in determining how separable and generalizing ability of classification problems is achievable, but it also provides interpretability and computational efficiency.

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(w^T x + b)}} \qquad (1)$$

Random Forest: RF It is an ensemble tree-based algorithm, which uses random feature selection and bootstrap sampling as its method of producing a large number of decision trees. It makes it more resistant to noisy and high-dimensional data, has less overfitting, and is more accurate in classification via majority voting.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \qquad (2)$$

KNN: The KNN is a distance classification algorithm that assigns classes to the data samples according to the major class of the closest data samples in the feature space. It plays a role in performance assessment because it provides a non-parametric base model that can capture local data structure and on the same note local data similarity patterns [14].

Voting Classifier: Voting Classifier: is an ensemble method that uses majority voting in order to integrate the predictions of many base learners. It improves the performance of generalization, the reliability of classification, and removes the shortcoming of single classifiers by making decisions in groups by exploiting model diversity.

$$\hat{y} = argmax_c \left(\sum_{i=1}^{n} II\ (\hat{y}_i = c) \right) \qquad (3)$$

Stacking Classifier: The Stacking Classifier is an ensembled method based on layer that is created by using a meta-learner to create final predictions and it uses several base models. This hierarchical combination increases the predictive performance and learns the best decision boundaries using the aggregate model outputs, thus improving the generalization and strength.

$$\hat{y} = g(Y_{base}) = g\left(f_1(x), f_2(x), \ldots, f_m(x)\right) \qquad (4)$$

4. EXPERIMENTAL RESULTS

Accuracy: The test accuracy depends on the ability of a test to discriminate between cases of a healthy and a patient. To ascertain the accuracy of a test, the ratio of true positive and true negative number of cases to all cases that a test has been evaluated should be calculated. This mathematically can be expressed as:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \qquad (5)$$

Precision: Precision measures the fraction of the instances or samples that are accurately correctly classified out of the ones that are correctly classified as positives. The accuracy, therefore, is calculated using the formula below:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \qquad (6)$$

Recall: Recall is a ML measure which determines the ability of a model to recount all relevant examples of a certain category. The proportion of positive observations that the model accurately predicts to the actual positives allow inferences about the completeness of a model in terms of its ability to capture instances of a particular class.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (7)$$

F1-Score: The F1 score is a figure of the precision of a machine learning model. It is a measure of the combination of the precision and recall rate of a model. The accuracy of a model would be gauged by the count of instances in which it has accurately forecasted the entire data.

$$F1\ Score = 2 * \frac{Recall\ \text{X}\ Precision}{Recall + Precision} * 100 \qquad (8)$$

Table.1 Performance Evaluation Table

ML Model	Accuracy	Precision	Recall	F1-Score
KNN	0.997	0.997	0.997	0.997
Logistic Regression	0.883	0.885	0.883	0.884
Random Forest	0.998	0.998	0.998	0.998
Voting Classifier	1.000	1.000	1.000	1.000
Stacking Classifier	1.000	1.000	1.000	1.000

Table 1 that uses Accuracy, Precision, Recall, and F1-Score as metrics presents the performance of the implemented algorithms. The findings show that the Voting and Stacking classifiers have the best performance in the entire evaluation measures as compared to the single models. Among the basic classifiers, RF is better than KNN and Logistic Regression. According to the selected performance measures, the table will allow making a clear comparative analysis of each algorithm.

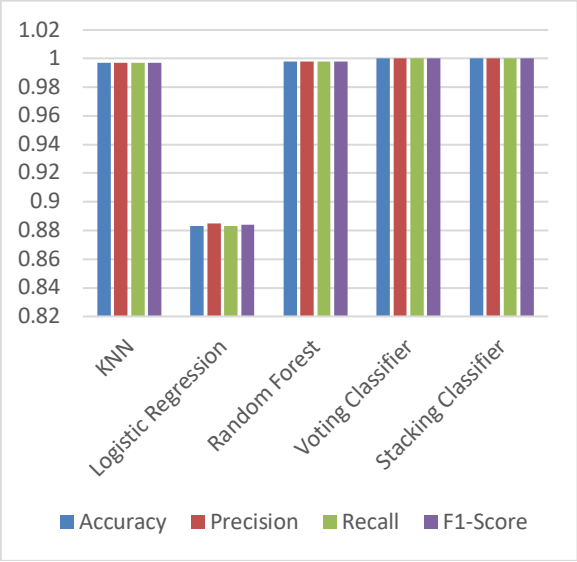

Fig.3 Comparison Graph

Figure 3 shows the comparative performance of all the classifiers based on Accuracy, Precision, Recall and F1-Score. The ensemble models (Voting and Stacking) receive almost perfect scores and exceed the single classifier, and the Logistic Regression achieves relatively lower values of the metrics.

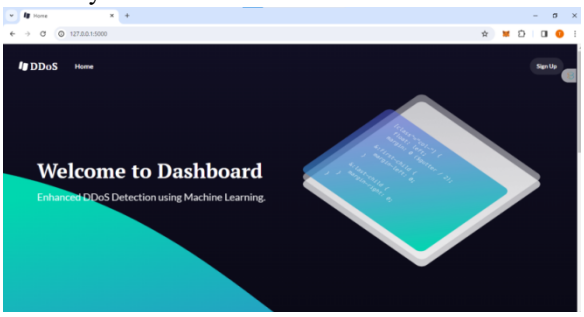

Fig.4 Home page

Figure 4 shows the homepage of the proposed DDoS detection system. The interface serves as a user-friendly entry point to the application and includes a welcome dashboard with branding of the system and a focus on better DDoS detection through machine learning.

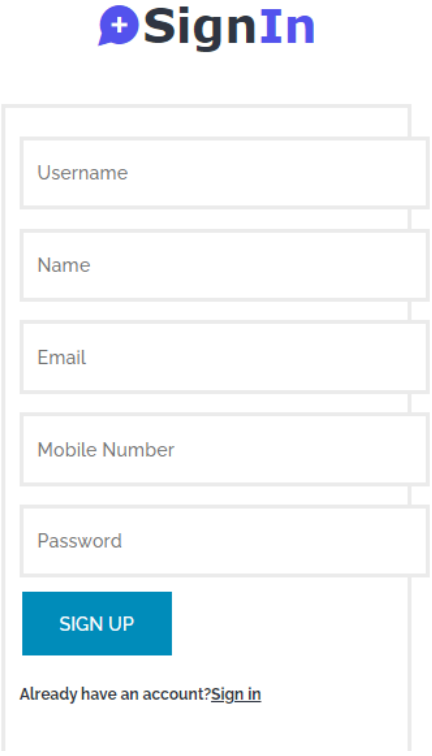

Fig.5 Signin page

Fig. 5 represents the process of registering the users and entails entering the personal information and credentials in the input fields provided to successfully create an account in the system.

Fig.6 Login page

In order to connect to the system and register within an account, the authentication process is depicted in Fig. 6 whereby the user enters an administrator username and a password to log in to their account.

Fig.7 User input

The actual process of data entry is shown in Fig. 7, which entails entry of certain network parameters in the respective fields as a way of enabling the system to produce a prediction through the values that are entered.

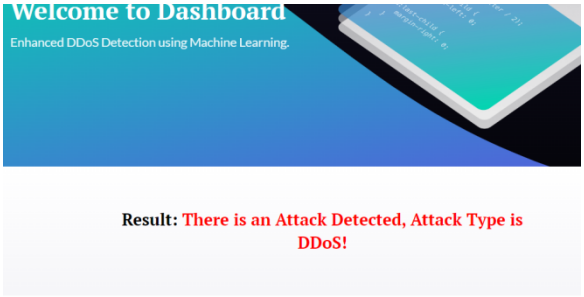

Fig.8 Predict result for given input

The ML model is tasked with the responsibility of processing the input data to determine and inform the user of a detected DDoS attack. The achievement of the final classification outcome is presented in Fig. 8.

5. DISCUSSION

The experimental findings support the claim that the ensemble learning models, namely the Voting and Stacking classifiers did better than the underlying base classifiers in the Accuracy, Precision, Recall and F1-Score measures. The Logistic Regression had a lower detection capacity than the Rand Forest that did very well by itself. These findings imply that the amalgamation of a number of classifiers enhances the stability of the decision-making and reduce the error of incorrect classification, especially in binary DDoS detection.

The fact is that the numerical analysis proves that the ensemble methods give more consistent results of classification with all the metrics of evaluation, although the graphical comparisons show the difference of the performance. This improvement in generalization ability can be ascribed to diversity in the models which are achieved by the compensation of individual weaknesses by several learners. Conversely, the differences in performance of the two methods used ensemble and Random Forest are relatively small, which suggests that the dataset might be biased in favour of tree-based learning models.

Even though the proposed method has achieved high classification scores, it has some limitations. The test was first performed on the NSL-KDD data that, though popular, might not be a correct reflection of the modern large-scale and highly dynamic network situations. Second, the researchers applied the holdout validation strategy that lacked cross-validation that can limit the robustness evaluation. Third, experimental validation of real-time deployment factors such as scalability, memory usage and latency was missing. These features are fundamental to intrusion detection systems that can work on the production level and require further research.

The novelty of the research is the systematic comparative evaluation of classical machine learning models and the ensemble methods in a single consistent experimental set-up utilizing a benchmark

dataset. In contrast to many of the recent works that focus mainly on hybrid neural architectures or deep learning, this study focuses on explaining interpretable ensemble strategies and computationally efficient models applicable to the practice deployment setting. In a practical sense, the results show that the ensemble classifier can enhance the reliability of detection even without the high level of computational resources that are usually required in the deep learning architecture. This makes the approach potentially suitable in the case of small to medium sized enterprise networks which are limited in terms of resource availability. In the future studies, the assessment should be expanded to modern datasets, cross-validation measures should be adopted, and the metrics of computational performance are to be evaluated to ensure that they are applicable in practice.

6. CONCLUSION

The paper presented a machine learning-based DDoS detection framework, which has a systematic comparative analysis between classical and ensemble classifiers. In this analysis, the NSL-KDD data were used. The experimental findings indicate that ensemble algorithm, especially Voting and Stacking classifier, performed better than single models on determinacy of detection thus exhibiting improved classification resilience and stability. Even though these findings are promising, it is crucial to note that there are several limitations. The analysis has been done using one benchmark dataset which might not be a true picture of how big networks use traffic today. Also, there is a potential limitation of the evaluation of generalization through holdout validation strategy, and real-time performance measures, such as computational complexity, scalability, and latency were not empirically tested. To enhance scalability of dynamic network environments, future studies ought to explore lightweight feature selection or model optimization methods, real-time deployment performance, integrating k-fold cross-validation as a robustness validation tool, and test on more contemporary data of intrusion detection (e.g., CIC-IDS, UNSW-NB15).

REFERENCES

[1] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, “DDoS attacks and machine-learning-based detection methods: A survey and taxonomy,” *Engineering Reports*, vol. 5, no. 12, e12697, 2023.

[2] X. Fu et al., “Deep learning techniques for DDoS attack detection: Concepts, analyses, challenges, and future directions,” *Expert Systems with Applications*, vol. 291, 128469, 2025.

[3] Y. Cao, Y. Gao, R. Tan, Q. Han, and Z. Liu, “Understanding internet DDoS mitigation from academic and industrial perspectives,” *IEEE Access*, vol. 6, pp. 66641–66648, 2018.

[4] A. Marzano et al., “The evolution of bashlite and mirai IoT botnets,” *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018.

[5] S. Sontowski et al., “Cyber Attacks on Smart Farming Infrastructure,” *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, 2020.

[6] B. S. Tran, T. H. Ho, T. X. Do, and K. H. Le, “Empirical performance evaluation of machine learning based DDoS attack detections,” in *Recent Advances in Internet of Things and Machine Learning*, Springer, 2022, pp. 283–299.

[7] A. A. Najar and S. M. Naik, “Applying supervised machine learning techniques to detect DDoS attacks,” in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, IEEE, 2022, pp. 1–7.

[8] M. Rusyaidi, S. Jaf, and Z. Ibrahim, “Detecting distributed denial of service in network traffic with deep learning,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 34–41, 2022.

[9] I. Ortet Lopes et al., “Towards effective detection of recent DDoS attacks: A deep learning approach,” *Security and Communication Networks*, 2021, Article ID 5710028.

[10] T. Alasmari, A. Alshomrani, and L. Hsairi, “CNN-LSTM based approach for DDoS detection,” in *2023 Eighth International Conference on Mobile and Secure Services (MobiSecServ)*, IEEE, 2023, pp. 1–6.

[11] A. M. Al-Eryani, E. Hossny, and F. A. Omara, “Efficient machine learning algorithms for DDoS attack detection,” in *2024 6th International Conference on Computing and Informatics (ICCI)*, IEEE, 2024, pp. 174–181.

[12] S. Das, D. Venugopal, and S. Shiva, “A holistic approach for detecting DDoS attacks by using ensemble unsupervised machine learning,” in *Future of Information and Communication Conference*, Springer, 2020, pp. 721–738.

[13] K. Kumari and M. Mrunalini, “Detecting Denial of Service attacks using machine learning algorithms,” *Journal of Big Data*, vol. 9, 56, 2022.

[14] E. S. Alghoson and O. Abbass, “Detecting Distributed Denial of Service Attacks using Machine Learning Models,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, 2021.