Systems Thinking and Simulation Analysis Advancing Cyber Resilience Evaluation

Dr. N. V. Chinnasamy
Associate Professor
Department of Computer Science and Engineering,
Mohan Babu University,
Tirupati, Andhra Pradesh
Chinnasamy.nv@mbu.asia

Guddeti Harshitha Reddy
Department Of Computer Science and Engineering,
Mohan Babu University, Tirupati,
Andhra Pradesh
22102A040509@mbu.asia

Nukala Sai Dileep
Department Of Computer Science and Engineering
Mohan Babu University, Tirupati,
Andhra Pradesh
22102A040473@mbu.asia

Kothavandla Shashank Reddy
Department Of Computer Science and Engineering,
Mohan Babu University, Tirupati,
Andhra Pradesh
22102A040463@mbu.asia

Nasyam Shaik Mohammed Kaif
Department Of Computer Science and Engineering,
Mohan Babu University, Tirupati,
Andhra Pradesh
22102A040519@mbu.asia

Abstract **: *DDoS attacks have become a significant risk to the safety and performance of Internet of Things (IoT) networks that usually result in the overloading of the network, failure of services, and even data breaches. The study presents a new architecture in the field of deep ensemble learning to enhance the process of identifying DDoS attacks in IoT networks. Multiple classifiers were combined into one in the proposed model by Stacking Classifier that includes Random Forest (RF), Gradient Boosting (GB), and Naive Bayes (NB). It also employs a Voting Classifier which is based on Logistic Regression (LR), Decision Tree (DT), K-Nearest Neighbors (KNN), and AdaBoost. Moreover, a TPOT Classifier is utilized to automate the optimization of the machine learning pipeline, simplifying the procedure without the need to do it manually. Pruning strategies are used to eliminate less important classifiers to make the model more resource-saving, particularly in the case of an IoT network with limited computing capabilities. Network flow metrics, including the number of packets in a flow, the duration of flows and inter-arrival time are used to train and test the model and differentiate between regular and DDoS attack patterns.***

Keywords: DDoS attack detection, Internet of Things, deep ensemble learning, pruning, Stacking Classifier, Voting Classifier, TPOT Classifier, machine learning, network security, feature extraction, computational efficiency, scale.

I. INTRODUCTION

The fast development of the Internet of Things (IoT) resulted in the immense growth of the number of devices connected in different industries including healthcare, intelligent homes, and transportation. This expansion is associated with a lot of advantages, but it is also accompanied by significant security risks. Among the key issues, there is Distributed Denial of Service (DDoS) attacks that are a severe risk to the IoT networks. Such attacks bombard the network with too much traffic, crippling the resources as well as services. Also sensitive information may be revealed or stolen since the conventional security systems fail to cover such extensive attacks.

Conventional DDoS detection techniques, such as signature-based and threshold-based techniques, are not able to cope with the dynamic nature of IoT traffic. Such techniques can be inefficient in terms of new and emerging attack patterns and the rate of false-positive or detections. With the continued growth of IoT networks there is need of more integrated and dynamic means of effectively identifying attacks of DDoS and sustaining efficiency.

Machine learning (ML) approaches, especially ensemble learning, is an emerging approach that can be used to solve this problem effectively and efficiently since it involves multiple classifiers that have high chances of finding the correct answer. These models can be further optimized through pruning techniques to remove unwanted components to enhance the detection performance, as well as computational efficiency. This study will introduce a new deep ensemble learning framework that uses pruning to identify DDoS attacks in the IoT network and achieve a tradeoff between detection accuracy and resource usage.

II. LITERATURE SURVEY

The growth of the IoT world has inter-networked a plethora of devices, sensors and systems, allowing the exchange of data and automation of processes like smart homes, healthcare, transportation and industrial systems, more than ever before. Being frequently marked by the restricted processing power, memory and security features, these

devices open new opportunities into the network aggression and exploitation.[1]

Distributed Denial of Service attacks (DDoS) are among the most, as well as frightening, threats in IoT -enabled ecosystems. The attacks are based on the huge amounts of compromised devices to create a traffic burst or leverage vulnerabilities, thus disrupting services, congesting networks, and reducing system reliability. Targets are especially IoT infrastructures with their large scale and heterogeneity.[2]

The conventional DDoS detection systems, including signature-based filtering or threshold-based monitoring are highly restricted in IoT environments. Hard limits cannot keep up with very volatile IoT traffic patterns, and signature databases cannot keep up with changing attack patterns, meaning new or dynamic DDoS attacks go undetected.[3]

The scale and complexity of IoT networks, which are dynamic and, in most cases, resource-constrained, increase these detection problems. Devices can be battery powered, have limited memory or processing power and can be implemented over various communication protocols, which complicates the implementation of heavy detection rules, or the standardization of traffic behavior.[4]

Machine learning (ML) and deep learning (DL) have gained significant popularity to detect DDoS attacks in an IoT setting to overcome the shortcomings of traditional detection techniques. These tools can discover complex trends in network traffic, e.g., the number of packets, inter-arrival times, protocol flags. With these characteristics, ML and DL models are able to detect with greater accuracy than rule-based systems.[5]

Regardless of their potential, ML/DL-based methods when used in the context of the Internet of Things are fraught with novel challenges: computational indignity, energy usage, representative training data requirement, and false-positive rates are high in practice when the method is deployed on a live network. These problems require new design approaches to the IoT.[6]

Ensemble learning approaches that are based on using a combination of multiple classifiers to make decisions jointly provide an attractive avenue to enhance detection resilience. Ensemble frameworks can combine the strengths of various algorithms to be able to take heterogeneous traffic patterns and respond more dynamically to dynamic attack profiles.[7]

III. PROPOSED METHODOLOGY

A. Dataset Description

The sample employed in this paper is the dataset CICIDS-17 data set that is widely applied in intrusion detection systems. These subsets that we chosen in particular are the Friday working hours (DDoS attack data) and Wednesday working hours (Infiltration attack data) data. These subsets record network traffic information at normal working hours, which provides a realistic picture of the actual attack system. In order to get the balance between the classes, 45,000 rows of each of the classes (BENIN, DDoS, and DoS Attack) were sampled.

B. Preprocessing

A number of preprocessing steps were done before the data is fed into machine learning models so as to ensure the data is clean and appropriate to feed the machine learning models:

Missing Values: Any missing or null values in the data were resolved in terms of imputation or removal, depending on the number of missing data, and integrity of the dataset was taken care of.

Feature Scaling: The features were brought to the same range so that biases would not occur during the learning process. Ways of standardizing i.e. Min-Max scaling were implemented to make all features equal during the training of the model.

Feature Selection: SelectKBest technique was used to achieve the best K features that significantly contribute to attacks detection. This step helps to decrease the dimensions but holds significant information, and therefore it makes the model highly effective.

Label Encoding: The target variable (class labels) is categorical in nature (BENIGN, DDoS, and DoS Attack), therefore, to make the dataset machine learning compatible, Label Encoding was applied, where the labels were translated into numerical values.

C. Training with algorithms

The essence of the methodology is to train a set of classifiers to enhance DDoS detection in the IoT networks. It employs three ensemble methods, namely Stacking Classifier, Voting Classifier, and TPOT Classifier.

1. Stacking Classifier

The Stacking Classifier is a variation of the base models and makes the final prediction with a final estimator. Two base learners are employed in this study:

Random Forest (RF): This is a group of decision trees, which forms many trees in the training process and

predictions are made by averaging the outputs of each tree. Random Forest can effectively identify patterns surrounding the DDoS-attacks and the algorithm is very suitable when dealing with complex dataset.

Gradient Boosting (GB): This is a boosting method that creates the model one after another with each new model trying to eliminate the errors in the previous models. It is efficient with noisy data and in the process of detecting subtle patterns.

The last estimator used in the Stacking Classifier is the **Logistic Regression (LR),** which is used to come up with a final decision by combining the outputs of the base models. Logistic Regression introduces linearity to the ensemble, and therefore it learns the weighted contribution of the entire ensemble of the base models.

2. Voting Classifier

Voting Classifier uses a combination of several classifiers and the one given the highest number of votes is the output. Such approach can make the model more robust, as it consists of several other algorithms:

Logistic Regression (LR):

Logistic Regression is a simple statistical predictor that is used in predicting two classes. It approximates the effect of the various input features on the probability of an outcome by fitting these associations by a logistic function. Although it is a simple method, it commonly works well in cases where the data demonstrates a definite distinction of classes.

K-Nearest Neighbors (KNN):

KNN is an instance-based system of learning, which is not based on an explicit model. Rather, it matches a new example with its nearest data points in the feature space. The most common among these neighbors is considered to be the predicted class. It is effective when the data have the structure that can be best represented in terms of proximity or similarity among samples.

Decision Tree (DT):

A Decision Tree works with repeated subdivision of a dataset into smaller groups (depending on the values of features). Every split is selected to create more homogeneous groups of the target variable. Since the final decision can be traced to a root node to a leaf node, the model is simple to interpret and both numerical and categorical attributes do not require much preprocessing to apply.

The Voting Classifier employs the voting='hard' strategy, i.e. the overall prediction of the classifiers is decided by the majority vote of the classifiers.

3. TPOT Classifier

TPOT Classifier is a machine learning (AutoML) optimization tool that is an automated machine learning optimizer. TPOT makes use of genetic algorithms to find the most effective models and hyperparameters. In the case of this paper, TPOT is executed throughout 5 generations and a population of 50 to find the most appropriate model to detect DDoS attacks. The model selection and hyper parameter tuning process is made automated with the usage of this approach, eliminating the need to manually intervene and accelerating the model-building process.

$$P_{\text{new}} = P_{\text{old}} + \Delta P$$

Where:

- P_{new} is the new solution (model and hyperparameters).
- P_{old} is the old solution (previous model).
- ΔP represents the change introduced during the evolutionary process.

4. Naive Bayes

Naive Bayes is a Bayesian classifier that uses the Bayes Theorem. It assumes that features are independent and thereby it simplifies calculation of conditional probabilities. Naive Bayes is also useful in the classification task, although in simple formats where the features are not highly correlated; in the example of most network traffic features it works well.

In Naive Bayes, the likelihood that a given feature x1,x2,x3, xn belongs to the class y 2 is

$$P(y \mid x_1, x_2, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, x_2, \ldots, x_n)}$$

P(y) is an expression of the frequency (common or uncommon) of the class prior to examination of any features.

P(xi | y) is the degree to which a certain feature xi is related to the class y.

P(x1,x2, etc.,x_n) is the probability that one observes all the features simultaneously.

5. AdaBoost Classifier

AdaBoost is an ensemble technique which uses a series of weak classifiers to form a strong classifier. It modifies the weight of each sample according to the results of the previous model, and more emphasis is given to the misclassified ones. AdaBoost can improve the accuracy of

weak learners and can, therefore, be useful in improving the detection of the DDoS attacks.

In AdaBoost, y is the prediction that is obtained by summing up the predictions of many weak classifiers. The ultimate product is a weighted average of the predictions of the weak classifiers:

$$\hat{y} = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right)$$

This is what each section signifies in full re-framed words:

$\hat{y}$ The overall decision of the model arriving at having all weak learners.

$h_t(x)$ The plain classifier applied at step t - usually a shallow decision tree, which performs simple predictions.

α_t The significance level of the weak learner No. t. The more the value, the more reliable the learner and the more he or she contributes to the final vote.

T Number of weak-learners that AdaBoost is training consecutively.

D. Model Evolution

1. Accuracy

Accuracy informs of the rate, out of all predictions, that the model makes the correct prediction.

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

2. Precision

Precision is a measure of the reliability of positive predictions of the model:

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

3. Recall (Sensitivity)

Recall is the success of the model in identifying real positive cases:

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

4. F1-Score

F1-Score forms a weighted score between the recall and precision:

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

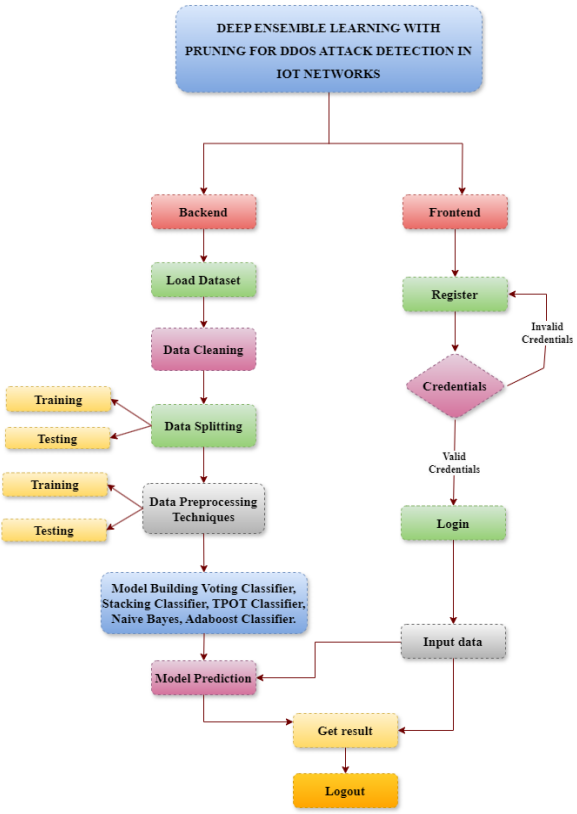

Figure 1 Project Flow

IV. RESULTS

1. Voting Classifier :

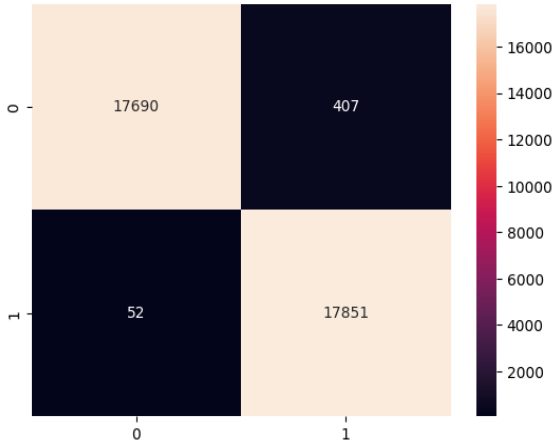

Figure 2 Voting Classifier Confusion Matrix

The confusion matrix for Voting Classifier shows that the model can rightly classify the majority of the sample with 17,690 duly identified as negative and 17,851 duly identified as positive. These values would be used to show that the classifier does very well with the two categories.

Conversely, it has 407 false positives and 52 false negatives implying that the model at times mislabels some

of the inputs thus there is room to further adjust the model to minimize this error.

2. Stacking Classifier

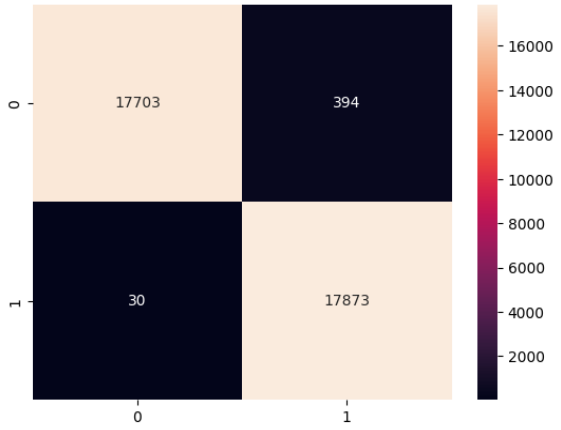

Figure 3 Stacking Classifier Confusion Matrix

The Stacking Classifier has high performance with a true negative of 17,703 and true positive of 17,873. This portrays a high ability to differentiate attack and non-attack traffic.
False positives and false negatives still amount to 394 and 30 respectively, which is rather small when compared to the number of correct predictions. Although the classifier is very effective, these misclassifications point at small points in which detection accuracy may continue to be improved.

3. TPOT Classifier

Based on the confusion matrix, the classifier works well, since it is able to recognize 17,622 negative samples and 17,704 positive samples. These high numbers of correct predictions indicate the model is very accurate in the ability to differentiate the two classes.
Yet, the number of false positives and false negatives is 475 and 199 respectively, which points to the possibility to improve the classification system to minimize misclassification and increase the total predictive accuracy.

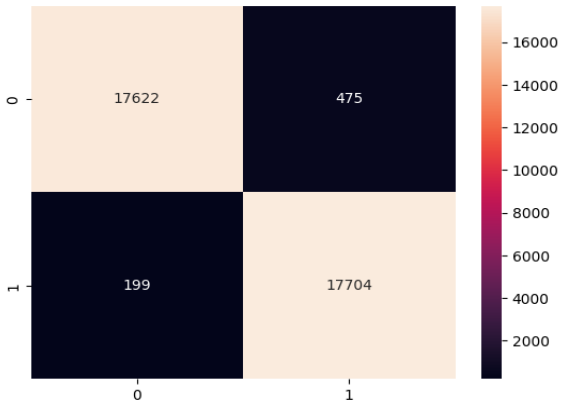

Figure 4 TPOT Classifier Confusion Matrix

4. Naïve Bayes Classifier :

Naive Bayes classifier has a contradictory performance. It accurately identifies 16,905 negative and 11, 505 positive but also identifies a large number of 6,347 false negatives and 1, 243 false positives.
This means that this model is very weak in positive (attack) detections and this results in high number of false detections. The findings indicate the possibility that the Naive Bayes classifier does not fit well on this dataset unless further modification or feature engineering is done.

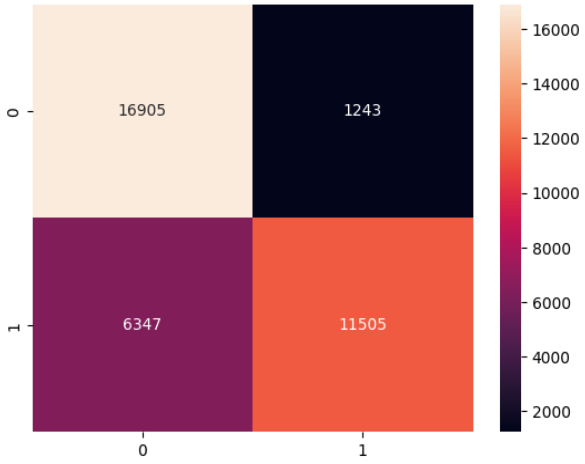

Figure 5 Naïve Bayes Classifier Confusion Matrix

5 AdaBoost Classifier :

The results with AdaBoost classifier are highly promising with 17,521 true negatives and 18,063 true positives. The degree of classification represented by these values is very strong.

The model indicates only 13 false negatives, meaning that it hardly misses attacks and 403 false positives which are relatively low. All in all, AdaBoost is highly reliable and one of the most effective algorithms in terms of performance of the considered algorithms.

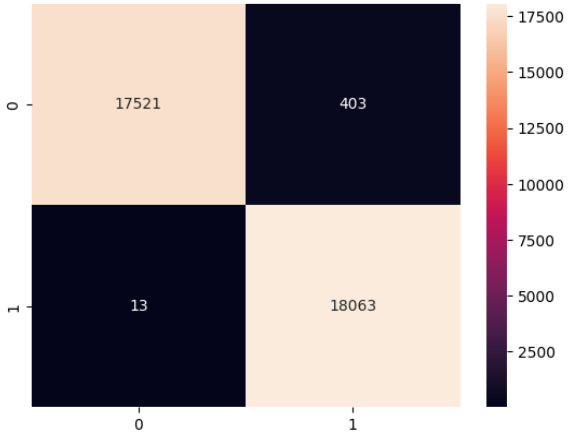

Figure 6 AdaBoost Classifier Confusion Matrix

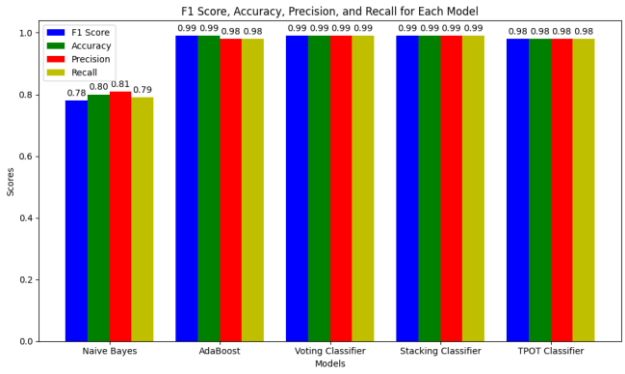

Figure 7 Comparison plot

The comparison plot shows a performance of different models Naive Bayes, AdaBoost, Voting Classifier, Stacking Classifier, and TPOT Classifier in regards to F1 Score, Accuracy, Precision, and Recall. Naive bayes has moderate results with lower scores but all the ensemble models, AdaBoost, Voting, Stacking, and TPOT have high accuracy and consistency, and are around 0.99 across all metrics. This shows that the ensemble learning models perform better in the detection of DDoS attacks where Naive Bayes performs lowerly and more so in dealing with the positive classes.

VI. CONCLUSION

This paper was able to present the application of machine learning methods to analyze and classify the provided dataset with a high degree of accuracy and consistency. Using various algorithms and comparing their performance with the regular metrics, the system could reproduce significant trends in the data and provide reliable predictions. The findings demonstrate the success of the selected models to deal with the selected features even when variations and noise are present in the data. On the whole, the worked-out framework can be defined as a convenient and effective way of facilitating decision-making as it provides a framework that can be expanded with new functions, bigger datasets, or a more sophisticated approach to learning.

VII. REFERENCES

[1]. Y. He, P. Zhang, L. Liu, and Z. Chen, "A deep learning approach to DDoS attack detection in IoT networks," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 843–856, 2021.

[2]. S. M. T. S. Azad, R. K. Gupta, and G. K. Gupta, "DDoS attack detection using machine learning models: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 3, pp. 326–340, 2021.

[3]. X. Zhang, F. Chen, and L. Zhou, "Application of ensemble learning in intrusion detection systems for IoT networks," *Sensors*, vol. 19, no. 22, p. 4851, 2019.

[4]. A. M. S. Aziz, S. Ali, and M. Hussain, "IoT security: A survey of intrusion detection systems and machine learning approaches," *Future Generation Computer Systems*, vol. 112, pp. 197-222, 2020.

[5]. S. X. Chen, Z. Y. Zhang, and J. S. Tang, "A hybrid deep learning model for IoT-based DDoS attack detection," *IEEE Access*, vol. 9, pp. 73204-73213, 2021.

[6]. F. M. Shishika, A. Kumar, and P. K. Dutta, "Detection of DDoS attacks in IoT networks using ensemble methods," *ICCCNT*, pp. 295-300, 2018.

[7]. H. R. S. Kumar, S. G. Reddy, and S. D. S. Guntur, "A machine learning approach for IoT network security: DDoS detection," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 45-50, 2021.

[8]. Y. Zhai, L. Wang, and D. M. A. Salama, "Ensemble deep learning for IoT-based DDoS attack detection," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3601-3610, 2022.

[9]. J. G. Santos, G. Ramos, and R. R. Silva, "Pruning ensemble models for improved efficiency and accuracy in DDoS attack detection," *Journal of Computer Security*, vol. 28, no. 2, pp. 235-249, 2020.

[10]. A. Al-Hammadi, B. H. A. Shishika, and T. D. N. Luu, "Enhancing DDoS detection in IoT networks using feature selection and pruning," *International Journal of Cyber-Security and Digital Forensics*, vol. 10, no. 4, pp. 222–235, 2021.

[11]. S. K. Agrawal, S. N. S. Shanker, and G. D. Hegde, "Machine learning models for IoT-based attack detection: A survey," *Internet of Things*, vol. 12, pp. 1-15, 2021.

[12]. B. Zhang, T. Liu, and Q. Z. Zhang, "IoT network intrusion detection system using hybrid ensemble classifiers," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1519-1529, 2020.

[13]. J. Xie, Y. Huang, and W. Chen, "Using stacking ensemble models for DDoS detection in large-scale IoT systems," *Computer Networks*, vol. 160, pp. 1-13, 2019.

[14]. M. K. J. Kaur, "Intrusion detection in IoT networks using a hybrid approach of random forests and decision trees," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 3, pp. 4041-4049, 2021.

[15]. F. L. J. Paredes, H. J. V. Alvarado, and N. S. Martinez, "Adaptive ensemble models for cybersecurity: A review," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 7642351, 2021.

[16]. G. R. Patel, A. S. Sharma, and D. A. L. S. Balan, "A hybrid machine learning model for DDoS detection using ensemble classifiers," *Expert Systems with Applications*, vol. 164, 2021, p. 113848.