Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 252 (2025) 490–499

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Evolutionary Computing and Mobile Sustainable Networks

A Smart Approach for Early Detection of DDoS Attacks: Artificial Neural Network and Random Forest Hybridization

Ishmam Ahmed Ongshu[a], Ahmed Wasif Reza[a,*], Md. Emad Uddin Aksir[a], Mohammed Tasiful Alam[a], Md. Mahfuzul Haq[a], Farhana Alam[a]

[a] *Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh*

Abstract

Advances in networking technology have made Distributed Denial of Service (DDoS) attacks a real danger to today's networks. Using logical reasoning, the network flow circumstances may be classified as an attack or a routine state to mimic DDoS detection. This research builds an Artificial Intelligence (AI) system using current improvements in Detection System (DS) and Artificial Neural Network (ANN) algorithms advances. It examines User Datagram Protocol (UDP) floods, ping floods, Transmission Control Protocol (TCP) floods, and land attacks to better understand attack behavior. The categorization model for DDoS attacks is constructed using machine learning approaches. Once trained and evaluated, the model can identify unlabeled benign or malicious network data. Experiments reveal that Decision Tree (DT), Random Forest (RF), Naïve Bayes, and ANN are more accurate in separating ordinary and attack traffic. The ANN is used to extract optimal features from Internet of Things (IoT) Intrusion Detection System (IDS) data. The DS Algorithm, a new RF optimizer, is employed for effective feature selection. Performance evaluation of the resulting model called Artificial Neural Network-Random Forest (ANN-RF), is done using the ”Application DDoS Layer Dataset”. RF was selected because it trains faster than DT. We have got 99.998% better accuracy than 99.930% which is the most efficient accuracy from previous work in this field. As per our results, the proposed work has detected smart accuracy and can detect it in real time while keeping the network connected at the same time. Furthermore, we do thorough empirical comparisons of various optimization techniques utilizing a variety of categorization performance metrics. The results confirmed that the proposed technique had a competitive performance across all datasets.

1. Introduction

The Internet has profoundly impacted the world during the last few decades. However, the phenomenal expansion of the Internet has resulted in a plethora of cyber-attacks. Cyber security is one of the most significant risks to today's society, which is estimated to be responsible for annual losses of hundreds of billions of dollars. The first reported Denial of Service (DoS) attacks were launched in 1974, using the assistance of a 13-year-old high school student

[*] Corresponding author. Tel.: +880-1780-099173
E-mail address: wasif@ewubd.edu

10.1016/j.procs.2025.01.008

who had recently learned of a command that could be performed on CERL PLATO. Since then, DoS has evolved into Distributed Denial of Service (DDoS), renowned as the most damaging cyber-attack type. PLATO was the first computerized shared learning system of its kind in the world. To interact with external devices, the command 'ext' (short for external) was used. However, if a system were not linked to an external device, the command 'ext' would cause the system to shut down. DDoS attacks on websites are carried out by combining several machines as an attack tool to carry out cyberattacks on a number of targets in order to maximize the functionality of the operation. The attack pattern is not statistically constrained because DDoS attacks have altered the conventional peer-to-peer attack mechanism. Furthermore, attackers use universal protocols and resources throughout the attack. The sorts of protocols and services utilized make it difficult to differentiate between an attack and a normal operation. There are several challenges in dealing with DDoS [1]. The scientific community and businesses have been studying DDoS detection and mitigation for years. The relevant literature suggests that various research has attempted to address this issue generically. For high and low-volume attacks, a separate collection of publications focuses on remedying the problem. Additionally, despite the Computer Emergency Response Team (CERT) and Request for Comments (RFC) many mitigation suggestions for DDoS attacks, these attacks continue to occur often. According to years of research, DDoS attacks are difficult to recognize and counteract due to continuing issues with configuration and delay caused by the lack of technologies that observe network dynamics without regular human intervention. Autonomous systems that are capable of operating (detecting and mitigating) based on behavior and attributes have piqued the interest of researchers. Artificial intelligence technologies, including Machine Learning (ML), have grown in popularity because they provide greater flexibility in the categorization process, resulting in better identification of hostile traffic. DDoS prevention is provided as a service in the industrial sector by large structures, which are frequently handled by specialist providers such as Akamai, Cloudflare, and Arbor Networks, who have vast processing resources and unique filtration methods. However, the industry faces a number of challenges, including a lack of stability in routing traffic through the Domain Name System (DNS), problems in identifying delayed attacks, and privacy issues that drive away certain consumers and governments [2]. The primary goal of this study is to improve the detection accuracy of DDoS attacks. Our expectations for accuracy in DDoS attack detection have not been met, as shown by our reading of relevant papers. Our goal is to increase detection accuracy, and this idea is a way to do so. According to our results, the proposed work has smart detection accuracy and was designed to detect in real time while keeping a network connection. Using this strategy is a good idea in our paper. Following the success of ML optimization strategies in feature selection, we employed the "Application-Layer DDoS Dataset." This dataset has been created primarily to address the challenges of DDoS attacks in application layers. A neural network and Random Forest (RF) technique is used to optimize the dataset. To achieve our goal, we will employ an Artificial Neural Network (ANN) approach for learning and extracting complex features. In addition, using a unique real-world incursion dataset, we intend to compare the efficacy of our suggested methodology with that of other popular methods. Finally, we suggest a new DDoS detection system that utilizes the advantages of both ANN and RF approaches.

2. Related Work

In the Cyber Attack Detection paper [1], the authors used RF, Gradient Boosting, SVM, and Logistic Regression with the CTU-13 dataset on Windows 10. The system effectively detects botnet traffic but struggles with traffic delays and discovering new botnets, achieving 90% accuracy. In [2], RF, DT, Linear Regression, and Adaboost were used for DoS/DDoS detection with CIC-D05, CICIDS2017, and CSE-CIC-IDS2018 datasets. The system had an accuracy of 99.93%, but distinguishing malicious and legitimate traffic remained a challenge, with risks to cloud servers. [3] used Linear Regression and the CICIDS2017 dataset for DDoS detection, achieving 93.85% accuracy. The limitation was relying on a single IDS and dataset, leading to lower performance. In [5], FS methods and eight algorithms (e.g., CapSA, GWO) were applied to multiple datasets (KDD99, NSL-KDD), achieving 99.997% accuracy. However, FS and complex DL models are needed for faster convergence. [6] used Naïve Bayes, RF, MLP, and J48 on the Alkasabeh dataset, reaching 98.1% accuracy but had issues with rerouting traffic and data preprocessing. Lastly, in [7], KNN, Naïve Bayes, DT, and RF were used for real-time DDoS detection in SDN, achieving 97.38% accuracy, though high failure rates (5.07% for KNN, 28.59% for Naïve Bayes) remained a concern for IT staff stress and server failures.

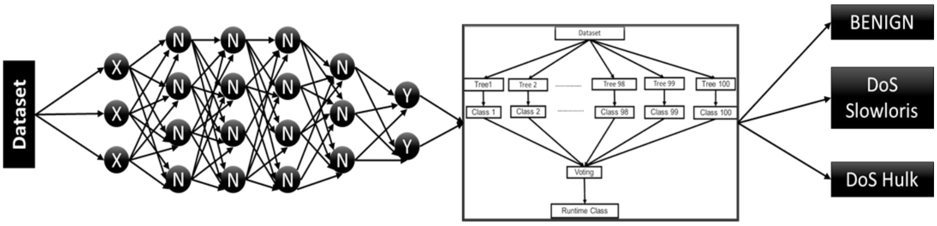


Fig. 1: Data Preprocessing Scenarios.

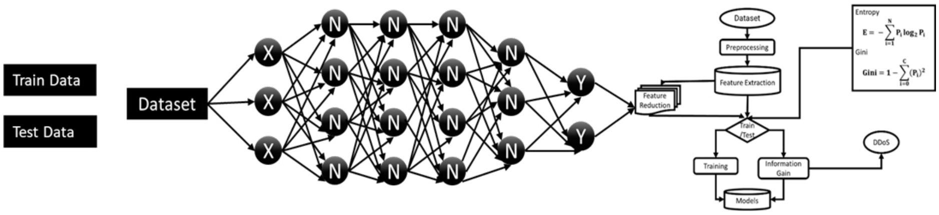


Fig. 2: Training and Testing Scenarios.

3. Materials and Methods

The 'application layer DDoS' dataset from Kaggle is designed to detect DDoS attacks. It contains 77 features, with 8,093,363 rows for training and 346,872 rows for testing. We used Python and the sklearn library to implement ANN, RF, DT, and Naïve Bayes on Windows 11. Our primary algorithm is ANN, which uses 5 layers with 30 neurons each to process key features. After evaluating feature importance with ANN, we applied Random Forest (RF) to further validate the importance. This combined approach is called Artificial Neural Network-Random Forest (ANN-RF). Figure 1 shows the data pre-processing, and Figure 2 illustrates the training and testing process.

For our paper, we have chosen a smart approach to improve the efficiency of our study. In Algorithm 1, we created a smart path by combining RF with DT. Additionally, we pre-processed our data set to improve accuracy. In our dataset, we have three types of output. The first is benign DoS, which is the starting phase detected by Algorithm 1. The second one, DoS Slowloris, detected by Algorithm 1 but not defined by ANN, is a medium level of threat, originating from one device in network security. However, the third one, DoS Hulk, is the most dangerous type, where many devices are used as network blocks. Therefore, determining the selected variables is a challenging task to obtain better accuracy. To attain a more accurate result, we selected some important variables. As a result, we have developed a smarter and more effective algorithm. The dataset used is the **Application-Layer DDoS Dataset** from Kaggle, specifically designed to address DDoS attacks in the application layer. Provide details such as the total number of samples, the number of features (e.g., 77 features), and how the data was split for training and testing (e.g., 80% training, 20% testing).

Our method utilizes equation 1 in RF and a large dataset. Currently, we have access to a subset of those variables. We have calculated the accuracy of our models using two 'while' loops. The initial condition of the first 'while' loop is that the complete dataset is not null. Training and testing data are separated in the second 'while' loop. If the condition is true, we finish the 'while' loop if the number (N) is greater than or equal to 100. For numbers less than or equal to 100, it uses an RF algorithm. The accuracy was computed using equation 2. The 'while' loop terminates if it encounters a null value during computation. If no null values are found, the current model is executed, identifying crucial elements. The algorithm will end by finding the largest number for N. The feature selection process begins with running the Artificial Neural Network (ANN) to process the data. ANN is used initially to extract patterns and understand the relationships between features. After this, Random Forest (RF) is applied to analyze the processed data and generate a feature importance chart. This chart identifies which parts of the data—i.e., which features—are most important, with each feature assigned an importance score. Using the RF chart, we focus on the features marked as most significant, analyzing and organizing them based on their importance. Once the important features are organized, we run RF again, this time with a refined set of features. This step helps further improve the model's accuracy, leading to a remarkable performance of nearly 100%. This two-step process of using ANN for initial data processing and RF for feature selection and refinement ensures both high accuracy and efficiency in detecting patterns and anomalies in the dataset.

i Gini: The Gini impurity is a loss statistic; therefore, higher values are less desirable for our model. Second, it is confined to the classifier form of DT models, since it needs distinct target values (or classes) to deliver a coherent value.

$$Gini = 1 - \sum_{i=1}^{C} P_i^2 \tag{1}$$

C represents the number of classes on the label and Pi represents the ratio of classes at the 'ith' node. It is used within the DT used in the algorithm and in which the value 0 is the best. And if its value is greater than 0, that is, 0.5 or more, the value will be worse [**?**].

ii Coefficient of variation:

$$C_x(P_x) = \frac{\sigma(P_x)}{\mu(P_x)} \tag{2}$$

$\sigma(P_x)$ is a statistical estimate of the variance and $\mu(P_x)$ is the variable's calculated mean value of the variable. It has been used to find the accuracy of a packet. This is what we used to find the packet_length_variance.

iii Sigmoid / Logistic Activation Function:

$$f(x) = \frac{1}{1 - e^{-}x} \tag{3}$$

where x is the input to the sigmoid function and e is Euler's number.

iv Quantile coefficient:

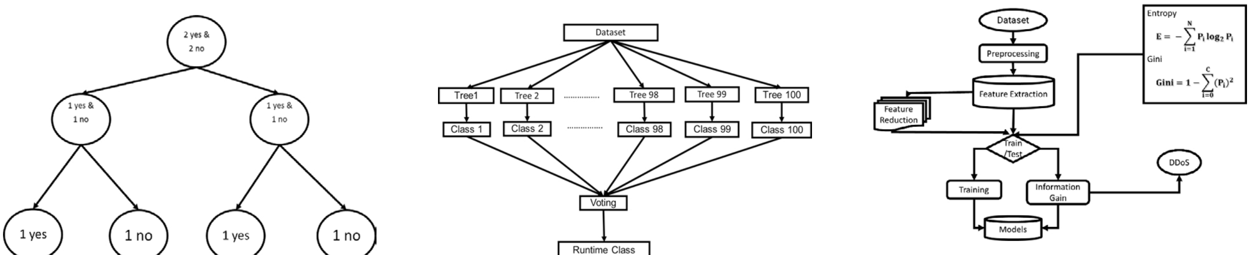
$$cvq = \frac{\hat{Q}_x(1-P) - \hat{Q}_x(P)}{\hat{Q}_x(1-P) + \hat{Q}_x(P)} \tag{4}$$

$$\hat{Q}_x = X_{(k)} + (X_{(k+1)} - X_{(k)}) * f \tag{5}$$

$\hat{Q}_x$ in equation-5 represents the sample p-quantile. Here equation 4, with being $\{X_1, ..., X_n\}$ the independent observational data ordered statistically, $\{X_1, ..., X_n\}$,$k = [p * n]$ and the index 'f' is the index fraction enclosed by X_k and X_{k+1}

v rate of change:

$$rte(X) = \frac{U_x}{S_x} \tag{6}$$



(a) Decision Tree.

(b) Random Forest

(c) Extraction, Transformation, and Labeling of Network Traffic

Fig. 3: Decision Tree and Random Forest Demonstration

In Equation-6, U_X is the number of unique values and S_X is the overall number of X values. The rate of change (ROC) is an important idea allowing investors to identify market trends. In the short term, security with high momentum or a positive ROC typically beats the market. Equations (4) and (6) have been used in our smart detection algorithm to select important variables.

Figure 3a shows the procedure using a DT. Here, we see that there are two instances of “Yes” and two instances of “No”, which we have divided into two parts, with one ”Yes” and one ”No” in each part. After that, if we further break down these parts into more than two categories, each with a “Yes” or “No” answer, we obtain a total of three possible ways to answer .

To improve our accuracy, we used 100 trees shown in Figure 3b. At first, we insert the data set and divide it into 100 separate trees, resulting in 100 potential outcomes. Our research shows the final judgment of which possibility is chosen. Subsequently, the runtime classes are processed, then we achieve the best accuracy [**?**].

Figure 3c explains how we extract, convert, and level network traffic. We preprocess the dataset by removing the features that are not essential. We run the data set through a feature reduction mode to eliminate any extraneous features. That is why we submit the data set to a service called ‘train/test’, which will put it through its paces using two different equations: Entropy and Gini. After that, the dataset is divided into two groups for training and information gathering. Regardless of whether it is a DDoS or not, we get more information about it. Finally, we build the models by combining training and information gain modes [9][10].

4. Results and Discussion

Measurement systems were evaluated using precision equation- 7 (PREC), recall (REC) Equation-8, and F1-Score Equation-9. PREC Equation- 7 measures accuracy, while REC Equation-8 assesses sensitivity to false positives. F1 in equation-9 is the harmonic mean of PREC and REC. True Positives (TP) indicate correct attack detection, True Negatives (TN) occur when no attack is present and none is detected. False Positives (FP) incorrectly flag normal traffic as an attack, and False Negatives (FN) miss actual attacks. These metrics help assess system performance [7]-[**?**]. Several inaccuracies are found in the six selected articles that we shall explore later. The most often used DDoS attack detection algorithms include RF, SVM, Naïve Bayes, and Logistic Regression. In most circumstances, the best accuracy of DDoS attack detection is 99.93%, but we achieve 99.998% using:

1. The ”Entropy” equation- 1 instead of the “Gini impurity” equation-1 .
2. We have set n_estimator to 100, which means that we use 100 random trees to train our model, and that helps to get better accuracy
3. We also use more features in our dataset, which is more effective than our previous dataset, which contains 77 features. That is also a reason for better accuracy.

$$Precision = (1 + \frac{FP}{TP})^{-}1 \tag{7}$$

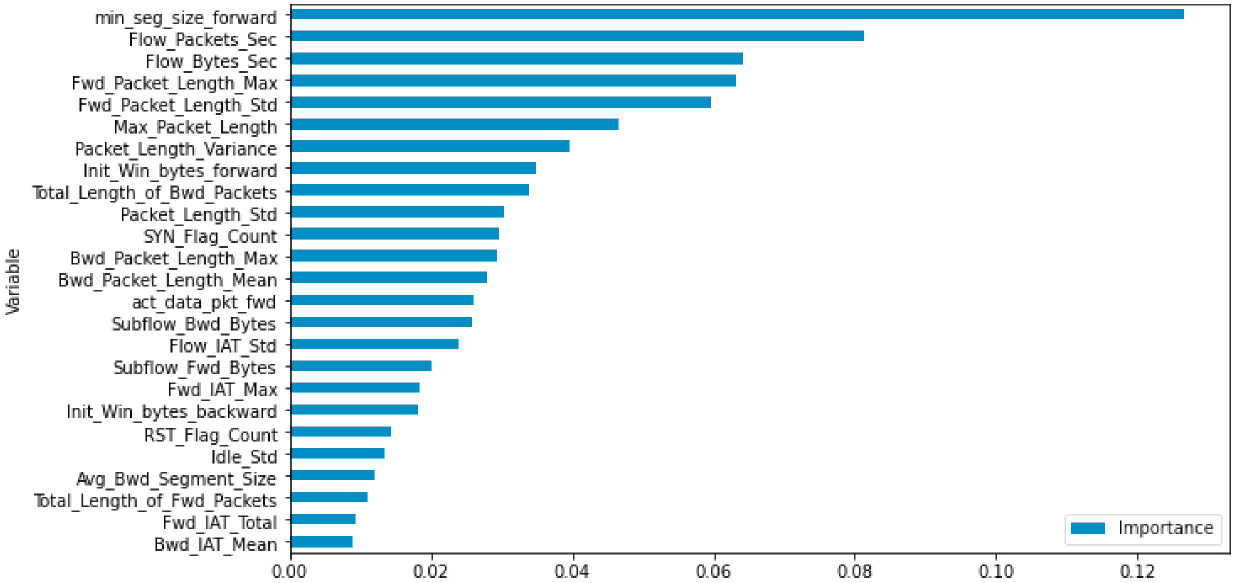


Fig. 4: No. of Variables vs. No. of Models.

Table 1: Comparison with other Hybrid Models.

Method	Session time	Memory Kill	Accuracy
ANN-SVM	16 hour 23min	82%	74%
ANN-KNN	11 hour 24min	55%	52%
ANN-DT	15 min	71%	99.991%
Our proposed model (ANN-RF)	12 min	42%	99.998%

$$Recall = (1 + \frac{FP}{TP})^{-}1 \tag{8}$$

$$F_{Measure} = 2 * (\frac{1}{Precision} + \frac{1}{Recal})^{-}1 \tag{9}$$

Figure 6 shows 25 important features of the RF Algorithm. Figure- 5 shows the important features of the DT Algorithm. In Figure- 6, we created a heat map using 17 features with importance values up to 0.95, utilizing Matplotlib, Seaborn, and Graphviz. This helped validate the decision tree (DT) using 0.05% of the dataset. Table 2 and Table 3 compare results across algorithms, covering three outputs: benign (malware classification), DoS Slowloris (persistent HTTP sessions), and DoS Hulk (obscured high-traffic volumes).

We ensured no overfitting by systematically dividing the dataset, where accuracy consistently improved with more data. Table -1 compares hybrid methods. ANN-SVM was the slowest (16 hours, 82% memory, 74% accuracy), ANN-KNN was faster (11 hours, 55% memory, 52% accuracy), and ANN-DT performed well (15 minutes, 71% memory, 99.991% accuracy). Our proposed ANN-RF method was the fastest (12 minutes, 42% memory) with the highest accuracy of 99.998%, demonstrating its superior performance and efficiency. To further prove the efficiency of our proposed method, the complexity analysis is given in the following:

Here, $n^4 > \log(n)$. So, the Big O notation of our method is Big $O(n^4)$. It is shown the time complexity

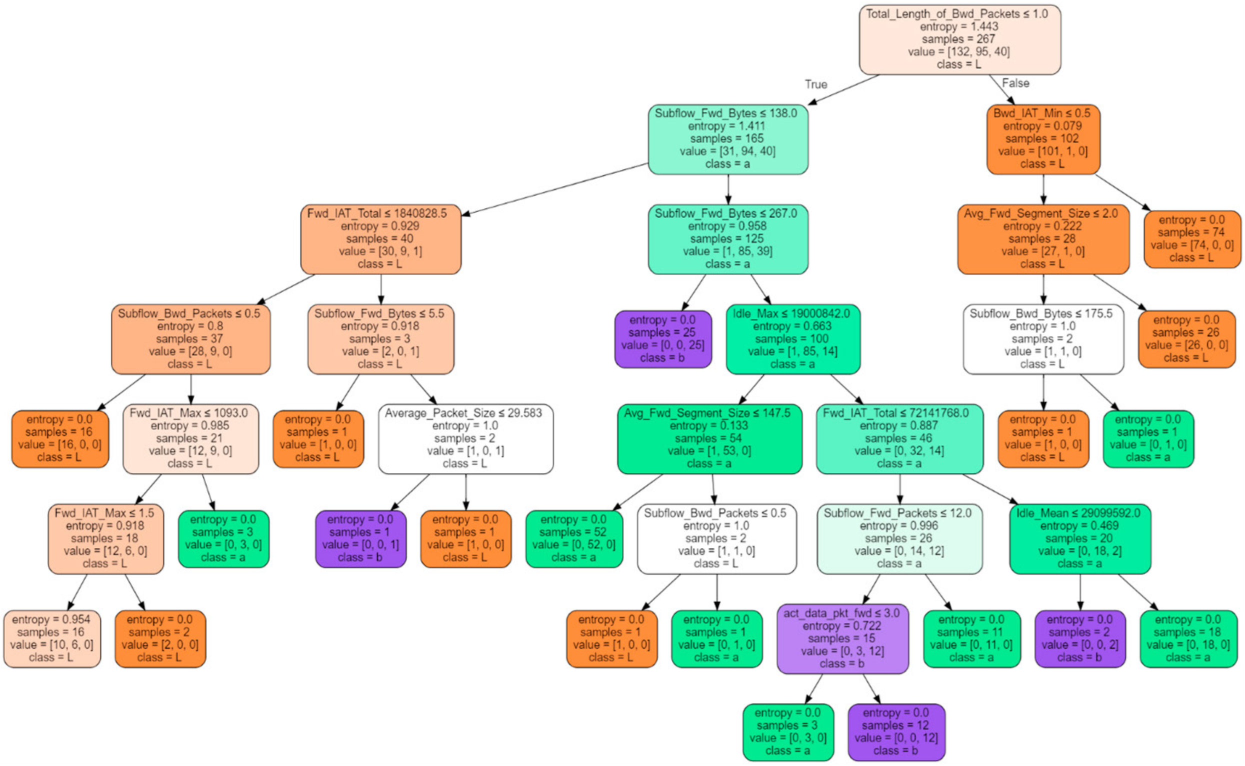


Fig. 5: Features of Decision Tree

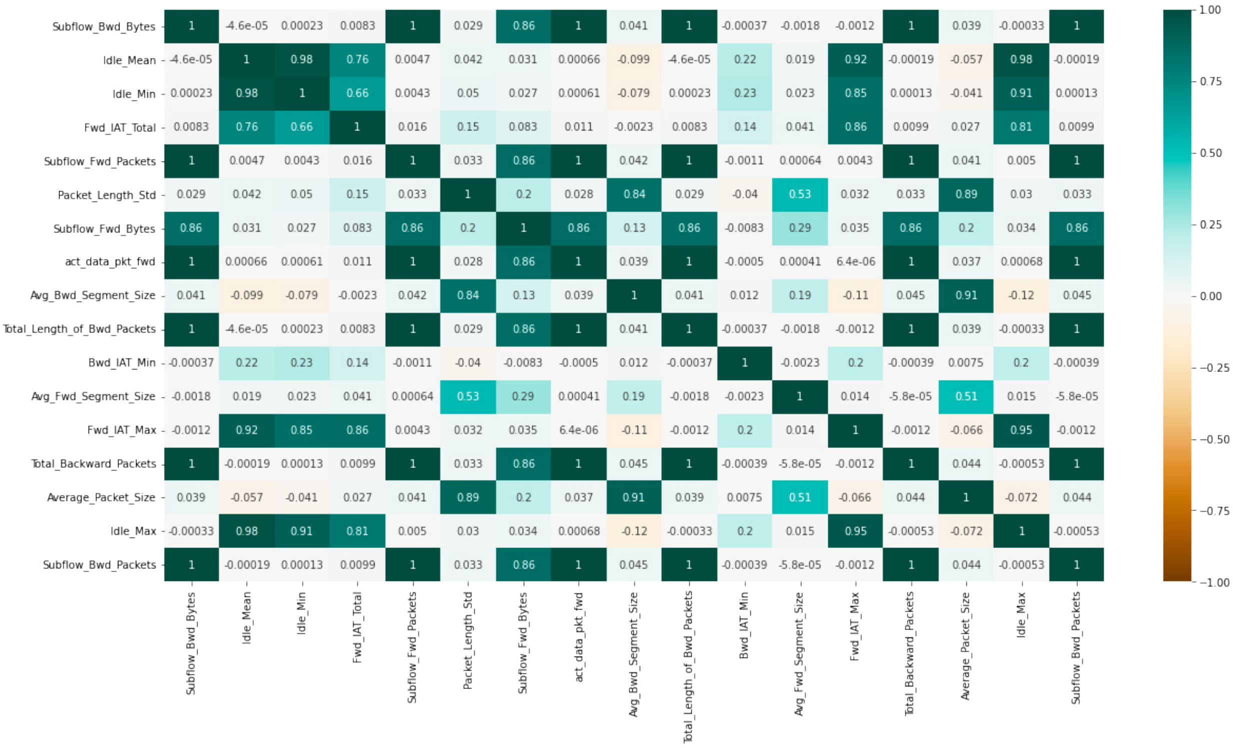


Fig. 6: Feature Correlation Heatmap.

Algorithm 1 Time Complexity Analysis Graph Pseudocode

```
$n$ = sample_number
$i$ = 0
**while** $i < n$ **do**
    Activation_function (values)
    $i2$ = 0
    **while** $i2 \leq n$ **do**
        Activation_function (neurons)
        **for** $j$ = 1 to 4 **do**
            Forward_propagation (values)
        **end for**
    **end while**
    **for** $k$ = 1 to $x$ **do**
        Generate_RF(samples) {Generating RF for $x$ samples}
    **end for**
    **if** $O(x_1) > O(x_2)$ **then**
        Big_O_notation = $O(x_1)$
    **else**
        Big_O_notation = $O(x_2)$
    **end if**
    **if** $n^4 > \log(\ln)$ **then**
        Final_Big_O_notation = $O(n^4)$
    **else**
        Final_Big_O_notation = $O(\log(\ln))$
    **end if**
**end while**
```

Table 2: Detection Accuracy

	Algorithms(ML)	Accuracy (%)	Number of DT	Activation function
1	RF	99.998	100	Entropy
2	DT	99.997	1	Entropy
3	Naïve Bayes	94.819	NULL	NULL
4	ANN	71.298	NULL	logistic

analysis of ANN-RF.

In Table- 1 , we have compared our paper with other journals that have been published in recent years and have made a revolution in the detection of DDoS attacks. The results of the accuracy after making changes are in Table 2. We tested four algorithms from the table. While ANN struggles with DoS Slowloris, Naïve Bayes achieved 94.819% accuracy. For our method, we used 100 decision trees with a random state of 2, logistic activation, and iterated 1000 times with 5 hidden layers (30 neurons each) and a learning rate of 0.01. The RF and DT algorithms showed strong individual performance, but combining them delivered even better results. We replaced Gini impurity with entropy, using Gini only within the internal DT, which boosted accuracy. Our 'Smart Approach Algorithm' achieved the highest accuracy of 99.998% (Table 3).

In Table 2, [4] reports a learning and prediction time of 470 minutes, with predictions taking over 15 minutes due to the complexity of the SVM algorithm. Despite the high accuracy of [2], it suffers from prolonged learning times, mainly due to the large dataset size.

On the other hand, [6] uses a customized J48 algorithm with an accuracy of 98.64%, lower than ours, and a slightly longer learning time of 8 minutes. In contrast, our proposed ANN-RF algorithm significantly reduces learning and prediction time to just 5 minutes, while achieving an accuracy of 99.998%. With over 77 features, our method not only improves detection rates in real time but also ensures efficiency and user-friendliness, which is why we refer to it as

Table 3: Comparison of the Best Existing Algorithms and the Proposed Approach

Reference Number	Algorithms	Features	Accuracy (%)	Learning and Predicting Time (s)
[4]	SVM	23	99.68	15 min+
[2]	RF	20	99.93	10 min+
[6]	J48	27	98.64	15 min
[5]	CapSA, FFA etc.	Not Mentioned	99.997	Not mentioned
Proposed Smart Approach	ANN-RF	77	99.998	5 min

'smart detection.' Improving the performance of the ANN-RF model can be achieved through several strategies. First, **hyperparameter tuning** is crucial, as adjusting parameters like the learning rate, number of neurons in the ANN, or the number of trees in the Random Forest can optimize the model's performance. Additionally, **feature engineering** plays a significant role by ensuring that the input data is of the highest quality; this might involve selecting the most relevant features or generating new, more informative ones from existing data. Another critical aspect is **data augmentation**, where synthetic data is generated to increase the dataset size, helping the model generalize better to unseen data. To ensure that the model's performance is consistent and not overly dependent on the training data, employing **cross-validation** techniques such as k-fold cross-validation can be highly effective in reducing overfitting.

Moreover, **reducing the complexity of the model** by limiting unnecessary layers or features can improve both the training time and the model's performance without sacrificing accuracy. Lastly, using hardware accelerators like **GPUs or TPUs** can significantly speed up both the training and inference processes, ensuring that the model runs efficiently, especially when dealing with large datasets. The proposed model demonstrates robustness against large-scale volumetric DDoS attacks, including the **MERIS** botnet, by effectively handling high traffic volumes and distinguishing legitimate from malicious traffic. The hybrid model's ability to manage complex traffic patterns makes it suitable for defending against such advanced attacks. We evaluated four different scenarios, distinguishing DDoS attacks like TCP, UDP, and HTTP floods from normal traffic [11]-[15]. Recent studies have focused on encryption techniques, often at the expense of detection time and quality. [5] also struggles with delayed convergence, requiring more complex models. Our method, however, relies on binary classification and a combination of RF and deep learning, providing faster and more accurate detection without compromising efficiency. While comparable in accuracy, our approach stands out by integrating machine learning and deep learning with a focus on speed, making it a faster and more practical solution.

5. Conclusion

DDoS attacks are a major danger to network security, affecting the accessibility of online services and resources. During the DDoS attack, real users experience delays as the system tries to deal with fake requests sent by bots. The proposed ANN-RF model, which combines the strengths of ANN and RF approaches, has provided a significant result of 99.998% in detecting DDoS attacks, which is much better than the previous best accuracy of 99.930%. It has also been proven to be much more efficient than other existing methods. According to our findings, it is possible to detect in real-time while maintaining network connectivity simultaneously. Confidential national data will be more secure by employing this approach of detecting DDoS attacks, thereby preventing the risk of a World Cyber War in the future. Furthermore, social media platforms have been used to steal our personal information. DDoS attacks may have a major effect on us if detected. Implement AI-based systems that can automatically detect and respond to attack patterns in real-time. Machine learning models can predict and mitigate potential threats before they fully materialize.

References

[1] F. S. De Lima Filho, F. A. F. Silveira, A. De Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, "Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning," Security and Communication Networks, vol. 2019, 2019, doi: 10.1155/2019/1574749.

[2] A. Delplace, S. Hermoso, H. N. Au, and K. Anandita, "Cyber Attack Detection thanks to Machine Learning Algorithms," Jan. 2020, doi: 10.48550/arxiv.2001.06309.

[3] S. Sambangi and L. Gondi, “A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression,” Proceedings 2020, Vol. 63, Page 51, vol. 63, no. 1, p. 51, Dec. 2020, doi: 10.3390/PROCEEDINGS2020063051.

[4] “DDOS Attack Detection Using Machine Learning.” https://www.jetir.org/view?paper=JETIR2006031 (accessed Jun. 21, 2022).

[5] M. Abd Elaziz, M. A. A. Al-qaness, A. Dahou, R. A. Ibrahim, and A. A. A. El-Latif, “Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm,” Advances in Engineering Software, vol. 176, Feb. 2023, doi: 10.1016/j.advengsoft.2022.103402.

[6] P. S. Saini, S. Behal, and S. Bhatia, “Detection of DDoS attacks using machine learning algorithms,” Proceedings of the 7th International Conference on Computing for Sustainable Global Development, INDIACom 2020, pp. 16–21, Mar. 2020, doi: 10.23919/INDIACOM49435.2020.9083716.

[7] S. Rajesh, M. Clement, S. S. B., A. S. S. H., and J. Johnson, “Real-Time DDoS Attack Detection Based on Machine Learning Algorithms,” SSRN Electronic Journal, Sep. 2021, doi: 10.2139/SSRN.3974241.

[8] M. Xu et al., “Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method,” Symmetry 2022, Vol. 14, Page 1095, vol. 14, no. 6, p. 1095, May 2022, doi: 10.3390/SYM14061095.

[9] Naveen Bindra and Manu Sood, “Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset,” Automatic Control and Computer Sciences, vol. 53, no. 5, pp. 419–428, Sep. 2019, doi: 10.3103/S0146411619050043.

[10] S. Vattikuti, M. R. Hegde, M. Manish, V. Bodduvaram, and V. Sarasvathi, “DDoS Attack Detection and Mitigation using Anomaly Detection and Machine Learning Models,” CSITSS 2021 - 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions, Proceedings, 2021, doi: 10.1109/CSITSS54238.2021.9683214.

[11] B. B. Rao and K. Swathi, “Fast kNN Classifiers for Network Intrusion Detection System,” Indian J Sci Technol, vol. 10, no. 14, pp. 1–10, Apr. 2017, doi: 10.17485/IJST/2017/V10I14/93690.

[12] R. Rajakumar and S. S. Devi, “Design of Intrusion Detection System using Machine Learning Techniques,” vol. 23, no. 3, pp. 18–30, 2021, doi: 10.9790/0661-2303021830.

[13] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, “IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization,” IEEE Access, vol. 9, pp. 123448–123464, 2021, doi: 10.1109/ACCESS.2021.3109081.

[14] A. Fatani, A. Dahou, M. A. A. Al-Qaness, S. Lu, and M. A. Elaziz, “Advanced feature extraction and selection approach using deep learning and aquila optimizer for iot intrusion detection system,” Sensors, vol. 22, no. 1, Jan. 2022, doi: 10.3390/s22010140.

[15] A. Dahou et al., “Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm,” Comput Intell Neurosci, vol. 2022, pp. 1–15, Jun. 2022, doi: 10.1155/2022/6473507.