Detection of DDOS Attacks using Ensemble and Probabilistic Classifiers

Pushpavalli K.[1], Meena R.[2], Ranganayaki E.[3]

[1]Assistant Professor, [2, 3]UG Scholar Department of Information Technology, Jerusalem College of Engineering, Chennai, India

E-mail: [1]pushpavalli.it@jerusalemengg.ac.in, [2]meenarit2021@jerusalemengg.ac.in, [3]ranganayakieit2021@jerusalemengg.ac.in

Abstract

Distributed Denial of Service (DDoS) attacks disrupt online services, leading to operational and financial losses. This project presents a machine learning-based detection system utilizing Logistic Regression, Decision Tree, Random Forest, and Gaussian Naive Bayes to analyze network traffic and identify anomalies. By employing data preprocessing and extracting critical features, the system enhances accuracy and efficiency in distinguishing between normal and malicious traffic. Each algorithm is evaluated based on performance, scalability, and computational efficiency. Future enhancements will focus on real-time monitoring and the exploration of more advanced models to improve detection capabilities.

Keywords: DDoS Attacks, Cybersecurity, Machine Learning, Traffic Analysis, Detection Accuracy, Performance Analysis, Real-time Detection.

1. Introduction

Distributed Denial of Service (DDoS) attacks remain a major cybersecurity threat, capable of overwhelming network resources and causing extended service disruptions. These attacks often result in downtime, financial loss, and compromised system integrity. This study specifically addresses key security concerns arising from evolving DDoS attack strategies,

DOI: https://doi.org/10.36548/jitdw.2025.2.002
Received: 24-03-2025. received in revised form: 27-04-2025, accepted: 15-05-2025, published: 27-05-2025

including volumetric floods, protocol-based exploits and application-layer attacks. A critical aspect of the security mechanism is traffic pattern analysis, which enables the identification of unusual spikes or irregular behavior in incoming requests, serving as a foundation for anomaly detection. Such attacks aim to exploit vulnerabilities in traffic-handling mechanisms to cripple targeted systems. The proposed system focuses on early detection and timely classification of malicious traffic, minimizing false positives and ensuring service continuity.

By integrating machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, and Gaussian Naive Bayes, the system effectively analyzes network traffic, identifies anomalies, and distinguishes between benign and malicious activity. The architecture incorporates data preprocessing, relevant feature extraction, and supervised learning models to enhance detection performance. Future enhancements may involve real-time threat monitoring and adaptive defense mechanisms to support scalability and robust protection.

2. Related Work

Machine learning methods have improved considerably the discovery and prevention of Distributed Denial-of-Service (DDoS) attacks based on their capacity to learn data patterns from massive traffic data and evolve in response to changing attack tactics. Initial methods were directed towards using simple classification models like decision trees, support vector machines, and k-nearest neighbors to detect anomalies within network traffic that point to DDoS activity [2][3]. These models demonstrated promise in enhancing detection rates over static rule-based systems. With time, the use of ensemble learning techniques improved performance even more. Scholars such as Kumar and Selvakumar suggested employing ensembles of neural classifiers to enhance accuracy for discriminating between normal and malicious traffic [5], and Katkar and Kulkarni experimented to prove the effectiveness of ensemble classifiers [6]. Ensemble models with adaptive behavior, for instance through neuro-fuzzy logic and hybrid models, were subsequently proposed to enhance real-time detection and system fault-tolerance [8]. Later studies focus on real-time performance and accuracy; for instance, Satyanarayana and Alasmi created a model that combines mitigation with detection capabilities to respond proactively [1], while Santhosh et al. tested different machine learning models to compare their performance against contemporary DDoS patterns [4]. Holistic detection techniques integrating several unsupervised models have also come forward as a strong approach to detect new and sneaky DDoS attacks [9]. Besides, researchers such as Hossain have also pointed to the use of new feature selection and reliable ensemble frameworks to improve detection even in network-rich scenarios [10]. In general, the evolution of research illustrates a transition from straightforward static classifiers to dynamic, hybrid, and ensembling models that provide more accuracy, flexibility, and scalability in identifying and countering DDoS attacks.

3. Proposed Work

The implementation of the proposed DDoS detection system was carried out using Python 3.10, utilizing Scikit-learn for building and evaluating machine learning models. The preprocessing phase involved handling missing values by applying mean imputation and removing duplicates to ensure data quality. Label encoding was used to convert categorical features into numeric format, and normalization was applied using the MinMaxScaler to bring all feature values into a common scale. Feature extraction was guided by domain-specific knowledge, focusing on key attributes such as packet size, connection duration, and protocol type, which are crucial indicators for identifying abnormal traffic behavior.

For training the models, the dataset was split into an 80:20 ratio for training and testing, respectively. All models were trained using their default parameters initially, and hyperparameter tuning was later applied specifically to improve performance. For instance, in the case of Random Forest, the number of estimators was set to 100 and max_depth was limited to 10 to reduce overfitting. The Decision Tree used the 'entropy' criterion for better information gain. K-fold cross-validation was employed to validate the consistency and robustness of the models across different data splits. Evaluation metrics such as accuracy, precision, recall, and F1-score were computed for each classifier to assess their detection capabilities. The entire pipeline was integrated into a user-friendly GUI built using Tkinter, with Matplotlib and Seaborn for plotting metrics and visualizing model performance.

The detection of DDoS Attack system follows a structured approach that integrates machine learning techniques to effectively identify and classify network traffic. The proposed system consists of multiple stages, beginning with data collection and preprocessing, followed by feature extraction, model training, evaluation, and real-time detection. Additionally, a

Graphical User Interface (GUI) is incorporated to facilitate user interaction, enabling seamless model selection, evaluation, and performance analysis.

The architecture of the DDoS Attack Detection System is thoughtfully structured into a sequential pipeline that ensures smooth and efficient detection and classification of network threats. The pipeline begins with data collection, which involves gathering raw network traffic from reliable sources. This traffic includes both benign (normal) and malicious (DDoS) entries, which provide a solid foundation for building an intelligent detection model.

The collected data is often unstructured and noisy, so it undergoes data pre-processing to clean and prepare it for analysis. During pre-processing, missing values are handled, duplicate records are removed, and feature normalization is performed to ensure uniformity across data samples.

Additional steps like label encoding or one-hot encoding may also be applied to convert categorical attributes into machine-readable formats, improving the quality and consistency of the input data. Furthermore, outlier detection techniques may be used to identify and eliminate abnormal values that could distort the training process or lead to inaccurate predictions.

Once the data is pre-processed, the system moves to the feature extraction phase, where meaningful attributes like source and destination IP, protocol type, duration, packet size, and connection state are derived from the raw data. These features are selected based on their ability to highlight anomalies and patterns typical of DDoS attacks. Advanced techniques such as correlation analysis and feature importance ranking may also be used to filter out redundant or less informative attributes, further refining the input set for training.

In the model training and evaluation stage, four prominent machine learning classifiers are employed Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and Gaussian Naive Bayes. Each algorithm learns to identify the subtle differences between normal and attack traffic by training on the labeled dataset. The training process is iterative, ensuring that the models generalize well to unseen data and do not overfit to specific traffic patterns. This is followed by a performance evaluation phase, where the accuracy of each model is computed to assess their detection capabilities.

To evaluate the effectiveness of the detection system, several performance-based parameters are considered, including accuracy, precision, recall, and F1-score. These metrics are critical for assessing how well the model distinguishes between legitimate and malicious traffic.

The Performance Analyzer, however, enables comparative evaluation through the use of bar charts and graphical visualizations of model accuracy and detection rates. This allows users to determine the performance of various machine learning models and make decisions on deploying them. Through the integration of machine learning algorithms, feature extraction methods, and an easy-to-use GUI, the system is guaranteed to be effective at detecting dynamic cyber threats while keeping administrators happy with ease of use.

The Graphical User Interface (GUI) for the DDoS detection system was developed using Python’s Tkinter library, which provides a lightweight and flexible environment for creating desktop-based applications. The interface is designed to be intuitive and user-friendly, allowing users to navigate various functionalities such as model training, dataset uploading, and performance evaluation with ease. For graphical representations and visual analysis, the system integrates Matplotlib and Seaborn libraries, which are used to generate comparative bar charts, accuracy graphs, and evaluation summaries in real-time.

The GUI also includes dynamic features such as dropdown menus to select models, file upload dialogs for dataset integration, structured display of evaluation metrics (accuracy, precision, recall, F1-score), and clearly labeled buttons to trigger training, evaluation, and result plotting processes. The interface is modular and well-structured, allowing for future expansion and the inclusion of additional models or functionality. It is fully compatible with Windows 10 (64-bit) operating systems and has been optimized to ensure responsive performance, making it suitable for deployment in machine learning-based network security systems.

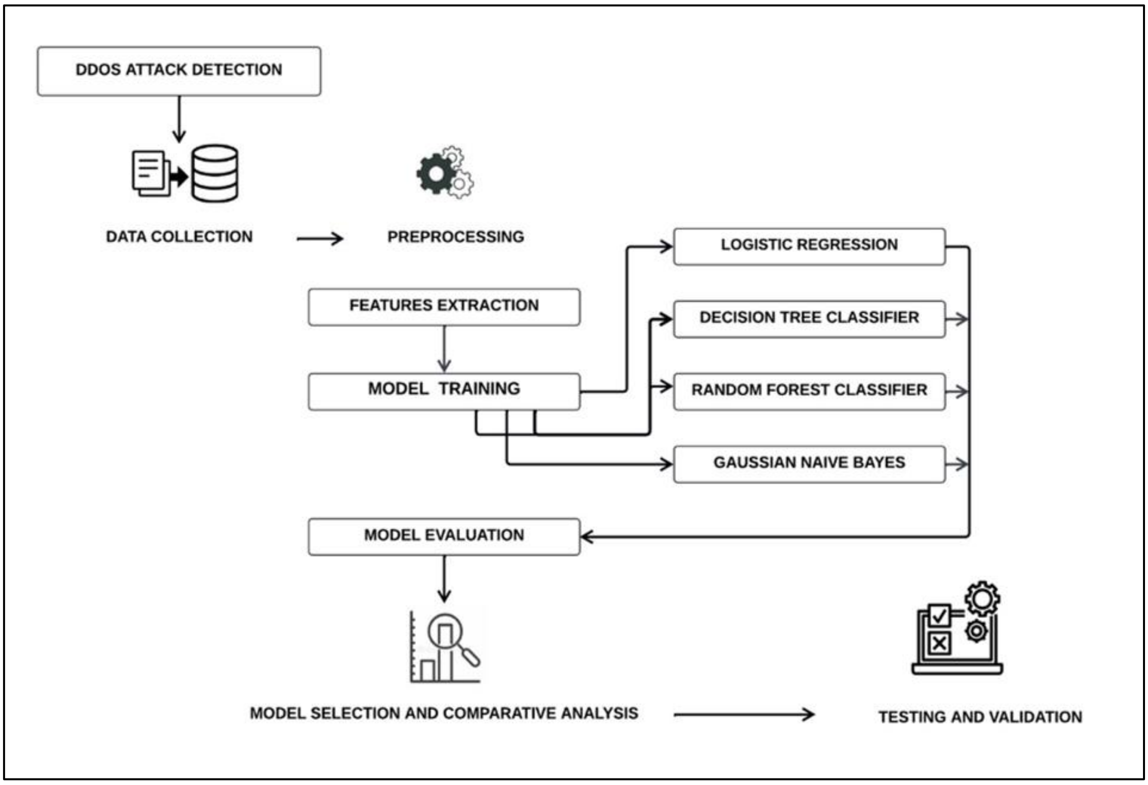


Figure 1. Architecture Diagram

4. Results and Discussion

The developed DDoS attack detection system successfully demonstrates the integration of multiple machine learning models within an intuitive GUI framework to support real-time network threat detection. The system was tested using a labeled dataset containing both benign and malicious entries, with the aim of classifying traffic accurately. To enhance reliability, the system implements a modular architecture where each model can be evaluated independently, ensuring flexibility in updates or model replacement without impacting the entire framework.

The GUI design provides users with easy navigation through various components like dataset uploading, model evaluation, metric display, and comparative analysis. The interface is divided into two components: Model Trainer and Evaluator, Performance Analyser.

The Evaluator module facilitated the training and testing of multiple models Logistic Regression, Decision Tree, Random Forest, and Gaussian Naive Bayes while offering an automated mechanism to display their outcomes side by side. Each algorithm was executed systematically, ensuring the consistency of results and enabling users to assess the behaviour of individual models when exposed to DDoS-specific traffic features.

The performance metrics such as accuracy, precision, recall, and F1-score were calculated for each classifier, offering a multi-dimensional view of model efficiency. Among the models evaluated, the Random Forest Classifier outperformed the others in terms of accuracy and stability, followed by Decision Tree, which also showed high precision but slightly lower recall. Logistic Regression and Gaussian Naive Bayes delivered moderate results, suitable for simpler or less computationally intensive tasks. The "Best Model" module automatically selected the top-performing algorithm based on predefined criteria, reducing user effort in decision-making. Furthermore, the detailed results such as confusion matrices and prediction outcomes allowed for a deeper interpretation of model sensitivity and robustness toward different traffic patterns. These insights are critical for enhancing the reliability of real-world DDoS detection systems, especially when deployed in dynamic and large-scale environments.

The Performance Analyser module played a significant role in translating raw performance metrics into visual insights. Through bar charts and comparative graphs, users were able to observe each model's accuracy and identify trends or anomalies in performance. The system also emphasized the importance of comparative analysis, ensuring the final model selection was evidence-based and backed by quantitative data.

Overall, the developed DDoS detection system successfully integrates machine learning models within an interactive GUI environment, enabling real-time detection of malicious network traffic. Through structured training and evaluation processes, the system effectively compares the performance of Logistic Regression, Decision Tree, Random Forest, and Gaussian Naive Bayes classifiers. The Random Forest model consistently demonstrated superior performance in terms of accuracy and robustness, making it the optimal choice for deployment. Performance metrics such as accuracy, precision, recall, and F1-score, along with comparative bar chart visualizations, provided comprehensive insights into the detection capabilities of each model.

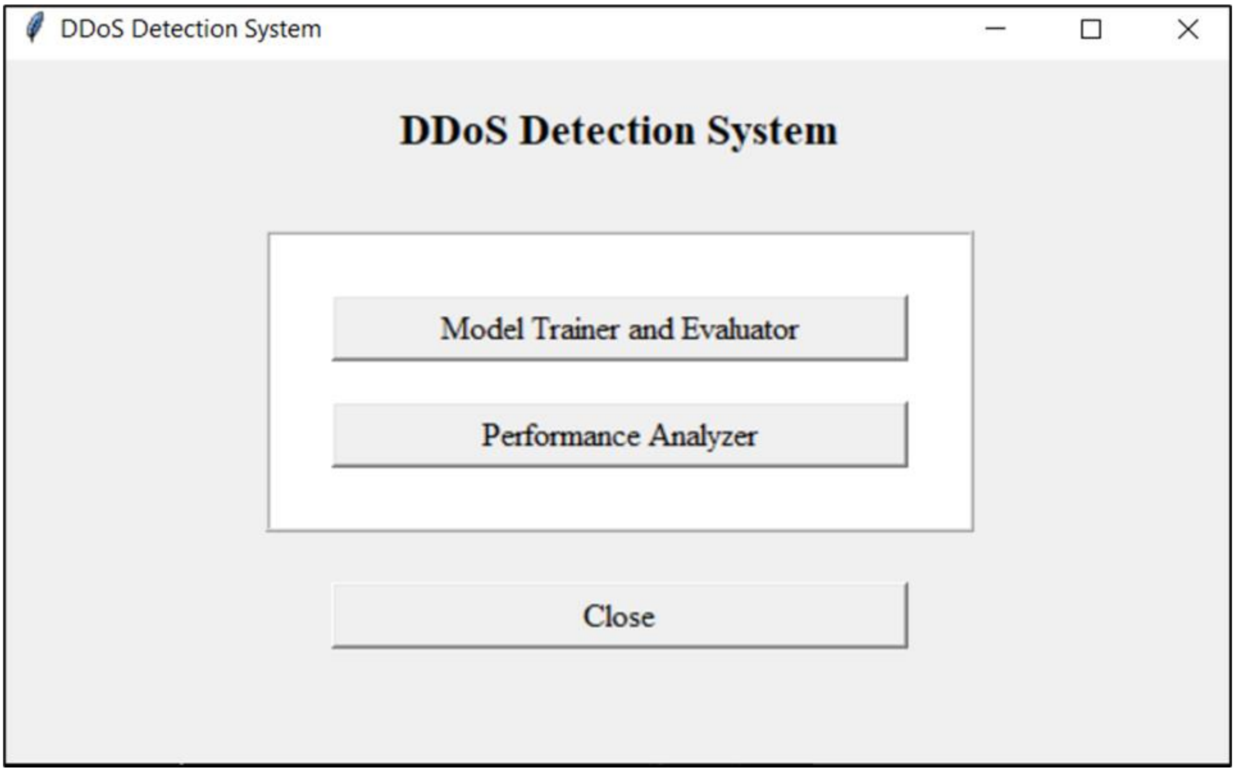


Figure 2. DDOS Detection GUI

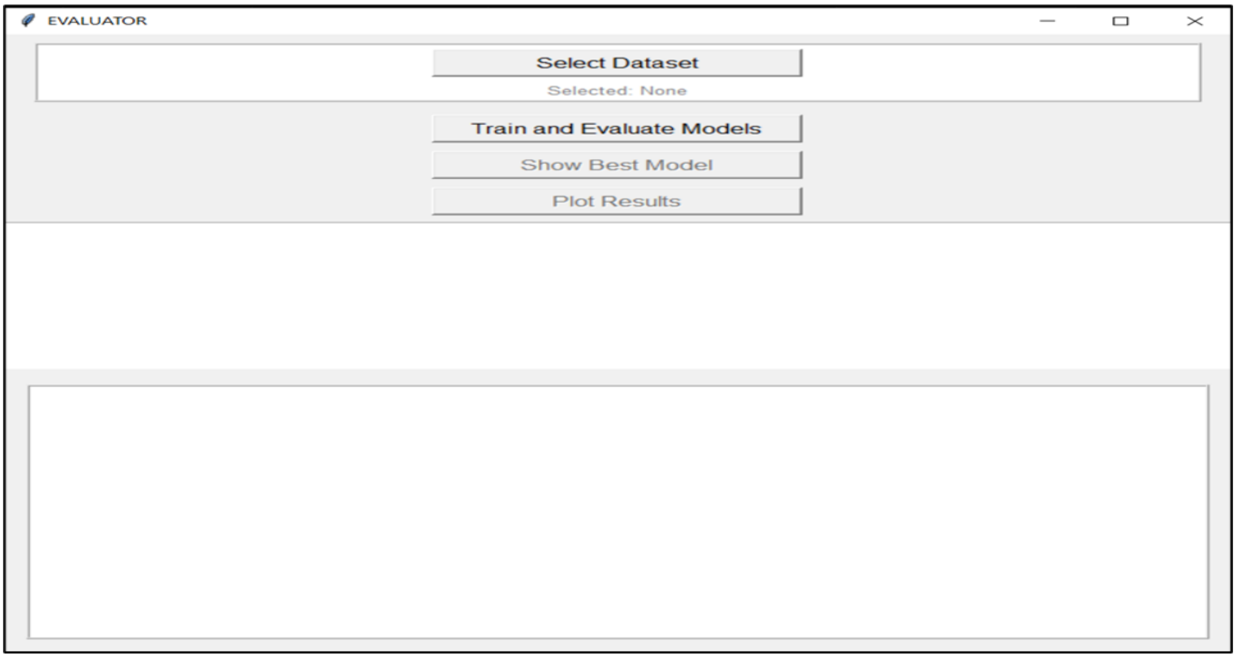
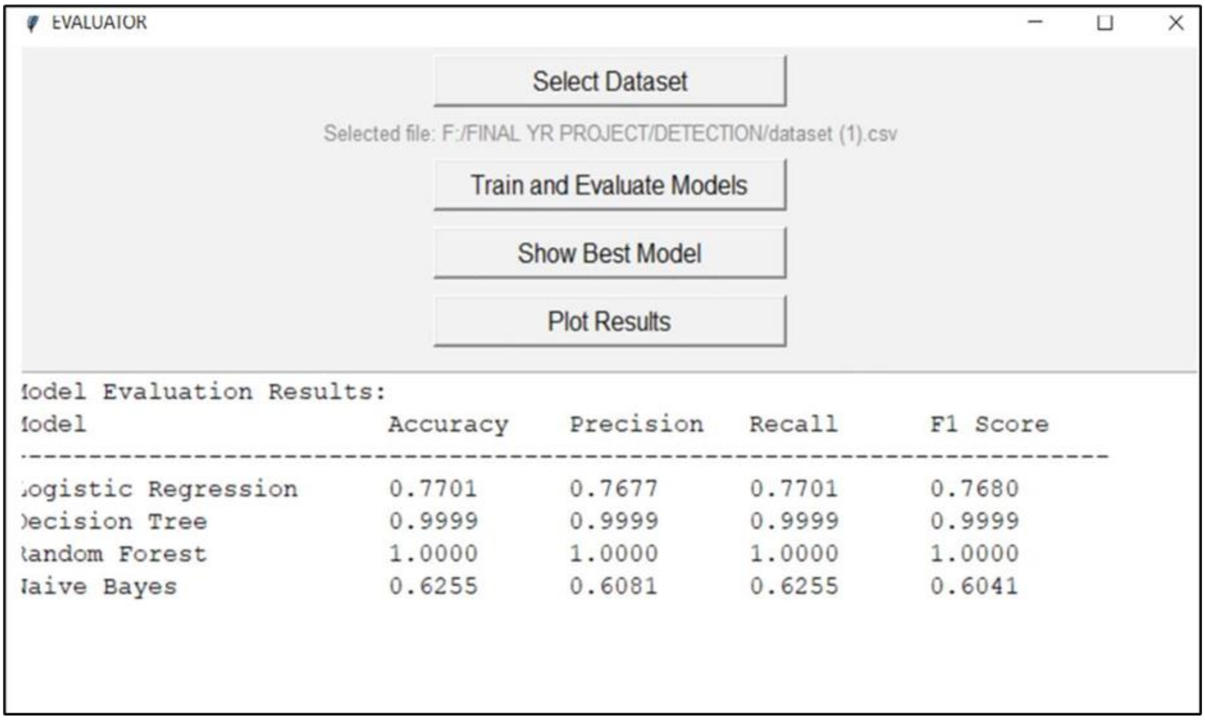
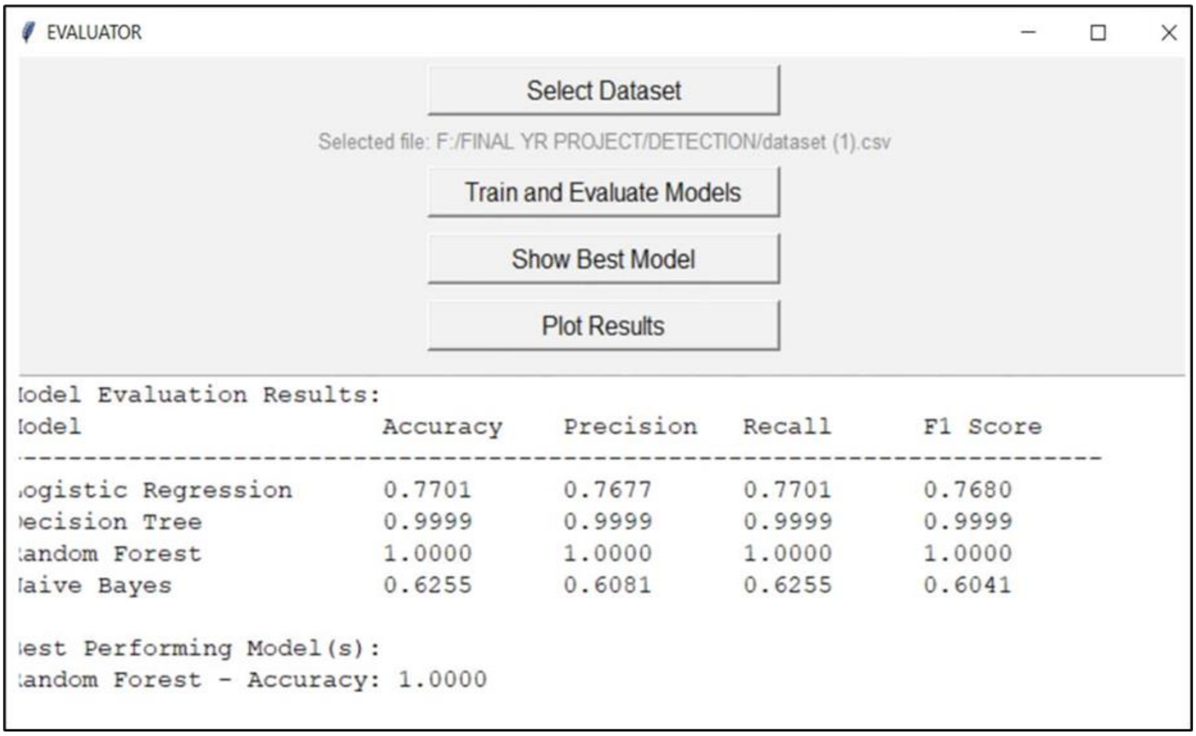


Figure 3. Model Trainer and Evaluator


EVALUATOR
Select Dataset
Selected file: F:/FINAL YR PROJECT/DETECTION/dataset (1).csv
Train and Evaluate Models
Show Best Model
Plot Results

```
Iodel Evaluation Results:
Iodel                    Accuracy    Precision   Recall      F1 Score
------------------------------------------------------------------------
.ogistic Regression      0.7701      0.7677      0.7701      0.7680
)ecision Tree            0.9999      0.9999      0.9999      0.9999
tandom Forest            1.0000      1.0000      1.0000      1.0000
Iaive Bayes              0.6255      0.6081      0.6255      0.6041
```

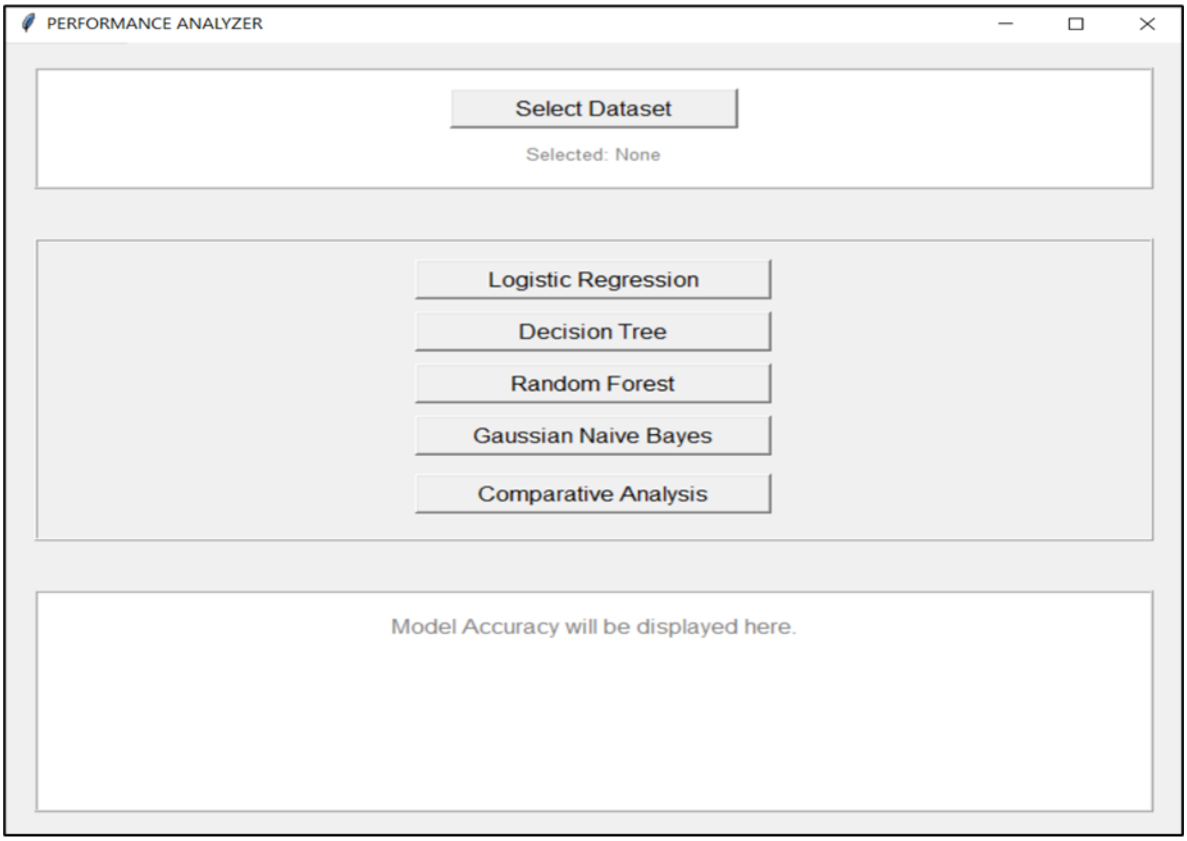
Figure 4. Evaluator: Evaluate Models


EVALUATOR
Select Dataset
Selected file: F:/FINAL YR PROJECT/DETECTION/dataset (1).csv
Train and Evaluate Models
Show Best Model
Plot Results

```
Iodel Evaluation Results:
Iodel                    Accuracy    Precision   Recall      F1 Score
------------------------------------------------------------------------
.ogistic Regression      0.7701      0.7677      0.7701      0.7680
)ecision Tree            0.9999      0.9999      0.9999      0.9999
tandom Forest            1.0000      1.0000      1.0000      1.0000
Iaive Bayes              0.6255      0.6081      0.6255      0.6041

iest Performing Model(s):
tandom Forest - Accuracy: 1.0000
```

Figure 5. Evaluator: Best Model



Selected: None

Model Accuracy will be displayed here.

Figure 6. Performance Anaylzer-The smoother Trend Directly Translates to Practical

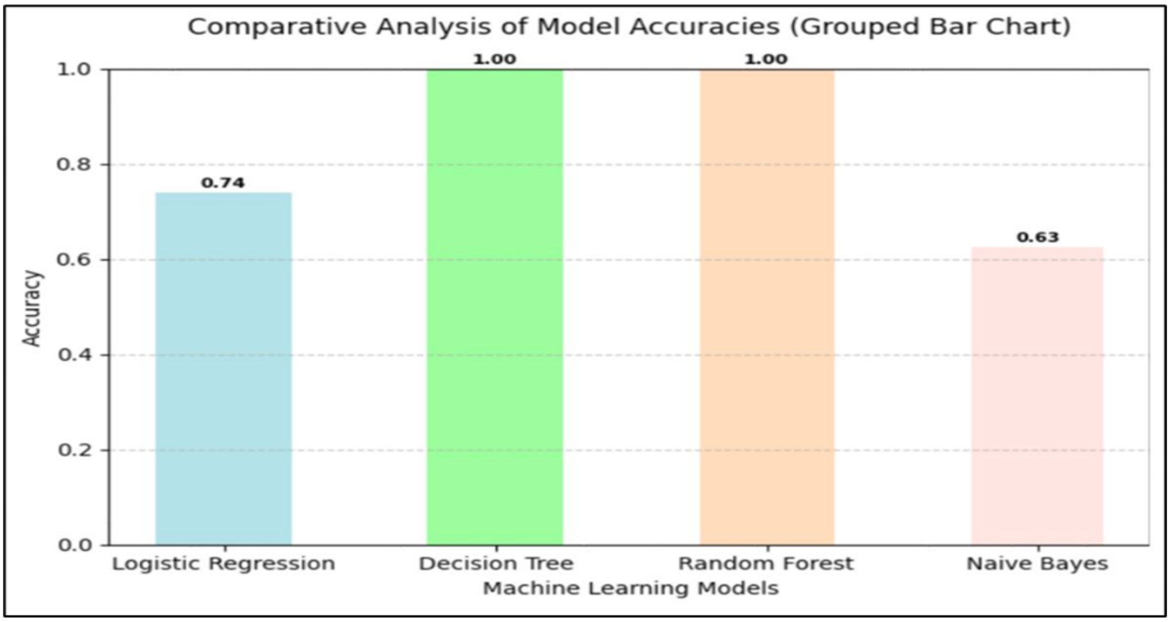


Figure 7. Performance Anaylzer: Comparative Anaylsis

As depicted in Figure 2, the DDoS Detection GUI is a single window with which to launch the system functionalities. Figure 3 shows information on the Model Trainer and Evaluator module, where dataset choosing and model evaluation can be performed. Figure 4 displays the results of evaluation for different machine learning models, and Figure 5 depicts choosing the best-performing model. The Performance Analyzer interface, shown in Figure 6, provides the option to assess one or more individual models or perform comparative analysis. Lastly, Figure 7 is a grouped bar chart comparing model accuracies and clearly shows the better performance of Decision Tree and Random Forest classifiers.

5. Conclusion

The developed DDoS detection system has demonstrated considerable success in identifying DDoS attacks through the analysis of network traffic utilizing various machine learning techniques. The integration of models such as Logistic Regression, Decision Tree, Random Forest, and Naive Bayes has facilitated effective classification of network traffic, allowing for the differentiation between benign and malicious requests. Key performance metrics, including accuracy, precision, recall, and F1-score, were employed to evaluate the models, and the results confirmed the efficacy of machine learning techniques in detecting malicious network traffic. The implementation of a Graphical User Interface (GUI) has significantly enhanced the system’s usability, making it accessible to non-expert users. The GUI enables seamless interaction for tasks such as dataset loading, model selection, training, evaluation, and result visualization, providing clear insights into the performance of various models. The overall system adopts a structured approach to DDoS attack detection, ensuring both accuracy and reliability. The robustness of the system is a critical consideration in addressing contemporary threats, such as the MERIS botnet, which is characterized by high-volume and distributed traffic spikes. While the current model does not explicitly simulate MERIS-scale attacks, the employment of ensemble classifiers like Random Forest enhances adaptability. Future iterations of the model may incorporate datasets or simulated traffic that replicate MERIS behavior for the purpose of stress-testing detection capabilities.

References

[1] Satyanarayana, D., and Aisha Said Alasmi. "Detection and mitigation of DDOS based attacks using machine learning algorithm." In 2022 International Conference on Cyber Resilience (ICCR), pp. IEEE, (2022) 1-5.

[2] Ajeetha, G., and G. Madhu Priya. "Machine learning based DDOS attack detection." In 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), vol. 1, pp. IEEE, (2019) 1-5.

[3] Saini, Parvinder Singh, Sunny Behal, and Sajal Bhatia. "Detection of DDoS attacks using machine learning algorithms." In 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), pp. IEEE, (2020) 16-21.

[4] Santhosh, S., M. Sambath, and J. Thangakumar. "Detection of DDOS attack using machine learning models." In 2023 International Conference on Networking and Communications (ICNWC), pp. IEEE, (2023) 1-6.

[5] Kumar, P. Arun Raj, and S. Selvakumar. "Distributed denial of service attack detection using an ensemble of neural classifier." Computer Communications 34, no. 11 (2011): 1328-1341.

[6] Katkar, Vijay D., and Siddhant Vijay Kulkarni. "Experiments on detection of Denial of Service attacks using ensemble of classifiers." In 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. IEEE, (2013) 837-842.

[7] Alguliyev, R. M., R. M. Aliguliyev, Y. N. Imamverdiyev, and L. V. Sukhostat. "An improved ensemble approach for DoS attacks detection." Радіоелектроніка, інформатика, управління 2 (45) (2018): 73-82.

[8] Kumar, P. Arun Raj, and S. Selvakumar. "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems." Computer Communications 36, no. 3 (2013): 303-319.

[9] Das, Saikat, Deepak Venugopal, and Sajjan Shiva. "A holistic approach for detecting DDoS attacks by using ensemble unsupervised machine learning." In Advances in Information

and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2, pp. 721-738. Springer International Publishing, 2020.

[10] Hossain, Md Alamgir. "Enhanced ensemble-based distributed denial-of-service (DDoS) attack detection with novel feature selection: a robust cybersecurity approach." Artificial Intelligence Evolution (2023): 165-186