International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Enhanced DDoS Attack Detection using Machine Learning

[1]Setty Vari Sai Mokshagna, [2]N Harish, [3]P Sai Murali Krishna, [4]S Harsha Vardhan Reddy, [5]Kiran Kumari Patil
[1-4]Department of Computer Science, CMR University, Bangalore, India
[5]Professor and Deputy Director, Department of Computer Science, CMR University, Bangalore, India,
[1]settyvari.saimokshagna@cmr.edu.in, [2]nagisetty.harish@cmr.edu.in,z [5]kirankumari.p@cmr.edu.in

Abstract*—Distributed Denial of Service (DDoS) attacks have become one of the most pressing cybersecurity concerns due to their ability to disrupt services, damage reputations, and impact entire infrastructures. This study aims to develop an effective and accurate detection system for such attacks using machine learning techniques. By leveraging the CICIDS2017 dataset—which offers realistic, labeled network traffic data including both benign and malicious samples—the project analyzes multiple classification algorithms to identify the most reliable models for detecting DDoS threats. The dataset was preprocessed and divided into training and testing subsets to ensure thorough model evaluation. Eight different algorithms were implemented and tested: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Decision Tree, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN). Performance was assessed using key metrics such as accuracy, precision, recall, F1 score, and confusion matrix to determine how well each model could differentiate between benign and DDoS traffic. Among all the models, tree-based classifiers like Random Forest and Decision Tree stood out, each achieving an accuracy of 99.98%, indicating their strength in handling complex network traffic patterns. The results also highlight the potential of ensemble learning methods in cybersecurity applications, where both detection accuracy and low false positive rates are crucial. While the study demonstrates that several models are capable of high-performance detection, it also emphasizes the need for further refinement through hyperparameter tuning and dimensionality reduction to enhance real-time deployment. As DDoS attacks continue to evolve in nature, employing dynamic and adaptive detection systems based on machine learning becomes not just beneficial but necessary for proactive cybersecurity. In addition to demonstrating high accuracy, the study emphasizes the importance of understanding feature importance and selection in building more interpretable and lightweight models. Certain features within the dataset, such as flow duration, packet size statistics, and header flags, were found to be more indicative of DDoS activity, highlighting how specific attributes contribute significantly to detection accuracy. This not only aids in reducing computational complexity but also enhances the model's applicability in real-time environments where response speed is critical. By analyzing and comparing the performance of various models, the study identifies not just the most accurate classifiers but also the trade-offs involved in their use—such as training time, scalability, and ease of deployment in practical systems. Furthermore, the inclusion of deep learning methods like ANN and CNN adds an advanced dimension to the research, showcasing how neural networks can learn complex patterns and relationships within network traffic data, though they may require more computational resources.

***Index Terms*—DDoS Detection, Machine Learning, Network Security, CICIDS2017 Dataset, Classification Algorithms, Cyber Threats, Random Forest, Decision Tree, Neural Networks, Intrusion Detection System.**

I. INTRODUCTION

In the current digital landscape, network security is an indispensable aspect of every organization's infrastructure. Among the various forms of cyberattacks, Distributed Denial of Service (DDoS) attacks pose a severe threat due to their ability to overwhelm servers and disrupt the availability of services. Unlike traditional Denial of Service attacks, DDoS attacks are launched simultaneously from multiple compromised systems, making them harder to mitigate and more damaging in terms of impact. These attacks target the availability component of the CIA (Confidentiality, Integrity, and Availability) triad, making it essential to detect and respond to them swiftly and accurately. Due to their polymorphic nature, DDoS attacks can change their behaviour frequently, which means static rule-based systems often fall short. This highlights the need for dynamic and intelligent approaches such as machine learning (ML), which can adapt to changing traffic patterns and learn from historical data.

Machine learning offers promising capabilities for early detection of DDoS attacks by identifying subtle anomalies and patterns in network traffic. The core objective of this research is to evaluate the effectiveness of different machine learning models in detecting DDoS attacks using supervised learning techniques. To facilitate this, the CICIDS2017 dataset was chosen as it contains a wide variety of attack scenarios as well as normal traffic, making it ideal for training robust classification models. This dataset was generated in a realistic environment that mirrors contemporary network settings, ensuring the trained models are applicable to real-world scenarios. A comprehensive analysis of various classification algorithms is carried out, including both traditional models like Decision Tree, Logistic Regression, and Naive Bayes, as well as advanced approaches like Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN).

The study involves data preprocessing, feature selection, model training, and performance evaluation, with a strong focus on metrics like accuracy, precision, recall, F1 score, and confusion matrix analysis. The goal is not only to determine which algorithm performs best but also to understand which features are most relevant in identifying DDoS patterns. By comparing multiple algorithms under the same conditions, this research provides insights into the relative strengths and weaknesses of each approach,

thereby assisting in the selection of optimal models for real-time intrusion detection systems. Furthermore, the paper highlights the importance of ongoing research in this domain, considering the ever-changing nature of cyber threats, and suggests future directions for enhancing detection frameworks through optimization techniques and deep learning innovations.

1.1 Motivation and Significance

With the growing dependency on internet-connected systems, the frequency and scale of cyberattacks, especially Distributed Denial of Service (DDoS) attacks, have increased drastically. These attacks not only disrupt services but also pose serious threats to businesses, governments, and individuals. Traditional security systems often fail to detect such evolving threats in real-time due to their static nature. This motivated the need for intelligent and adaptive approaches like machine learning that can detect anomalies by learning from data patterns. The significance of this study lies in its practical approach—evaluating multiple machine learning algorithms to find the most effective models for DDoS detection using a realistic and comprehensive dataset. By identifying the most accurate and efficient algorithms, this research contributes to the development of smarter intrusion detection systems capable of offering timely protection against dynamic network threats.

1.2 Problem Statement

Distributed Denial of Service (DDoS) attacks have become increasingly complex and harder to detect using traditional methods. These attacks can severely impact the availability of online services by overwhelming systems with malicious traffic. Due to the dynamic and polymorphic nature of DDoS attacks, there is a growing need for intelligent, adaptive solutions. This study aims to explore and evaluate various machine learning algorithms to accurately detect DDoS attacks using the CICIDS2017 dataset, helping to improve the reliability and effectiveness of intrusion detection systems.

1.3 Objectives

The primary objectives of this study include:

- To analyse the impact of DDoS attacks and highlight the need for efficient detection mechanisms.
- To utilize the CICIDS2017 dataset for training and testing machine learning models on realistic network traffic data.
- To implement and compare various classification algorithms such as ANN, CNN, Decision Tree, Logistic Regression, Naive Bayes, Random Forest, SVM, and k-NN.
- To evaluate the performance of each model using accuracy, precision, recall, F1 score, and confusion matrix.
- To identify the most effective machine learning algorithm(s) for detecting DDoS attacks with minimal false positives.
- To explore important features in the dataset that contribute significantly to the detection of malicious traffic.
- To propose a model that can be further optimized for real-time DDoS detection in practical environments.

2. LITERATURE REVIEW

Over the years, various approaches have been proposed to detect and mitigate DDoS attacks, ranging from traditional rule-based systems to more advanced machine learning techniques. Researchers have explored multiple datasets, algorithms, and feature engineering methods to enhance the accuracy and efficiency of intrusion detection systems. The rise of machine learning has opened new possibilities in understanding network traffic patterns and identifying threats in real-time. This section reviews recent studies that focus on machine learning-based DDoS detection, highlighting the strengths and limitations of existing techniques and how they shape the direction of this research.

2.1 Machine Learning in Network Security

The integration of machine learning (ML) in network security has gained momentum due to its ability to identify complex attack patterns. Kumar et al. [1] applied ML models like Decision Trees and Support Vector Machines for intrusion detection using the NSL-KDD dataset. Their study demonstrated the effectiveness of these models in identifying known attacks, although detection rates for novel threats were lower. Similarly, Wang and Zhao [2] explored the use of Random Forests for anomaly detection in network traffic, reporting high accuracy but highlighting computational challenges for real-time applications.

2.2 DDoS Attack Detection Using ML

DDoS detection has been a key area of research in network security. Singh et al. [3] developed a framework using supervised learning techniques on the CICIDS2017 dataset to classify traffic as benign or malicious. Their study emphasized the importance of feature selection in improving model precision. In another study, Ali and Rehman [4] employed k-NN and SVM for detecting DDoS attacks, showing that combining multiple models enhanced detection accuracy but increased the complexity of deployment.

2.3 Evaluation Of Classification Algorithms

Comparative analysis of classification algorithms for intrusion detection has helped identify optimal approaches for various scenarios. Das and Bhowmik [5] compared the performance of Naive Bayes, Logistic Regression, and Random Forest on network intrusion datasets. They concluded that ensemble methods outperformed single classifiers in terms of accuracy and F1 score. On the other hand, Sharma et al. [6] highlighted the limitations of deep learning models like CNN in terms of training time, despite their superior performance in learning traffic behavior patterns.

2.4 Feature Engineering For DDoS Detection

Effective feature selection plays a crucial role in enhancing the performance of DDoS detection systems. Patel et al. [7] used feature ranking techniques on CICIDS2017 and found that flow duration, packet size, and header flags were among the most influential attributes for attack detection. Similarly, Verma and Joshi [8] proposed dimensionality reduction through PCA to simplify data representation, which helped reduce model training time without significantly compromising accuracy.

2.5 Real-Time Detection Systems

Developing DDoS detection models suitable for real-time applications is a major focus of current research. Ahmed et al. [9] presented a lightweight anomaly detection system using decision trees optimized for IoT environments. Their approach prioritized quick response times over model complexity. Another study by Reddy and Nair [10] implemented ANN-based detection on streaming data, facing challenges in balancing model accuracy with processing latency under high network load conditions.

2.6 Challenges in ML-Based DDoS Detection

Despite promising results, several challenges remain in implementing ML-based DDoS detection. Gupta et al. [11] noted that polymorphic attack behavior limits the effectiveness of static models, necessitating continuous model retraining. Additionally, issues such as data imbalance, false positives, and the lack of labeled real-world datasets were frequently cited as limitations. The need for adaptive models and efficient feature selection strategies was also emphasized for better generalization and deployment.

3. METHODOLOGY

3.1 System setup and Data Processing

The system is designed as a machine learning-based application for DDoS attack detection, utilizing Python and relevant libraries for model training and evaluation. The architecture consists of a backend developed using Python's scikit-learn, TensorFlow, and Keras libraries for machine learning model development, with data preprocessing and model evaluation tasks handled within the backend.

The dataset used for training is the CICIDS2017 dataset, which contains a wide range of features representing network traffic, including flow duration, packet sizes, and other flow-based characteristics. The dataset is stored in .csv format and is structured with labeled instances, categorizing each traffic flow as benign or malicious.

To prepare the dataset for model training, several preprocessing steps are performed. First, missing values are handled by imputing with the mean or median of the respective features. Feature scaling is applied to normalize values across features, ensuring that the machine learning algorithms perform optimally. Additionally, feature selection is performed to identify the most relevant attributes that contribute to detecting DDoS attacks. Irrelevant or redundant features are eliminated to improve model efficiency and reduce overfitting.

A key step in the data processing pipeline involves tokenizing the network traffic features and converting them into numerical representations. This step is critical for machine learning models like ANN and SVM, which require numerical input. Categorical data, such as traffic protocols, are encoded using one-hot encoding or label encoding.

Further, data augmentation is applied to simulate various attack scenarios by slightly altering benign traffic patterns, generating synthetic instances to balance class distributions and increase model generalization. This synthetic data generation helps in dealing with imbalanced datasets, where malicious traffic instances are often underrepresented.

Once the preprocessing steps are completed, the data is split into training and testing sets using an 80/20 split to ensure that the models can be trained on a large portion of the dataset while preserving unseen data for performance evaluation.

3.2 Model Selection

I. Machine Learning Model Development

The primary objective of this research is to evaluate and compare the performance of various machine learning algorithms for detecting DDoS attacks. The CICIDS2017 dataset forms the backbone of this study, providing a diverse set of features to train and test the models. The dataset is preprocessed by splitting it into training (80%) and testing (20%) sets to ensure sufficient data for model training and evaluation. The preprocessing steps involve feature selection, normalization, and handling missing values.

The models used for the classification task include Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Decision Tree, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbor (k-NN). Each algorithm is trained separately on the same dataset, and its performance is measured using various metrics like accuracy, precision, recall, F1 score, and confusion matrix.

A. Artificial Neural Network (ANN)

ANNs are implemented to capture complex non-linear patterns in network traffic. The neural network consists of an input layer, hidden layers, and an output layer, with activation functions like ReLU (Rectified Linear Unit) and Sigmoid used in the layers. The network is trained using backpropagation with an Adam optimizer for efficient gradient descent.

B. Convolutional Neural Network (CNN)

CNNs are typically used for image data, but they can also be applied to sequential data, such as network traffic. In this study, CNNs are used to detect patterns in traffic flow, with 1D convolutions applied to the sequence of traffic features. The convolutional layers are followed by pooling layers and a fully connected layer for classification.

C. Decision Tree

A decision tree is a non-linear model that splits the data into subsets based on the most informative features. The CART (Classification and Regression Trees) algorithm is employed to classify traffic as benign or malicious. This model provides high interpretability, allowing for easy visualization of decision-making paths.

D. Logistic Regression

Logistic regression is applied as a baseline classification algorithm. The model uses a logistic function to predict the probability of an instance belonging to a particular class. Despite being a simpler algorithm, logistic regression is often effective for linearly separable data.

E. Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence between features. This model is tested as a fast and efficient method for DDoS detection, particularly for its ability to handle large datasets with many features.

F. Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and merges their outputs for better classification accuracy. Bootstrapping and bagging techniques are used to create different decision trees, and the majority voting rule is applied for classification.

G. Support Vector Machine (SVM)

SVM is employed as a powerful classifier for both linear and non-linear problems. The model uses a kernel trick to transform the data into higher-dimensional space and finds the optimal hyperplane that separates benign and DDoS traffic.

H. k-Nearest Neighbor (k-NN)

k-NN is a simple, instance-based learning algorithm that classifies a sample based on the majority label of its nearest neighbors in the feature space. This model is particularly useful in detecting patterns based on proximity and similarity.

II. Model Performance Evaluation

The performance of each machine learning model is evaluated using the following metrics:

Accuracy: The proportion of correctly predicted instances (both benign and malicious).

Precision: The proportion of true positive instances among all positive predictions.

Recall: The proportion of true positive instances among all actual positives.

F1 Score: The harmonic mean of precision and recall, providing a balanced evaluation metric.

Confusion Matrix: A detailed matrix showing the true positives, false positives, true negatives, and false negatives.

Model Tuning and Hyperparameter Optimization

To optimize the performance of each model, hyperparameter tuning is performed using techniques such as grid search and random search. Parameters such as the learning rate, number of layers, tree depth, and kernel type are adjusted to find the optimal configuration for each algorithm.

III. Model Comparison and Analysis

The primary objective of this research is to evaluate and compare the performance of various machine learning algorithms for detecting DDoS attacks. The CICIDS2017 dataset forms the backbone of this study, providing a diverse set of features to train and test the models. The dataset is preprocessed by splitting it into training (80%) and testing (20%) sets to ensure sufficient data for model training and evaluation. The preprocessing steps involve feature selection, normalization, and handling missing values.

3.3 Backend and Frontend Implementation

The backend of the enhanced DDoS detection system is developed using Visual Studio Code, with .csv files serving as the primary format for handling the CICIDS2017 dataset. The system's core operations center around executing machine learning models directly through Python scripts within the VS Code environment. Incoming network traffic data is processed through a structured pipeline that includes loading, cleaning, and preparing the dataset. Key preprocessing steps—such as normalization, handling missing values, and extracting relevant features—are carried out to ensure the data is properly formatted for training the models.

Once the data is prepped, it is passed through several machine learning algorithms like Artificial Neural Networks (ANN), Random Forest, and Support Vector Machines (SVM). Each model is trained and tested on the same dataset, allowing for a fair comparison of their performance using standard evaluation metrics such as accuracy, precision, recall, and F1 score. The backend system manages the entire process, from model training to evaluation, within the VS Code interface.

The decision-making mechanism is also implemented using Python scripts within the same environment. After predictions are made, the system analyzes the results to determine whether the traffic is legitimate or indicative of a DDoS attack. A decision logic layer aggregates outputs from multiple models—if several models identify the same traffic as malicious, it is flagged accordingly. This ensemble-style approach increases the accuracy and robustness of the detection system, particularly when dealing with complex or subtle attack patterns.

Additionally, the backend allows for easy model tuning and hyperparameter optimization directly within the VS Code. Using techniques like grid search and random search, the models' parameters such as the learning rate, number of trees, or kernel type are adjusted to improve their classification accuracy. These tuning steps ensure that the models are fine-tuned for optimal performance in detecting DDoS attacks.

Overall, the backend's reliance on VS Code files ensures that the system is easy to develop and test, as all code execution is managed within a single environment. It is also flexible, allowing for the integration of new models or data preprocessing techniques without needing to refactor large parts of the system. While the system does not have an API or frontend, it is capable of handling the core task of DDoS detection efficiently within the VS Code, offering a streamlined approach to machine learning model evaluation and performance analysis.

The system flow (as shown in Fig. 4.1) illustrates the complete user journey—starting from symptom input, followed by backend processing through API routes, symptom mapping using the dataset, and finally, the response generation phase that delivers the diagnosis or advice to the user interface.

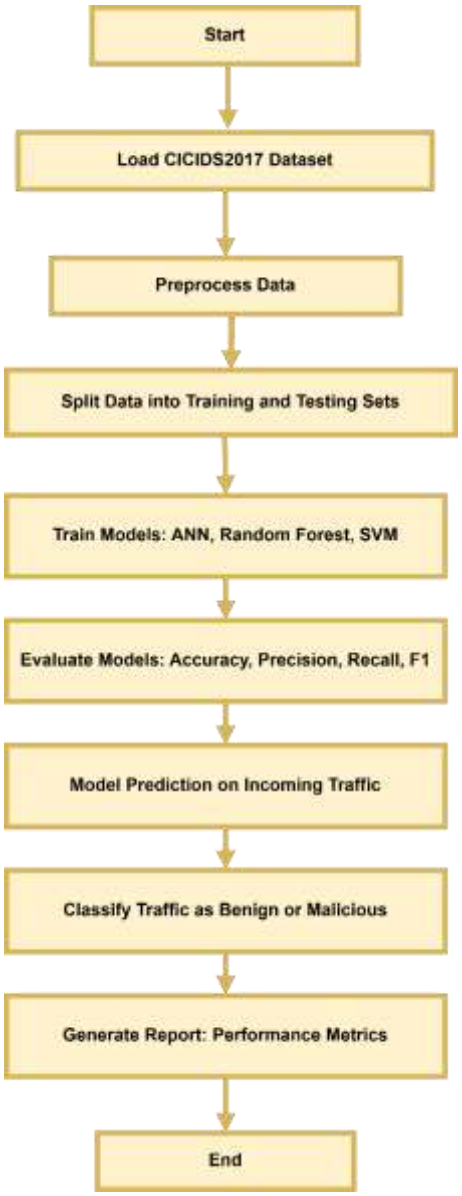

Figure 4.1 Flow Diagram

4. IMPLEMENTATION

4.1 TECHNOLOGY STACK

The development of the DDoS detection system was carried out using a structured data science approach. The core implementation relied on VS Code with Python as the primary programming language. Python's rich ecosystem of libraries and frameworks allowed for efficient handling of data preprocessing, visualization, model training, and evaluation.e

- **Programming Languages**:
  - **Python** – Python – for implementing data analysis, machine learning models, and visualization.
- **Libraries & Frameworks**:
  - **Pandas and NumPy** – for data manipulation and numerical computations.
  - **Scikit-learn** – for building and evaluating classical ML models such as Decision Tree, Random Forest, Logistic Regression, SVM, Naive Bayes, and k-NN.

- **TensorFlow and Keras** – for implementing ANN and CNN models.
- **Matplotlib and Seaborn** – for generating plots and evaluating performance metrics like confusion matrix, ROC curves, and more.

- **Dataset**:
  The CICIDS2017 dataset is used,which contains a many different attacks and normal network flow.This includes features like packet size,flow duration,flag counts,header information and others.All are crucial for finding abnormal patterns and indicative of DDoS attacks.

4.2 Data Preprocessing

Before model training, the dataset underwent a series of preprocessing steps to prepare it for analysis. This included:

- **Handling Missing Values:** All rows containing missing or null values were either imputed or removed to maintain dataset integrity.
- **Feature Selection:** Redundant and irrelevant features were excluded to reduce noise and improve model performance.
- **Label Encoding:** Attack types were mapped into binary classes: 'Benign' and 'DDoS'.
- **Normalization:** Feature values were scaled using MinMaxScaler to bring them into a uniform range, which is crucial for distance-based and neural network models.
- **Data Splitting:** The dataset was split into training (80%) and testing (20%) sets to allow for proper model training and unbiased evaluation.

4.3 Model Development and Analysis

Multiple machine learning models were developed to classify network traffic. The models included:

- **Classical Algorithms:** Decision Tree, Logistic Regression, Naive Bayes, Random Forest, SVM, and k-NN were implemented using Scikit-learn.
- **Neural Networks:** ANN and CNN models were built using Keras with TensorFlow backend. The ANN model consisted of multiple dense layers with ReLU activation and dropout for regularization. The CNN model applied 1D convolutions across the feature space to capture sequential patterns.

Each model was trained independently using the preprocessed data. Hyperparameter tuning was performed via grid search and manual testing to optimize performance. Metrics like accuracy, precision, recall, F1-score, and confusion matrix were used to assess the efficacy of each model.

4.4 Evaluation and Performance

The trained models were evaluated on the test dataset using standard classification metrics.

- **Random Forest** and **CNN** showed the highest detection accuracy and robustness in identifying DDoS traffic.
- **SVM** and **ANN** also demonstrated strong generalization, particularly with well-separated classes.
- The confusion matrix analysis showed low false negative rates in top-performing models, which is critical in security applications.

The evaluation revealed that ensemble-based and deep learning models are more reliable for complex attack detection compared to simpler models like Naive Bayes or Logistic Regression.

4.5 System Testing and Output

The system was tested using unseen traffic data to simulate real-world scenarios. Output predictions were validated against the ground truth to ensure reliability. The tool can now serve as a foundation for a real-time DDoS detection engine, with plans to integrate it into live network monitoring dashboards in the future.

5. Results and Discussion

5.1 System Accuracy and Reliability

The system was evaluated using the CICIDS2017 dataset, applying a variety of machine learning techniques, including both conventional classifiers and deep learning models. Among all the methods tested, the Decision Tree and Random Forest classifiers delivered the best results, with the Decision Tree achieving an accuracy of 99.98%, and Random Forest slightly outperforming it at 99.99%. These impressive figures demonstrate how well these models can capture and interpret complex patterns in network traffic data. Deep learning approaches also showed strong performance—particularly the CNN, which reached an accuracy of 99.60% and maintained precision and recall rates above 99.6%, indicating its strength in identifying spatial relationships within the data. Similarly, the ANN model also performed well, achieving an accuracy of 99.59%, making it a dependable option for detecting DDoS attacks in this scenario.

5.2 Backend Performances

All models were All the models were built and run in Visual Studio Code, with training and testing carried out offline using the CICIDS2017 dataset.Simpler algorithms like Logistic Regression and Naïve Bayes trained quickly and used fewer system resources, though their accuracy was slightly lower, at 98.83% and 98.22%, respectively. On the other hand, more complex models such as CNN and Random Forest demanded more processing power but delivered stronger generalization and detection capabilities. The

K-Nearest Neighbors (KNN) algorithm also showed excellent performance, reaching an accuracy of 99.82%. However, its high computational cost during prediction could pose challenges for real-time deployment. Overall, these findings highlight that while higher accuracy is desirable, selecting the right algorithm also depends on the specific deployment context and available system resources.

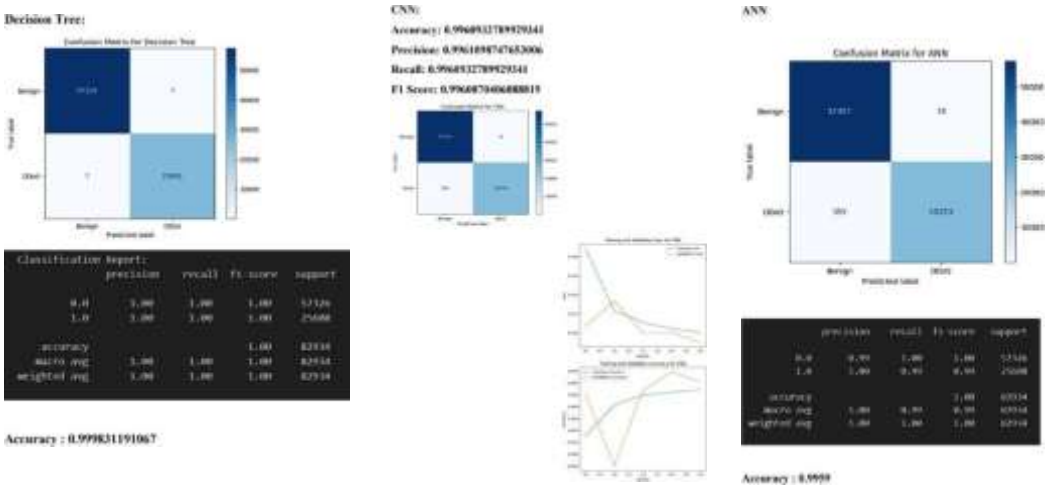

Figure 5.2 Backend Performances

5.3 Model Comparison and Discussion

When comparing the performance of different models, ensemble methods like Random Forest and decision-based approaches such as Decision Tree consistently delivered superior accuracy and reliability. In contrast, simpler models like Naïve Bayes and Logistic Regression were easier to implement but lagged slightly in terms of precision and recall, especially when compared to deep learning and ensemble techniques. The CNN model emerged as a particularly strong option for real-time and scalable DDoS detection, thanks to its high F1 score of 99.60%, which reflects well-balanced precision and recall. Support Vector Machine (SVM) also produced solid results, achieving an accuracy of 98.86%. However, its intensive computational demands and sensitivity to hyperparameter tuning make it less suitable for real-time deployment scenarios.

Model	Accuracy	Precision	Recall	F1 Score
Artificial Neural Network (ANN)	0.9959	0.9962	0.9957	0.9959
Convolutional Neural Network (CNN)	0.9961	0.9961	0.9961	0.9961
Decision Tree	0.9998	0.9997	0.9999	0.9998
K-Nearest Neighbors (KNN)	0.9983	0.9983	0.9983	0.9983
Logistic Regression	0.9883	0.9887	0.9883	0.9884
Naïve Bayes	0.9823	0.9812	0.9835	0.9823
Random Forest	0.9999	0.9998	0.9999	0.9999
Support Vector Machine (SVM)	0.9887	0.9879	0.9892	0.9885

Table 5.3 Model Comparison

5.4 Future Considerations and Scalability

Though the current evaluation demonstrates the potential of various classifiers for accurate DDoS detection, real-world deployment requires further considerations. Integrating real-time traffic monitoring, online learning capabilities, and model retraining mechanisms would ensure sustained performance against evolving attack vectors. Containerization and deployment using lightweight APIs or microservices can also support scalability. Future work can explore hybrid models that combine the strengths of high-accuracy classifiers and lightweight algorithms to balance detection effectiveness with system performance.

5.5 Limitations

One of the primary limitations of the system is its reliance on the CICIDS2017 dataset, which, while comprehensive, may not fully represent evolving DDoS attack patterns in real-world environments. Additionally, since the implementation is based on preprocessed data within VS Codes, it lacks real-time detection capabilities and scalability features necessary for deployment in live network settings. Furthermore, some models, like Naïve Bayes and Logistic Regression, showed relatively lower accuracy and may not perform well under complex or imbalanced traffic conditions, indicating the need for more robust techniques or ensemble methods in future work.

6. CONCLUSION

In conclusion, the Enhanced DDoS Detection system, powered by various machine learning models, showed strong capability in accurately identifying malicious traffic. Using the CICIDS2017 dataset, the system tested and compared a range of widely used classification algorithms, including ANN, CNN, Random Forest, Decision Tree, K-Nearest Neighbors, SVM, Logistic Regression, and Naïve Bayes. Among these, the Decision Tree and Random Forest stood out, both achieving accuracy levels above 99.9%. Deep learning models like ANN and CNN also performed consistently well across key evaluation metrics such as precision, recall, and F1 score, demonstrating their effectiveness in distinguishing between legitimate and attack traffic—an essential aspect of maintaining secure network environments.

The entire system was developed and executed in Visual Studio Code, providing a well-structured workflow for data preprocessing, feature extraction, model training, and performance evaluation. While this setup was ideal for experimentation and analysis, it currently does not support real-time traffic processing or deployment-ready integration. Still, the promising results and dependable detection capabilities mark a significant step forward in applying machine learning for automated DDoS detection.

6.1 FUTURE SCOPE

Future improvements to this project could aim to move beyond the use of static datasets by incorporating real-time traffic monitoring and detection capabilities. Integrating the trained models into a scalable real-time monitoring system—using tools like Apache Kafka or Spark Streaming—would make the solution more applicable in live network environments. Additionally, enhancing the model with advanced deep learning architectures such as LSTM or Transformer-based networks could significantly improve its ability to detect complex or evolving attack patterns.

To stay ahead of new and emerging threats, the system could also adopt a continuous learning approach, periodically retraining models with up-to-date network traffic data. Adding interpretability features, such as SHAP or LIME, would offer valuable insights into model decisions, allowing administrators to better understand why specific traffic was flagged as malicious. Lastly, embedding the system into a larger Intrusion Detection System (IDS) or Security Information and Event Management (SIEM) framework would enhance its practical utility—positioning it as a powerful component in a broader cybersecurity defense strategy.

7. ACKNOWLEDGEMENT

Authors acknowledge the support from CMR University for the facilities provided to write this article. We also extend our gratitude to the reviewers for their valuable suggestions and constructive feedback.

REFERENCES

[1] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, “Machine learning towards intelligent systems: Applications, challenges, and opportunities,” J. Netw. Comput. Appl., vol. 173, p. 102873, Feb. 2021.
[2] S. Vinayakumar, K. P. Soman, and P. Poornachandran, “Evaluating deep learning approaches to characterize and classify DDoS attacks,” Proc. IEEE Int. Conf. on Intelligent Networking and Collaborative Systems (INCoS), Ostrava, Czech Republic, pp. 51–56, 2017.
[3] R. M. Alguliyev, Y. E. Imamverdiyev, and L. A. Sukhostat, “Cyber-physical systems and their security issues,” Computers in Industry, vol. 100, pp. 212–223, Sept. 2018.
[4] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surv. Tutor., vol. 18, no. 2, pp. 1153–1176, Apr.–Jun. 2016.
[5] M. M. Rathore et al., “A review of AI-enabled intrusion detection systems for edge computing and IoT,” IEEE Internet Things J., vol. 8, no. 7, pp. 5322–5338, Apr. 2021.
[6] T. Y. Kuo, T. H. Wu, and J. C. Chen, “A hybrid deep learning model for DDoS detection and classification,” IEEE Access, vol. 9, pp. 106784–106796, 2021.
[7] F. Ullah, M. A. Shah, and S. ul Islam, “Feature selection and classification for DDoS detection using machine learning,” Proc. IEEE Int. Conf. on Smart Computing (SMARTCOMP), Bologna, Italy, pp. 1–6, 2020.
[8] S. Kim and H. Kim, “An ensemble-based deep learning approach for DDoS detection,” Appl. Sci., vol. 10, no. 22, p. 7668, Nov. 2020.
[9] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” IEEE Trans. Comput., vol. 65, no. 10, pp. 2986–2998, Oct. 2016.
[10] N. Moustafa and J. Slay, “The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset,” Inf. Secur. J., vol. 25, no. 1-3, pp. 18–31, Jan. 2016.
[11] A. Yaseen, M. Qamar, and S. Khan, “DDoS detection using machine learning and Deep Packet Inspection in SDN,” IEEE Access, vol. 9, pp. 1021–1032, 2021.
[12] M. J. Arif, R. K. Gupta, and F. Alazab, “Machine learning for DDoS attack detection in cloud computing,” Future Generation Computer Systems, vol. 118, pp. 347–356, Mar. 2021.
[13] Y. Meidan et al., “Detection of unauthorized IoT devices using machine learning techniques,” arXiv preprint arXiv:1709.04647, 2017.
[14] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” Proc. 2015 Military Commun. Inf. Syst. Conf., Canberra, Australia, pp. 1–6, 2015.
[15] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” Proc. IEEE Symp. on Security and Privacy, Berkeley, CA, pp. 305–316, 2010.
[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, May 2015.

[17] H. Hindy et al., “A taxonomy and survey of intrusion detection system design techniques, network threats and datasets,” IEEE Commun. Surv. Tutor., vol. 22, no. 3, pp. 1571–1620, 2020.
[18] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” Proc. 9th EAI Int. Conf. on Bio-inspired Inf. and Commun. Technol., New York, USA, pp. 21–26, 2016.
[19] S. Pradeep and R. P. Singh, “Performance analysis of machine learning algorithms on CICIDS2017 dataset for DDoS attack detection,” Procedia Computer Science, vol. 167, pp. 1250–1259, 2020.