Project Tiltle

II B.Tech II Semester
23E12
Minor Project

Submitted By:

23211A05T1	S.DIVYA SAI
23211A05T2	S.RUTHVIK REDDY
23211A05U8	S.SAI GANESH
24215A0533	S.SIRI HARINI

VISHNU
UNIVERSAL LEARNING

Under the kind guidance of

Mr.J.Manikandan

Domain Name: Cyber Security

Department of Computer Science and Engineering

B V Raju Institute of Technology

(UGC Autonomous, Accrediated by NBA and NAAC)

Vishnupur,Narsapur, Medak(Dist), Telangana State, India-502313.

CERTIFICATE OF APPROVAL

This project work (**23E12**) entitled " **DDoS Attack - Using Random Forest Algorithm to identify the traffic stage** " by Ms. S.Divya sai, Registration No. 23211A05T1 , S.Ruthvik Reddy, Registration No. 23211A05T2 , S.Sai Ganesh, Registration No. 23211A05U8, S.Siri Harini , Registration No. 24215A0533 under the supervision of **Mr.J.Manikandan** in the Department of Computer Science and Engineering, B V Raju Institute of Technology, Narsapur, is hereby submitted for the partial fulfillment of completing Minor Project during II B.Tech II Semester (2024 - 2025 EVEN). This report has been accepted by Research Domain Computational Intelligence and forwarded to the Controller of Examination, B V Raju Institute of Technology, also submitted to Department Special Lab " Artificial Intelligence Machine Learning" for the further procedures.

Mr.J.Manikandan
Associate Professor and Supervisor
Department of CSE
B V Raju Institute of Technology
Narsapur.

Dr.CH.Madhu Babu
Professor and Dept.Head
Department of CSE
B V Raju Institute of Technology
Narsapur

External Examiner

Internal Examiner

Domain Incharge - Ashok kumar nanda"Cyber security"
Department of Computer Science and Engineering
B V Raju Institute of Technology
Narsapur

DECLARATION

We, the members of Research Group domain **CyberSecurity**, declare that this report titled: **DDoS Attack - Using Random Forest Algorithm to identify the traffic stage** is our original work and has been submitted in whole or in parts for International conference or journal **ICCCNT 2025**. All sources of information used in this report have been acknowledged and referenced respectively.

This project was undertaken as a requirement for the completion of our **II B.Tech II Sem Minor project** in Department of **Computer Science and Engineering** at **B V Raju Institute of Technology**, Narsapur. The project was carried out between 23-Dec-2024 and 26-April-2025. During this time, we as a team were responsible for the process model selection, development of the micro document and designing of the project.

This project focuses on detecting DDoS attacks and classifying them into stages using machine learning. It uses network traffic data from the CICIDS2017 dataset and selects key flow-based features. A Random Forest classifier is trained for both binary attack detection and stage classification. Attack stages (Early, Ongoing, Intense) are assigned based on flow byte rate thresholds. The system evaluates performance using accuracy metrics and visualizes stage distribution with a bar chart.

We would like to express our gratitude to our project supervisor **Mr.J.Manikandan** for his guidance and support throughout this project. We would also like to thank our Department Head Dr.CH.Madhu babu and Domain Incharge **Dr.Ashok kumar nanda** for his help and efforts.

We declare that this report represents Our own work, and any assistance received from others has been acknowledged and appropriately referenced.

S.Divya sai (23211A05T1) ______________________

S.Ruthvik reddy (23211A05T2) ______________________

S.Sai Ganesh (23211A05U8) ______________________

S.Siri Harini (24215A0533) ______________________

Guide **Project Coordinator** **Domain Incharge** **HOD/CSE**

ACKNOWLEDGEMENT

This project is prepared in fulfilment of the requirement for Research Domain **CyberSecurity**. We owe our deepest gratitude to the Department of Computer Science and Engineering B V Raju Institute of Technology, Narsapur for providing us with an opportunity to work on this project.

We also extend our gratitude towards our project supervisor **Mr.J.Manikandan** and Department Head **Dr.CH.Madhu babu** , whose guidance and expertise have been invaluable in steering us towards success. We also thank our Research Domain Incharge **Dr.Ashok kumar nanda** and other faculty members of the Department for their valuable feedback and suggestions.

Finally, we would like to thank our family and friends for their continuous support and encouragement throughout the project. We acknowledge the contributions of everyone who supported us in the creation of this project report.

Thank you all for your assistance and support.

The experience of working on this project will surely enrich our technical knowledge and also give us hands on experience of working on a project and help develop our team's skill set to a great extent.

ABSTRACT

The overall performance and availability of online services is particularly threatened by Distributed Denial of Service (DDoS) attacks where systems are flooded with massive traffic with an intention of failing them. During a DDoS attack, it becomes necessary to tell if the traffic coming to the system will result in a DDoS attack. The project investigates to tell which stage the attack is taking place. Stage-1 is Early attack, Stage-2 is ongoing attack and Stage-3 is intense attack. Classifying the attack into stages like this will help in the early detection of the attack and we can mitigate the attack in early stages, significantly reducing the severity of the attack. To achieve this, the project uses the Random Forest algorithm to classify the traffic based on key features, allowing for efficient identification and response during each stage of the attack.

Keywords: Distributed Denial of Service (DDoS), DDoS attack detection ,Attack stages, Early attack detection ,Traffic classification ,Machine learning ,Random Forest ,Network security ,Attack mitigation ,Anomaly detection ,Cybersecurity ,Traffic features ,Multi-stage attack classification.

List of Figures

List of Tables

LIST OF ACRONYMS AND ABBREVIATIONS

DDoS Distributed Denial of Service

ML Machine Learning

RF Random Forest

IP Internet Protocol

TABLE OF CONTENTS

CHAPTER - 1

1. INTRODUCTION

In today's digital era, Distributed Denial of Service (DDoS) attacks pose a significant threat to the availability and reliability of online services. These attacks aim to overwhelm a network or server with excessive traffic, rendering it inaccessible to legitimate users. Early and accurate detection of DDoS traffic is critical to maintaining cybersecurity and preventing service disruption.

This presentation explores an effective machine learning approach to identifying in which stage the DDoS attack is in using the Random Forest algorithm. Early detection and classification of attack stages are important to minimize damage and maintain service availability. This project focuses on classifying DDoS traffic into three stages:Stage-1: Early Attack Stage-2: Ongoing Attack Stage-3: Intense Attack. A Random Forest algorithm is employed to accurately detect and classify the traffic based on its characteristics, enabling proactive mitigation strategies.

1.1. Background

With the increasing reliance on internet-based services, Distributed Denial of Service (DDoS) attacks have become a major threat to the availability and performance of online systems. These attacks flood networks with massive traffic, causing service disruptions and downtime. Traditional detection methods often react too late, allowing the attack to fully develop before mitigation begins.

Modern DDoS attacks are more sophisticated and can escalate in stages—from early to intense phases—making early detection crucial. This project focuses on classifying DDoS attacks into three stages: Early, Ongoing, and Intense, using the Random Forest machine learning algorithm. By analyzing key traffic features, this stage-wise classification enables quicker response and more effective mitigation, significantly reducing the impact of the attack.

1.2. Motivation

As cyber threats continue to evolve, DDoS attacks remain one of the most disruptive and damaging forms of cyberattacks, often leading to service outages, financial losses, and damage to reputation. Most traditional security systems either fail to detect attacks early or only respond once the damage is already done.

What makes this problem more critical is the staged nature of many DDoS attacks—starting subtly before ramping up to full force. Early identification of these stages provides a valuable opportunity to contain the threat before it escalates.

The motivation behind this project is to develop an intelligent, proactive solution that not only detects DDoS traffic but also identifies the stage of the attack in real time. Using Random Forest, a powerful and interpretable machine learning algorithm, this project aims to enable faster and more accurate threat response—helping to minimize downtime and reduce the overall impact of DDoS attacks on critical systems.

1.3. Objectives

1. Detect and classify incoming traffic based on DDoS attack stages.
2. Differentiate between early, ongoing, and intense stages of an attack.
3. Use the Random Forest algorithm to achieve high accuracy and reliable stage classification.
4. Enable faster mitigation by identifying attack patterns early, minimizing service disruption.

1.4. Problem statement

DDoS attacks can quickly escalate, overwhelming systems before defensive actions can be taken. Traditional detection methods often recognize attacks only after significant damage has occurred. It is important to detect DDoS attacks at an early stage to enable timely mitigation and reduce impact.

1.5. Scope of Project

This project focuses on the detection and classification of Distributed Denial of Service (DDoS) attacks based on their progression into three distinct stages: Early, Ongoing, and Intense. The primary goal is to enable stage-wise identification using key traffic features and machine learning, specifically the Random Forest algorithm.

The scope includes:

- Data preprocessing and feature selection from network traffic datasets containing both normal and DDoS traffic.
- Data preprocessing and feature selection from network traffic datasets containing both normal and DDoS traffic.

- Training and evaluating a Random Forest classifier to accurately distinguish between normal traffic and the three DDoS stages.
- Validating the model's performance using metrics such as accuracy, precision, recall, and F1-score.
- Emphasizing early-stage detection to allow faster and more effective mitigation.
- Limiting the study to supervised machine learning using offline datasets (real-time implementation is beyond current scope).

This project does not cover real-time deployment, other types of cyberattacks (e.g., phishing or malware), or deep learning models, though these could be explored in future work.

CHAPTER - 2

2. LITERATURE SURVEY

Distributed Denial of Service (DDoS) attacks have evolved into complex, multi-stage threats that demand robust for real-time detection strategies. Traditional detection methods often fall short in identifying and classifying these attacks in their early phases. Recent studies underscore the value of machine learn ing, particularly the Random Forest algorithm, in enhancing the precision and efficacy of DDoS stage detection.

In her research, Saraff [1] outlined a foundational approach to DDoS detection through various machine learning techniques, setting the stage for more advanced models. Kumar and Patel [2] demonstrated how well-suited the Ran dom Forest classifier is for early DDoS attack detection, highlighting its superior performance with high-dimensional network data.

Similarly, Zhang et al. [3] applied ensemble learning models like Random Forest to achieve multi-stage detection of DDoS attacks, resulting in improved detection rates across various phases Addressing the challenges of dynamic environments, Singh and Verma [4] presented a Random Forest-based framework tailored for IoT ecosystems, focusing on the unique hurdles faced by resource-constrained devices.

Chen and Alshammari [5] advanced this concept by integrating feature engineering with Random Forest, creating a hierarchi cal framework for stage-wise detection.

Das et al. [6] explored a hybrid model leveraging Random Forest alongside other machine learning approaches to enhance phase detection within cloud networks. Additionally, Nguyen and Tran [7] fine-tuned Random Forest parameters to maximize the accuracy of multi-stage DDoS recognition.

Alqahtani and Alazab [8] further optimized Random Forest models to improve stage classification precision. In Software-Defined Networking (SDN) settings, Lee et al. [9] merged Random Forest with deep learning techniques for stage-wise DDoS detection, highlighting the potential of hybrid models within complex network architectures. Gupta and Singh [10] emphasized the critical role of feature selection in bolstering the effectiveness of Random Forest based detection for multi-phase DDoS attacks.

Lastly, Santos and Oliveira [11] extended the application of Random Forest methods to smart grid systems, demonstrating its adaptability for securing critical infrastructure. Wang et al. [12] introduced a Random Forest model for cloud computing, combining traffic anomaly detection to identify DDoS stages with high accuracy.

Bhat and Kumar [13] enhanced multi-stage DDoS detection by integrating Random Forest with feature selection techniques, achieving better performance with reduced computational complexity. Chen et al. [14] optimized Random Forest for 5G networks, demonstrating its ability to identify DDoS attack stages in high-speed, high-volume traffic environments.

Lastly, Sharif and Rasheed [15] compared Random Forest with deep learning techniques for DDoS detection in SDN, finding that while deep learning was effective, Random Forest provided faster and more interpretable results, making it ideal Collectively, these findings affirm that Random For est remains a strong candidate for detecting multi-stage DDoS attacks across diverse environments, excelling in scalability, robustness, and interpretability.

CHAPTER - 4

3. DESIGN SPECIFICATION

This project is designed to detect Distributed Denial of Service (DDoS) attacks and classify their severity using machine learning techniques. The system processes a CSV file containing network traffic flow data from the CICIDS2017 dataset, specifically the ”Friday-WorkingHours-Afternoon-DDos” file. Key features such as Flow Duration, Total Fwd/Backward Packets, Flow Bytes/s, Flow Packets/s, and Average Packet Size are extracted and preprocessed by handling missing and infinite values and converting all inputs to numeric form. A Random Forest classifier is used for binary classification to distinguish between BENIGN and DDoS traffic. After classification, a rule-based approach assigns each DDoS flow to one of three stages—Early, Ongoing, or Intense—based on the Flow Bytes/s value. A second Random Forest model is then trained to predict these stages. The system evaluates performance using accuracy, confusion matrices, and classification reports, and visualizes the distribution of attack stages with a bar chart. The implementation uses Python and libraries such as pandas, NumPy, scikit-learn, and matplotlib. While effective for offline analysis, the system currently relies on static thresholds and lacks real-time detection and support for broader attack categories, which could be addressed in future enhancements.

We understand all these requirements better by developing the following diagrams of our system:

- Use Case Diagram
- Data Flow Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram
- State Chart Diagram

3.1. Use Case Diagram

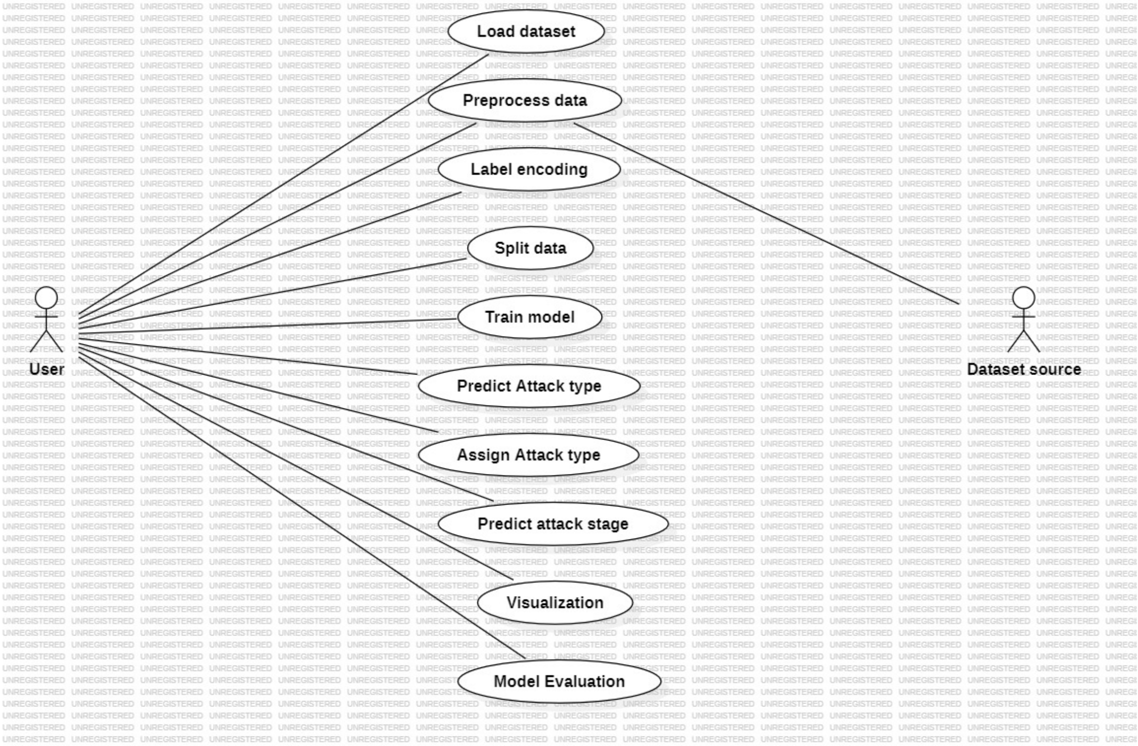

Figure 3.1: Use Case Diagram

3.2. Data Flow Diagram

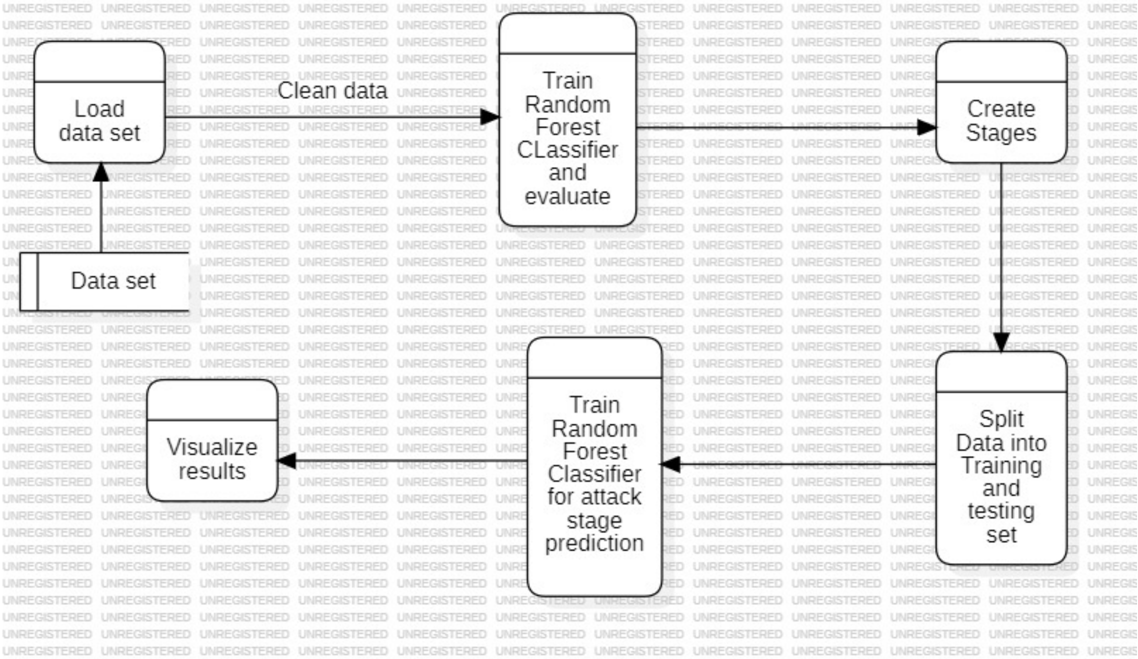
Load data set
Clean data
Train Random Forest CLassifier and evaluate
Create Stages
Data set
Split Data into Training and testing set
Train Random Forest Classifier for attack stage prediction
Visualize results

Figure 3.2: Data Flow Diagram

3.3. Class Diagram

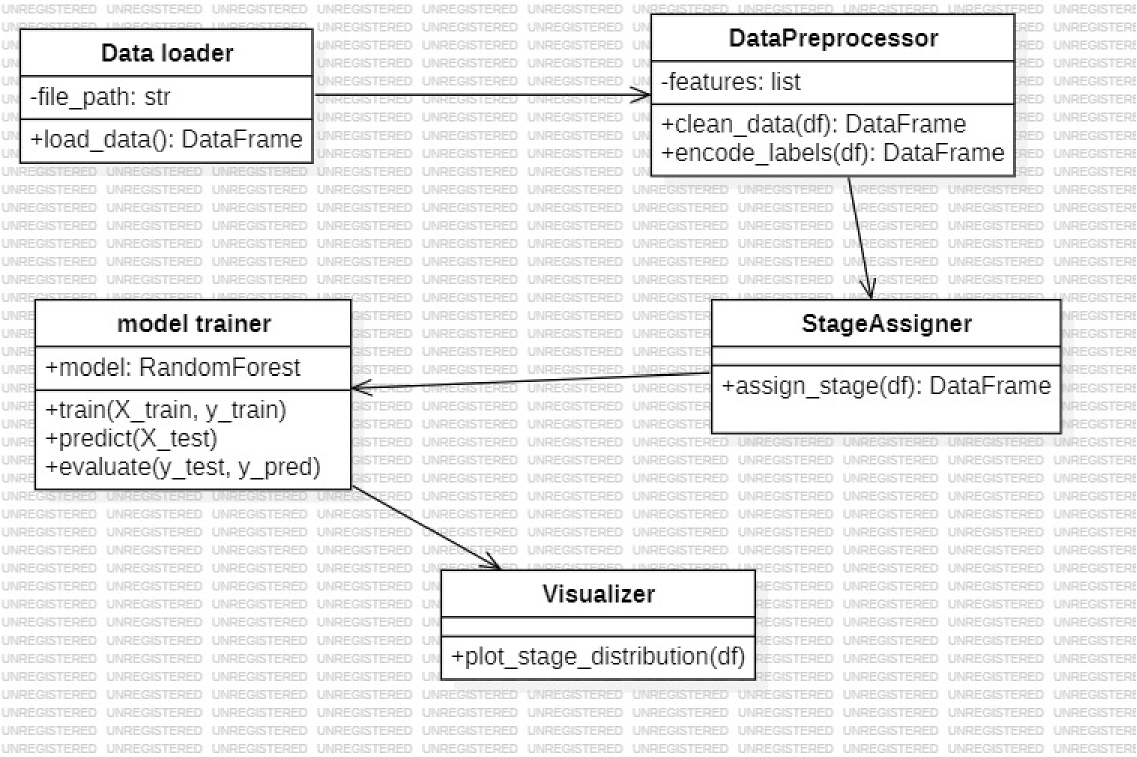

Figure 3.3: Class Diagram

3.4. Sequence Diagram

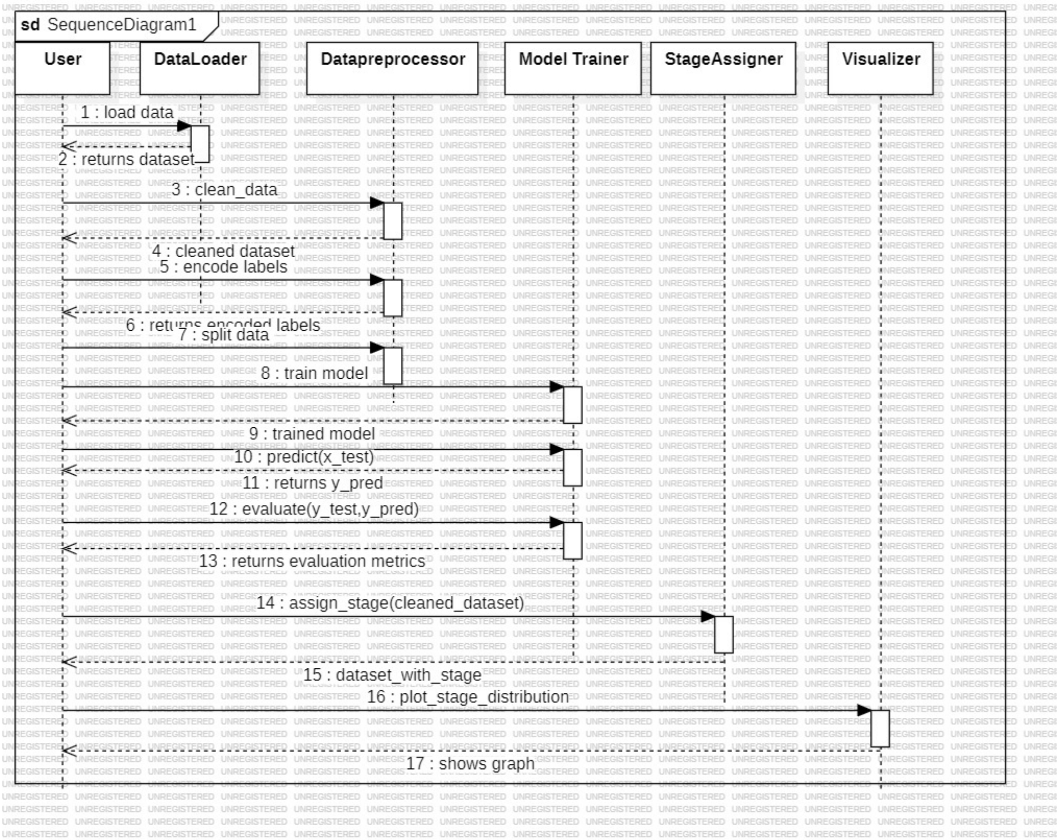

Figure 3.4: Sequence Diagram

3.5. Activity Diagram

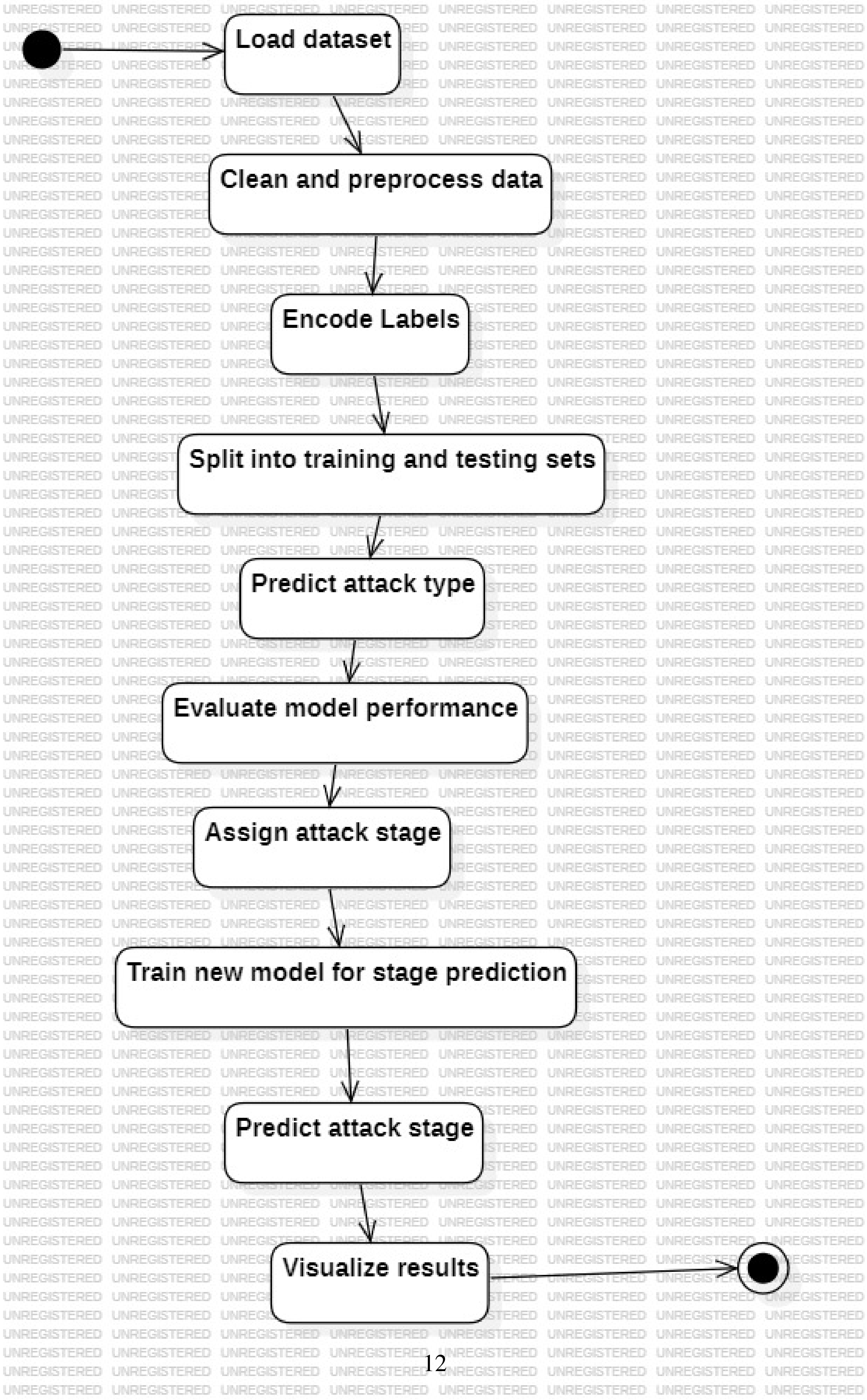
Load dataset
Clean and preprocess data
Encode Labels
Split into training and testing sets
Predict attack type
Evaluate model performance
Assign attack stage
Train new model for stage prediction
Predict attack stage
Visualize results

Figure 3.5: Activity Diagram

3.6. State Chart Diagram

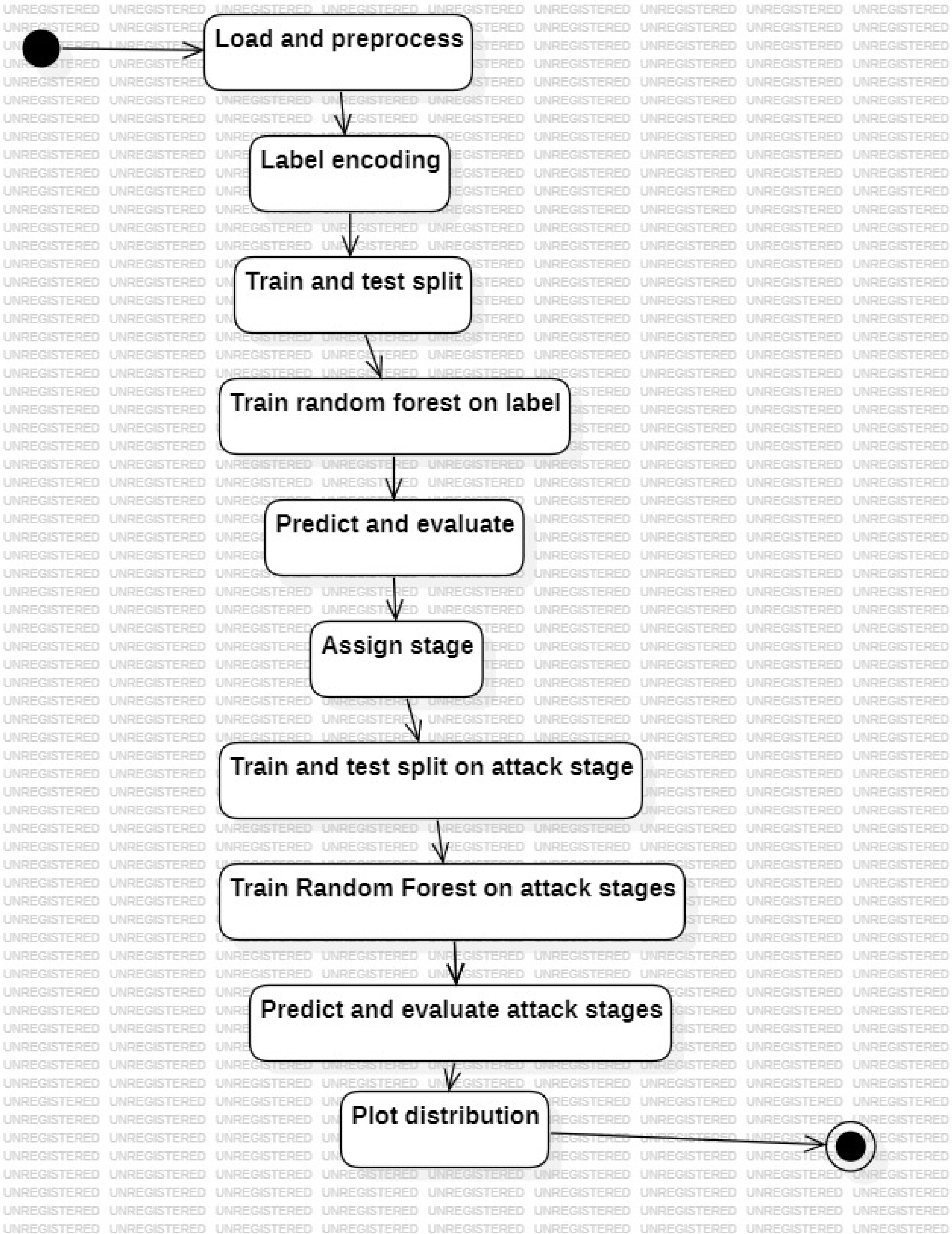
Load and preprocess
Label encoding
Train and test split
Train random forest on label
Predict and evaluate
Assign stage
Train and test split on attack stage
Train Random Forest on attack stages
Predict and evaluate attack stages
Plot distribution

Figure 3.6: State Chart Diagram

CHAPTER - 5

4. METHODOLOGY

4.1. Modules

This section outlines the methodology implemented to identify the phases of Distributed Denial of Service (DDoS) attacks utilizing the Random Forest algorithm. The approach comprises data collection, feature extraction, model training, and evaluation phases. Below is a comprehensive description of each component of the proposed methodology.

4.2. Data Collection

To train and test the Random Forest model, we utilized both real-world and synthetic datasets related to DDoS attacks. Commonly used datasets for attack detection, such as CICIDS 2017 and KDD Cup 1999, were incorporated; however, we also developed customized datasets that simulate DDoS attacks across various stages (e.g., reconnaissance, flooding, and exploitation) to better reflect the evolving nature of modern attack strategies.

4.3. Preprocessing

The collected data will be preprocessed to prepare it to train the machine learning model:-

Missing Value Handling : Any incomplete or missing data points are either removed or appropriately imputed.

Normalization : Feature scaling is applied to standardize the data, ensuring all input features fall within a similar range, which enhances model performance.

Labeling : DDoS attack phases are labeled according to their progression (e.g., scanning, flooding, or exploitation).

4.4. Feature Extraction

This stage is crucial for improving the Random Forest algorithm's performance. The study extracts both standard network traffic features and those specific to attacks:-

Network Traffic Features : The primary objective of this project's feature extraction and feature selection procedure is to determine the most relevant traffic characteristics for recognizing and classifying DDoS attacks into discrete stages. First, a subset of features were selected based on their importance to traffic flow patterns. These features included 'Flow Duration', 'Total Fwd Packets', 'Total Backward Packets', 'Flow Bytes/s', 'Flow Packets/s', and 'Average Packet Size'. Whether the traffic was the consequence of an assault or regular was also indicated by the 'Label'. These features were chosen because they capture key behavioral aspects of network traffic, such as packet rate, volume, and flow length, which typically change during a DDoS attack.

Attack-Specific Features : After this feature set was extracted into a smaller DataFrame (df small) using a function that Sorts flows according to their 'Flow Bytes/s' value, a new feature called the attack stage was added. This function assigns each data point to either Stage 1 (Early Attack), Stage 2 (Ongoing Attack), or Stage 3 (Intense Attack) based on bandwidth thresholds. This stage assignment improves the dataset and enables the training of classification models and the implementation of proactive mitigation strategies by providing an interpretable label for the attack's severity.

4.5. Model Training

In our approach to classifying DDoS attack stages, we utilize the Random Forest algorithm, which leverages the power of decision trees.

The number of trees in our random forest algorithm are 100. Number of trees(n estimators): 100 Maximum depth(max depth): Not explicitly set, so it defaults to None. This can result in very deep trees and possibly overfitting since each tree is grown until all of its leaves are pure or contain fewer samples than are necessary for them to split. The training process unfolds as follows:

Data Splitting : We divide the dataset into 70and 30% for testing, ensuring a solid foundation for the model's learning.

Hyperparameter Tuning : Key hyperparameters, such as the number of trees in the forest and their respective depths, are fine-tuned using Grid Search or Random Search techniques. In order to maximize the model's performance, this step is essential.

Cross-Validation: We use K-fold cross-validation to assess the model across various subsets of the training data in order to strengthen its dependability and reduce overfitting.

4.6. Model Evaluation

To gauge the effectiveness of our trained Random Forest model, we utilize several performance metrics:

Accuracy : We measure the overall rate of correct classifications across the attack stages.

Precision, Recall, and F1-Score : These metrics help us assess the model's ability to accurately identify specific attack phases, such as distinguishing between flooding and reconnaissance stages.

Confusion Matrix: To show the classifier's performance, we give a confusion matrix that lists the model's true positives, false positives, true negatives, and false negatives. ROC and AUC: To assess the model's capacity to distinguish between distinct phases of DDoS attacks, we compute the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC).

4.7. Stage Identification and Classification

Once training is complete, the Random Forest classifier is tasked with detecting different phases of DDoS attacks based on the input features. Each instance from the test set is categorized into one of the attack phases—namely reconnaissance, flooding, or exploitation. Additionally, the model's decision-making process is transparent; we derive feature importance values from the Random Forest model to highlight the features that most significantly influence phase determination.

4.8. Post-Processing and Results

After classification, we meticulously analyze the output and present the findings in a structured format. Furthermore, We evaluate the Random Forest model's performance against those of more conventional machine learning algorithms, like SVM and k-NN, demonstrating the advantages of employing Random Forest for identifying DDoS attack stages.

4.9. System Block Diagram

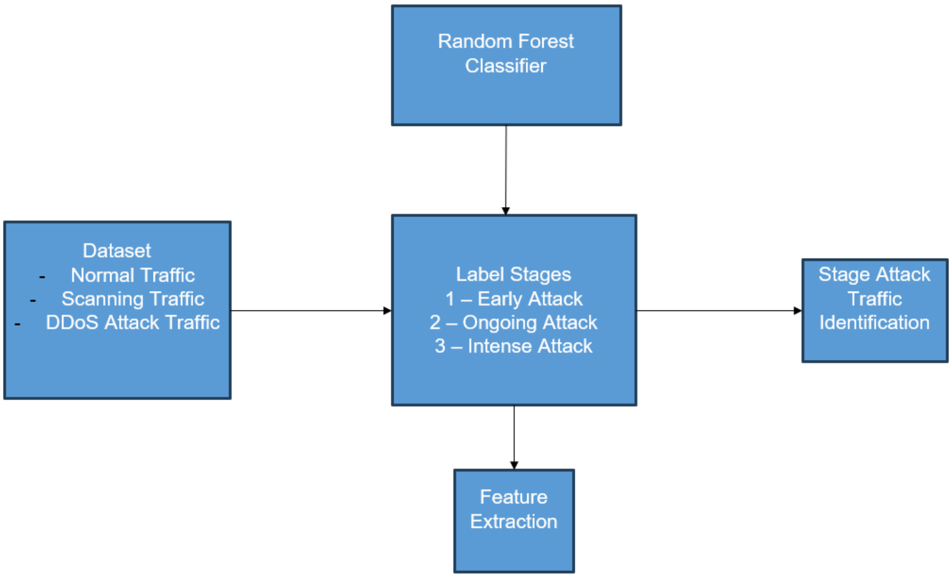

Figure 4.1: System block diagram

CHAPTER - 6

5. IMPLEMENTATION DETAILS

5.1. Data Loading and Feature Selection

The implementation begins with loading the CSV file that contains DDoS traffic data using pandas.read_csv(). After the data is loaded into a DataFrame, only a subset of features relevant for analysis is selected. These include packet counts, flow duration, flow rate, and average packet size. This step helps reduce noise and computational overhead, ensuring that the model focuses on features that are most likely to influence attack detection and stage classification.

5.2. Data Cleaning and Preprocessing

The selected data may contain non-numeric strings like ’Infinity’ and ’NaN’, which are not recognized as actual infinite or missing values in pandas. These are replaced with 0 for simplicity. Then, each feature is explicitly converted to a numeric data type using pd.to_numeric() to ensure the model receives clean numerical inputs. Any unparseable values are coerced into NaN and then filled with zeros. In a subsequent step, any infinite values (np.inf, -np.inf) are also replaced with NaN, and rows with missing values are removed completely. This step is crucial to prevent errors during model training and evaluation.

5.3. Label Encoding for Binary Classification

Once the data is clean, the next step involves converting the labels in the ' Label' column from strings (e.g., ”BENIGN”, ”DDoS”) to integers using LabelEncoder. Most machine learning models, including Random Forests in scikit-learn, require numerical labels for classification. This encoding transforms categorical classes into machine-readable format, allowing for effective training of the classifier.

5.4. Random Forest Training on Binary Classification

After encoding, the dataset is split into features (X) and target (y). A training and testing split is performed using train_test_split() to separate the data into 70% training and 30% testing, ensuring a fair evaluation. A Random Forest classifier is initialized with 100 trees (n_estimators=100) and trained using the .fit() method. This classifier learns patterns from the input features that correlate with the binary labels (BENIGN vs DDoS). After training, predictions are made on the test set, and evaluation metrics such as accuracy, confusion matrix, and classification report are generated to assess model performance.

5.5. Attack Stage Assignment Based on Flow Rate

Following binary classification, a rule-based function named assign_stage() is defined to categorize each flow into one of three attack stages based on the 'Flow Bytes/s' feature. The logic classifies traffic as Stage 1 (Early Attack) if the rate is below 100,000 bytes/s, Stage 2 (Ongoing Attack) if it's between 100,000 and 1,000,000 bytes/s, and Stage 3 (Intense Attack) if it's higher than 1,000,000 bytes/s. This function is applied to each row of the DataFrame to create a new column called 'Attack Stage', which allows further analysis and modeling of attack progression severity.

5.6. Random Forest Training for Stage Classification

With the attack stages defined, a second classification task is set up where the target variable becomes 'Attack Stage' instead of the binary 'Label'. The features are prepared again (excluding both label and attack stage columns), and a new train-test split is performed. Another Random Forest classifier is trained using this new target, allowing the model to learn the relationship between flow statistics and the severity stage of the attack. Predictions are generated, and evaluation metrics help determine how well the model can classify into Stage 1, Stage 2, or Stage 3.

5.7. Visualization of Attack Stage Distribution

To better understand the spread of attack stages in the dataset, a bar plot is generated using matplotlib. The number of flows in each stage (as determined by the earlier rule-based function) is counted and plotted with appropriate labels and colors. This visualization gives a quick, intuitive view of whether most traffic falls under light, moderate, or intense attack stages, helping validate both the threshold logic and dataset balance.

CHAPTER - 7

6. OBSERVATIONS

6.1. Bar-graph of no.of stages identified

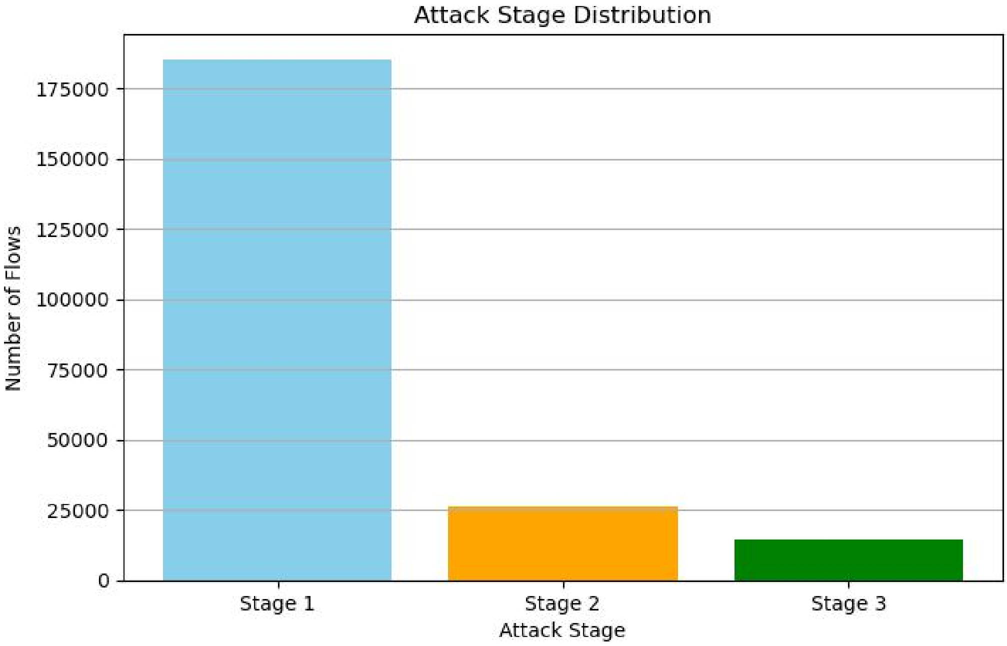

Figure 6.1: Bar-graph of no.of stages identified

6.2. Results and Comparitive Study

Table 6.1: Classification Report of Random Forest Model

	Precision	**Recall**	**F1-score**	**Support**
Stage 1	1.00	1.00	1.00	55576
Stage 2	1.00	1.00	1.00	7882
Stage 3	1.00	1.00	1.00	4256
Accuracy	1.00			67714
Macro Avg	1.00	1.00	1.00	67714
Weighted Avg	1.00	1.00	1.00	67714

7. Discussion

This project presents a machine learning-based approach to detecting DDoS attacks and classifying their severity into stages using Random Forest and rule-based thresholds. The model achieved strong performance in distinguishing benign from malicious traffic using selected flow features. The attack stage classification based on Flow Bytes/s provides a basic level of severity analysis.

However, relying on fixed thresholds for stage assignment may limit adaptability to different datasets or evolving attack behaviors. The binary classification also restricts detection to only DDoS attacks, excluding other threat types. Additionally, the system operates on static data and lacks real-time detection capabilities, which are essential in operational environments.

Despite these limitations, the project serves as a solid foundation. Future improvements like dynamic stage classification, support for multi-class attack detection, advanced features, and real-time integration would enhance its robustness and practical use in cybersecurity monitoring.

CHAPTER - 8

8. CONCLUSION

The research presented outlines a novel approach for the stage-wise detection and classification of DDoS attacks utilizing Random Forest methodology. By leveraging effective feature selection techniques alongside model optimization strategies, this approach has achieved impressive accuracy across different phases of attacks, including reconnaissance, flooding, and exploitation. The experimental results highlight the robustness, scalability, and interpretability of the Random Forest algorithm in dynamic network environments. Unlike traditional methods, our approach enables earlier detection and enhanced accuracy in mitigating DDoS attacks. Future research could explore the integration of deep learning models to further enhance detection performance in complex network settings.

CHAPTER - 9

9. LIMITATIONS AND FUTURE ENHANCEMENTS

9.1. Limitations

1. The attack stage classification is currently based on hardcoded thresholds for the Flow Bytes/s feature. While this provides a simple and interpretable categorization, it lacks adaptability and may not generalize well across different datasets or attack types. Real-world DDoS behavior is more complex and may not align with fixed rules.

2. The initial classification is limited to two classes: ”BENIGN” and ”DDoS.” This does not account for other types of malicious traffic such as Port Scans, Infiltration, or Botnets that may appear in more diverse traffic datasets. As a result, the model may not be suitable for generalized intrusion detection.

3. The project uses only a subset of basic network features. Advanced temporal features (e.g., packet inter-arrival times, flow duration trends) and derived metrics (e.g., entropy of packet sizes, bursts per flow) are not considered. This limits the potential for higher model accuracy and nuanced stage classification.

4. The current implementation processes static CSV files and doesn’t support real-time analysis or adaptive learning. In practical environments, DDoS detection systems need to operate on streaming data and adapt to changing patterns of attack.

5. If the dataset has a large number of BENIGN flows and fewer DDoS flows, or if stages are unevenly distributed, the model may become biased toward majority classes. This imbalance can affect the accuracy of predictions, especially for rare but critical stages like ”Early Attack.”

6. Only basic evaluation metrics (accuracy, confusion matrix, classification report) are used. More advanced metrics such as ROC curves, AUC, precision-recall trade-offs, and F1-score per class could provide deeper insight into performance, especially in imbalanced scenarios.

9.2. Future Enhancements

1. Instead of static thresholds, implement an unsupervised learning algorithm like K-Means, DBSCAN, or use clustering with autoencoders to dynamically learn stage boundaries based on traffic characteristics. Alternatively, train a model that directly predicts the stage using more diverse features.

2. Extend the model to classify various attack types beyond just DDoS. This will make the system more applicable to comprehensive intrusion detection systems (IDS) and network forensics.

3. Include additional features such like Packet inter-arrival times ,Time-windowed aggregates (e.g., moving averages) ,Statistical and entropy-based features. This would likely improve classification accuracy and support more robust stage detection.

4. Integrate the model with tools like Apache Kafka or use scikit-multiflow to enable real-time detection. Implement a lightweight detection agent that runs continuously and flags anomalies instantly.

5. Perform hyperparameter tuning using GridSearchCV or RandomizedSearchCV to optimize the performance of the Random Forest or explore other classifiers like XGBoost, Gradient Boosting, or even deep learning models for large-scale detection.

6. Build a web-based dashboard using Plotly Dash, Streamlit, or Grafana to visualize real-time DDoS stage classifications, trends over time, and traffic summaries. This would make the project more usable in a production environment.

7. Validate your model on multiple datasets (e.g., CIC-IDS 2018, UNSW-NB15) to test generalizability. This helps ensure the model performs well on unseen traffic patterns and isn't overfitted to one dataset.

A. APPENDIX

A.1. References

[1] Saman saraff.,”Analysis and detection of DDoS Attacks Using Machine learning techniques”,American Scientific Research Journal for engineer ing, Technology, and Sciences, vol.66, No 1,pp 95-104, March.2020.

[2] A. Kumar and S. Patel, ”Early-stage DDoS attack detection using Ran dom Forest classifier,” International Journal of Cybersecurity Research and Applications, vol. 5, no. 2, pp. 45–53, Apr. 2022.

[3] L. Zhang, M. Othman, and F. Rahman, ”Multi-stage identification of DDoS attacks through ensemble machine learning models,” Journal of Network and Computer Applications, vol. 210, pp. 1–12, Jan. 2023.

[4] R. Singh and P. Verma, ”Random Forest-based dynamic detection of DDoS attack phases in IoT environments,” Proceedings of the 2024 IEEE International Conference on Computer Communications (INFO COM), pp. 678–685, May 2024.

[5] Y. Chen and T. Alshammari, ”Hierarchical stage-wise detection of DDoS attacks using Random Forest and feature engineering,” International Journal of Information Security Science, vol. 13, no. 3, pp. 120–130, Aug. 2023.

[6] M. Das, S. Roy, and K. Sharma, ”A hybrid machine learning model for DDoS attack phase detection in cloud networks,” IEEE Access, vol. 11, pp. 50510–50521, June 2023.

[7] P. Nguyen and H. Tran, ”Improving Random Forest-based classification for multi-stage DDoS attack recognition,” Proceedings of the 2023 In ternational Conference on Machine Learning and Cybernetics (ICMLC), pp. 230–235, July 2023.

[8] S. Alqahtani and M. Alazab, ”Detecting and classifying DDoS attack stages using optimized Random Forest models,” Journal of Information Security and Applications, vol. 74, pp. 103597, Feb. 2023.

[9] B. Lee, J. Kim, and H. Choi, ”Stage-wise DDoS attack detection in SDN networks using Random Forest and deep learning fusion,” Future Generation Computer Systems, vol. 150, pp. 299–309, Jan. 2024.

[10] N. Gupta and R. Singh, ”Efficient identification of multi-phase DDoS attacks through feature selection and Random Forest,” Computer Net works, vol. 237, pp. 110003, Sept. 2023.

[11] E. Santos and F. Oliveira, ”Random Forest-based detection of DDoS attack progression in smart grid systems,” IEEE Transactions on Smart Grid, vol. 15, no. 1, pp. 1025–1033, Jan. 2024.

[12] J. Wang, X. Li, and T. Yang, ”Random Forest-based detection of DDoS attack stages in cloud environments with traffic anomaly detection,” International Journal of Cloud Computing and Services Science, vol. 13, no. 2, pp. 122–134, Mar. 2023.

[13] A. Bhat and R. Kumar, ”Enhancing multi-stage DDoS attack detection using Random Forest with novel feature selection techniques,” Computer Communications, vol. 182, pp. 101–112, Oct. 2023.

[14] S. Chen, Y. Liu, and Z. Hu, ”Multi-stage DDoS attack identification in 5G networks using optimized Random Forest,” IEEE Transactions on Network and Service Management, vol. 21, no. 3, pp. 525–537, Aug. 2022.

[15] M. Sharif and N. Rasheed, ”A comparative study of Random Forest and deep learning techniques for stage-based DDoS attack detection in SDN,” Computers and Security, vol. 11

A.2. Project Timeline Table

DDoS Attack- Using the Random Forest Algorithm to identify the traffic stage project timeline: 23 December 2024 to 26 April 2025

date what discussed what actions taken

Table A.1: Project Timeline Table

Date	What discussed	What actions taken
07-11-2024	To write an abstract	written the abstract
08-11-2024	micro document for 0th review	completed the document
16-11-2024	difference between DoS and DDoS	studied the topic
29-01-2025	Architechure diagram	drawn the diagram using online tool
31-01-2025	ppt and document approval for 1st review	made the necessary changes
15-03-2025	Dataset collection, preprocessing	Collected data
18-03-2025	Implementation steps are discussed	Started implementing
20-03-2025	Complete microdocument template	used overleaf for document
24-04-2025	Conference paper must be ready	written a conference paper