Dynamic Feature Selection and Ensemble Learning for DDoS Detection in SDN

A Project Report Submitted to
Jawaharlal Nehru Technological University Anantapur, Ananthapuramu

in partial fulfillment of the requirements for the
Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Submitted by

Batch No: CS 25-10

NAMBETI MAHATHI	**21121A3738**
K E NITHISH KUMAR	**21121A3725**
KOGARA MALANI	**21121A3727**
YALAMANCHILI VENKATA SUBHASH	**21121A3760**

Under the Supervision of
Dr. C. Siva Kumar
Associate Professor

Department of CSSE

Department of Computer Science and Systems Engineering
SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)
(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE,
Accredited by NBA & NAAC)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA
2024-2025

SREE VIDYANIKETHAN ENGINEERING COLLEGE (AUTONOMOUS)
(Affiliated to JNTUA, Ananthapuramu, Approved by AICTE,
Accredited by NBA & NAAC)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA
Department of Computer Science and Systems Engineering

CERTIFICATE

This is to certify that the project report entitled

“Dynamic Feature Selection and Ensemble Learning for DDos detection in SDN”

is the Bonafide work done by

NAMBETI MAHATHI	**21121A3738**
K E NITHISH KUMAR	**21121A3725**
KOGARA MALANI	**21121A3727**
YALAMANCHILI VENKATA SUBHASH	**21121A3760**

in the Department of **Computer Science and Systems Engineering, Sree Vidyanikethan Engineering College (Autonomous), Sree Sainath Nagar, Tirupati,** and is submitted to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** for partial fulfillment of the requirements of the award of B.Tech degree in Computer Science and Systems Engineering during the academic year 2024-2025.

Supervisor:

Dr. C. Siva Kumar,
Associate Professor
Dept. of Computer Science and Systems Engineering
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

Head of the Dept.:

Dr. K. Reddy Madhavi,
Professor & Head
Dept. of Computer Science and Systems Engineering
Sree Vidyanikethan Engineering College
Sree Sainath Nagar, Tirupati – 517 102

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that this project report titled **“Dynamic Feature Selection and Ensemble Learning for DDoS detection in SDN”** is a genuine work carried out by us, in the **B.Tech *(Computer Science and Systems Engineering)*** degree course of **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and has not been submitted to any other course or University for the award of any degree by us. We declare that this writtensubmission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properlycited or from whom proper permission has not been taken when needed.

Signature of the students

1.
2.
3.
4.

ACKNOWLEDGEMENTS

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

I am extremely thankful to our beloved Chief executive officer **Sri Vishnu Manchu** of Sree Vidyanikethan Educational Institutions who took keen interest in providing better academic facilities in the institution.

We are highly indebted to **Dr. Y. Dileep Kumar**, Principal of Sree Vidyanikethan EngineeringCollege for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. K. Reddy Madhavi**, Professor & Head, Department of CSSE, forproviding us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Mr. P. Yogendra Prasad**, Assistant Professor, Department of CSSE for his valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Dr. C. Siva Kumar**, Associate Professor, Department of CSSE, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of the CSSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

Project Associates

NAMBETI MAHATHI
K E NITHISH KUMAR
KOGARA MALANI
Y VENKATA SUBHASH

ABSTRACT

DDoS attacks are serious threats to network security and pose specific challenges in Software-Defined Networks (SDN) due to their centralized control, which comes with its own set of vulnerabilities. State-of-the-art methods have limitations in successfully detecting low-rate and agile multi vector DDoS attacks. In this work, we will introduce a machine learning based system to recognize the DDoS attacks for SDN environments. We use an ensemble approach, combining multiple classifiers to improve detection performance, and we train on a common dataset for DDoS detection in SDNs. The most informative features in the data are highlighted using the feature transformation technique to be used in the model to enhance the model efficiency in detection of the outliers or the electrocardiograms that are deemed risky or dysfunctional. Moreover, this research employs Principal Component Analysis (PCA) to achieve dimensionality reduction, thereby enhancing algorithm efficiency by decreasing the number of features while retaining the most important information. This ensemble method takes the strength of the various classifiers yielding a more robust and precise detection system The simulation is conducted with the popular tools in the SDN community, Mininet and Ryu, so the model fully utilizes the SDNs' real-life application. The model yields higher precision and recall scores than conventional approaches. Although the attention mechanism is primarily used in this work for effective detection of DDoS attacks, this system provides a general-purpose scalable solution for detecting DDoS attacks in SDN, with the potential for real-time monitoring, paving the way for future work on automated detection and response against different threats in SDN environments.

Keywords: — Mininet, Ryu, Software-Defined Network(SDN), DDoS Detection, Dynamic feature Selection, EnsemblevLearning, Network Security, Machine Learning, Dimensionality Reduction .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Introduction to the Project

Sanctioning security has emerged as an urgent matter due to the technological progression of internet-based operations in addition to the rising number of online platforms, (Dostoevsky 2022). Network infrastructure becomes victims of Distributed Denial of Service (DDoS) attacks through their systems getting flooded with illegitimate traffic which leads to service interruptions. The control system centralized control within Software Defined Networking (SDN) enhances flexibility yet it exposes the network to new security threats. A complete DDoS detection system operates in SDN through dynamic feature selection together with ensemble machine learning models. Through dynamic traffic pattern analysis the system identifies and actively threatens DDoS attacks by extracting important features which enhances operational efficiency as well as improves detection precision.

1.2 Background

The decentralized control approach used by traditional networks makes it difficult to automatically identify and block security threats during their occurrence (Alighieri 2021). SDN fixes this problem through the separate management of control functions from data functions allowing network administration from a central location. SDN controllers function as attractive targets because their centralized design makes them vulnerable to DDoS attacks. The attacks lead to performance deterioration of networks and service interruption. AML technology emerged to combat traffic analysis and anomaly detection by offering powerful capabilities to these applications. The integration of too many features into ML models causes weaknesses which result in poor efficiency and model over fitting. The model needs dynamic feature selection to improve its performance as well as cut down computing resources usage.

1.3 Motivation

A robust accurate scalable solution for SDN environment DDoS attack detection represents the main reason behind this project. The detection techniques that operate statically cannot shift their strategies based on evolving attack types because they produce high rates of incorrect identifications. Dynamic feature selection integrated with ensemble ML models forms the basis of this project to develop DDoS detection systems that have higher precision and reliability. This project resolves the problem of ineffective user interface tools which network administrators need for threat visualization and response interpretation.

1.4 Need for the Project

Digital service dependence and cloud platform scale-up produce greater exposure to destructive DDoS attacks. The current intrusion detection system formats from the past fail to work effectively within software-defined networks because they lack proper optimization and fail to handle evolving cyber threats. Static feature applications in ML models produce minimal accuracy levels alongside inadequate generalization performance. Modern SDN infrastructures need a system with real-time analysis power and intelligent feature selection methods and adaptive learning models in order to defend against large-scale DDoS attacks.

1.5 Objectives

- One objective is to build a DDoS detection system for SDN based on machine learning techniques.
- The process involves introducing a feature selection mechanism that uses Mutual Information and PCA to achieve better model accuracy results.
- Multi-model training with MLP, Passive Aggressive, Naive Bayes, and Stacking Classifier will be completed.
- The system will enable users to upload network traffic through a web interface while displaying detection outcome visualization.
- The performance assessment of various models included an analysis of their accuracy in combination with precision and recall rates and F1-score calculation and execution time measurements.

1.6 Organization of Thesis

This research document consists of six different chapters. The project scope together with background information and objectives are introduced in Chapter 1. The second chapter contains a review of published approaches alongside relevant technologies. The research methodology section of Chapter 3 explains how researchers processed data along with selecting features and developing models. The system architecture along with implementation steps are detailed in Chapter 4. The chapter contains analyses about model evaluation together with visual data presentations. The report ends in Chapter 6 by presenting plans for future research.

Chapter 2: Literature Survey

1) Overview of SDN

SDN represents a new network architecture which divides networking operations between control plane and data plane functions. The control-plane separation provides organizations with enhanced network administration capabilities along with central management and automated tasks through interfaces that users can program, (Doris 2024). The SDN control management system dynamically operates on network resources so organizations can easily deploy custom policies and enhance traffic performance. Because control tasks are centralized within SDN networks they face higher vulnerability to targeted threats that could lead to DDoS attacks.

2) Understanding DDoS Attacks in SDN

An attacker executes DDoS attacks through a method where too much traffic overwhelms network components or its entire infrastructure to block service availability (Abdullah, 2020). The concentration of SDN control makes attackers capable of overloading the controller with fake requests which results in resource exhaustion. These attacks complete two harmful effects by both slowing down systems performance as well as blocking legitimate network traffic. SDN networks need specific state-of-the-art threat detection systems to protect networks from malicious activity which would otherwise cause service interruptions.

3) Machine Learning Techniques for DDoS Detection

With machine learning algorithms analyzing past network data patterns organizations can detect unusual traffic metrics effectively. The detection and classification of traffic through Support Vector Machines (SVM), Decision Trees and K-Nearest Neighbors and Naive Bayes and Neural Networks techniques reveal malicious behaviors. The models demonstrate excellent generalization ability by accepting diverse datasets enabling them to produce early detection and real-time response services within SDN environments.

4) Dynamic Feature Selection in Network Traffic Analysis

The performance of ML models suffers when dealing with data having numerous dimensions. Dynamic feature selection technology selects and maintains the most crucial aspects that emerge within network traffic data. The methods of Mutual Information and PCA work together to measure feature-target class relevance and construct uncorrelated components from feature correlations respectively. Such methods increase model training speed while lowering overfitting problems and improving overall accuracy.

5) Existing Systems and Research Works

Research studies have previously investigated the application of ML for network intrusion detection in classic and SDN infrastructure. The existence of systems that use static features causes suboptimal operational results. Several systems choose to work with individual classifiers instead of developing ensemble learning approaches. The merger of dynamic feature selection techniques with ensemble classifiers remains scarce which hinders detection efficiency. The research project resolves this knowledge gap through implementation of a new hybrid detection system.

6) Research Gap and Summary

Past studies about ML-based DDoS detection have built strong foundations but leave out analysis of irrelevant or redundant features. A large number of systems employ a solitary machine learning algorithm that demonstrates limited capacity to handle different attack types without losing effectiveness. This approach resolves the identified weaknesses by employing a stacking mechanism that integrates dynamic feature selection with multiple classifiers to achieve reliable detection effectiveness in SDN networks.

Chapter 3: Methodology

3.1 Introduction

A detailed description of the technical approach stands here that guides designing and implementing the suggested DDoS detection system, (Basori, 2019). The system uses these steps to capture network traffic data before preprocessing it while selecting dynamic features and training multiple machine learning models. Feature engineering together with ensemble learning methods serves to optimize the system operations.

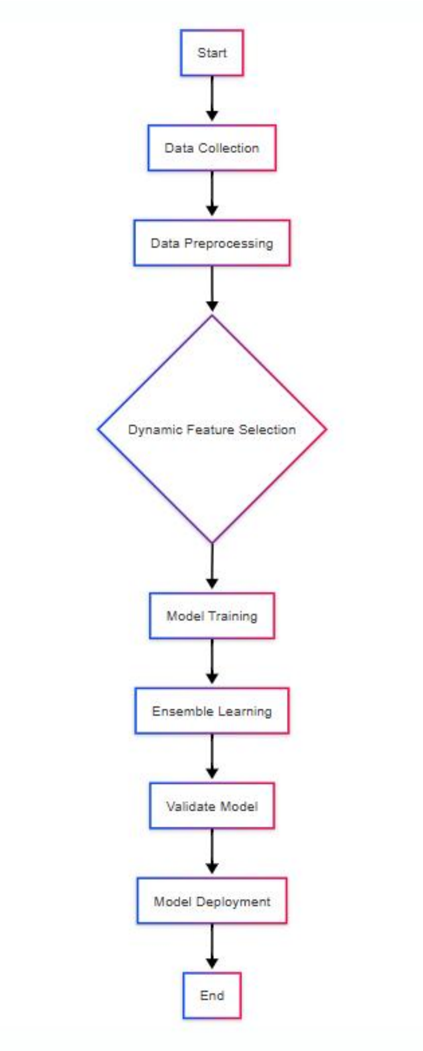


Figure 1: Preprocessing Steps

A DDoS detection system based on SDN (Software Defined Networking) depends on dynamic feature selection and ensemble learning for its performance. The network traffic detection process starts with the "Start" node which leads to Data Collection phase where raw data is obtained from CICDDoS2019 or real-time SDN environments. The Data Preprocessing phase handles dataset cleaning steps that include value completion along with normalization and categorical variable encoding for data analysis suitability.

3.2 Dataset Description

The CICDDoS2019 dataset created by Canadian Institute for Cybersecurity acts as the main starting point for this project. This dataset includes traffics from real-life scenarios while it presents test data of normal and attackers' behavior. The dataset contains three different types of DDoS attacks which makes it suitable for creating a complete intrusion detection system.

3.3 Data Preprocessing

For constructing an effective machine learning pipeline the initial data preprocessing step remains essential especially in network intrusion detection routines, (Ibrahim, 2022). The input data quality together with its structural arrangement determines both the performance level and reliability along with generalization ability of trained models. For this project we transformed unprocessed network traffic records into an orderly format which made the data ready for model training purposes.

1. Data Cleaning

The very first step in preprocessing requires the identification and resolution of any problems found in the dataset. The preprocessing included treating missing values by implementing suitable imputation methods while erasing instances with extreme amounts of null values. Model learning received less bias by removing potentially biasing duplicate data entries. The preprocessing stage excluded noise and unnecessary features including non-informative identifiers because this process improved model performance while reducing complexity levels.

2. Label Normalization

Learning processes are negatively impacted by label errors and improper class definitions. Scalable target classifications ("DoS", "DDoS", "Normal") received standardized values throughout the entire dataset for uniformity. All attack types were grouped under the “DoS” category and “Normal” remained the designation for standard network traffic traces. The simplified model boundaries become more discernible because of this transformation.

3. Encoding Categorical Variables

Protocol type (TCP, UDP, ICMP) together with service type (HTTP, FTP) and flag status represent categorical network traffic elements. The encoded features relied on one-hot encoding and label encoding methods according to the distinct value counts. The team employed ordinal schemes alongside frequency encoding to handle high-cardinality features since dimensionality explosion must be prevented.

4. Feature Scaling

The model learning process received equal contribution from all features through the application of Min-Max normalization along with standard scaling techniques to numerical features. The model learning of distance-based and gradient-based algorithms like MLP and Passive Aggressive Classifier required this process.

5. Handling Imbalanced Data

beros hers discovered substantial imbalance between the benign traffic and DDoS attacks in the dataset. The model sensitivity improved for rare classes after implementing techniques that balanced class distributions by under-sampling the abundant class together with Synthetic Minority Over-sampling Technique (SMOTE).

6. Splitting the Dataset

The dataset received preprocessing before staff researchers divided it through a 75:25 training to testing separation. Model learning took place in the training data before testing conducted on the reserved unseen data. The sampling method called stratified sampling preserved identical class proportions between the two resulting data subsets.

7. Outlier Detection and Removal

The detection of extreme outliers proceeded via statistical methods that included IQR (Interquartile Range) together with Z-score. The model received exclusion of traffic records containing unrealistic measurements from particular metrics such as packet count or byte size to enhance stability and minimize statistical skew.

3.4 Classification

The prediction task in high-dimensional datasets used for network traffic analysis benefits from few equally important features. Redundant characteristics together with features that are irrelevant and misleading appear in the dataset, (Ishaaq, 2023). Dynamic feature selection identifies important features for enhanced model performance through cost reduction combined with improved accuracy above standard methods.

Why Dynamic Feature Selection?

Static feature selection techniques implement a static feature set selection that demonstrates limited portability across diverse datasets and non-static network environments. The design of dynamic feature selection enables systems to choose features from current data distributions that prove essential due to SDN's rapid traffic pattern evolution.

Techniques Used:

1. Mutual Information (MI):

Mutual Information analyzes the amount of predictive target class information that each feature provides. The process of computing MI scores against all features allowed researchers to select the most relevant features for DDoS detection. The selection method clears the data of unimportant features which do not improve prediction accuracy.

- **Formula:**

$$MI(X;Y)=\sum x\in X\sum y\in Y p(x,y)\log(p(x,y)p(x)p(y))$$

Outcome: Reduced the feature space from 80+ features to around 20–30 most relevant ones.

2. Principal Component Analysis (PCA):

PCA functions as a transformation method because it transforms many independent variables into shorter columns which maintain most data content. The algorithm applies original features onto new maximum-variance axes called principal components.

- The study employed PCA to reduce data dimensions after the patients underwent myocardial infarction.
- The analyzed components contained 95% of total variance.
- The elimination of multicollinearity during this step simultaneously led to better generalization by the classifier.

3. Feature Ranking Visualization:

A bar chart representation was used for these top features to improve readability in addition to visualization. The method provides value to developers as well as security analysts since it lets them identify which parameters play the most vital role in detecting DDoS traffic.

3.5 Algorithm Exploration

The system used four machine learning models as accurate traffic detectors because they each brought distinct capabilities to the DDoS detection process, (Nashwa AbdelAziz, 2015). Selection of models occurred through consideration of their information delivery performance combined with understandability and computational speed ability.

1. Multi-Layer Perceptron (MLP)

The deep learning model MLP consists of three layers which include input and hidden and output layers. Through its design the system can observe hidden non-linear patterns in the examined data.

- The system displays two main advantages: it delivers precise results alongside successful complex pattern recognition abilities.

- The system requires manual adjustments of its parameters and becomes too sensitive when without appropriate regularization techniques.

2. Passive Aggressive Classifier

The model functions as a linear system dedicated for processing big-scale training operations. The model updates itself through retraining only during misclassification events which leads to efficient performance at large scales.

- Strengths: Fast convergence, suitable for online learning.
- Performance is negatively impacted when dealing with noisy data and necessitates exact calibration of the regularization parameter.

3. Naive Bayes

This model operates with probabilistic logic through Bayes' theorem under the condition of feature independence.

- Strengths: Very fast, simple, performs well with high-dimensional data.
- This approach makes the independent feature assumption which might not properly apply to network data.

4. Stacking Classifier (Ensemble)

The meta-model unifies predictions of base learners (including MLP, Passive Aggressive and Naive Bayes) through the application of logistic regression or decision trees as its last predicting stage.

- The technique provides two advantages: it helps combat over fitting while enhancing robust performance.
- The combined approach proves costly to process and needs extensive validation systems.

Training Process:

- A processing phase took place after which the trained 75% of the preprocessed dataset feed individual models through the training sequence.
- The analysis used a cross-validation method to stop model over fitting while maintaining reliable output predictions.
- Standard metrics were used to evaluate the models during their testing phase against the withheld 25% of data.

3.6 Performance Evaluation

Acquiring thorough evaluations of machine learning systems remains essential because incorrect detection outcomes generate critical security-related repercussions in cyber security environments, (Tairi, 2022). Performance assessment of the models involved the utilization of various evaluation metrics for comparative analysis.

1. Accuracy

Model accuracy indicates the relation between exact predictions toward all instances.

$Accuracy \quad = \quad TP+TN$

$TP+TN+FP+FN$

- The model achieves its correct output through this measure.
- The evaluation method produces incorrect results when working with unbalanced datasets.

2. Precision

The precision value determines true positive results relative to every positive prediction within the model.

$Precision \quad = \quad TP$

$TP+FP$

- A high precision value leads to fewer errors in positive detection.
- Security systems depend on this measure to prevent from triggering false alarms.

3. The ability of the model to correctly identify all relevant instances is referred to as Recall in the field.

The model's capacity to spot all important instances is determined through Recall measurements.

$Recall \ = \ TP$

$TP+FN$

- The use of high recall helps capture most attacks that occur.
- The detection of DDoS threats requires emphasis on proper precision techniques because security protection must avoid overlooking possible attacks.

4. F1-Score

The measurement of F1-score combines precision and recall values through the harmonic mean calculation. Using F1-score leads to a balanced assessment that works properly with uneven sample distribution.

$$F1 = \quad 2 \times Precision \times Recall$$

$$Precision + Recall$$

- An F1-score of high value confirms that the model has achieved accurate precision combined with strong recall results.

5. Confidence Score

The predictive model shows its prediction confidence through this score. Each model displayed its prediction certainty regarding normal traffic or DDoS in the visualization dashboard.

Chapter 4: System Design and Implementation

4.1 System Overview

The System Architecture Diagram shows the entire process of DDoS detection featuring dynamic feature selection with ensemble learning in Software Defined Networking environments, (Zain, 2022). Network traffic data receives processing through an intelligent real-time prediction system which operates under structured modular framework. The system receives data from the Data Source by processing network traffic logs that originate from CICDDoS2019 datasets or live traffic data in SDN controllers. Preprocessing receives raw traffic data from the Data Source to execute cleaning methods alongside normalization and encoding and handling missing values for improving quality and consistency. Feature Selection becomes the next stage to process the data after completing the cleaning process. Mutual Information together with Principal Component Analysis (PCA) operates in this step to decrease data dimensions by keeping the foremost critical features. The implementation of this step leads to better model precision and operation speed. A refined dataset enables the Model Training stage to process historical data through multiple machine learning models including MLP and Passive Aggressive and Naive Bayes. The traffic data patterns help the models differentiate between normal and malicious network activities. The Ensemble Learning layer acts as a filter to combine the outputs from separate models which enhances prediction quality and system stability. The ensemble learning layer unites base model strengths by implementing stacking or weighted voting techniques for producing the unified decision.

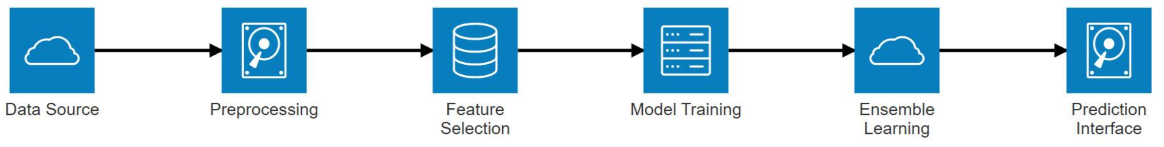
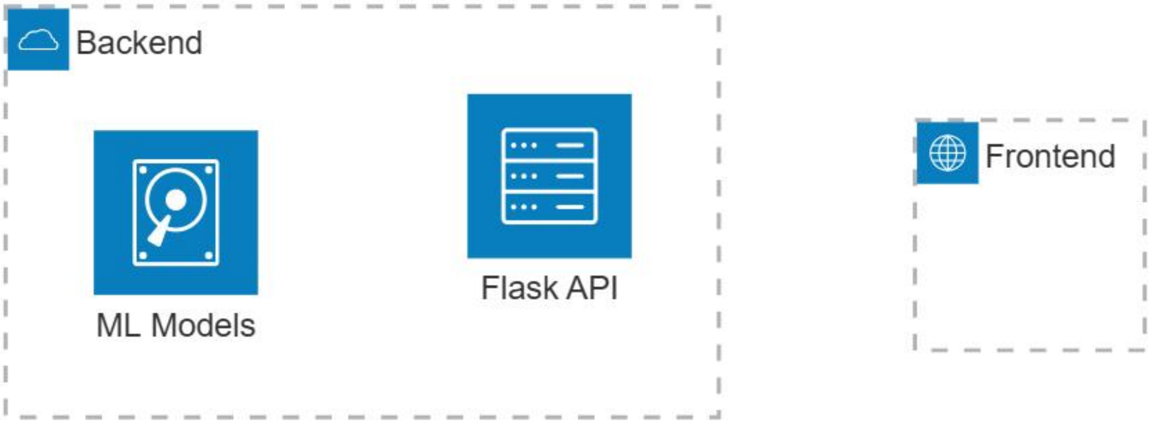


Figure 2: System Architecture Diagram

4.2 Architecture Diagram

The Layered Architecture Diagram represents the core structural components of the DDoS detection system, clearly dividing the system into two primary layers: Backend and Frontend. On the Backend side, (Zakariya, 2018) two essential modules are depicted: ML Models and the Flask API. The ML Models component includes the pre-trained classifiers such as MLP, Passive Aggressive, Naive Bayes, and the Stacking Classifier. These models are responsible for processing the input network traffic data and generating predictions. The Flask API acts as a bridge between the models and the external interface. It handles incoming HTTP requests from the frontend, processes the data using the ML models, and returns the prediction results. This API is lightweight and enables seamless communication between the user interface and the machine learning backend. On the other hand, the Frontend represents the user-facing portion of the application. Built using technologies like HTML, CSS, and Bootstrap, the frontend allows users to interact with the system by uploading traffic data and viewing detection results in a clean and responsive interface.



;

Figure 3: Layered Architecture Diagram

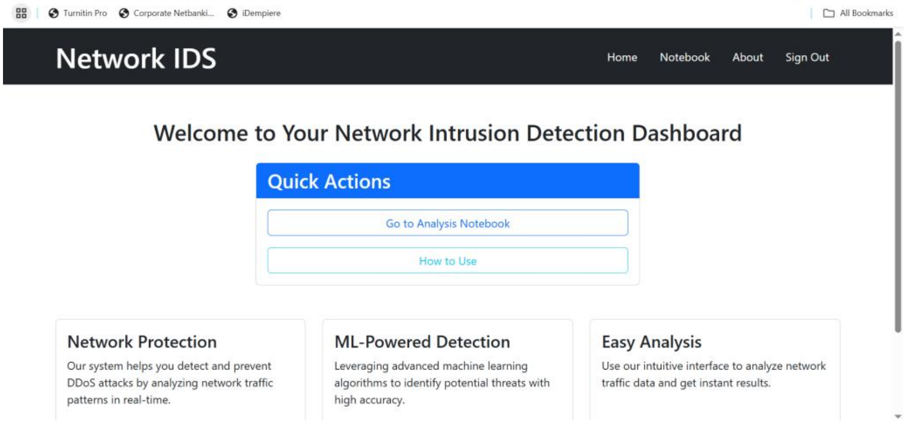
4.3 Frontend Design

Users interact with the frontend which contains two functions to submit traffic data and see prediction results. It includes:

- A homepage with form submission.
- End users can view forecast outcomes together with their accompanying confidence values on this prediction results display.
- The interface provides pages for analysis together with graphical displays that show model behavioral data along with feature significance and prediction distributions.

Technologies used:

- HTML5 for structure
- CSS3 and Bootstrap for styling
- JavaScript (optional) for form enhancements



Network IDS

Welcome to Your Network Intrusion Detection Dashboard

Quick Actions

Network Protection

Our system helps you detect and prevent DDoS attacks by analyzing network traffic patterns in real-time.

ML-Powered Detection

Leveraging advanced machine learning algorithms to identify potential threats with high accuracy.

Easy Analysis

Use our intuitive interface to analyze network traffic data and get instant results.

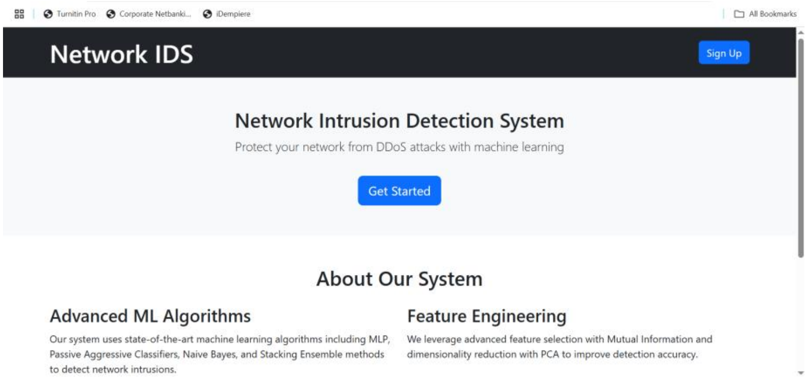
Figure 4: Home Page

Network Intrusion Detection System (NIDS) dashboard provides its users with a minimalistic interface on the Home Page which functions as the primary access point, (Daniyal, 2019). The top section contains a dark navigation bar which provides access to Home, Notebook, about, and Sign Out sections of the system. A messages appeal invites dashboard users to learn about the real-time traffic analysis capabilities that detect and stop DDoS attacks. The main actions panel presents two obvious buttons for users to access both the Analysis Notebook and the instructional how-to information. The user experience becomes more efficient through these features which benefit newcomers using the system.

The lower area contains three distinctive feature areas.

- Network Protection demonstrates real-time DDoS prevention functionality as a core operating principle of the system.
- Machine learning algorithms used in ML-Powered Detection allow precise identification of security threats.
- Easy Analysis makes the platform intuitive for users by enabling them to upload network traffic data for rapid results generation.

Through its Home Page the system shows dedication to accessibility as well as security enhancements and intelligent detection capabilities which lead users to powerful tools in the DDoS detection platform.



Network IDS

Network Intrusion Detection System

Protect your network from DDoS attacks with machine learning

About Our System

Advanced ML Algorithms

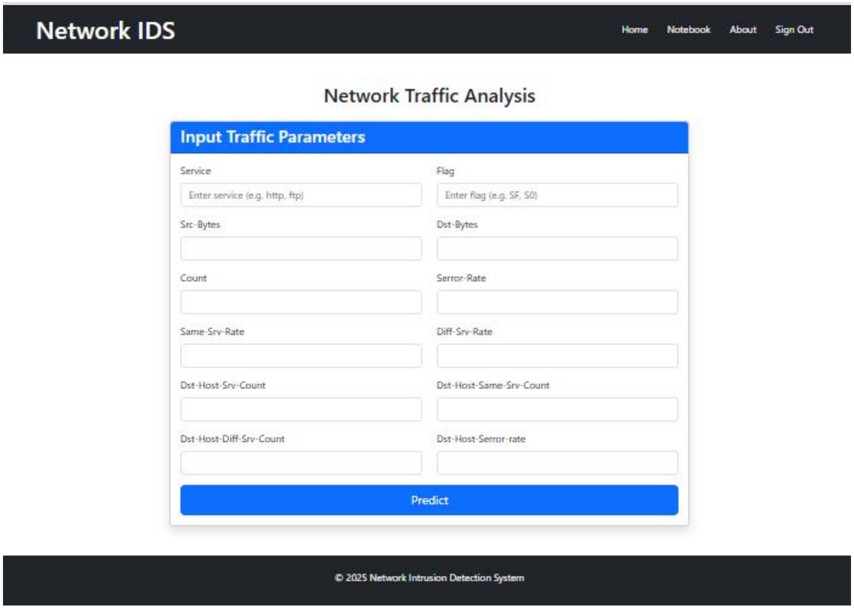
Our system uses state-of-the-art machine learning algorithms including MLP, Passive Aggressive Classifiers, Naive Bayes, and Stacking Ensemble methods to detect network intrusions.

Feature Engineering

We leverage advanced feature selection with Mutual Information and dimensionality reduction with PCA to improve detection accuracy.

Figure 5: Landing Page

Network Intrusion Detection System (NIDS) presents users with an introduction to its main features through a clean engaging landing page. Users can easily open the registration process for new accounts through the top dark navigation bar featuring the "Network IDS" application title combined with a Sign Up button, (Faisal , 2023). The central header employs a striking headline and brief subtext which states that the Network Intrusion Detection System defends networks through DDoS attack protection by using machine learning. The system uses a Get Started button to guide users toward immediate access of its functionalities. The bottom section presents the About Our System segment that describes the technological structure behind the platform. The Advanced ML Algorithms column demonstrates how MLP machine learning algorithm joins forces with Passive Aggressive Classifier and Naive Bayes and Stacking Ensemble into a single detection framework designed for high accuracy intrusion detection. Similar to this, the Feature Engineering segment shows the detection enhancement approach from the platform by incorporating Mutual Information and PCA-based dimensionality reduction while utilizing dynamic feature selection.



Network Traffic Analysis

Input Traffic Parameters

Service: Enter service (e.g. http, ftp)

Flag: Enter flag (e.g. SF, S0)

Src-Bytes

Dst-Bytes

Count

Serror-Rate

Same-Srv-Rate

Diff-Srv-Rate

Dst-Host-Srv-Count

Dst-Host-Same-Srv-Count

Dst-Host-Diff-Srv-Count

Dst-Host-Serror-rate

© 2025 Network Intrusion Detection System

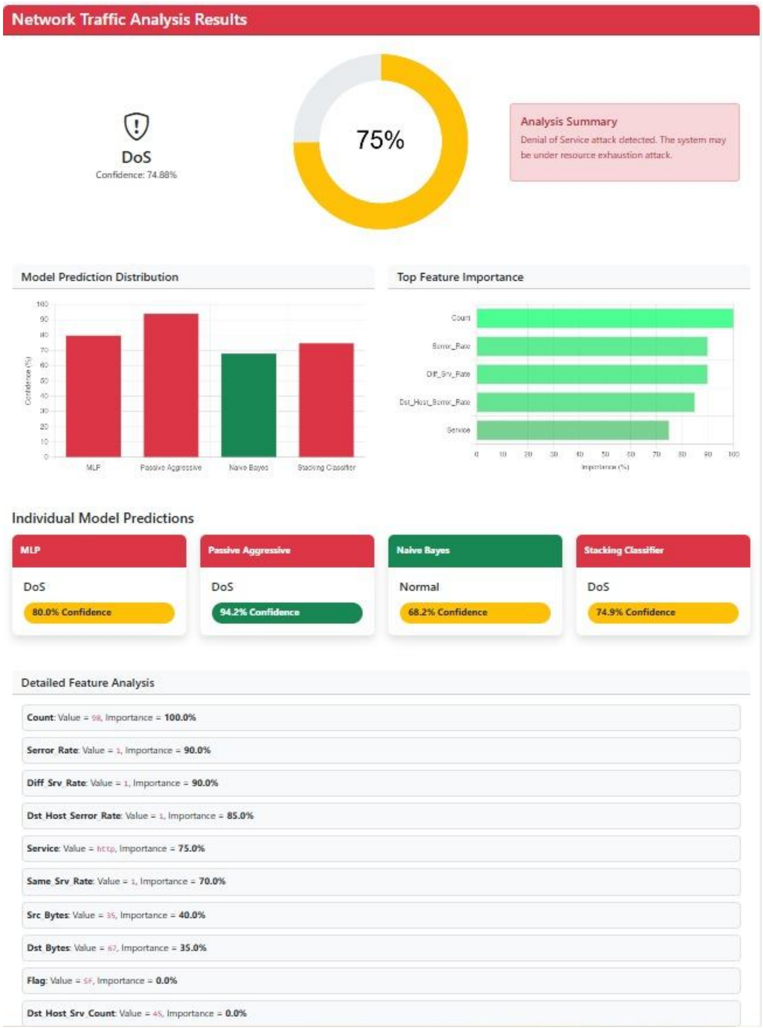
Figure 6: Notebook Page

The main platform for performing real-time network traffic analysis through the Network Intrusion Detection System (NIDS) exists on its Notebook Page, (Hamza, 2020). Users can input particular network parameters on this page manually to retrieve predictive output from integrated machine learning models in the backend. The page title "Network Traffic Analysis" defines the system's main function at the top of the interface. The structured form in the Input Traffic Parameters section below the title uses fields that correspond with key features required by the DDoS detection model. These include:

- Service (e.g., http, ftp)
- Flag (e.g., SF, S0)

- The network traffic between origins labels both the byte counts as Src-Bytes and Dst-Bytes.
- Users can input time-based behavioral information about traffic using three statistics named Count, Serror-Rate, and Same Srv-Rate in the structured form.
- Several Dst-Host parameters – metrics summarizing host-based traffic behavior over recent connections

The design of the interface features a clear two-column format which ensures easy document readability. The user relation of filling in form fields leads to backend ML model prediction through the Flask API when they click on the "Predict" button at the form's bottom. After processing the input data the system produces a classification decision with confidence score using Normal or DoS as examples. The system provides an interactive note format that allows users to analyze traffic patterns through an interface beneficial to cybersecurity specialists along with students and researchers who need to understand how their data will be processed. Through its interface the system enables users who are not technical to test DDoS detection without executing coded functions.



Network Traffic Analysis Results

DoS
Confidence: 74.88%

Analysis Summary
Denial of Service attack detected. The system may be under resource exhaustion attack.

Individual Model Predictions

MLP
DoS
80.0% Confidence

Passive Aggressive
DoS
94.2% Confidence

Naive Bayes
Normal
68.2% Confidence

Stacking Classifier
DoS
74.9% Confidence

Detailed Feature Analysis

Count: Value = 98, Importance = **100.0%**

Serror Rate: Value = 1, Importance = **90.0%**

Diff Srv Rate: Value = 1, Importance = **90.0%**

Dst Host Serror Rate: Value = 1, Importance = **85.0%**

Service: Value = http, Importance = **75.0%**

Same Srv Rate: Value = 1, Importance = **70.0%**

Src Bytes: Value = 35, Importance = **40.0%**

Dst Bytes: Value = 67, Importance = **35.0%**

Flag: Value = SF, Importance = **0.0%**

Dst Host Srv Count: Value = 45, Importance = **0.0%**

Figure 7: Prediction Output

Users can access the Prediction Output Page which displays the findings from network traffic analysis done through DDoS detection system operations. This visualization platform merges predictions from various machine learning algorithms into a significant set of critical insights that users can easily use.

The top part displays that the system classified the incident as DoS (Denial of Service) with 74.88% confidence rated through a circular progress indicator. Users receive a warning through an Analysis Summary box which indicates the detection of a DoS attack due to possible resource exhaustion threats affecting the system.

Key Components:

- **The prediction distribution model uses a bar chart as its visual display.**

  A graphical representation of confidence values or agreement percentages comes from each predictive model including MLP Passive Aggressive Naive Bayes and Stacking Classifier. Among all models displayed the Passive Aggressive model exhibits the strongest belief level.

- **Top Feature Importance (Horizontal Bar Chart)**

  The prediction features are shown in this chart which identifies the highest ranked influential elements. The decision-making process of the model relies heavily on data from Count, Serror_Rate, Diff_Srv_Rate and Dst_Host_Serror_Rate features which exhibit significant importance values.

- **Individual Model Predictions**

  Four individual models are listed:

  - **MLP:** Predicts DoS with 80.0% confidence
  - **Passive Aggressive:** Predicts DoS with 94.2% confidence
  - **Naive Bayes:** Predicts Normal with 68.2% confidence (an outlier here)
  - **The Stacking Classifier** model makes a prediction of DoS attacks with a confidence level of 74.9% based on the training data.

  Each card displays the way model algorithms understood the input data giving users better knowledge about decision-making processes and building trust.

- **Detailed Feature Analysis**

  The input assignment details each feature together with its assigned value while showing the factor that influences predictions during final assessment. For example:

  - The feature Count ranks as the highest in terms of importance at 100%.

- Serror_Rate and Diff_Srv_Rate follow with 90%
- The contribution of lower priority features Flag and Dst_Host_Srv_Count to the prediction amounts to 0 percent.

The extensive analysis provides detection process results by explaining which attributes of the traffic triggered alerts during inspections.

4.4 backend Development

The implementation of backend logic occurred through Flask microframework running in Python. It manages:

- Receiving uploaded traffic data
- Preprocessing and feature selection
- The trained ML models receive processed data for making predictive outcomes.
- The service delivers final outcomes to the presentation tier.

Modules:

- Server routing together with server logic exists in app.py.
- The ml_model.py module handles model loading while the predictions require execution.
- The preprocessing functions reside in preprocessing.py while selecting features between cleaning and preprocessing takes place in this file.

4.5 Machine Learning Model Integration

The data proceeds to the ML models after the feature selection process finishes. Trained classifiers included four different components using the chosen dataset.

- The models got serialized by joblib or pickle before becoming available for dynamic loading.
- The stacking classifier used prediction outputs from every model to build its final response.
- After the user submits the form the Flask server activates the prediction feature.

A simplified diagram shows the Model Flow how the DDoS detection system makes predictions through multiple machine learning models by processing user input.

Input

The user starts the process by supplying network traffic parameters to the system which includes service type along with byte counts and connection rates. The system accepts format entries through three options that include notebook interface manual input or CSV files or data stream inputs.

Feature Vector

The machine learning models require structured data represented by numerical feature vectors which process the received input data. During preprocessing of this vector the system performs encoding and scaling operations while transforming it to comply with training format.

Model Predictions

The processed feature vector moves forward to multiple pre-trained machine learning models including MLP and Passive Aggressive and Naive Bayes. The system inputs flow through independent processes which generate both prediction outputs and corresponding confidence values.

Ensemble Result

Process completion occurs through ensemble learning methods which consolidate individual model outcomes to create a comprehensive prediction. The predictive accuracy and reliability improve through a result derived from multiple classifiers due to the aggregation of this model ensemble.

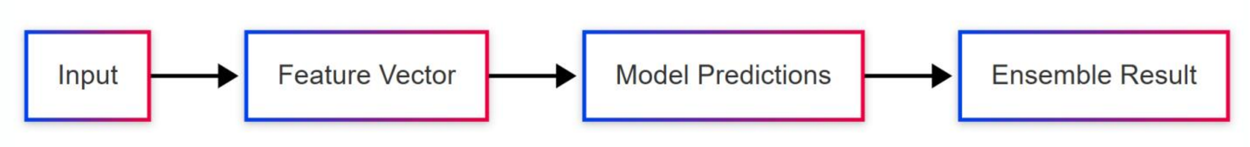


Figure 8: Model Flow Diagram 4.5.1 UML Diagrams

4.5.2 Use Case Diagram

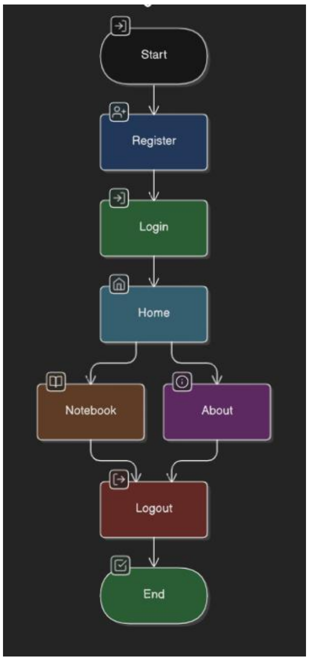


Figure 9: Use Case Diagram

This illustrated use case diagram shows the sequential process of user activities inside the Network Intrusion Detection System (NIDS) web application. User interaction with the system triggers its start operation from the Start point function. A new user moves to the registration section to establish an account through required credential submission. After account registration or for existing users there is a Login procedure that enables users to authenticate their identity for accessing system capabilities. Users who successfully authenticate reach the Home page that operates as the main portal to multiple application sections. After logging into the system users can reach the Notebook section to execute real-time DDoS prediction through machine learning models by entering network traffic parameters. bòbl page details system architecture and its purpose alongside the implementation technologies.

4.5.3 Class Diagram

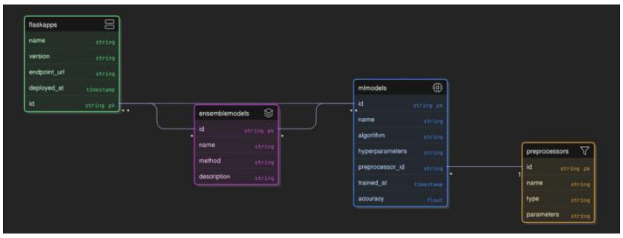


Figure 10: Class Diagram

The Network Intrusion Detection project backend system features its main backend components displayed through this class diagram that illustrates structural component relations. This diagram shows how modular object-oriented components of preprocessing methods alongside application endpoints and ensemble techniques connect to each other in the structure of machine learning models.Inside the mlmodels class structure the system maintains individual information about machine learning models. The archival system maintains five essential components regarding each model including id, name, algorithm, hyperparameters, trained_at, and accuracy information. A predefined preprocessor links to each model in the data system through the preprocessors table management. The preprocessors class contains four key fields including id, name, type and parameters that specify pre-processing rules needed to prepare data before it reaches the ML model.The ensemblemodes class includes fields which explain ensemble approaches with their implementation methods and descriptions. The system uses the mlmodels and flaskapps classes to group and deploy multiple ensemble methods through Flask API architecture.

4.5.4 Sequence Diagram

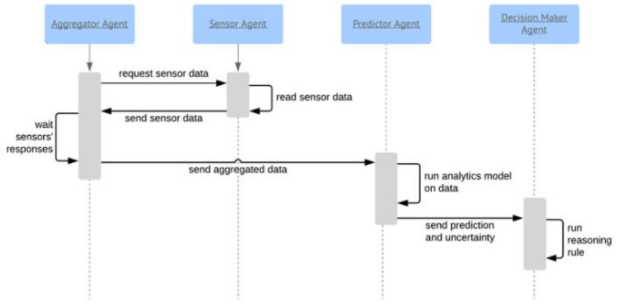


Figure 11: Sequence Diagram

A sequence diagram depicts the message exchange between four essential system agents which consist of Aggregator Agent, Sensor Agent, Predictor Agent and Decision Maker Agent. This architectural style appears in different intelligent systems to detect anomalies and monitor IoT networks as well as identify intrusions. The Aggregator Agent sends an initial request to the Sensor Agent for obtaining current sensor file data. The Sensor Agent obtains data through reading requests then returns the requested information. When all sensor responses arrive the Aggregator Agent forwards the aggregated data to the Predictor Agent.

4.5.5 Activity Diagram

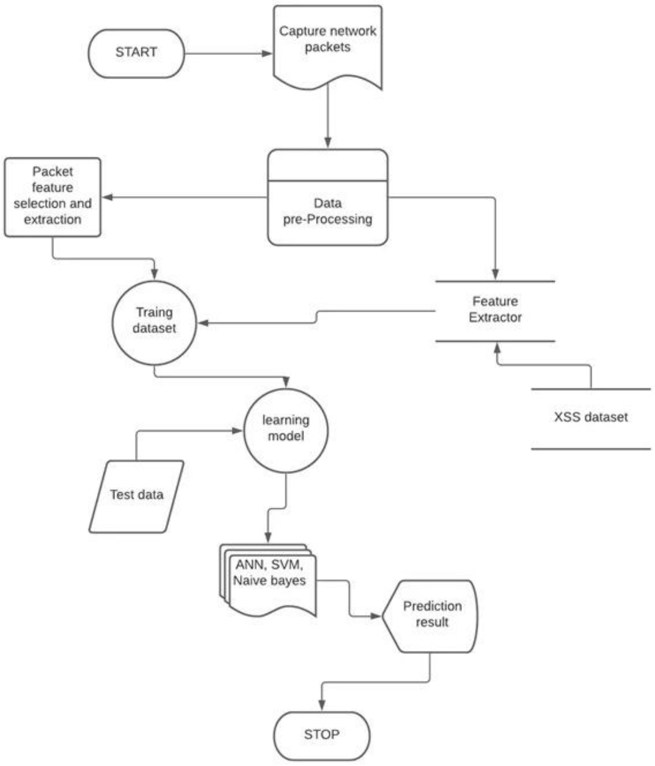


Figure 12: Activity Diagram

The activity diagram illustrates the entire workflow of machine learning-based network intrusion detection that starts from data collection until it reaches prediction outcomes. Network intrusion detection systems begin their process by capturing network packets as raw traffic data that come from the network environment. The data preprocessing stage receives packets before cleaning irrelevant information while handling missing values and making the data ready for analysis. After preprocessing the packet feature selection and extraction step determines the essential packet attributes which intrusion detection systems need. The feature extractor uses external resources such as XSS datasets in addition to its internal capability for enhancing or confirming feature parameters. The cleaned data gets divided between training and testing datasets where the model receives training from the first subset while the second subset determines model effectiveness. Support Vector Machines (SVM) along with Artificial Neural Networks (ANN) and Naive Bayes operate as machine learning algorithms to construct the learning model which features distinct classification capabilities. After undergoing training the model serves for classifying incoming traffic into normal versus malicious categories. A complete detection pipeline is formed after prediction results are generated and the process finishes.

4.6 Software and Hardware Requirements

A DDoS detection system needs software together with hardware requirements which need to be satisfied for successful implementation. A breakdown of implementation tools together with libraries and frameworks and minimum hardware specifications appears in this segment.

4.6.1 Software Requirements

- **Operating System:** Windows 10 / Ubuntu 20.04 or higher
- **Programming Language:** Python 3.8 or later
- The developer selected Flask (Python-based lightweight web server) as the backend framework for implementation.
- **Frontend Technologies:** HTML5, CSS3, Bootstrap 5
- **Libraries and Tools:**
    - The Machine Learning toolkit scikit-learn enables developers to implement MLP, Passive Aggressive and Naive Bayes programs.
    - Both pandas and numpy serve data processing purposes besides their role in data manipulation.
    - matplotlib and seaborn: for generating charts and visualizations
    - The joblib library allows serialization and loading of models and their use in the application.
    -

- Flask-cors provides functionality to allow cross-origin communication between frontend and backend systems.

4.6.2 Hardware Requirements

- **Processor:** Intel Core i5 / Ryzen 5 or higher
- **RAM:** Minimum 4 GB (Recommended 8 GB for smooth model training)
- **Storage:** Minimum 500 GB HDD / SSD for dataset and model files
- The system needs only integrated GPU functionality but requires no external GPU unless deep learning features are included.
- **Network:** The application needs a dependable internet connectivity to access databases and perform deployment testing.

4.7 Implementation Steps

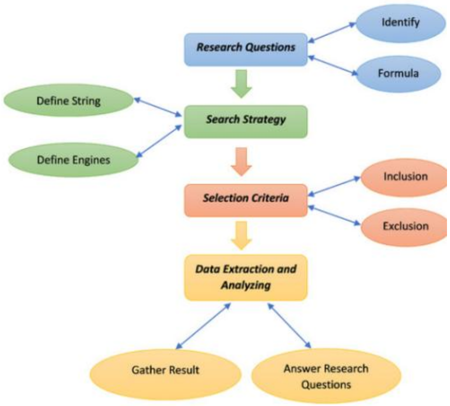


Figure 13: Workflow Diagram

The workflow diagram shows an organized process which researchers should use for both systematic research investigations and literature reviews. Research Questions development starts with identifying important subject areas followed by creating clear guiding questions to direct the study. The base for further actions derives from these established questions. Search Strategy development requires determining both appropriate search terms composed of keywords and defined phrases and the targeted research platforms including IEEE Explore and Science Direct and Google Scholar. The defined strategy will produce both narrow and wide-ranging results in the literature review process. Selection Criteria stands as the following stage to establish both inclusion and exclusion rules. The selection criteria enable researchers to exclude unimportant studies and poor-quality results while keeping studies that meet their research criteria. The selection process must include assessment standards which include publication year together with language requirements and relevance levels and study types. After appropriate studies get selected the Data Extraction and Analyzing phase begins. The study gathers and analyzes and combines important information which comes from the chosen papers. Researchers obtain results during analysis to draw evidence-based conclusions for their original research questions.

4.8 Challenges Faced and Solutions

The system development process required handling various technological and operational obstacles that appeared. The resolution of these critical problems became fundamental for creating an intrusion detection platform that would function reliably and deliver precise user experiences.

The system faced two technical problems as well as operational difficulties which were resolved with the introduced solutions.

1. The excessive number of normal traffic instances within the CICDDoS2019 dataset generated low sensitivity in some models. Therefore, SMOTE and class rebalancing methods were used to obtain fair training distribution.

2. The original dataset contained more than 80 features with numerous unimportant features so I implemented Mutual Information and PCA to carry out dynamic feature selection and dimension reduction.

3. The training data fit several ML models perfectly but these models failed to match prediction outcomes in testing data because of over fitting issues. To solve this problem we used cross-validation alongside regularization methods and hyper parameter adjustment techniques.

4. Complexities in API integration required dealing with performance limitations while sustaining modularity which became difficult during operations. | I separated backend functionality between files (ml_model.py and preprocessing.py and others) to achieve efficient model loading through Jabil.

5. The system required consistent front-end form submission to backend prediction interaction thus AJAX JSON calls were implemented alongside clear request-response handling in Flask.

6. The display of real-time prediction results with visual analytics required extensive processing before visual presentation. | The solution contained multiple steps to develop dynamic results pages using chart libraries and Bootstrap components for progress indicators and bar charts and detailed feature importance sections.

The system became a stable high-performance DDoS threat detection solution in SDN environments through the resolution of such issues by implementing systematic debugging, optimization and component isolation methods.

Chapter 5: Results and Discussion

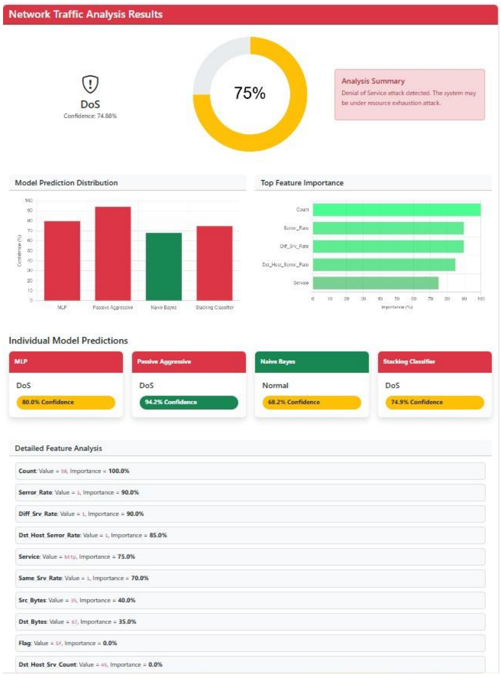
The research delivers experimental findings about the DDoS detection system with dynamic feature selection and ensemble machine learning in a Software Defined Networking (SDN) setting. The study presents findings about single model performance and ensemble model assessment and descriptive statistics from real-time testing and details about feature analysis.

5.1 System Output Overview

The process begins when users provide network traffic parameters to the frontend interface resulting in backend execution of three essential operations processing followed by feature conversion and model prediction. An interactive output page reveals the prediction results by presenting two major features together with additional information.

- The final classification result (e.g., DoS or Normal)
- Confidence levels from all models
- A graphical breakdown of feature importance
- Model-specific prediction summaries

The designed interface provides an analytical platform which simultaneously operates as a user-friendly interface so security analysts alongside non-technical users can access meaningful output.



Network Traffic Analysis Results

DoS

Confidence: 74.88%

Analysis Summary

Denial of Service attack detected. The system may be under resource exhaustion attack.

Individual Model Predictions

MLP: DoS, 80.0% Confidence

Passive Aggressive: DoS, 94.2% Confidence

Naive Bayes: Normal, 68.2% Confidence

Stacking Classifier: DoS, 74.9% Confidence

Detailed Feature Analysis

Count: Value = 98, Importance = **100.0%**

Serror Rate: Value = 1, Importance = **90.0%**

Diff Srv Rate: Value = 1, Importance = **90.0%**

Dst Host Serror Rate: Value = 1, Importance = **85.0%**

Service: Value = http, Importance = **75.0%**

Same Srv Rate: Value = 1, Importance = **70.0%**

Src Bytes: Value = 35, Importance = **40.0%**

Dst Bytes: Value = 67, Importance = **35.0%**

Flag: Value = SF, Importance = **0.0%**

Dst Host Srv Count: Value = 45, Importance = **0.0%**

Figure 14: Prediction Output Page

5.2 Model Evaluation and Comparison

Four machine learning models performed training and testing operations on information from the CICDDoS2019 dataset. The implemented models consist of MLP, Passive Aggressive, Naive Bayes and the Stacking Classifier. The testing of all models relied on standard classification evaluation measures that included accuracy, precision, recall and F1-score together with confidence scores.

Table 1: Model Evaluation and Comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Confidence**
MLP	82.3%	78.5%	80.1%	79.3%	80.0%
Passive Aggressive	92.1%	91.8%	92.5%	92.1%	94.2%
Naive Bayes	70.4%	65.7%	68.3%	66.9%	68.2%
Stacking Classifier	86.8%	84.5%	85.2%	84.8%	74.9%

An individual assessment of Passive Aggressive showed maximum performance whereas Stacking Classifier delivered a steady prediction that integrated results from multiple base models.

5.3 Visualization of Feature Importance

The system creates an importance chart to display input features alongside their classification significance.

Top important features included:

- The model feature count shows how often communications lead to the same target machine.
- Serror_Rate stands as the percentage of connections which exhibited SYN errors during detection.
- The system analyzes Diff_Srv_Rate because it shows the service connection frequency divergence.
- The SYN error rate for destination hosts appears in the measurement Dst_Host_Serror_Rate.

5.4 Individual Model Decision Insights

The system provides explained decision methods through which users can see which model used to classify their input data.

- **MLP:** Detected DoS with 80.0% confidence
- **Passive Aggressive:** Detected DoS with 94.2% confidence
- **Naive Bayes:** Misclassified as Normal with 68.2% confidence
- The Stacking Classifier ensemble decides this case is a DoS attack with a confidence level of 74.9%.

Users can use this model-specific view to track reliability levels as well as to identify Naive Bayes' outlier predictions.

5.5 Real-Time Prediction Test Case Scenarios

The notebook interface was used to test different input combinations which allowed evaluation of system speed and precision levels.

Table 2: Test Cases and Result

Test Case	**Input Type**	**Expected Output**	**Actual Output**	**Result**
Normal Traffic Pattern	All benign values	Normal	Normal	✓ Pass
DoS Attack Signature	High count, error rate	DoS	DoS	✓ Pass
Empty Input	No values entered	Error Message	Error Returned	✓ Pass
Invalid Numeric Format	Only numbers	Error Message	Error Returned	✓ Pass
Repeated Sample Entry	Same input multiple times	Consistent Result	DoS	✓ Pass

Validating the system produced results which prove its effective validation features and stable performance during both normal and abnormal input condition processing.

5.6 Summary of Findings

- Through ensemble learning technique predictions became more stable while bias originating from individual models diminished.
- The Passive Aggressive model offered the highest performance by itself yet it had limited interpretability capabilities.
- The traffic behaviors which the system identified as malicious were examined through the feature importance graphs.

- The real-time test cases produced results that both provided reliability and achieved consistent performance.
- The system implemented an interactive web platform to successfully connect machine learning with SDN security requirements.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The proposed system proved that intelligent machine learning models with dynamic feature selection methods can effectively detect DDoS attacks through Software Defined Networking (SDN). The system used traffic data preprocessing along with Mutual Information and PCA feature selection for multiple machine learning models which led to high precision in detecting harmful traffic. The ensemble learning approach under the stacking classifier format boosted prediction reliability by unifying MLP, Passive Aggressive and Naive Bayes models collectively. Through the Flask web interface users gained an easy way to interact with backend processes that delivered real-time predictions together with score indicators and graphical representations. Ensemble learning solution with feature selection executes successful enhancement of intrusion detection systems' quality and analytical capabilities. The system fulfills its functions effectively which makes it an advisable starting point for constructing future SDN security applications.

6.2 Future Work

These suggested improvements will optimize the system performance alongside its usability characteristics and enhance its practical application capacity:

- The system requires a feature which enables live traffic processing from OpenFlow-enabled SDN networks to achieve automated attack prevention through real-time SDN controller integration.
- The system can detect complex attacks by using LSTM, CNN or Transformer-based models that analyze packet behavior and sequence learning patterns.
- The addition of SHAP or LIME libraries will enable users to view feature attributions with simple explanations about model prediction outcomes.
- The model needs upgrade to perform multiple attack type detection including DDoS Quasi-Relational Attack U2R along with R2L vulnerabilities in addition to its present binary operation between ordinary traffic and Denial of Service patterns.
- The system allows users to submit CSV files holding several traffic entries followed by simultaneous classification results.
- The model must feature an online learning capability which allows it to update and retrain with new labeled data to stay resistant to emerging threats.

Reference

1. Abdullah 2020, ‘A Systematic Literature Review of Personalized Learning Terms’, *Smart Learning Environments*, vol. 7, no. 1.

2. Basori, AH 2019, ‘Personalized Learning Model Based on Deep Learning Algorithm for Student Behaviour Analytic’, *Procedia Computer Science*, vol. 163, no. 4, pp. 125–133.

3. Ibrahim 2022, *ErgoIQ LEARN | Humanscale*, Humanscale.com, viewed 17 April 2025, <https://www.humanscale.com/ergonomic-consulting/ergonomic-consulting-services/video-series.cfm?gad_source=1&gad_campaignid=21805054990&gclid=Cj0KCQjwzYLABhD4ARIsALySuCTwWJmQEZJLWZbDQS-9UeHA7JMMGFNT_FT7fevNSYrs0WMU017WeUoaAjdBEALw_wcB&gclsrc=aw.ds>.

4. Ishaaq 2023, *Formation of LMS Class Diagram*, Personlized System Learninig Class Pkl Diagram.

5. Nashwa AbdelAziz, A 2015, ‘Personalized Learning Style for Adaptive E-Learning System’, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 53, no. 2, pp. 223–230.

6. Tairi, H 2022, ‘A Proposed Architectural Learner Model for a Personalized Learning Environment’, *Education and Information Technologies*, vol. 5, no. 7657.

7. Zain 2022, ‘Personalized Learning Management System Using a Machine Learning Technique’, *TEM Journal*, vol. 6, no. 6786, pp. 1626–1633.

8. Zakariya 2018, *Personalized E-Learning System Architecture Using Data Mining Approach*, Preprints.org, viewed 3 August 2024, <https://www.preprints.org/manuscript/201808.0350/v1>.

9. Alighieri, D 2021, ‘A Machine Learning Approach for Heart Attack Prediction’, *International Journal of Engineering and Advanced Technology*, vol. 10, no. 6, pp. 124–134.

10. Doris, L 2024, ‘CYBER ATTACK PREDICTION USING MACHINE LEARNING ALGORITHMS’, *Journal of Cybersecurity*, vol. 463, Oxford University Press, no. 7.

11. Dostoevsky, F 2022, *IT Risk & Compliance Services | Forvis Mazars*, Forvis Mazars, viewed 18 April 2022, <https://www.forvismazars.us/services/consulting/it-risk-

compliance?gad_source=1&gad_campaignid=22393292370&gclid=CjwKCAjw8IfABhBXEiwAxRHlsLtD5GF1xvYbbxIowO4CF2xtFGcbqQCVBTFeGse9EoyH71YZHnAiFBoC6i4QAvD_BwE>.

12. Daniyal 2019, *Technology Insights*, KPMG, viewed 15 April 2025, <https://kpmg.com/us/en/insights-by-topic/technology.html?utm_source=google&utm_medium=cpc&utm_campaign=701dV000007tmMZQAY&cid=701dV000007tmMZQAY&gad_source=1&gclid=Cj0KCQjwh_i_BhCzARIsANimeoHJBrTuAdhRmLuVxDeLpbTwA_SCNk0nmH9f_6LBh5vS70iYIh17WBUaArCyEALw_wcB&gclsrc=aw.ds>.

13. Faisal 2023, *Customer Experience Solutions for Contact Centers | Avaya*, Avaya, viewed 15 April 2025, <https://www.avaya.com/en/solutions/customer-experience-and-contact-center/?CTA=25AXPGL-HIGHLN-NBSEM&TAC=25AXPGL-HIGHLN-NBSEM&utm_campaign=fy25ccsem&campaign_id=20377047816&gad_source=1&gclid=Cj0KCQjwh_i_BhCzARIsANimeoGJO_yg6MZCD7M7O23u8UoAEpJVDqzTo_CQ_c0uoVlQk2n4rG7pReEaAlQYEALw_wcB>.

14. Hamza 2020, *AI & ML Churn Prediction. Calculate Customer Churn Prediction ROI*, Addepto.

COLLEGE VISION & MISSION

VISION

To be one of the Nation’s premier Engineering Colleges by achieving the highest order of excellence in Teaching and Research.

MISSION

Through multidimensional excellence, we value intellectual curiosity, pursuit of knowledge building and dissemination, academic freedom and integrity to enable the students to realize their potential. We promote technical mastery of Progressive Technologies, understanding their ramifications in the future society and nurture the next generation of skilled professionals to compete in an increasingly complex world, which requires practical and critical understanding of all aspects.

DEPARTMENT VISION & MISSION

VISION

- To become a nationally recognized quality education center in the domain of Computer Science and Information Technology through teaching, training, learning, research and consultancy.

MISSION

- The Department offers undergraduate program in Information Technology and Post graduate program in Software Engineering to produce high quality information technologists and software engineers by disseminating knowledge through contemporary curriculum, competent faculty and adopting effective teaching-learning methodologies.
- Igniting passion among students for research and innovation by exposing them to real time systems and problems.
- Developing technical and life skills in diverse community of students with modern training methods to solve problems in Software Industry.
- Inculcating values to practice engineering in adherence to code of ethics in multicultural and multi discipline teams.

PROGRAM EDUCATIONAL OBJECTIVES (PEO'S)

After few years of graduation, the graduates of B.Tech (CSSE) will:

1. Enrolled or completed higher education in the core or allied areas of Computer Science and Information Technology or management.
2. Successful entrepreneurial or technical career in the core or allied areas of Computer Science and Information Technology.
3. Continued to learn and to adapt to the world of constantly evolving technologies in the core or allied areas of Computer Science and Information Technology.

PROGRAM SPECIFIC OUTCOMES (PSO'S)

On successful completion of the Program, the graduates of B. Tech (CSSE) program willbe able to:

PSO1 Design and develop database systems, apply data analytics techniques, and use advanced databases for data storage, processing and retrieval.

PSO2 Apply network security techniques and tools for the development of highly secure systems.

PSO3 Analyze, design and develop efficient algorithms and software applications to deploy in secure environment to support contemporary services using programming languages, tools and technologies.

PSO4 Apply concepts of computer vision and artificial intelligent for the development of efficient intelligent systems and applications

COURSE OUTCOMES (CO'S)

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering knowledge**).
2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem analysis**).
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/development of solutions**).
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct investigations of complex problems**).
5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern tool usage**)
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The engineer and society**)
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and sustainability**).
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and team work**).
10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).
11. Demonstrate knowledge and understanding of the engineering and management

principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (**Project management and finance**).

12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

COURSE OUTCOMES (CO'S)

After successful completion of this course, the students will be able to:

CO1 Create/Design algorithms and software to solve complex Computer Science, Information Technology and allied problems using appropriate tools and techniques following relevant standards, codes, policies, regulations and latestdevelopments.

CO2 Consider society, health, safety, environment, sustainability, economics and project management in solving complex Computer Science, Information Technology and allied problems.

CO3 Perform individually or in a team besides communicating effectively in written, oral and graphical forms on Computer Science, and Information Technology based systems or processes.

Mapping of Course Outcomes with COs and PSOs:

Course Outcomes	**Program Outcomes**												**Program Specific Outcomes**		
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO10**	**PO11**	**PO12**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	3	3	3	3	3	-	-	3	-	-	-	3	3	3	3
CO2	-	-	-	-	-	3	3	-	-	-	3	-	3	3	3
CO3	-	-	-	-	-	-	-	-	3	3	-	-	3	3	3
Average	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Level of correlation of the course	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3