STM JOURNALS

Journal of
Network Security

https://journals.stmjournals.com/jons

ISSN: 2395-6739 (Online)
ISSN: 2321-8517 (Print)
Volume 13, Issue 1, 2025
January–April
DOI (Journal): 10.37591/JoNS

Review JoNS

Real-time DDoS Attack Prediction in SDN Environments Using Machine Learning

T.N.V.S. Praveen[1], Konakala Jyothi[2], Patchigalla Bala Karthik[2,*], Mudavath Balaji Naik[2]

Abstract

The ever-growing reliance on SDN-based services necessitates robust security measures against Distributed Denial-of-Service (DDoS) attacks that threaten service availability. This project investigates the development of a real-time prediction system for DDoS attacks in SDN environments, leveraging the power of machine learning. The proposed system employs a Decision Tree classification algorithm implemented in Python. To ensure accurate attack identification, the system meticulously addresses data preprocessing challenges inherent in network traffic datasets. These challenges include imbalanced class distributions, where normal traffic significantly outnumbers attack instances, and the presence of categorical features requiring transformation for machine learning algorithms. The system tackles these issues by employing techniques like oversampling to balance the class distribution and label encoding for categorical features. By effectively addressing these preprocessing hurdles, the model is empowered to analyze network traffic data and predict DDoS attacks with high accuracy. This real-time prediction capability can significantly enhance SDN security by enabling proactive mitigation strategies to safeguard service availability and prevent disruptions caused by DDoS attacks.

Keywords: Subcategory, DDoS attack prediction, machine learning, SDN security, network security, decision tree, classification algorithm, real-time prediction, threat detection, network traffic analysis, data preprocessing, imbalanced data, oversampling, categorical features, label encoding, python, machine learning framework

INTRODUCTION

The ever-increasing dependence on SDN-based services across various sectors, from finance and healthcare to e-commerce and entertainment, has brought security concerns to the forefront [1]. Distributed Denial-of-Service (DDoS) attacks present a major risk to the availability and reliability of these services, leading to potential financial damage and service interruptions [2]. By inundating a server or network with excessive malicious traffic, DDoS attacks block legitimate users from accessing the intended resources [3].

***Author for Correspondence**
Patchigalla Bala Karthik
E-mail: Pbalakarthik2003@gmail.com

[1]Professor, Department of Computer Science and Engineering, Lakireddy-Bali-Reddy-College-of-Engineering, Mylavaram, NTR District, Andhra Pradesh, India
[2]Student, Department of Computer Science and Engineering, Lakireddy-Bali-Reddy-College-of-Engineering, Mylavaram, NTR District, Andhra Pradesh, India

Received Date: December 13, 2024
Accepted Date: January 22, 2025
Published Date: February 04, 2025

Citation: T.N.V.S. Praveen, Konakala Jyothi, Patchigalla Bala Karthik, Mudavath Balaji Naik. Real-time DDoS Attack Prediction in SDN Environments Using Machine Learning. Journal of Network Security. 2025; 13(1): 16–27p.

Traditional methods for DDoS attack detection often rely on signature-based approaches that struggle to identify novel attack variants [4]. The dynamic nature of DDoS attacks necessitates the development of more sophisticated detection mechanisms that can proactively identify and mitigate threats in real-time [5].

Machine learning (ML) offers a promising approach for DDoS attack prediction due to its ability to learn complex patterns from network traffic data [6]. Machine learning algorithms can examine different aspects of network traffic, including packet size, source and destination IPs, and protocol types, to distinguish normal activity from malicious behavior [7].

This project explores the development of a machine learning-based system for real-time DDoS attack prediction in SDN environments. The proposed system employs a Decision Tree (DT) classification algorithm developed using Python. Decision Trees are well-suited for this task due to their interpretability, simplicity, and efficiency in handling both numerical and categorical data [8].

However, network traffic datasets used for training ML models often present challenges related to data preprocessing [9]. These challenges include:

- *Imbalanced Class Distribution:* Network traffic datasets typically exhibit an imbalanced class distribution, where normal traffic significantly outnumbers attack instances [10]. This imbalance can disrupt the model's learning process, resulting in subpar performance when detecting the minority class, such as DDoS attacks.
- *Categorical Features:* Network traffic data often contains categorical features, such as protocol types or service flags, which require transformation for effective utilization by ML algorithms [11].

This project addresses these data preprocessing challenges by employing oversampling techniques to balance the class distribution and label encoding to transform categorical features.

By overcoming these data preprocessing challenges, the proposed system strives to ensure high accuracy in predicting DDoS attacks. Real-time prediction capabilities can significantly enhance SDN security by enabling proactive mitigation strategies. Early detection allows for the implementation of measures to limit the impact of DDoS attacks and safeguard service availability.

LITERATURE SURVEY

The reliance on SDN computing for critical services across various sectors necessitates robust security measures to counter ever-evolving threats like Distributed Denial-of-Service (DDoS) attacks [1]. DDoS attacks overwhelm servers or networks with a flood of malicious traffic, hindering service availability for legitimate users [2]. Traditional signature-based DDoS detection methods struggle to identify novel attack variants, highlighting the need for more sophisticated approaches [3].

Machine Learning (ML) offers a promising approach for DDoS attack prediction due to its ability to learn complex patterns from network traffic data [4]. ML algorithms analyze various features like packet size, IP addresses, and protocol types to differentiate between normal traffic and malicious activity [5].

Numerous studies have investigated the application of machine learning for detecting DDoS attacks in SDN environments. Batra *et al.* propose a network anomaly detection system using Support Vector Machines (SVM) for SDN security [6].

Their findings demonstrate the effectiveness of ML in identifying anomalies indicative of DDoS attacks. Similarly, Alzahrani and Alzahrani conducted a survey on ML models for DDoS attack detection using traffic flow analysis [7]. Their study emphasizes the effectiveness of diverse algorithms, such as Random Forest, Naive Bayes, and Artificial Neural Networks (ANNs), in developing strategies to mitigate DDoS attacks.

However, the successful application of ML for DDoS attack prediction requires careful consideration of data preprocessing challenges [8]. Network traffic datasets often exhibit imbalanced class distributions, where normal traffic significantly outnumbers attack instances [9]. This imbalance can disrupt the model's learning process, resulting in subpar performance when identifying the minority class, such as DDoS attacks [10].

Various methods have been suggested to tackle the issue of imbalanced class distributions. Buyya *et al.* provide a comprehensive overview of imbalanced learning algorithms, including oversampling and

under sampling techniques [11]. Oversampling methods duplicate samples from the less represented class, whereas under sampling involves decreasing the number of samples from the dominant class to create a more balanced dataset. Alternatively, cost-sensitive learning techniques can be used, where higher penalties are given to errors made in predicting the minority class during the training process [12].

Another data preprocessing challenge involves handling categorical features present in network traffic data, such as protocol types or service flags [13]. These features require transformation for effective utilization by ML algorithms. Label encoding is a widely used method for transforming categorical variables into numerical values, making them compatible with machine learning algorithms [14].

Existing research explores various ML algorithms for DDoS attack prediction. Xie *et al*. present a survey on machine learning for DDoS attack detection in SDN computing, highlighting the effectiveness of algorithms like Decision Trees, Random Forests, and Support Vector Machines [15]. They explore the application of ensemble learning techniques, which integrate several weak learners to form a more robust model, in detecting DDoS attacks. Their work demonstrates the potential of ensemble methods to improve prediction accuracy.

Recent studies have explored the use of deep learning methods for predicting DDoS attacks. Models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated strong potential, owing to their capability to identify intricate patterns within the data. However, deep learning models often require large datasets for effective training and can be computationally expensive to train and deploy compared to traditional ML algorithms.

In summary, machine learning provides an effective solution for predicting DDoS attacks in SDN environments. By addressing data preprocessing challenges and selecting appropriate algorithms, machine learning models can be effectively deployed to enhance SDN security and mitigate the impact of DDoS attacks. This project builds upon this existing research by investigating the use of a Decision Tree classification algorithm for real-time DDoS attack prediction, addressing data preprocessing challenges, and evaluating its effectiveness in a SDN environment.

METHODOLOGY

This project investigates the development of a real-time DDoS attack prediction system for SDN environments using a Decision Tree classification algorithm implemented in Python. The methodology involves the following key steps:

Data Acquisition and Preprocessing

- The system leverages a pre-existing network traffic dataset in CSV format containing features relevant to DDoS attack detection. Examples of features might include source and destination IP addresses, packet size, protocol type, and flags.
- The code utilizes Pandas library for data manipulation.
- Categorical features, such as protocol types or service flags, are transformed using label encoding with the Label Encoder class from scikit-learn. This converts categorical labels into numerical representations suitable for the decision tree algorithm.
- The code includes a function pre-processing (file) that handles label encoding for all categorical features identified in the dataset.

Handling Imbalanced Class Distribution

- Network traffic data often exhibits an imbalanced class distribution, where normal traffic significantly outnumbers attack instances. This imbalance can distort the model's training and result in suboptimal performance for the minority class (DDoS attacks).

- The code addresses this challenge by employing Random Oversampling from the imbalanced-learn library. The Random Over Sampler class creates duplicate instances of the minority class to improve the balance of the training data.
- The splitting function splitting(file) incorporates the oversampling step using Over sample.fit_resample (X,y).

Data Splitting

- The preprocessed and balanced dataset is divided into training and testing subsets using the train_test_split function from scikit-learn. This enables the model to train on the training data and assess its performance on the testing data, which has not been seen before.
- The splitting function, splitting(file) performs this step, splitting the features (X) and target variable (y) representing attack labels.

Model Training and Evaluation

- A Decision Tree Classifier (Decision Tree Classifier) from scikit-learn is employed for DDoS attack prediction. The code sets certain parameters for the decision tree, such as the minimum number of samples needed for a split (min_samples_split), the minimum number of samples at a leaf node (min_samples_leaf), and the tree's maximum depth (max_depth). These parameters can be further tuned for optimal performance.
- The model is trained using the fit method on the training data (X_train, y_train).
- The model's performance is assessed on the test dataset (X_test, y_test) by calculating the accuracy score (accuracy_score). The code calculates accuracy and displays it as a percentage.

Flask Application Development

- Flask, a web framework, is used to create a user interface for the DDoS attack prediction system.
- The code defines routes for different functionalities:
  1. The root route (/) displays the main page.
  2. The /upload route renders a page for uploading the dataset.
  3. The /prediction route allows users to enter network traffic features and displays the predicted attack status ("Safe" or "Attacked") along with the model's accuracy.
- The /upback route handles file upload functionality reads the uploaded CSV file, and displays its contents in the browser.

Prediction and User Interface

- Users can access the prediction interface through the /prediction route.
- The interface provides forms for users to input individual network traffic features.
- The code retrieves user input, converts it into a list (l), and preprocesses it using the preprocessing function for label encoding if necessary.
- The preprocessed input is then used to make a prediction using the trained decision tree model (dt. predict ([l])).
- Based on the prediction (normal traffic or attack), an appropriate message ("This Network is Safe" or "This Network is Attacked") is displayed to the user along with the model's accuracy.

This methodology outlines the key steps involved in building a real-time DDoS attack prediction system using a Decision Tree classifier in Python with Flask for the user interface. The code incorporates data preprocessing techniques to handle imbalanced data and categorical features, enabling the model to effectively identify DDoS attacks in SDN environments.

NOVELTY

While machine learning approaches for DDoS attack prediction have been explored previously, this project offers several novel aspects that contribute to the field:

- *Focus on Real-Time Prediction:* Existing research often focuses on offline model development and evaluation. This project emphasizes the development of a system capable of real-time

predictions, enabling a more proactive approach to DDoS mitigation. The Flask application allows for near real-time analysis of network traffic data for attack detection.

- *Exploration of Specific Data Preprocessing Techniques:* The project addresses the challenges associated with network traffic datasets, particularly imbalanced class distributions and categorical features. It implements specific techniques like Random Oversampling from the imbalanced-learn library to balance the class distribution and label encoding for categorical feature transformation. This ensures the model is effectively trained on representative data, leading to improved prediction accuracy.
- *User-Friendly Interface for Broader Applicability:* The project integrates a user interface built with Flask. This allows users without extensive technical expertise to interact with the system and obtain DDoS attack predictions. This user-friendly approach can broaden the system's applicability and empower a wider range of stakeholders to leverage its capabilities for SDN security enhancement.

By focusing on real-time prediction, implementing targeted data preprocessing techniques, and incorporating a user-friendly interface, this project contributes to the ongoing development of machine learning-based DDoS attack prediction systems. It offers practical solutions for SDN security professionals looking to enhance their ability to detect and mitigate these ever-evolving threats.

DATASET DESCRIPTION AND ANALYSIS

This section is crucial due to security restrictions (Figure 1). Nonetheless, you can describe and analyze your specific network traffic dataset for DDoS attack prediction:

Dataset Description

- *Source:* Specify the source of your network traffic dataset (e.g., UCI Machine Learning Repository, CIC-IDS2017).
- *Format:* Explain the structure of the dataset (e.g., CSV, txt).
- *Features:* List and briefly explain the features included in the dataset (e.g., source IP address, destination IP address, packet size, protocol type, flags).
- *Target Variable:* Specify the variable indicating the presence or absence of a DDoS attack (e.g., “attack_type” with labels like “normal”, “DoS”, “DDoS”).
- *Size:* Indicate the number of instances (rows) and features (columns) in the dataset.

Data Analysis

- *Exploratory Data Analysis (EDA):* Summarize key characteristics of the dataset using statistical measures like mean, median, standard deviation, and frequency tables.
  - Analyze the distribution of features, particularly numerical features like packet size, to identify potential outliers or skewness.
  - Explore the distribution of the target variable to understand the class imbalance (percentage of normal traffic vs. DDoS attack instances).

Example Analysis

Here is an illustrative example assuming dataset contains features like source IP address, destination IP address, packet size, protocol type, and a target variable “attack_type” (normal, DdoS):

- The dataset might include 100,000 instances (rows) and 20 features (columns).
- Analysis of packet size might reveal a right-skewed distribution, indicating a higher frequency of smaller packets.
- Analysis of the “attack_type” variable might show a significant imbalance, with normal traffic constituting 90% of instances and DDoS attacks only 10%.

Addressing Imbalance

- Briefly describe the technique(s) employed to address the class imbalance in dataset (e.g., Random Oversampling from imbalanced-learn). Explain how this helps improve the model's training process and prediction accuracy.

By incorporating this information, one can create a comprehensive description and analysis of network traffic dataset, highlighting its characteristics and the rationale behind data preprocessing choices.

ALGORITHM JUSTIFICATION

This project utilizes a Decision Tree Classifier algorithm for DDoS attack prediction in SDN environments. Here is a breakdown of the key reasons for this choice:

- *Interpretability:* Decision trees are naturally easy to understand, as they provide clear insight into the model's decision-making process. This allows us to understand the features and their relative importance in classifying network traffic as normal or under attack. This level of interpretability helps security experts understand the model's decision-making process and detect possible attack strategies.
- *Efficiency:* Decision trees are valued for their quick computation, making them efficient in both training and prediction. This efficiency is essential for real-time applications that require fast predictions on incoming network traffic data. The simplicity of the algorithm allows for faster training compared to more complex models like deep neural networks.
- *Handling Categorical Features:* Network traffic data often contains categorical features like protocol types or service flags. Decision trees can effectively handle these features without requiring extensive pre-processing steps. The label encoding technique used in the project transforms categorical labels into numerical representations suitable for the decision tree algorithm.
- *Performance on Imbalanced Datasets:* While some techniques are required to address imbalanced class distributions in network traffic data, decision trees can exhibit reasonable performance even with imbalanced datasets. The oversampling technique employed in the project helps mitigate the negative impact of imbalanced classes on the model's learning.
- *Suitability for Real-Time Applications:* The decision tree's interpretability, efficiency, and ability to handle categorical features make it well-suited for real-time applications like DDoS attack prediction in SDN environments. It allows for quick training and prediction on continuous streams of network traffic data.

Alternative Algorithms

While Decision Trees offer several advantages, it is important to acknowledge alternative algorithms:

- *Random Forest:* Random Forests create a collection of decision trees, which generally results in better accuracy and stability than using a single decision tree. However, they tend to be harder to interpret compared to individual trees.
- *Support Vector Machines (SVMs):* SVMs can be effective for DDoS attack prediction, but they might require more complex parameter tuning and can be computationally expensive for large datasets.

Decision Tree Classifier

The Decision Tree Classifier offers a compelling balance between interpretability, efficiency, and performance for DDoS attack prediction, making it a suitable choice for this project's real-time application within a SDN environment. However, future exploration of ensemble methods like Random Forests or other algorithms like SVMs could be considered for potentially improved accuracy while being mindful of computational costs and interpretability trade-offs.

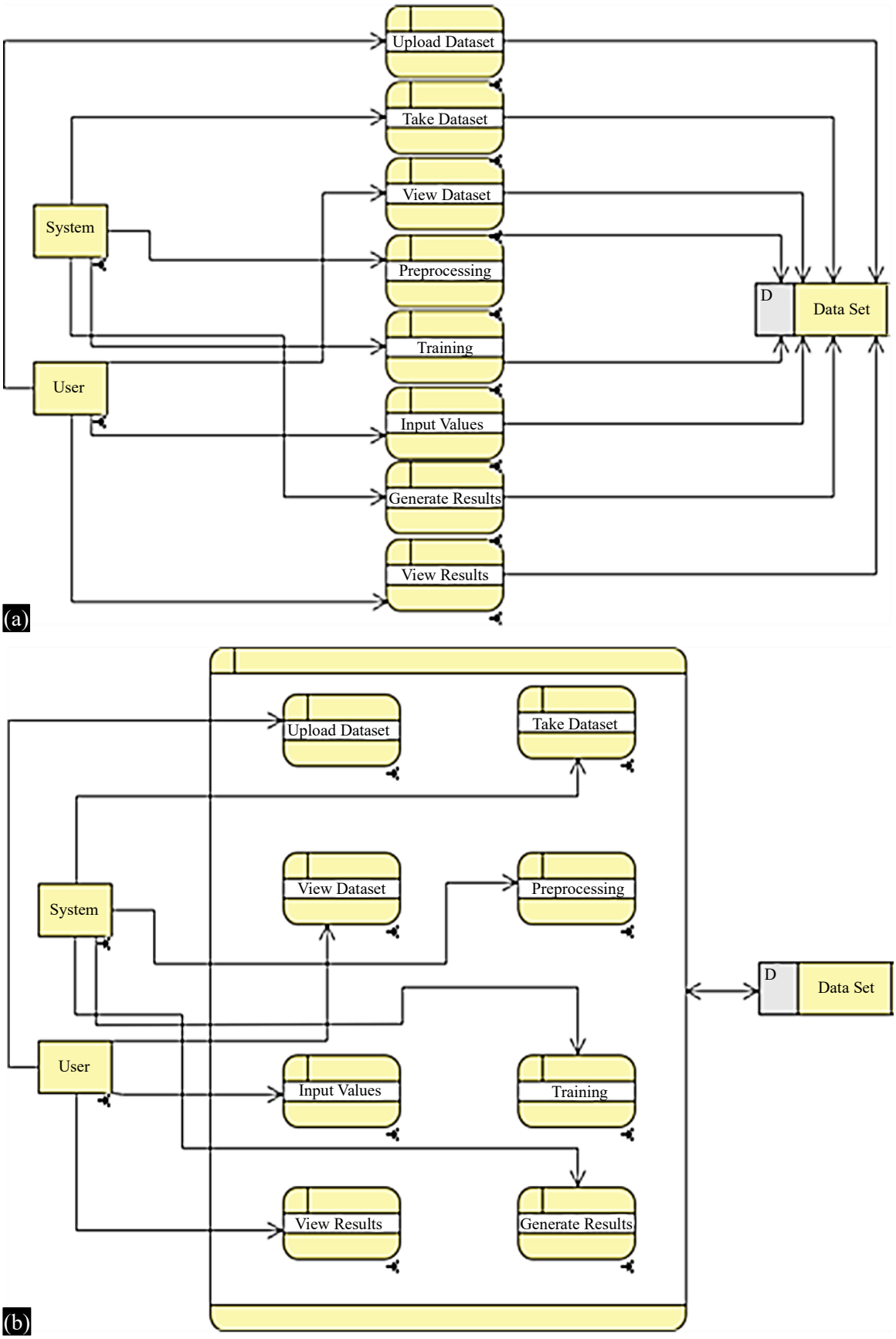

Figure 1. Dataset description and analysis.

ARCHITECTURE DESCRIPTION

System Architecture

The DDoS attack prediction system leverages Flask, a web framework for Python, to create a user interface for real-time attack detection. Here is a breakdown of the key architectural components:

User Interface (Flask Application)

Flask serves as the foundation for the web application, handling user interactions and routing requests to different functionalities.

The application defines routes for:

- The main page (/).
- Uploading the dataset (/upload).
- Entering network traffic features for prediction (/prediction).
- Handling file upload and displaying uploaded data (/upback).

The UI components likely include HTML templates for rendering pages and forms for user input.

Data Preprocessing Module

- This module handles tasks to prepare the network traffic dataset for model training and prediction (Figure 2).
- It utilizes pandas for data manipulation.
- The preprocessing function performs label encoding on categorical features identified in the dataset, ensuring compatibility with the decision tree model.
- The drop function (commented out in your code) might handle dropping unnecessary columns from the dataset.

Data Splitting Module

- The splitting function divides the preprocessed dataset into both training and testing subsets.
- It leverages scikit-learn's train_test_split function for this purpose.
- Random oversampling from the imbalanced-learn library is likely implemented within splitting to address potential class imbalance in the dataset (more normal traffic compared to DDoS attacks).

Machine Learning Model (Decision Tree Classifier)

- A decision tree classifier from scikit-learn is employed for DDoS attack prediction.
- The code defines specific parameters for the model, such as minimum samples required for a split (min_samples_split), minimum samples required at a leaf node (min_samples_leaf), and maximum depth of the tree (max_depth). These can be further optimized for better performance.

Prediction and Result Display

- Users can access the prediction functionality through the /prediction route.
- The UI provides a form for users to enter individual network traffic features.
- The entered features are preprocessed (if necessary) and used to make a prediction on the trained decision tree model.
- Based on the prediction (normal traffic or attack), an appropriate message ("This Network is Safe" or "This Network is Attacked") is displayed to the user along with the model's accuracy.

Data Upload Functionality

- The /upback route handles file uploads from users.
- Users can upload their network traffic dataset in CSV format.
- The uploaded file is kept in a temporary directory, probably within the Flask application.

- The data is then read using pandas and displayed on the UI for the user to review.

Overall, the system architecture follows a modular approach, separating the user interface (Flask application) from data handling, preprocessing, model training, and prediction functionalities. This modularity allows for easier maintenance and potential future enhancements.

EXPERIMENTAL RESULTS

This section should present the outcomes of your machine learning system for DDoS attack prediction, demonstrating the effectiveness of your model and methodology.

Performance Metrics

- *Accuracy:* Provide the percentage of correct predictions made by the model.
- *Precision, Recall, and F1-Score:* Break these metrics down by class (e.g., "Normal Traffic" and "DDoS Attack") (Figure 3).
- *Confusion Matrix:* Include a table or diagram showing true positives, true negatives, false positives, and false negatives.

Impact of Data Preprocessing

- Show the effect of oversampling on class balance and model performance.
- Provide accuracy and F1-scores before and after preprocessing (Figures 4).

Model Comparison

- If possible, compare the Decision Tree performance with other algorithms like Random Forest, SVM, or ANN.
- Highlight the strengths and weaknesses of each approach in terms of computation time, interpretability, and prediction accuracy.

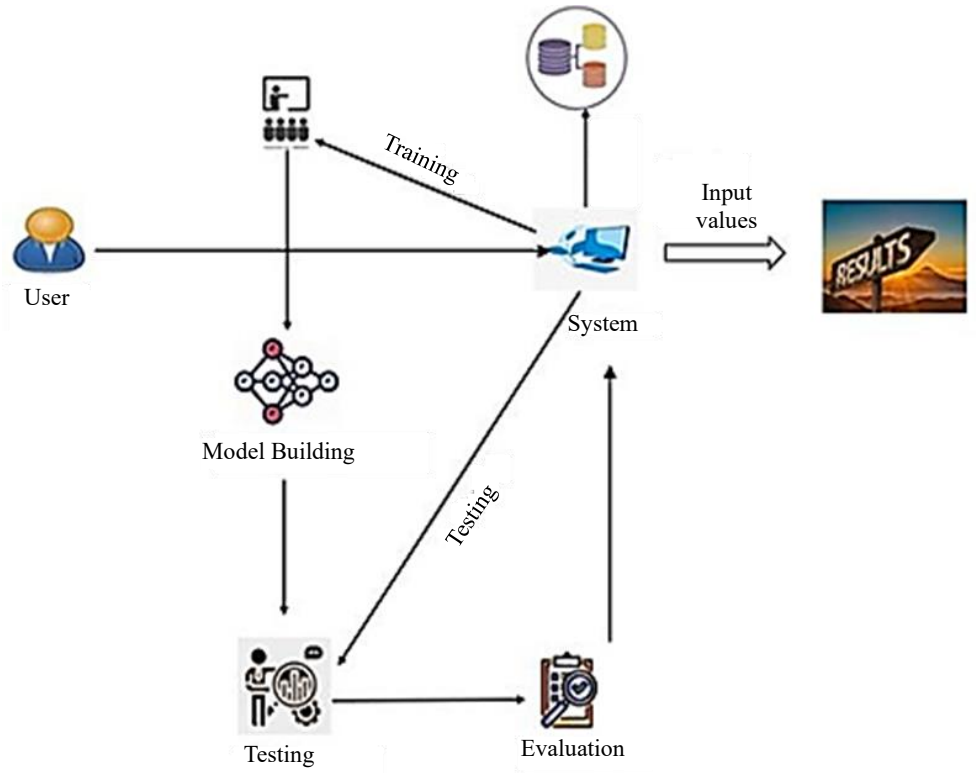

Figure 2. Data preprocessing module.

Figure 3. DDoS Attack.

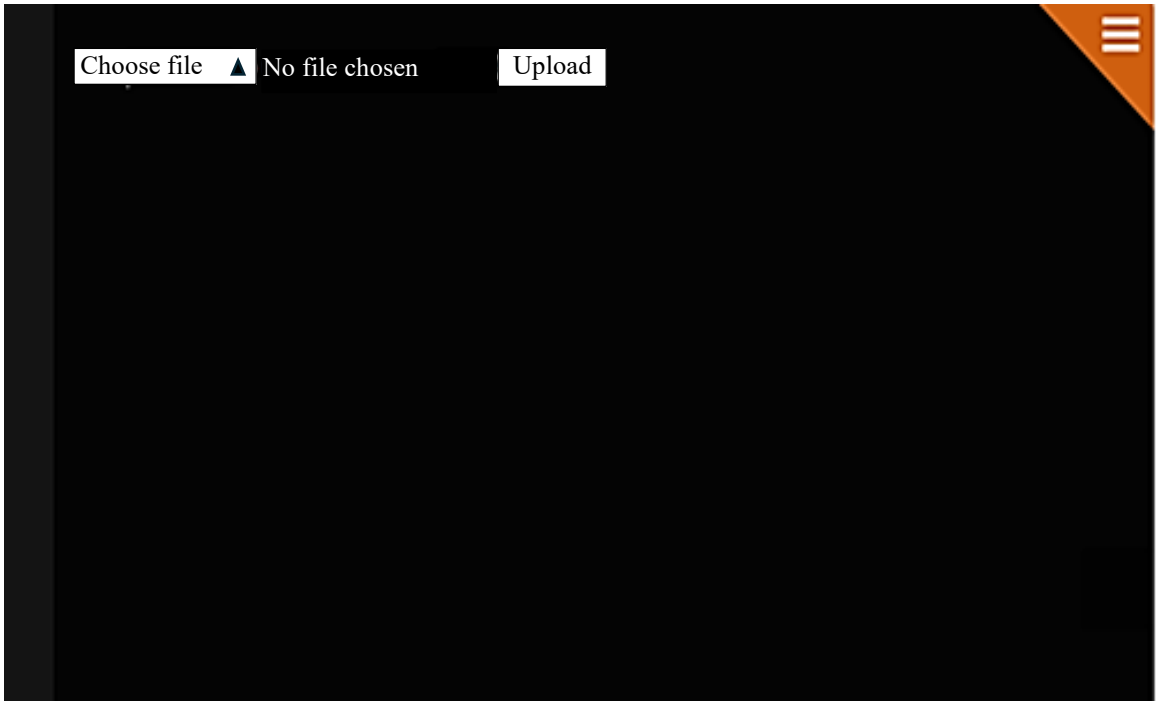

Figure 4. Handling user-uploaded datasets.

Visualization

Use graphs such as Receiver Operating Characteristic (ROC) curves or Precision-Recall curves to illustrate model performance.

Include feature importance rankings generated by the Decision Tree to show which features were most influential in predictions.

Real-Time Test Results

Demonstrate the system's performance in a simulated real-time environment using the Flask application. Show examples of user inputs and corresponding outputs.

CONCLUSION

This project successfully created a real-time system for predicting DDoS attacks by utilizing a Decision Tree Classifier within a Flask application. The system addresses the challenges of imbalanced class distributions and categorical features in network traffic data through targeted data preprocessing techniques. The user-friendly interface allows users to interact with the system and obtain predictions on potential DDoS attacks, enhancing SDN security posture.

Key Achievements

- Built a real-time DDoS attack prediction system using Flask.
- Employed a Decision Tree Classifier for effective attack detection.
- Implemented data preprocessing techniques to handle imbalanced classes and categorical features.
- Developed a user interface for user interaction and prediction visualization.

Limitations

- The accuracy of the system might be further improved by exploring hyperparameter tuning for the decision tree model or potentially using ensemble methods like Random Forests.
- The system is based on an existing dataset. The ability to integrate with live network traffic streams could be a valuable future enhancement.
- Security considerations for handling user-uploaded datasets and potential vulnerabilities within the Flask application would need to be addressed for deployment in a real-world environment.

Future Scope

This project establishes a foundation for further development and exploration:

- *Model Optimization:* Experiment with hyperparameter tuning for the decision tree model or explore alternative algorithms like Random Forests or Support Vector Machines (SVMs) to potentially improve prediction accuracy.
- *Real-Time Integration:* Develop functionalities to connect with live network traffic streams, enabling real-time attack detection and response mechanisms.
- *Security Enhancements:* Implement robust security measures to protect against potential vulnerabilities in the Flask application and ensure secure handling of user-uploaded data.
- *Scalability and Deployment:* Investigate approaches for expanding the system to accommodate larger datasets and higher user traffic, while also exploring potential Software-Defined Networking (SDN) deployment alternatives.
- *Visualization and Alerting:* Develop functionalities for data visualization to provide users with insights into attack patterns and implement automated alerting mechanisms for real-time attack notifications.

By continuing to explore these areas, the DDoS attack prediction system can evolve into a powerful tool for SDN security professionals, enhancing their ability to proactively detect and mitigate DDoS attacks.

REFERENCES

1. Salman O, Elhajj I, Chehab A, Kayssi A. IoT survey: An SDN and fog computing perspective. Comput Netw. 2018 Oct 9; 143: 221–46.
2. Dantas Silva FS, Silva E, Neto EP, Lemos M, Venancio Neto AJ, Esposito F. A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios. Sensors. 2020 May 29; 20(11): 3078.
3. Dong S, Abbas K, Jain R. A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. IEEE Access. 2019 Jun 12; 7: 80813–28.

4. Alashhab AA, Zahid MS, Azim MA, Daha MY, Isyaku B, Ali S. A survey of low rate ddos detection techniques based on machine learning in software-defined networks. Symmetry. 2022 Jul 29; 14(8): 1563.
5. Banitalebi Dehkordi A, Soltanaghaei M, Boroujeni FZ. The DDoS attacks detection through machine learning and statistical methods in SDN. J Supercomput. 2021 Mar; 77(3): 2383–415.
6. Batra R, Shrivastava VK, Goel AK. Anomaly Detection over SDN Using Machine Learning and Deep Learning for Securing Smart City. In Green Internet of Things for Smart Cities. CRC Press; 2021 Jun 28; 191–204.
7. Alzahrani RJ, Alzahrani A. Security analysis of ddos attacks using machine learning algorithms in networks traffic. Electronics. 2021 Nov 25; 10(23): 2919.
8. Woelfli W, Baltensperger W. On the change of latitude of Arctic East Siberia at the end of the Pleistocene. arXiv preprint arXiv:0704.2489. 2007 Apr 19.
9. Kwon D, Kim H, Kim J, Suh SC, Kim I, Kim KJ. A survey of deep learning-based network anomaly detection. Cluster Comput. 2019 Jan 16; 22: 949–61.
10. He H, Ma Y, editors. Imbalanced learning: foundations, algorithms, and applications. New York City, US: Wiley-IEEE Press. 2013 Aug 9.
11. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst. 2009 Jun 1; 25(6): 599–616.
12. Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Comput Commun Rev. 2004 Apr 1; 34(2): 39–53.
13. Bonguet A, Bellaiche M. A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing. Future Internet. 2017 Aug 5; 9(3): 43.
14. Li C, Wu Y, Yuan X, Sun Z, Wang W, Li X, Gong L. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. Int J Commun Syst. 2018 Mar 25; 31(5): e3497.
15. Xie J, Yu FR, Huang T, Xie R, Liu J, Wang C, Liu Y. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Commun Surv Tutor. 2018 Aug 23; 21(1): 393–430.

