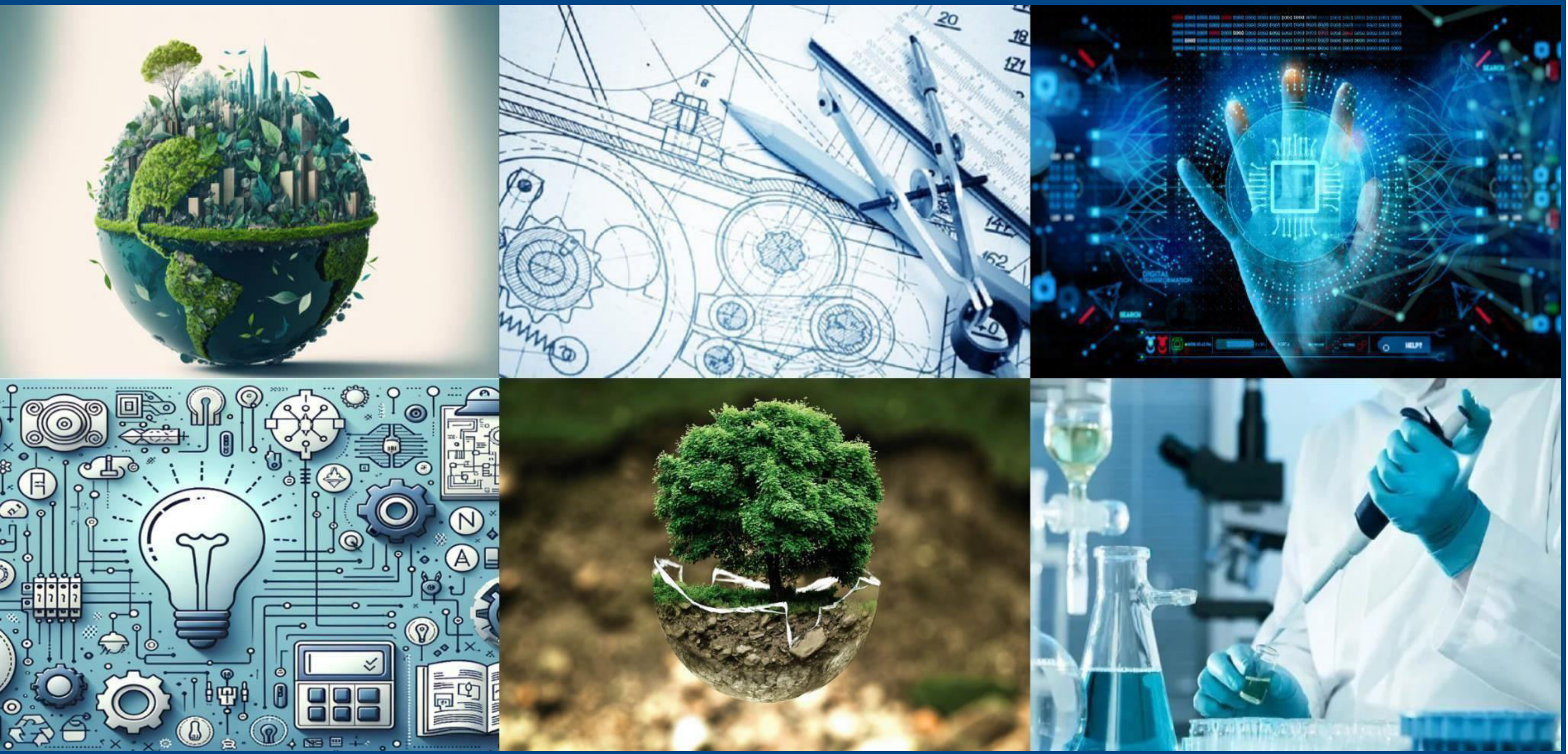
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 8, August 2025

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

FLOWSENTINEL: TRAFFIC ANOMALY SMART SHIELD

Gunasekaran K, G Charan Kumar Reddy

Assistant Professor, Department of MCA, AMC Engineering College, Bengaluru, India

Student, Department of MCA, AMC Engineering College, Bengaluru, India

ABSTRACT: This project presents a machine learning-based web application for DDoS (Distributed Denial of Service) attack detection using essential network traffic features. The system allows users to manually input network parameters & upload a CSV file containing multiple traffic records. Using a trained Random Forest model, it predicts whether the given data indicates normal traffic or a potential DDoS attack. The backend is powered by Flask, while the frontend is built with React.js, offering a clean & intuitive UI. This tool helps in early detection of DDoS patterns & improves network security monitoring without requiring advanced technical expertise. This project presents a machine learning-based web application for Distributed Denial of Service (DDoS) attack detection using essential network traffic features. The system allows users to manually input key network parameters & records of network traffic. A trained Random Forest model evaluates these inputs & classifies them as either normal or DDoS traffic. The backend is powered by Flask, & the frontend is built using React.js with professional CSS styling. The interface is designed to be clean, responsive, & accessible to both technical and non-technical users.

KEYWORDS: DDoS, Machine Learning, Flask, React.js, Random Forest, Network Security.

I. INTRODUCTION

DDoS (Distributed Denial of Service) attacks are attempts to disrupt regular network traffic by overwhelming the target with a flood of internet traffic. Traditional systems rely on rule-based or threshold systems which often fail to adapt to complex, evolving attack patterns. With advancements in artificial intelligence, it is now possible to implement a smarter and automated DDoS detection mechanism. This project introduces a solution that combines a machine learning model with a user-friendly interface to help detect and classify DDoS attacks based on traffic behaviour using time-related features.

Increasing complexity of network systems, DDoS attacks have become one of the most common & damaging threats to web services. This project introduces a web-based tool that leverages machine learning to detect possible DDoS attacks.

II. LITERATURE SYRVEY

DDoS detection has been a widely researched topic in the field of network security and machine learning. Over the years, several approaches have been proposed, ranging from rule-based systems to intelligent algorithms that adapt to evolving traffic patterns. The literature highlights the evolution from traditional methods to AI/ML-based techniques for effective and scalable DDoS detection.

1.Signature-Based Detection Systems Signature-based intrusion detection systems (IDS), such as Snort, rely on known attack patterns. These systems maintain a database of attack signatures & match incoming traffic against them. While effective for known threats, they fail to detect zero-day or evolving DDoS attacks.

2.Threshold and Statistical Methods Earlier systems focused on threshold-based detection—identifying traffic spikes or abnormal packet rates beyond a fixed limit. While simple and fast, they are prone to false positives during traffic bursts from legitimate users. Statistical methods (mean, variance) helped reduce noise but lacked adaptability.

3.Machine Learning for Anomaly Detection Recent advancements have shifted toward using supervised learning methods. Techniques such as Decision Trees, Support Vector Machines (SVM), and Random Forests learn from labeled traffic datasets and can detect complex attack patterns.

4.Deep Learning Approaches Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been applied to sequence-based traffic data. These models show improved accuracy in handling high-dimensional and temporal data but require high computational resources and large datasets.

5.**Ensemble Learning Models** Ensemble models, such as Random Forest and Gradient Boosting, combine multiple weak classifiers to improve accuracy. They are particularly effective for structured datasets and are robust to overfitting. Our system utilizes Random Forest due to its interpretability and high detection performance

6.**Hybrid Detection Systems** Some modern systems combine signature-based and anomaly-based detection to balance between speed and intelligence. These hybrid systems offer real-time capabilities while adapting to evolving attack patterns.

EXISTING SYSTEM

Traditional DDoS detection relies on static firewalls, threshold-based rules, and signature-based IDS, which often fail to detect evolving attack patterns. These systems generate high false positives & require manual monitoring. They lack real-time adaptability & struggle with dynamic traffic. Hence, they are inadequate for modern, automated attack detection.

PROPOSED SYSTEM

The proposed system uses a Machine Learning-based approach to detect DDoS attacks in real time. It employs a trained Random Forest model that analyzes specific traffic features such as Flow Duration, Idle Mean, Idle Std, and IAT metrics. Users can input data manually are upload a CSV file for batch prediction. The frontend is built using React.js, and the backend is developed with Flask. Predictions are displayed with a clean and intuitive UI. This system improves accuracy, automation, and accessibility over traditional detection methods.

III. SYSTEM ARCHITECTURE

The system architecture includes a React.js-based frontend that allows users to input network traffic data manually or through CSV file upload. This data is transmitted to a Flask-powered backend, which preprocesses the input and forwards it to a trained Random Forest model

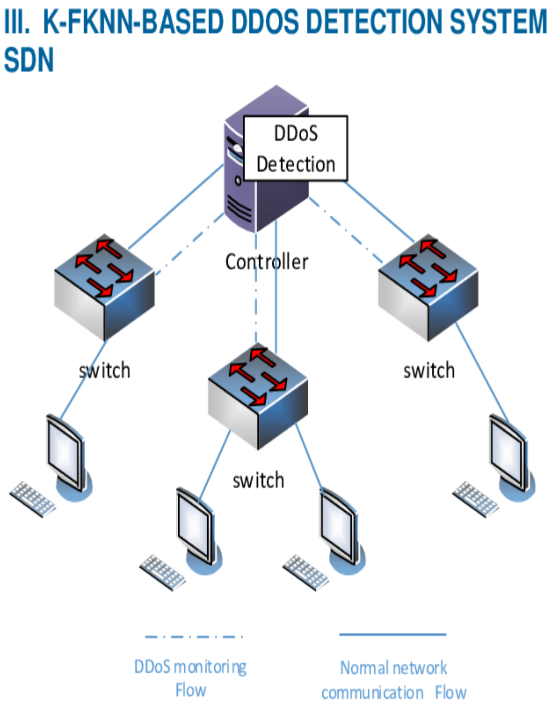

III. K-FKNN-BASED DDOS DETECTION SYSTEM
SDN
DDoS Detection
Controller
switch
switch
switch
DDoS monitoring Flow
Normal network communication Flow

Fig 3.1 System Architecture

1.METHODOLOGY

The process begins with data acquisition, during which historical network traffic data is gathered. Key features such as Flow Duration, Idle Mean, Idle Std, Flow IAT Mean, and Flow IAT Std are selected, as they are strong indicators of abnormal traffic behavior associated with DDoS attacks. On the frontend, a React.js application is developed that allows users to either manually input values or upload a CSV file. Axios is utilized to send HTTP requests to the Flask server. The interface is styled professionally using Tailwind CSS and CSS modules to ensure an engaging and responsive design. In the data preprocessing stage, the collected data is cleaned and normalized. Missing values are handled, and scaling techniques are applied to ensure uniformity across all features, enhancing the performance of the machine learning model.

2.DESIGN AND IMPLEMENTATION

The DDoS detection system is designed with a modular and scalable architecture that integrates machine learning, backend API services, and a user-friendly frontend interface. The design ensures both real-time responsiveness and ease of use for non-technical users. At the core of the system, a pre-trained Random Forest Classifier is used for its effectiveness in detecting attack patterns based on flow-related network features. The model is trained on labeled network traffic data, focusing on key features such as Flow Duration, Idle Mean, Idle Std, Flow IAT Mean, and Flow IAT Std. After training, the model is saved as a .pkl file for deployment. The backend is built using Flask, a lightweight Python web framework. It exposes RESTful endpoints that can handle both single record (manual input) and multiple records (CSV file) submissions. Upon receiving data, the backend processes and validates it, applies necessary preprocessing, loads the trained model, and returns prediction results in JSON format. The frontend is implemented using React.js, offering a clean and responsive user interface. Users can input traffic data manually through input fields or upload a .csv file for batch prediction. The frontend communicates with the backend through Axios, which is a promise-based HTTP client. It displays prediction results with clear status indicators, including DDoS probability confidence, to help users quickly assess the threat. Styling is done using Tailwind CSS and modular CSS files, ensuring an attractive layout, consistent theming, and smooth transitions across pages like Home, DDoS Detection, and Result Display. Reusable components like Navbar and Footer maintain consistency across the application. Robust error handling, user alerts, loading indicators, and reset functionality are implemented to enhance user experience. The modular design allows for easy updates, such as switching to more advanced models or adding new features like traffic visualization or real-time monitoring dashboards. The DDoS detection system is designed with a modular and scalable architecture that integrates machine learning, backend API services, and a user-friendly frontend interface. The design ensures both real-time responsiveness and ease of use for non-technical users.

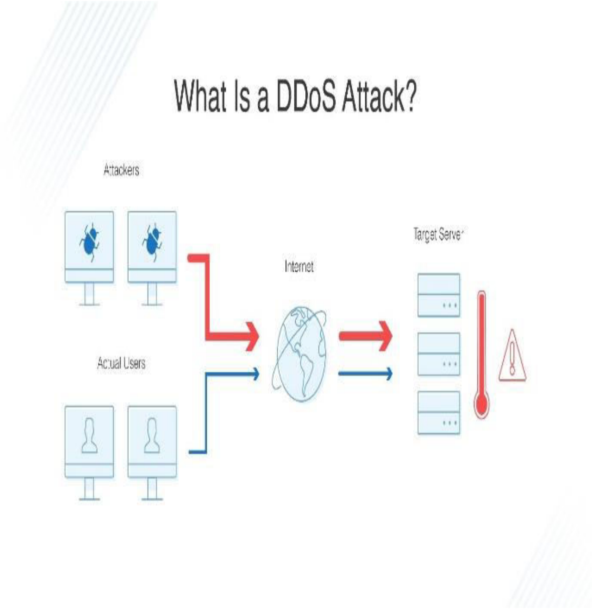


Fig 5.1 Flowchart of Working System

3.OUTCOME OF RESEARCH

The developed DDoS Detection System successfully demonstrates the application of machine learning in identifying abnormal traffic patterns that signal potential attacks. The Random Forest classifier achieved high accuracy during testing, confirming its effectiveness in distinguishing between normal and malicious traffic using selected flow-based features. Users are able to interact with the system via a responsive React interface, choosing between manual input for single traffic entries or CSV uploads for batch analysis. The system provides real-time feedback on predictions, including the confidence level of each result, helping users make informed decisions quickly. This outcome proves the viability of integrating ML models with web-based platforms for cybersecurity applications, offering a foundation for future expansion and practical deployment in real-world environments.

IV. RESULT AND DISCUSSION

The DDoS Detection System successfully integrated machine learning with a web-based interface to detect and classify potential DDoS traffic in both real-time and batch modes. Upon evaluation, the trained **Random Forest Classifier** demonstrated strong performance, showing a high degree of **accuracy, precision, and recall** when tested on labeled network traffic data. This indicates the model's robustness in distinguishing between normal and malicious traffic patterns based on just a few critical features. In **manual prediction mode**, users were able to input values for features such as Flow Duration, Idle Mean, Idle Std, Flow IAT Mean, and Flow IAT Std. The system returned an immediate prediction along with a **confidence score**, helping users interpret how likely a given traffic pattern is to be a DDoS attack. This mode is useful for researchers or analysts who want to test specific scenarios. In **CSV upload mode**, users could upload bulk traffic logs, and the system would process multiple rows in one go, returning predictions for each row with excellent responsiveness. This batch-processing capability makes the system practical for analyzing large datasets, especially in enterprise environments or academic research. The **backend**, built with Flask and Python, provided a clean and efficient REST API. It handled data validation, model inference, and JSON formatting effectively. The system architecture ensured minimal latency and high scalability. Through testing, the model performed well under varied input conditions. However, performance is dependent on the quality of the input data, and future improvement could involve **data normalization**, **error handling enhancements**, and **support for additional features** to further boost model performance. The project validates how machine learning can be embedded into real-world applications for network security, showcasing how even a simple ML model with a user-friendly interface can significantly aid in **cyber threat detection**.

V. CONCLUSION

This project successfully demonstrates the integration of machine learning with a user-friendly web application for real-time detection of Distributed Denial of Service (DDoS) attacks. By leveraging a Random Forest Classifier trained on key traffic features, the system accurately distinguishes between normal and malicious network activity. The dual functionality—manual input and CSV-based batch prediction—makes the tool accessible to both individual users and network analysts.

REFERENCES

1.Garcia, S., Grill, M., J., & Zunino, A. (2014). "An empirical comparison of botnet detection methods. "Computers & Security, 45, 100–123.

2.Sommer, R., & Paxson, V. (2010).IEEE Symposium on Security and Privacy.

3.Breiman, L. (2001). *Random Forests.* Machine Learning, 45(1), 5–32.

4 Scikit-learn Developers. (2023). *Scikit-learn: Machine Learning in Python*. Retrieved from https://scikit-learn.org/

5.Flask Documentation Team. (2023). *Flask Web Framework Documentation*. Retrieved from https://flask.palletsprojects.com/

6.React.js Documentation. (2023). *React – A JavaScript library for building user interfaces.* Retrieved from https://reactjs.org/s

INNO SPACE
SJIF Scientific Journal Impact Factor

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com