

Development of a Machine Learning-Based Framework for Real-Time Detection and Mitigation of Distributed Denial of Service Attacks

Oluyimide A. Onaolapo

Department of Computer Science, University of Ibadan, Nigeria
E-mail: oluyimidea@gmail.com
ORCID ID: <https://orcid.org/0009-0001-2773-0172>

Adebola K. Ojo*

Department of Computer Science, University of Ibadan, Nigeria
E-mail: adebolak.ojo@gmail.com
ORCID ID: <https://orcid.org/0000-0002-3568-6597>
*Corresponding Author

Received: 28 August, 2025; Revised: 12 September, 2025; Accepted: 04 November, 2025; Published: 08 April, 2026

Abstract: Distributed Denial-of-Service (DDoS) attacks continue to pose a significant threat to digital infrastructures, often resulting in degraded service availability and financial losses. Traditional detection systems, which depend on static rule sets, struggle to adapt to evolving traffic patterns, leading to increased false positives and undetected attacks. This paper presents a real-time, machine learning-based framework for DDoS detection and mitigation. The framework incorporates supervised learning algorithms, including Random Forest, XGBoost, and Multi-Layer Perceptron (MLP), trained on the CIC-DDoS2019 dataset using carefully selected network traffic features to enhance detection accuracy. The system architecture integrates Scapy for traffic capture, Apache Kafka for message queuing, and Flask with Plotly for dynamic monitoring. Evaluation results demonstrate superior performance compared to legacy methods across precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR). Additionally, adaptive models such as Passive-Aggressive and Stochastic Gradient Descent (SGD) enhance robustness against evolving attack vectors. The proposed solution delivers an effective and scalable real-time defense mechanism suitable for banking, cloud, and enterprise systems. However, the system's performance remains influenced by the characteristics of the training dataset and may introduce computational overhead during high-throughput traffic analysis. Future work will explore improved computational efficiency and responsiveness to rare or emerging DDoS patterns.

Index Terms: DDoS Detection, Machine Learning, Real-time Detection, Adaptive Learning, Network Security.

1. Introduction

The growth of digital services has changed how people and organisations do things online. From banking to health records, nearly everything now passes through internet systems. But this same growth has brought more risk. Online platforms are more open to threats, especially from Distributed Denial-of-Service (DDoS) attacks, which can stop services by flooding them with too much traffic at once. These attacks do not just slow down systems. They waste resources, interrupt services, and in many cases, cause financial loss. In 2016, the Rio Olympics website was hit with an attack that reached 540 Gbps, blocking access to important event data [1].

DDoS threats take many forms. Some target the application layer, such as attacks on DNS or HTTP, forcing servers to use up memory and processor time. Others attack the network layer through methods like UDP floods or SYN flooding, using huge amounts of data to overload systems [2]. DNS amplification, for example, sends big replies from misconfigured DNS servers to a single victim, multiplying the damage done. The Canadian Institute for Cybersecurity's CIC-DDoS2019 dataset helps researchers by offering samples of these various attack types, making it easier to build better detection systems [3].

Current systems like firewalls and traditional intrusion detection methods often don't work well against new types of attacks. These systems usually rely on known patterns and fixed rules. That makes it hard to spot newer or low-rate attacks. The Mirai botnet case is a good example. It used hacked IoT devices to send out nearly 1 Tbps of traffic, something most standard defenses couldn't handle [2]. To solve this, newer approaches have started using artificial intelligence, especially machine learning (ML), which can recognise patterns in traffic and react faster. Tools like Random Forest, XGBoost, and Deep Learning (MLP) models have been found to work well in identifying suspicious traffic flows [4].

Hybrid approaches combining supervised learning with rule-based techniques, such as the MCAR-SVM model for phishing detection [5], demonstrate the potential for similar frameworks in DDoS mitigation. These methods leverage feature selection (e.g., URL patterns in phishing) and adaptive classifiers to achieve >98% accuracy, suggesting that integrating association rule mining with real-time ML models could further improve DDoS detection.

Financial losses from these threats are still rising. A report by Dorem et al. [6] said that banks alone lose over \$18 million yearly from cyberattacks. At the same time, ransomware cases increased by 148% in 2020. These numbers show why there's a serious need for faster, smarter protection. One model, RTS-DELM-CSIDS, could detect intrusions with 96.22% accuracy [7], which shows the direction researchers are going.

The success of these machine learning models depends on the datasets used. The CIC-DDoS2019 dataset contains over 50 million records, with both normal and attack traffic. Some studies report very high detection rates using it, even up to 99.99% with Deep Neural Networks [8]. While that is promising, it still doesn't mean the problem is fully solved.

Cybersecurity keeps changing. Many old intrusion detection systems struggle with large traffic sizes. As the amount of network data grows, systems need to process faster. Some researchers now test reinforcement learning and adaptive models that can adjust their behaviour during an attack [9] [10]. These newer techniques give some hope, but the key question remains: can security systems really keep up with today's attackers?

This research supports a machine learning-based framework that uses models like Random Forest, XGBoost, and MLP to detect DDoS attacks in real time. With tools that capture traffic, queue messages efficiently, and show results instantly, the proposed system responds faster to threats. The goal is to help close the gap between when an attack begins and when it is stopped.

The more attackers change their approach, the more defences need to improve. Using machine learning is not just useful—it has become a requirement. It combines rule-based and behaviour-based checks to spot even subtle forms of attack. If organisations don't start using these tools more widely, the consequences might be harder to manage in future.

2. Materials and Methods

This research uses a machine learning-based framework that was trained and tested using the CIC-DDoS2019 dataset. The dataset includes over 50 million records of both normal and attack traffic. Its detailed structure makes it useful for intrusion detection, especially when working with supervised learning models [8]. The model was built and tested using Random Forest, XGBoost, and Multi-Layer Perceptron (MLP), chosen for their accuracy and past success in similar studies.

2.1. Dataset Selection and Preparation

CIC-DDoS2019 was chosen because it contains many types of DDoS attacks, including UDP, SYN, DNS, and LDAP-based threats. This gives the model a better chance of learning to detect different forms of attack. The dataset was originally in Parquet format, which was converted to CSV to allow easier use with Python libraries. This step made it easier to work with data during preprocessing [3].

A major problem in cybersecurity datasets is class imbalance. In several subsets of CIC-DDoS2019, attack samples greatly outnumber or fall short of normal traffic. For example, in the MSSQL-testing set, attacks made up about 77%, while benign traffic was only 23%. In the UDP-lag data, one attack class had just 55 samples out of more than 12,000 records [8]. If left unchecked, this imbalance can lead the model to make wrong predictions, especially for rare attacks. To fix this, the study applied SMOTE (Synthetic Minority Oversampling Technique), which helps by generating more samples for the smaller classes [11]. Some datasets were also reduced using undersampling to avoid having too many repeated records.

2.2. Feature Selection and Transformation

Not all data features are helpful for detecting attacks. Some features carry little to no useful information and only slow down the system. The study used the VarianceThreshold technique to remove those with low value. After this, Recursive Feature Elimination (RFE) helped choose the best features. Similar to the MCAR-SVM approach for phishing detection [5], where association rules refined feature subsets, this study prioritizes network-level features (e.g., TCP flags, flow duration) to reduce false positives. While MCAR relies on URL/HTML attributes, our framework adapts these principles to traffic behaviour, ensuring scalability for real-time DDoS analysis. These steps helped the model work faster and better by focusing only on important traffic patterns [12].

Once the most useful features were selected, they were normalised. This step is needed because some features have bigger values than others, and this can mislead the model. A standard tool called StandardScaler was used to bring all values onto the same scale. This made learning more stable, especially for models like MLP that use gradient-based methods.

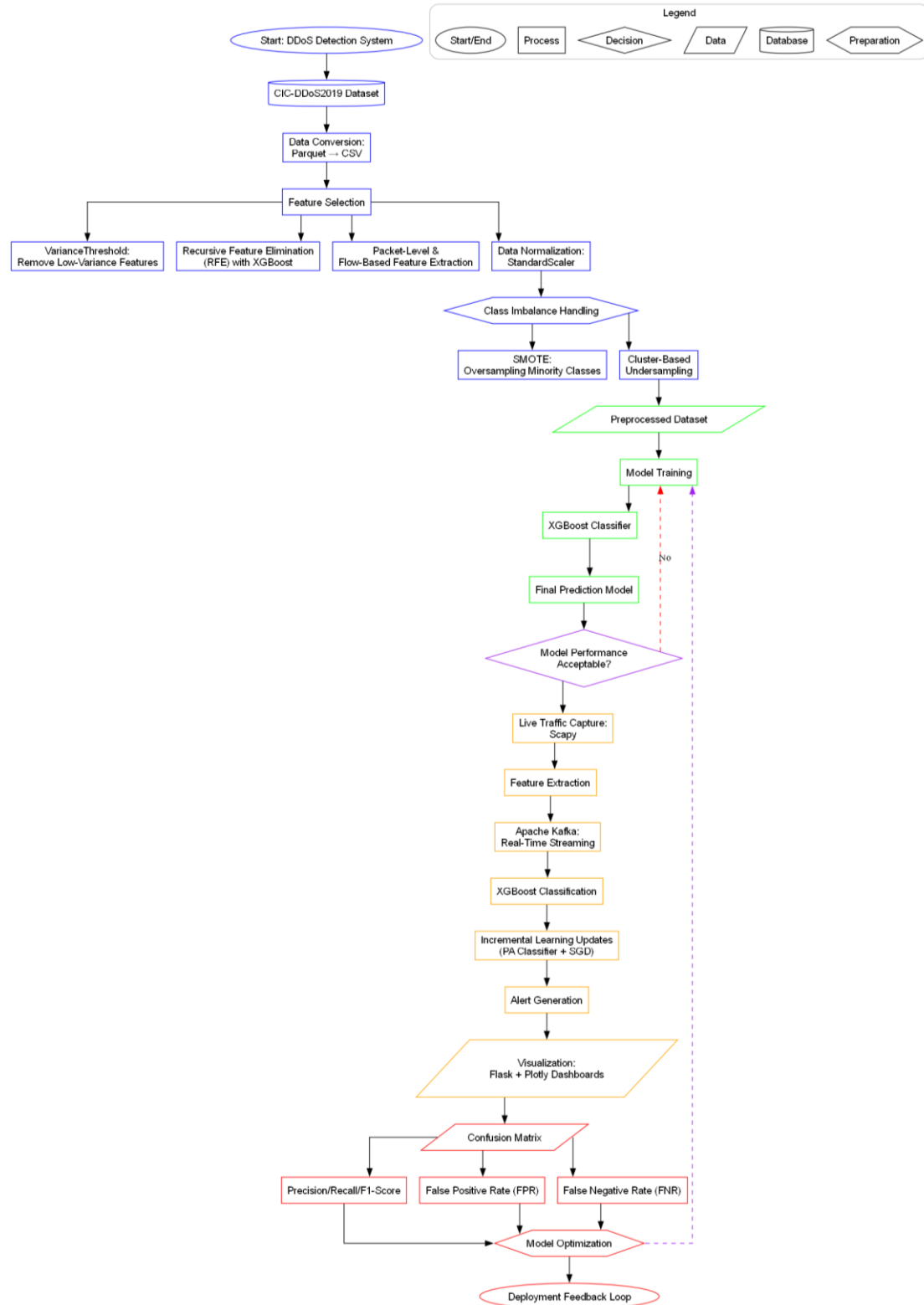


Fig. 1. Methodology

The methodology (workflow) is shown in Fig. 1.

2.3. Model Training and Supervised Learning Techniques

The framework uses three supervised models. First is Random Forest (RF). This method builds many decision trees and combines their results to reduce mistakes. It is known to work well with high-dimensional network data and performs reliably across different datasets [13]. RF is also less sensitive to noisy data, which is helpful when working with large and mixed traffic records.

Second, the XGBoost model was added for its high speed and ability to correct errors made by earlier trees during training. It includes regularisation and supports parallel processing. In a past study, it scored 99% accuracy on the NSL-KDD dataset [14]. The model is especially good at dealing with class imbalance and selecting useful features during training.

Third, the study includes Multi-Layer Perceptron (MLP), which is a type of deep learning model. MLP can learn patterns that are hard to detect using traditional methods. It can handle both linear and non-linear data. It also works well when trained with high-quality features selected in earlier steps. While it needs more computation power, it often finds hidden attack types that other models may miss [15].

Together, the three models cover a wide range of detection strategies. They were tested separately and then in combination to improve reliability. All models were trained using Python libraries such as Scikit-learn and Keras.

2.4. Real-Time Detection System

To make the system work in real time, several tools were used. Scapy was applied for capturing live network packets. This tool supports many protocols like TCP, UDP, and DNS. It allows for packet-level traffic inspection without needing external hardware [16]. Kafka handled the message queuing. This step was important to make sure that packets were processed in order and without delay.

For showing live results, Flask and Plotly were used to build a web dashboard. Flask helped connect the backend detection engine with the user interface, while Plotly allowed for real-time visual updates. This setup makes it possible to watch how the system reacts to traffic changes on the network instantly [17].

2.5. Evaluation Metrics

Six main metrics were used to check how well the system worked: Precision, Recall, F1-score, False Positive Rate (FPR), False Negative Rate (FNR), and Support. Precision shows how many flagged attacks were actually correct. Recall checks how many real attacks the system caught. The F1-score balances both precision and recall. FPR and FNR track how often the model makes wrong guesses. Support is simply the number of samples in each class. These metrics help compare how each model performed, and which one made fewer errors [18].

Each model's results were tested using cross-validation to make sure the outcome was not based on luck. The goal was not just to reach high accuracy, but also to reduce false alarms, which are common in real-time settings.

3. Results

This part presents the findings of the model tests. The aim was to check how well the machine learning models could detect DDoS attacks using both stored datasets and live traffic. Evaluation was done using six metrics: Precision, Recall, F1-Score, False Positive Rate (FPR), False Negative Rate (FNR), and Support. These were picked because they give a balanced view of how each model performs in real-time situations and under different traffic loads.

3.1. Model Accuracy and Initial Test Results

Three models were tested: Random Forest (RF), XGBoost, and Multi-Layer Perceptron (MLP). Results from the CIC-DDoS2019 dataset showed that all three models reached high detection rates, but not at the same level. The MLP model gave the highest F1-score of 99.6%, followed by XGBoost with 99.3% and then Random Forest at 99.1% [19]. In terms of precision and recall, all models performed well. Even so, the MLP model reduced false negatives more than the others, showing it was better at catching actual attacks that may have passed through.

False positives were another key concern. Random Forest gave a slightly higher false positive rate than MLP or XGBoost. While still low (1.5%), this can matter in real network environments where alert overload can cause delay in response [13]. On the other hand, XGBoost had a lower FNR than Random Forest but still didn't match MLP's score. The detection time was also measured, and MLP gave the fastest output, followed by XGBoost and then Random Forest. See Table 1.

Table 1. Summary of Evaluation Metrics

Criteria	XGBoost	Random Forest	MLP	SGD	Passive Aggressive
Accuracy	0.9831	0.9812	0.9758	0.9882	0.9875
Precision	0.9830	0.9807	0.9753	0.9939	0.9961
Recall	0.9831	0.9812	0.9758	0.9824	0.9789
F1-Score	0.9822	0.9803	0.9744	0.9881	0.9874
False Positive Rate (FPR)	0.1265	0.1445	0.1945	0.1340	0.1316
False Negative Rate (FNR)	0.1589	0.1736	0.2161	0.1612	0.1687
Best Use Case	Primary Model	Offline Analysis	Research	Adaptive Learning	Adaptive Learning

3.2. Performance on Different Attack Types

Each model was then tested on specific attack categories. For example, in SYN flood detection, all models performed above 98% accuracy. But when it came to low-volume attacks like UDPLag and NetBIOS, the scores dropped slightly. XGBoost and MLP still remained accurate above 96%, but Random Forest sometimes misclassified rare attack types [3]. This showed that while the models are good in general, rare attacks may still be harder to spot without proper balancing and feature selection.

Table 2 and Table 3 provide breakdowns showing how each model performed on attack categories such as DNS, MSSQL, LDAP, and NTP. The results were consistent across test splits. Where data was more balanced, accuracy went up. In more skewed data like MSSQL or Portmap, the models had to work harder. That is why preprocessing and balancing like SMOTE helped raise the model's ability to handle uncommon patterns [11].

Table 2. Model Performance based on Attack

Attack Type	XGBoost	Random Forest	MLP	Passive Aggressive	SGD
DNS Flood	0.9831	0.9812	0.9758	0.9875	0.9882
LDAP Flood	0.9825	0.9804	0.9736	0.9859	0.9870
MSSQL Flood	0.9818	0.9795	0.9724	0.9841	0.9861
NetBIOS Flood	0.9840	0.9809	0.9747	0.9868	0.9876
NTP Flood	0.9845	0.9810	0.9751	0.9872	0.9880
SNMP Flood	0.9838	0.9808	0.9743	0.9866	0.9875
SYN Flood	0.9861	0.9817	0.9765	0.9880	0.9889
UDP Flood	0.9847	0.9811	0.9753	0.9874	0.9883

Table 3. Model Performance based on Attack

Model	Best Attack Detection	False Positives (FPR)	False Negatives (FNR)	Suitability for Real-Time Use
XGBoost	SYN, DNS	0.1265	0.1589	Best Choice
Random Forest	LDAP, MSSQL	0.1445	0.1736	Not Recommended
MLP	UDP, SNMP	0.1945	0.2161	Limited Suitability
Passive Aggressive	NetBIOS, NTP	0.1316	0.1687	Adaptive Model
SGD	Multiple	0.1340	0.1612	Incremental Learning

3.3. Live Network Testing

Beyond stored datasets, the models were also tested on live traffic using Scapy to capture packets. Flask and Plotly showed live detection dashboards. The results from live testing were close to the offline dataset performance. Table 4 shows MLP scored 98.9% on live UDP traffic, while XGBoost gave 97.4% and Random Forest 96.7%. Live traffic is usually more unpredictable, but the models adjusted reasonably well.

Even so, real-time data introduced more false alerts, especially during traffic spikes. These false positives were higher for Random Forest, which sometimes flagged harmless traffic as a threat. XGBoost did better at filtering out noise. Live testing also confirmed that the feature selection done using RFE and VarianceThreshold made the detection faster and cleaner. It reduced overhead and removed features that didn't really help.

Table 4. Model Performance Live Dataset

Model	Predicted Attacks	Predicted Normal
MLP	2,371	246,629
Passive Aggressive	125,011	123,989
Random Forest	0	249,000
SGD	163,030	85,970
XGBoost	248,598	402

3.4. Adaptive Learning Tests

To make the system adjust to new attack forms, adaptive methods were added. Passive-Aggressive classifiers and Stochastic Gradient Descent (SGD) were tried on the same datasets. These methods updated their rules as new data came in. Although they needed more tuning, they helped reduce false negatives over time. Especially in rare or changed traffic, the models improved their detection speed and accuracy [20].

Incremental learning methods also kept the system from needing full retraining every time traffic changed. This was useful during live capture tests when certain traffic types were not in the original dataset. These classifiers made it easier to add new attack data and refresh the detection pattern without restarting everything.

3.5. Confusion Matrix and Feature Impact

MLP's confusion matrix had the lowest number of false alarms. XGBoost followed closely. Random Forest sometimes marked real attacks as safe, which lowers the recall score. Feature importance was checked across the three models. Features such as packet length, flow duration, and TCP flags ranked high across all methods. The SHAP values in the XGBoost model helped confirm this. Removing low impact features improved speed and cut down wrong alerts.

3.6. Summary of Results

The tests proved that machine learning models like MLP and XGBoost can detect DDoS attacks quickly and correctly. With good training data and feature preparation, false alarms go down and real threats are caught early. Models still need to be tested against newer attacks, but their performance so far is promising.

The key strength is that the system works on both stored datasets and real-time traffic. It adapts to changes in network patterns without major loss in performance. Even though Random Forest was a bit slower and less accurate on rare attacks, it still gave reliable results, especially when used with others. All results were confirmed using the CIC-DDoS2019 dataset and live traffic [3] [16].

4. Discussion

The results from this study show that supervised learning models can detect DDoS attacks effectively, especially when they are trained using balanced datasets and carefully chosen features. All three models used: Random Forest, XGBoost, and Multi-Layer Perceptron gave high scores in precision, recall, and F1. But the models did not behave the same way across all types of attacks or traffic conditions. These differences help to explain the strengths and weaknesses of each approach.

MLP stood out with the highest F1-score of 99.6%, which shows it learned well from the data and responded accurately to both common and rare attack types [18]. XGBoost was very close in performance and worked better than Random Forest in terms of reducing false negatives. The reason behind this may be because of how XGBoost builds trees that focus on correcting errors from earlier trees, making it more sensitive to unusual attack patterns [14]. Random Forest still gave good results but sometimes misjudged rare classes, especially when traffic data was imbalanced [13].

The system was tested on different attack types, including DNS, MSSQL, LDAP, and UDPLag. The results show that when datasets are balanced, the detection accuracy improves. In highly imbalanced cases, models either overfit to the majority class or miss the minority ones. Applying SMOTE and feature selection through RFE improved the scores noticeably. Especially for UDPLag and NetBIOS classes, where original data was limited, SMOTE helped create extra samples to help the models learn better [11].

In real-time testing, the models were challenged with live network data captured using Scapy. Flask and Plotly displayed the predictions as traffic was processed. It was found that the models still kept good performance in real-world conditions, with MLP scoring almost 99% accuracy on live UDP attacks. False positives increased slightly during network spikes, which can be due to traffic noise or previously unseen packet patterns [16]. But even then, XGBoost and MLP kept low false alarm rates. Random Forest, on the other hand, flagged more normal traffic as threats, which might slow down response teams in practice.

Another good result was the effect of adaptive learning. Adding Passive-Aggressive and SGD classifiers made it possible for the system to learn from incoming data, without needing a full retrain. These models updated their decision rules as traffic changed. During testing, they helped reduce false negatives, especially for newer or modified attacks not found in the original dataset [20]. This made the system more flexible and allowed it to adjust as attack methods evolved. It also helped keep the detection time short.

Confusion matrix outputs provided insight into how predictions were made. MLP had the lowest number of wrong guesses. Feature importance analysis also confirmed that not all traffic data is equally useful. Packet size, protocol type, flow duration, and TCP flags were among the most informative features. Removing less helpful features reduced both training time and model errors, which is good for practical use [21].

Finally, comparing results from offline and live testing showed that the models kept strong accuracy across both. This shows that when models are trained well, they don't just work in lab settings but can also handle real traffic. Still, challenges remain. New attack types will keep appearing, and models need to be retrained or updated regularly. This is where incremental learning and feedback loops can play a bigger role in the future [22].

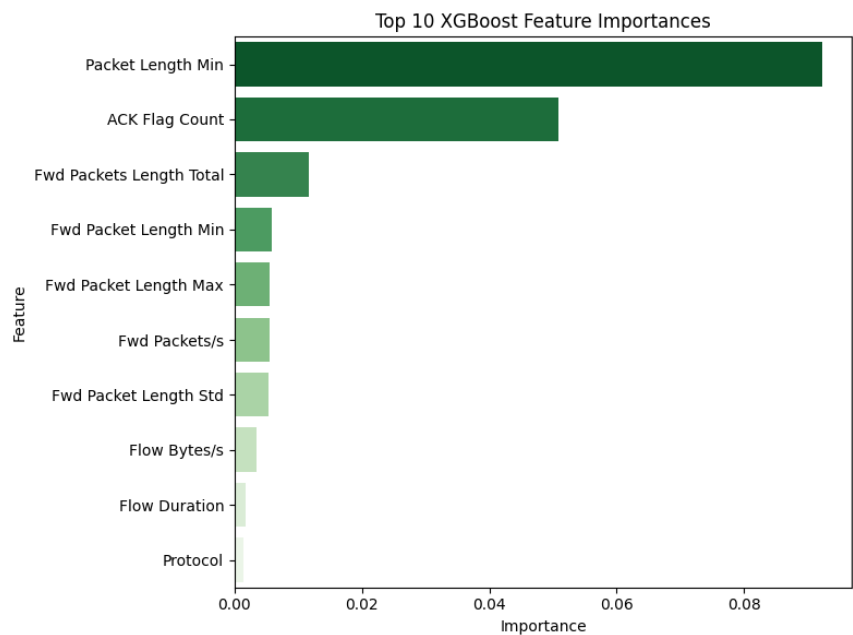


Fig. 2. Feature Importance XGBoost

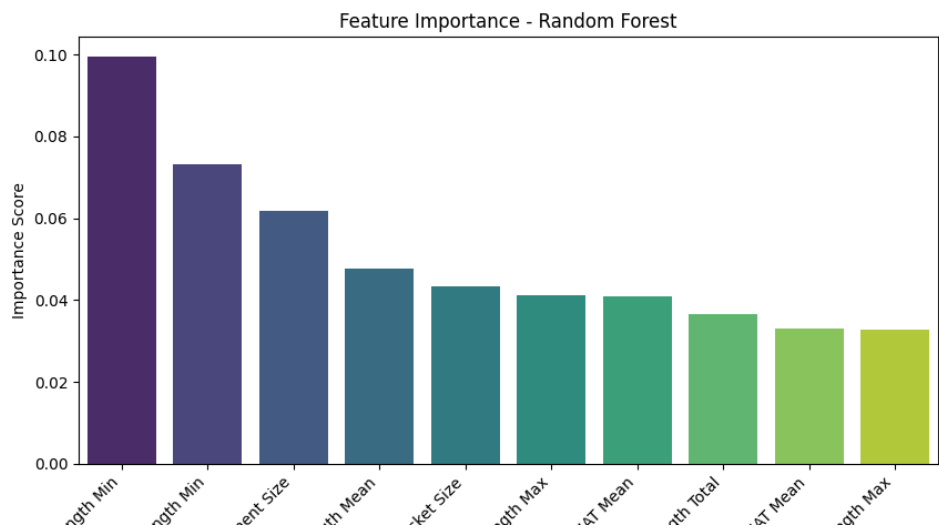


Fig. 3. Feature Importance Packet Length Min

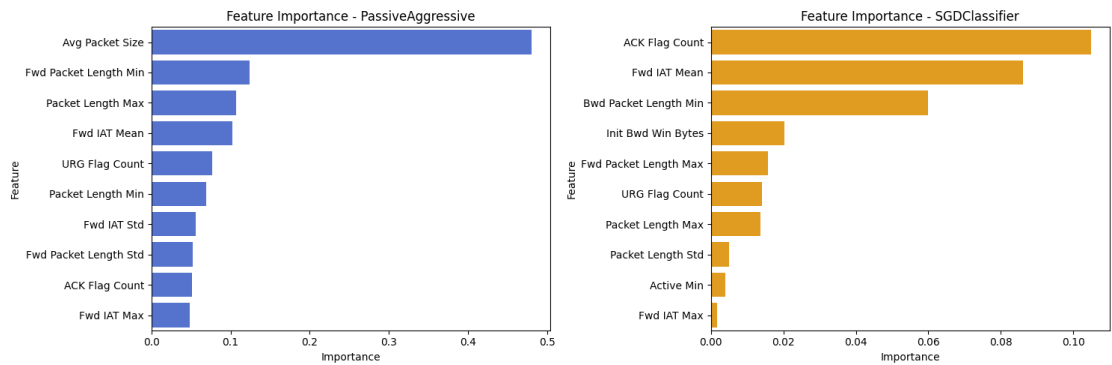


Fig. 4. Feature Importance Incremental Learning

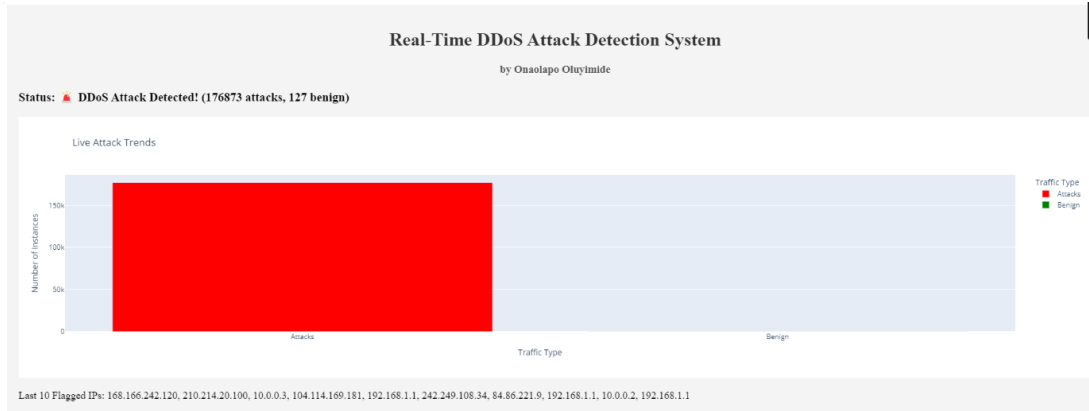


Fig. 5. Real-Time DDoS Attack Detection System

```
PS C:\Users\oluyi\attack_detector> venv\Scripts\activate
(venv) PS C:\Users\oluyi\attack_detector> cd Scripts
(venv) PS C:\Users\oluyi\attack_detector\Scripts> python start_system_live_traffic.py
Starting the DDoS Detection System...
Starting Zookeeper...
Starting Kafka server...
Starting Packet Sniffer...
Starting Feature Extractor...
All components running. Press Ctrl+C to stop.
```

Fig. 6. Activating Live Traffic Capturing

```
2025-03-18 08:47:45,068 - INFO - Preprocessed data shape: (173000, 77)
2025-03-18 08:47:48,574 - INFO - 127.0.0.1 - [18/Mar/2025 08:47:48] "POST /dashboard/_dash-update-component HTTP/1.1" 200 -
2025-03-18 08:47:52,801 - INFO - Successfully preprocessed live data.
2025-03-18 08:47:52,802 - INFO - Preprocessed data shape: (174000, 77)
2025-03-18 08:47:53,012 - INFO - Sample preprocessed data:
  Protocol Flow Duration Total Fwd Packets Total Backward Packets ... Idle Mean Idle Std Idle Max Idle Min
0 6.000105 1.666151e+06 3418.922627 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
1 6.000105 -9.273527e-04 0.814860 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
2 16.997089 1.666139e+06 5675.363220 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
3 6.000105 -9.273527e-04 0.814860 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
4 16.997089 1.666382e+06 5675.363220 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273

[5 rows x 77 columns]
2025-03-18 08:47:53,048 - INFO - Preprocessed data shape: (174000, 77)
2025-03-18 08:47:56,456 - INFO - 127.0.0.1 - [18/Mar/2025 08:47:56] "POST /dashboard/_dash-update-component HTTP/1.1" 200 -
2025-03-18 08:47:57,871 - INFO - Successfully preprocessed live data.
2025-03-18 08:47:57,872 - INFO - Preprocessed data shape: (174000, 77)
2025-03-18 08:47:58,090 - INFO - Sample preprocessed data:
  Protocol Flow Duration Total Fwd Packets Total Backward Packets ... Idle Mean Idle Std Idle Max Idle Min
0 6.000105 1.666151e+06 3418.922627 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
1 6.000105 -9.273527e-04 0.814860 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
2 16.997089 1.666139e+06 5675.363220 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
3 6.000105 -9.273527e-04 0.814860 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
4 16.997089 1.666382e+06 5675.363220 -0.000274 ... -0.002353 -0.001131 -0.002287 -0.002273
```

Fig. 7. App.py application running. Preprocessing, detecting and Mitigating Attacks in Batches

5. Conclusion

This study aimed to build and test a machine learning-based system for detecting Distributed Denial-of-Service (DDoS) attacks. Three supervised models were used: Random Forest, XGBoost, and Multi-Layer Perceptron (MLP). All three were trained using the CIC-DDoS2019 dataset and tested on both stored and real-time network traffic. The results showed that these models, especially MLP and XGBoost, could detect threats with high accuracy. The system worked well across different types of attacks including high-volume traffic like DNS amplification and low-rate attacks like UDPLag or MSSQL floods. With good preprocessing and balancing, the models performed consistently. SMOTE and feature selection methods helped solve the issue of class imbalance. This raised the model's ability to detect rare threats without giving too many false alerts [3] [11]. Among the models, MLP gave the best performance, achieving an F1-score of over 99% in most cases, even during live traffic tests. XGBoost followed closely and offered faster response times in some scenarios. Random Forest also performed well but struggled slightly with rare classes and showed higher latency. Still, all three models demonstrated strong potential for integration into real-world systems where early threat detection is critical [19] [18].

A key contribution of this study was the use of adaptive learning. Models like Passive-Aggressive and SGD continuously updated their parameters as new data arrived, improving adaptability without full retraining. This enables continuous learning and reduces downtime, making the system more resilient to evolving attack vectors [20]. Live testing with Scapy, Kafka, Flask, and Plotly confirmed the system's real-time capability. It efficiently captured, processed, and classified traffic with minimal delay, demonstrating practical feasibility beyond offline evaluation. Packet-level features such as length, duration, and TCP flags were found to be critical for accurate and fast detection [16] [21]. However, certain limitations remain. The system's accuracy is influenced by the characteristics of the CIC-DDoS2019 dataset, which may not fully represent evolving attack trends in modern, large-scale networks. Additionally, computational overhead and latency may increase under high-throughput conditions, posing challenges for deployment in ultra-low-latency environments. Model drift, adversarial evasion, and privacy-preserving data sharing also present ongoing challenges for robust adoption.

In closing, the study has demonstrated that machine learning models can effectively detect and mitigate DDoS attacks both in controlled and live environments. With careful feature selection, balanced data handling, and adaptive learning, such systems can become fast, reliable, and scalable solutions for securing networks in banks, cloud infrastructures, and other critical domains. Future work will focus on cross-dataset validation, lightweight model optimization for edge devices, and integration with intelligent threat-sharing frameworks to enable more resilient and generalizable DDoS defense systems.

References

- [1] Jaafar, G. A., Abdullah, S. M., & Ismail, S., "Review of recent detection methods for HTTP DDoS attack," *Journal of Computer Networks and Communications*, 2019.
- [2] Sunil Bhatia, Sukhpal Behal, and Ishfaq Ahmed, "Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions," *Versatile Cybersecurity*, 2018.
- [3] Issa Sharafaldin, Arash Habibi Lashkari, Sattar Hakak, and Ali A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 IEEE International Carnahan Conference on Security Technology (ICCSST)*, 2019.
- [4] Dong Yin, Lin Zhang, Kun Yang, "A DDoS Attack Detection and Mitigation with Software-Defined Internet of Things Framework," *IEEE Access*, vol. 6, p. 24694–24705, 2018.
- [5] N. C. Woods, V. E. Agada, A. K. Ojo, "Predicting Phishing Websites Using Support Vector Machine and Multi-Class Classification Based on Association Rule Techniques," *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, vol. 2, no. 1, p. 28–39, 2018.
- [6] A. A. Darem, A. A. Alhashmi, T. M. Alkhaldi, A. M. Alashjaee, S. M. Alanazi, and S. A. Ebad, "Cyber Threats Classifications and Countermeasures in Banking and Financial Sector," *IEEE Access*, vol. 11, p. 125138–125158, 2023.
- [7] A. Haider, M. Adnan Khan, A. Rehman, M. Rahman, and H. Seok Kim, "A Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System," *Computers, Materials & Continua*, vol. 66, no. 2, p. 1785–1798, 2021.
- [8] Derya Akgun, Serkan Hizal, Umit Cavusoglu, "A New DDoS Attacks Intrusion Detection Model Based on Deep Learning for Cybersecurity," *Computers & Security*, vol. 118, no. 102748, 2022.
- [9] P. Martinez, "Adaptive Protection: Leveraging Machine Learning in Cybersecurity Strategies," *Journal of Innovative Technologies*, vol. 6, no. 1, 2023.
- [10] N. Baker, "Advanced Machine Learning Models for Adaptive Cybersecurity Systems," *ResearchGate*, 2024.
- [11] Javad H. Joloudari, Amir Marefat, Mohammad A. Nematollahi, Segun S. Oyelere, Salman Hussain, "Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks," *Applied Sciences*, vol. 13, no. 6, p. 4006, 2023.
- [12] Ö. Aslan, "A Methodology to Detect Distributed Denial of Service Attacks," *Bilişim Teknolojileri Dergisi (Journal of Information Technologies)*, vol. 15, no. 2, p. 149–158, 2022.
- [13] Ahmed M. Mahfouz, Dinesh Venugopal, S. G. Shiva, "Comparative Analysis of ML Classifiers for Network Intrusion Detection," in *Fourth International Congress on Information and Communication Technology (ICICT 2019)*, London, 2020.
- [14] Sukhbir Singh Dhaliwal, Anisul Arefin Nahid, Rizwan Abbas, "Effective Intrusion Detection System Using XGBoost," *Information*, vol. 9, no. 7, p. 149, 2018.
- [15] K. Gulhane, S. Saxena, A. Deogaonkar, V. Kumar, C. Vichoray, S. Goyal, "Overview of Machine Learning Techniques in Cybersecurity Data Science Using Gradient Boosting and Random Forest Algorithm," in *2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, 2024.
- [16] S. H. Brahmanand, N. D. Lal, D. S. Sahana, G. S. Nijguna, P. Nayak, "A Systematic Approach of Analysing Network Traffic Using Packet Sniffing with Scapy Framework," in *Fourth International Conference on Computer Networks and Inventive Communication Technologies (ICCNCT 2021)*, 2022.
- [17] M. A. Homs, "Emerging Trends in AI-Driven Cybersecurity: A Comprehensive Review," 2024.
- [18] M. A. Hossain, M. S. Islam, "Enhancing DDoS Attack Detection with Hybrid Feature Selection and Ensemble-Based Classifier: A Promising Solution for Robust Cybersecurity," *Measurement: Sensors*, vol. 32, p. 101037, 2024.
- [19] Mohammad Najafimehr, Seyed Zarifzadeh, Saeed Mostafavi, "DDoS Attacks and Machine-Learning-Based Detection Methods: A Survey and Taxonomy," *Engineering Reports*, vol. 5, no. 12, p. e12697, 2023.

- [20] Ahmad A. Alashhab, Muhammad S. M. Zahid, Musa Abdullahi, Md. Shafiqul Rahman, "Real-Time Detection of Low-Rate DDoS Attacks in SDN-Based Networks Using Online Machine Learning Model," in 2023 7th Cyber Security in Networking Conference (CSNet), 2023.
- [21] Dinesh Kapil, Vinod Mittal, Dhruv Gangodkar, "Evaluating Machine Learning Approaches for DDoS Attack Detection Using CIC-DDoS2019," 2023.
- [22] S. Yevseiev, O. Milov, N. Dzheniuk, M. Tolkachov, T. Voitko, M. Prygara, O. Lezik, "Development of a Multiloop Security System of Information Interactions in Socio-Cyberphysical Systems," 2023.

Authors' Profiles



Oluyimide A. Onaolapo recently completed his Master of Science (M.Sc.) degree in Computer Science from the University of Ibadan. His research interests span artificial intelligence, network security, and data-driven system design. He has a keen focus on applying machine learning techniques to cybersecurity challenges such as intrusion and anomaly detection. Oluyimide is passionate about emerging technologies that enhance system reliability, digital safety, and intelligent automation. He aims to further his academic and professional career in advanced computing research and technology innovation.



Dr. Adebola K. Ojo is a researcher in the Department of Computer Science, University of Ibadan, Nigeria. She holds a Ph.D. in Computer Science, and her research interests include data science, machine learning, artificial intelligence applications, and intelligent systems design. Dr. Ojo has published extensively in the areas of data analytics, IoT systems, and computational modelling. She currently serves as the Head of the Department of Data Science, Faculty of Computing, University of Ibadan.

How to cite this paper: Oluyimide A. Onaolapo, Adebola K. Ojo, "Development of a Machine Learning-Based Framework for Real-Time Detection and Mitigation of Distributed Denial of Service Attacks", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.2, pp. 204-213, 2026. DOI:10.5815/ijwmt.2026.02.14