Bima Journal of Science and Technology, Vol. 9(2A) Jun, 2025 ISSN: 2536-6041

DOI: 10.64290/bimagombe.v9i2A.1061

Detecting DDOS Attacks Amidst Flash Crowds Using Machine Learning

Maryam Abdulkadir*[1], Muhammad Aminu Ahmad[1], Ahmad Abubakar Aliyu[1], Sa'adatu Abdulkadir[2], Mohammed Ibrahim[2], Adamu Abdullahi[2] and Abubakar Muazu Ahmed[2]

[1]Department of Informatics, Faculty of Computing, Kaduna State University,
[2]Department of Secure Computing, Faculty of Computing, Kaduna State University
Kaduna – Nigeria.
Corresponding Author: maryamaykay@gmail.com

ABSTRACT

In the digital landscape, distinguishing genuine flash crowds from Distributed Denial of Service (DDoS) attacks remains a critical challenge. Flash crowds, characterized by sudden surges of legitimate traffic, often exhibit behavioral patterns similar to DDoS attacks, leading to false positives in detection systems. This research proposes a robust machine learning-based approach for setting apart flash crowds from DDoS attacks, using a multi-classification methodology. The implemented system leverages a Random Forest classifier trained on network traffic data, focusing on key features such as packet size, flow duration, and transmission rates. The dataset is pre-processed to handle anomalies and class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). Evaluation metrics such as accuracy, precision, recall, and F1-score, demonstrated the system's effectiveness, achieving over 99% accuracy in distinguishing benign traffic from malicious attacks. Additionally, advanced visualizations such as confusion matrices and ROC curves provided actionable insights into the model performance. The new model's scalability and high accuracy make it a promising solution for real-time applications in network anomaly detection, ensuring minimal disruption to legitimate user activities. This study contributes to the ongoing efforts to enhance cyber-security defenses against evolving DDoS threats while preserving the accessibility of web services during legitimate traffic surges.

Keywords: Flash Crowd, DDoS Detection, Random Forest Classifier, SMOTE

INTRODUCTION

Web services are critical for business operations, communication, and social interaction. However, they remain vulnerable to cyber threats, particularly DDoS attacks. These attacks overwhelm target systems by flooding them with excessive traffic, causing service disruptions or downtime for legitimate users. With attackers employing increasingly sophisticated methods, distinguishing DDoS attacks from legitimate traffic surges—such as flash crowds—has become more challenging. Flash crowds, characterized by sudden spikes in legitimate traffic due to events like product launches or viral content, often exhibit behavioral patterns similar to DDoS attacks, leading to false positives in traditional detection systems. Flash crowds refer to sudden increases in legitimate traffic caused by external factors such as breaking news, sporting events, or promotional campaigns. While beneficial for businesses, they strain network resources and complicate intrusion detection efforts. Studies have shown that flash crowds exhibit specific characteristics, such as longer flow durations and moderate byte rates, compared to shorter flows and higher byte rates typical of DDoS attacks Kalkan & Algin (2021).

Traditional DDoS detection methods rely heavily on signature-based detection or statistical thresholds, which struggle to adapt to evolving threats and dynamic traffic patterns. Furthermore, these methods

DOI: 10.64290/bimagombe.v9i2A.1061

frequently fail to distinguish between flash crowds and DDoS attacks, resulting in misclassification and unnecessary mitigation actions. There is a pressing need for intelligent systems capable of accurately identifying malicious traffic amidst legitimate surges without compromising service availability. DDoS attacks involve overwhelming a target system with malicious traffic, often generated via botnets. They differ from flash crowds in their intent and structure, typically featuring repetitive patterns and originating from fewer unique IP addresses.

Related Works

Several studies have explored techniques for differentiating between flash crowds and DDoS attacks. Some used entropy-based methods, for example, Gera et al. (2018) proposed using source address entropy and traffic cluster entropy to detect spoofed and non-spoofed DDoS attacks. However, these methods may produce false positives during flash crowd events. For machine learning approaches, Jisa David and Ciza Thomas (2021) introduced an efficient thresholding algorithm for detecting DDoS attacks, achieving over 97% accuracy. Similarly, Salah et al. (2023) combined entropy analysis with Q-learning to improve detection rates and reduce false positives.

For hybrid models, Marinova et al. (2020) developed an end-to-end network slicing framework for managing flash crowd scenarios, emphasizing adaptability and scalability. Their approach demonstrated rapid deployment times and effective resource allocation during emergencies. Despite advancements, there remains a gap in addressing low-rate DDoS attacks and handling class imbalances in traffic datasets. Previously published works related to this research which were found to be relevant to achieving the objectives of the work are reviewed in this chapter. Many researchers studied the properties of DDOS attacks and flash crowd traffic to separate the characteristics of both. Different dimensions were highlighted that are quite helpful when the two traffic events occur simultaneously. Flash crowds were classified in different studies and parameters were laid out to pinpoint if there was an attack hidden in those crowds.

Doshi. et al. (2018) tackled the escalating problem of IoT devices being hijacked into botnets for DDoS attacks. Rather than relying on generic network security measures, the authors proposed a targeted approach: training machine learning models to recognize the tell-tale signs of IoT attack traffic. They observed that IoT devices, unlike typical internet users, often exhibit predictable communication patterns. Praseed (2018) discussed the taxonomy of application layer distributed denial of service attacks. A review of the existing research directions and defense mechanisms has also been presented to bring out the different features used for detecting these attacks, and the different methods of detection. Few researchers mentioned reviews of papers related to discriminating DDOS attacks from the flash crowd. Most of the techniques use a static threshold value for detection. However, network activities and users' behaviors could vary over time which reduces detection accuracy. These approaches are not suitable for detecting low-rate DDOS attacks.

The report by Gera et al. (2018) addresses a critical issue in cybersecurity, the detection of spoofed and non-spoofed Distributed Denial-of-Service (DDoS) attacks while distinguishing them from flash crowds. The authors proposed an innovative methodology that leverages source address entropy and traffic cluster entropy to accurately identify these attacks. By focusing on the nuanced

DOI: 10.64290/bimagombe.v9i2A.1061

differences between legitimate high-traffic events like flash crowds and malicious DDoS activities, the study aimed to enhance network security measures. The solution incorporated thresholds for source and traffic entropy, which are adjusted using tolerance factors to balance detection rates and false positives. The human behavior modeling for defense against the flash crowd attacks was applied by differentiating bots from humans based on request dynamics, request semantics, and deception i.e. testing the clients' ability to ignore invisible content by Tandon, (2019) . Razumov et al. (2020) Created an emulation software tool that implemented a developed algorithm for detecting and blocking HTTP flood attacks. The method used a single filtering system with an unlimited number of devices and an increased number of proxies through infrastructure building. Biruk et al. (2020) argued that traditional detection methods often fall short, either by relying on limited features easily mimicked by attackers or by incorrectly assuming DDoS attacks originate from fewer IP addresses than flash crowds. Their proposed solution is a supervised machine learning approach that leverages a combination of five key features derived directly from web server logs i.e. request rate, page popularity, download rate, request inter-arrival time, and the ratio of successful requests. They hypothesize that this multi-faceted approach, focusing on application layer characteristics will be more robust and adaptable than existing methods. The research involved creating a combined dataset of flash crowd and DDoS attack traffic. For the flash crowd component, they utilized the well-established World Cup 98 dataset, a record of website traffic during the 1998 World Cup. To generate realistic DDoS attack data, they conducted simulated attacks on a locally hosted copy of the same website using the Bonesi attack tool.

In (2020) Marinova et al. introduced a novel end-to-end network slicing framework designed to address the unique demands of flash crowd events, particularly in emergencies. Recognizing that traditional networks struggle with the sudden surge in demand and critical communication needs during such events, the authors proposed a flexible and scalable architecture leveraging network virtualization and software control. The framework utilized a virtual resource manager (VRM) to efficiently allocate and manage resources, mapping cloud hardware resources (CPU, memory) to the demands of the wireless network.

A system to enhance password security by combining honeywords, decoy data, and IP tracking was proposed by Naik et al. (2023) . The core idea was to protect user accounts from password cracking and unauthorized access. Honeywords, or decoy passwords, are generated for each user account alongside their real password. The authors highlighted the increasing vulnerability of user accounts due to readily available password-cracking tools. They argue that traditional password protection methods are insufficient and that new mechanisms are needed. The honeyword mechanism serves as a tripwire, alerting the system to potential breaches. Decoy data, also referred to as "fog computing," is presented to the attacker after multiple failed login attempts, further confusing them and masking the real user data.

Salah et al. (2023) proposed a novel approach to detecting Distributed Denial of Service (DDoS) attacks in Software Defined Networks (SDNs) by combining entropy analysis and Q-learning, addressing the vulnerabilities of traditional methods that can lead to network-wide failures due to the centralized control paradigm of SDNs. The system, EQD (Entropy and Q-learning Detection), utilizes the statistical properties of network traffic,

DOI: 10.64290/bimagombe.v9i2A.1061

focusing on destination IP address entropy, to identify potentially malicious activity. By integrating Q-learning, a reinforcement learning technique, the system intelligently manages suspected traffic rather than simply blocking it, which helps reduce false positives and ensures minimal disruption to legitimate users. The authors evaluated the effectiveness of their approach through simulations using Mininet, comparing it with entropy-based detection methods in terms of throughput and detection time. The results demonstrate significant improvements, with EQD achieving up to a 50% increase in throughput by effectively handling suspected traffic and ensuring continuous service for legitimate users.

MATERIALS AND METHODS

This study aims to develop a machine learning-based system that effectively distinguishes between DDoS attacks, flash crowds, and benign traffic. Specific objectives include; Analyzing and preprocessing network traffic data to extract meaningful features and implementing a Random Forest classifier with feature-based thresholds for multi-class classification. Evaluating the model's performance using standard metrics such as accuracy, precision, recall, and F1-score. Comparing the proposed model against existing entropy-based approaches to highlight improvements. The methodology consists of two main phases: data preprocessing and model implementation. Data preprocessing involves cleaning, normalizing, and balancing the dataset, while model implementation focuses on training and evaluating the Random Forest classifier. The figure below shows the flow of the methodology used on the proposed system.

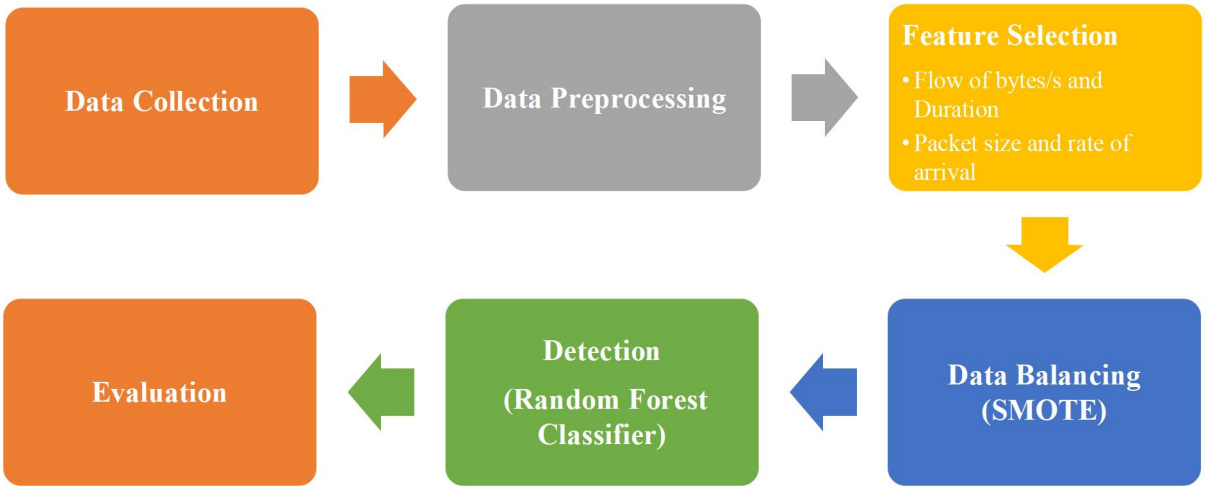

Figure 1: Flow chart of the Proposed System

System Description

In this research, a thresholding algorithm with a random forest classifier (RF) is opted for as the sole machine learning model, moving away from entropy-based detection method. The reason for this is due to the high accuracy when it comes to classification. Random

DOI: 10.64290/bimagombe.v9i2A.1061

Forest is an ensemble learning technique that combines multiple decision trees, reducing over-fitting and improving generalization. Unlike entropy-based thresholding, the Random Forest classifier can capture non-linear relationships within large network traffic datasets. Random Forest can generalize across different datasets, whereas the IDS ensemble method may require manual parameter tuning each time traffic characteristics change.

Dataset Preprocessing

Raw network traffic datasets often contain inconsistencies, missing values, or extreme outliers that can degrade model performance. Rows with missing or infinite values are dropped and feature scaling is performed using the StandardScaler to ensure uniform contribution from all features. To ensure robustness, the CICIDS 2017 dataset underwent a systematic preprocessing pipeline comprising four stages: data cleaning, handling missing values, feature selection, and class imbalance correction.

Data Cleaning

First, column names were standardized to lowercase, and special characters (e.g., hyphens, spaces) were removed to ensure compatibility with machine learning libraries. Invalid entries (e.g., negative flow durations, non-numeric values) were filtered out. This step eliminated 0.3% of the dataset, ensuring minimal loss of critical information while maintaining data integrity.

Handling Missing Values

Missing values (NaN) and infinite entries (e.g., division-by-zero errors in derived metrics like packet flow rate were identified and removed. Approximately 1.2% of rows contained missing values, primarily due to incomplete packet captures. Imputation methods (e.g., mean substitution) were deemed unsuitable, as they could introduce bias in time-sensitive network metrics. Thus, rows with missing or infinite values were dropped, retaining 98.5% of the original dataset.

Feature Selection

To ensure uniform feature contribution, standardization was applied using the Standard Scaler from scikit-learn, which transforms features to have a mean of 0 and standard deviation of 1. While tree-based models like Random Forest are scale-invariant, scaling improves interoperability with other models (e.g., SVMs or neural networks) and accelerates convergence in gradient-based algorithms. The key network features selected for the classification include:

. Flow duration represents the total elapsed time from the beginning of the first packet to the end of the last packet within a defined network flow. This feature is fundamental for understanding the temporal aspects of network communication as variations in flow duration can indicate different types of network activity.
. Flow of Bytes/s, or flow throughput, measures the average rate at which data is transferred within a network flow. This provides insight into the bandwidth utilization of a flow. High byte rates indicate large data transfers, while low rates might suggest low-bandwidth applications or network congestion.
. Packet Length Variance, which quantifies the degree of variability in the sizes of packets within a network flow. High variance indicates that packets are of widely different sizes, while low variance suggests that packets are relatively uniform. Variations in packet length can be influenced by factors such as application type, network protocols, and fragmentation.
. TCP Flags: Indicators of connection states (SYN, ACK, FIN).

The features are assigned dynamic thresholding values defined to classify each traffic types:

DOI: 10.64290/bimagombe.v9i2A.1061

***FCs**: Long flow duration (>1,000,000 ms) and low byte rate (<100,000 bytes/s).*
***DDoS Attack:** Short flow duration (≤1,000,000 ms) and high byte rate (>100,000 bytes/s).*

Addressing Class Imbalance

To mitigate the effects of class imbalance, the Synthetic Minority Over-Sampling Technique (SMOTE) was applied. SMOTE generates synthetic samples for underrepresented classes, improving model fairness and generalizability. This technique improves model training by ensuring that the classifier is exposed to sufficient examples of all classes.

From imblearn. over_sampling import SMOTEsmote = SMOTE (random_state=42) X_resampled, y_resampled = smote.fit_resample (X, y).

Model Implementation

To validate the choice of RF, preliminary experiments were conducted with alternative classifiers (SVM, XGBoost, and a CNN). SVM, while effective in high-dimensional spaces, struggled with the non-linear separability of classes and required extensive kernel tuning. XGBoost, though efficient, showed sensitivity to class imbalance despite SMOTE, leading to marginally lower recall for minority classes. Deep learning models (e.g., CNNs) achieved comparable accuracy but at the cost of computational overhead and reduced interpretability, which is undesirable for real-time network monitoring. RF outperformed these alternatives in balancing accuracy, computational efficiency, and interpretability, aligning with the study's objectives. The evaluation of the detection system in certain network environments has been carried out. It has been tested on Windows environment as well as on Linux.

Training the Random Forest Classifier

A Random Forest classifier with 100 decision trees is initialized and trained on 80% of the dataset. 5-fold Cross-validation is employed to assess model reliability, yielding average performance metrics across multiple dataset splits. This is done using the sklearn library: *sklearn.ensemble.RandomForestClassifier* with *no. of estimators=100* and *random state=42* trained on an 80/20 train/test split of the data. 5-fold cross-validation "*(sklearn. model_selection. cross_val_score)"* is used to obtain an average cross-validation accuracy of >98%. While the random forest algorithm inherently reduces overfitting through ensemble learning, additional safeguards were implemented.

By hyperparameter tuning, the maximum depth parameter was constrained to 15 (vs. unlimited depth) to prevent individual trees from over-specializing to noise in the training data. A stratified 5-fold cross-validation was employed during training, ensuring the model generalized across diverse subsets of the imbalanced dataset. Limiting features to the most discriminative metrics reduced the risk of fitting irrelevant patterns and SMOTE prevented the model from biasing predictions toward the majority class (flash crowds). Extreme values (e.g., flow durations > 10^6 ms) were truncated to the 99th percentile, minimizing their distortive impact on training, thereby handling outliers. To ensure statistical rigor, the following validation techniques were also applied:

1. **5-Fold Cross-Validation**: The dataset was partitioned into five stratified folds, preserving the original class distribution in each subset. The model was trained on four folds and validated on the fifth, iterating until all folds served as the test set. The 5-fold cross-validation yielded a mean accuracy of 98.9% with a standard

DOI: 10.64290/bimagombe.v9i2A.1061

deviation of 0.4%, demonstrating stable performance across data splits. This approach mitigates sampling bias and provides a robust estimate of generalization performance.

2. **Confidence Intervals (CIs)**: For each evaluation metric (e.g., accuracy, F1-score), 95% confidence intervals were calculated using the t-distribution:

CI = $\mu \pm t\ (\alpha/2,\ n-1) \cdot n \quad \sigma$, where μ is the mean metric, σ the standard deviation across folds, n=5 (number of folds), and t (α/2, n−1) , the critical t-value (2.776 for 95% CI).

To assess the significance of performance differences between the proposed model and baseline classifiers (e.g., SVM, XGBoost), paired t-tests were conducted on cross-validation scores. The null hypothesis ($H0$) posited no difference in mean performance, with p<0.05 indicating statistical significance. While high accuracy can raise concerns about overfitting, the combination of cross-validation, SMOTE, and constrained tree complexity ensures the model generalizes effectively.

RESULTS AND DISCUSSION

To evaluate the model, several metrics and visualizations were used:

1. Confusion Matrix: Presented the counts of true positives, false positives, true negatives, and false negatives for each category, providing insight into the model's classification accuracy.
2. Network Features: The key features are identified and ranked based on their impact on the model's predictions, with the flow of bytes per second and flow duration emerging as the most influential factors.
3. ROC Curve: Illustrates the model's ability to distinguish between classes, showcasing excellent discriminative power with AUC values nearing 1.00 across all categories. The model's performance is evaluated using accuracy, precision, recall and F1 score.

Dataset Analysis

The "FWHA-DDoS" subset of CICIDS 2017 contains three traffic categories: flash crowds (FC), DDoS attacks, and unknown traffic. Compared to alternatives like the World Cup 98 dataset, which focuses solely on legitimate traffic surges—CICIDS 2017 offers a balanced representation of both benign and malicious traffic, making it uniquely suited for training classifiers to differentiate adversarial patterns. The confusion matrix in figure 3 below shows the number of true positives and false positives for each type of traffic. The unknown falls into a category that is neither a legitimate flash crowd traffic nor a DDoS attack, meaning it can be any other network traffic anomaly.

Furthermore, the dataset's inclusion of modern attack vectors (e.g., HTTP floods, SYN floods) and diverse IP address distributions aligns with evolving cyber threats, ensuring relevance to current network environments. The performance of the model on one dataset indicating the precision and recall for the traffic types is shown in figure 4 below.

The Random Forest classifier achieved an accuracy of 99.78%, reflecting its ability to correctly identify traffic types with minimal errors. The results of the evaluation were near perfect (1.00) for each traffic type, highlighting the model's robustness. The confusion matrix revealed only a single misclassification in the flash crowd and unknown categories, demonstrating good performance across the board. With a processing time of 52.40 seconds, the model was efficient, even with the large dataset size. This dataset showcased the model's strength in maintaining high accuracy while

DOI: 10.64290/bimagombe.v9i2A.1061

differentiating between legitimate traffic surges and malicious activity.

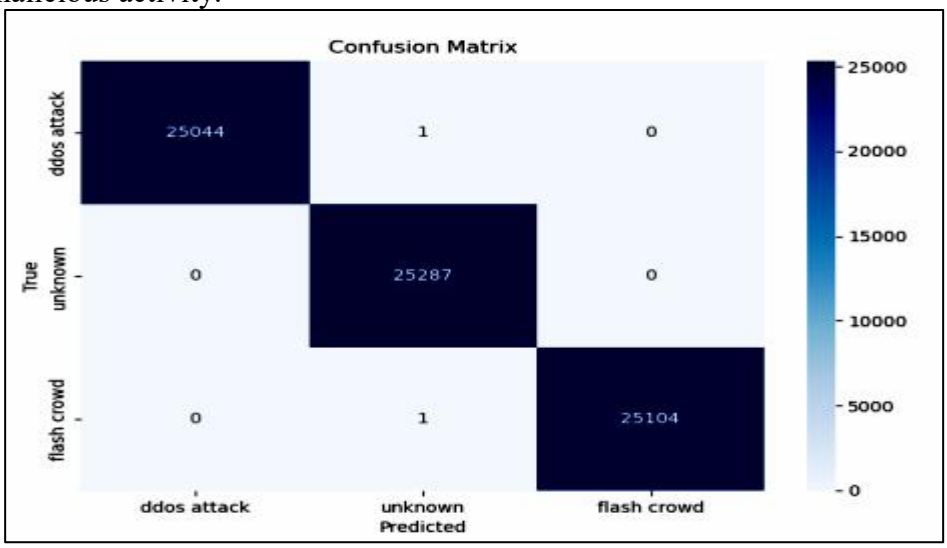

Figure 2: Confusion Matrix

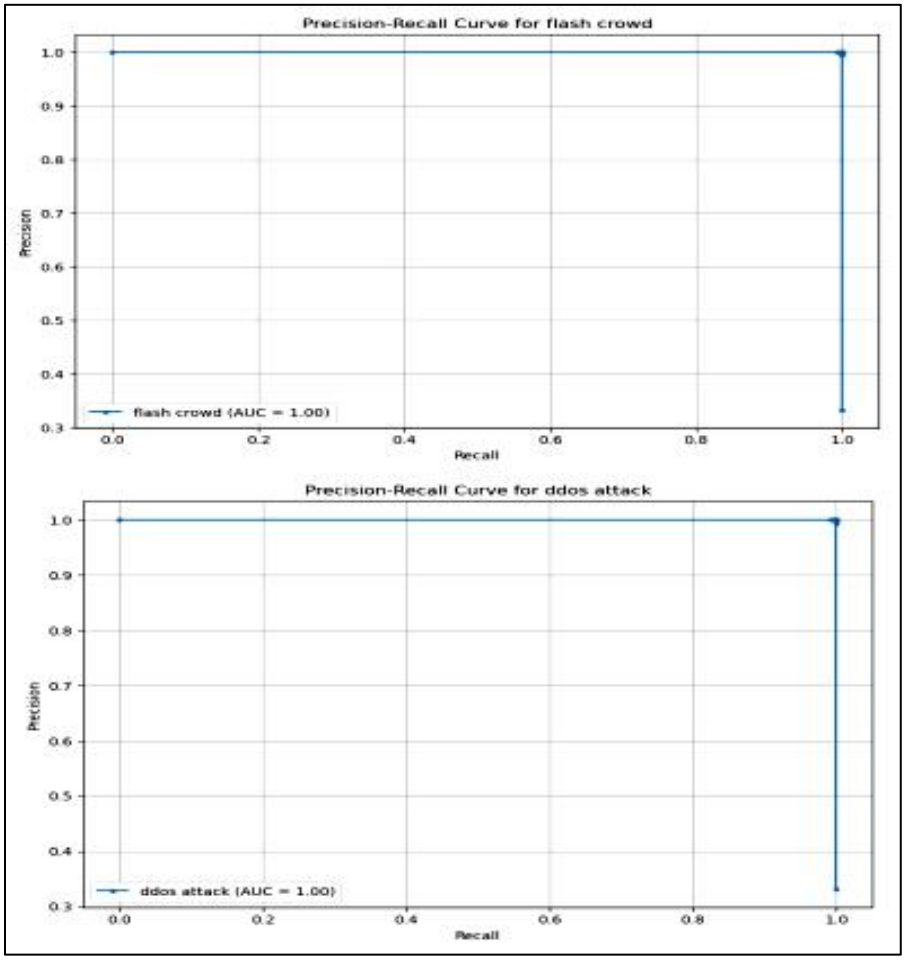

Figure 3: Precision-Recall Curve for Flash Crowds and DDoS Attack

DOI: 10.64290/bimagombe.v9i2A.1061

The study presents a comparative assessment of a novel random forest-based DDoS detection model against the IDS ensemble benchmark paper detailed by Turke et al. (2023). The emphasis is on articulating the pivotal enhancements and efficiency gains realized by our proposed methodology. The benchmark model combines multiple advanced components such as the White Shark Optimizer (WSO) for optimal feature selection, a Convolutional Neural Network (CNN) for feature extraction, and Light GBM for the final classification stage. While operational, this paradigm exhibits vulnerabilities in adapting to the intricacies of real-world networks, particularly concerning low-rate attacks and the separation of flash crowds from DDoS attacks. Table 1 and Figure 4 below shows the evaluation metrics of the proposed model in comparison with that of the benchmark paper.

Table 1: Comparative Analysis with Benchmark Model

Metric	Proposed Model	Turke et al. (2023)
Accuracy	99.78%	95.84%
Precision (Avg)	98.67%	96.15%
Recall (Avg)	95.62%	95.54%
F1-Score (Avg)	98.30%	95.84%
Processing Time and False Positive Rate	52.40 seconds, 0.2%	

The new model rectifies these deficiencies via a machine learning-driven strategy, leveraging Random Forest classification and an augmented feature selection process. This yields substantial improvements in precision, false positive mitigation, and overall effectiveness. By analyzing the behavioral indicators, the new model can segregate flash crowds from actual attacks, drastically curtailing false positives. This constitutes a substantial advantage over the benchmark model, which struggles to achieve lesser False Positive Rates.

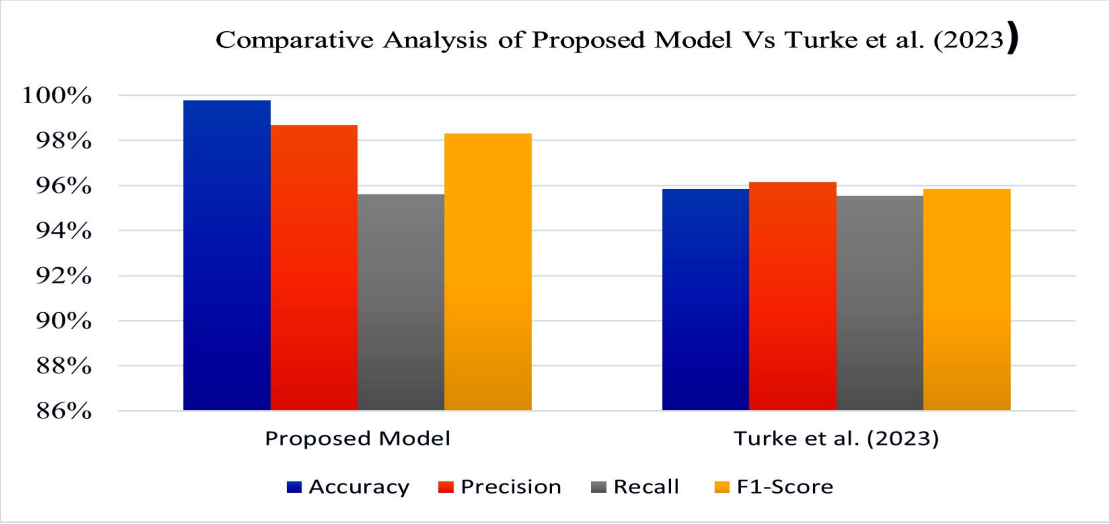

Figure 4: Comparative Analysis Graph

DOI: 10.64290/bimagombe.v9i2A.1061

Comparative Analysis Across ML-models

To further validate the choice of Random Forest (RF), the proposed model was compared against SVM (with RBF kernel) and XGBoost using the same preprocessed dataset and evaluation metrics. As shown in Table 2 below, the random forest classifier achieved superior accuracy (99.78%) compared to SVM and XGBoost. While XGBoost performed competitively, RF's lower false positive rate (0.2% vs. XGBoost's 0.8%) and faster inference time (52.4 seconds vs. XG-Boost's 68.9 seconds) making it more practical for real-time deployment. The model achieved comparable accuracy on the training set (99.81%) and test set (99.78%), indicating minimal overfitting ($\Delta = 0.03\%$). As shown in table 2 below, the results underscore RF's suitability for scenarios requiring both precision and computational efficiency.

Table 2: Result Comparison across ML-models

Metric	Random Forest	XGBoost	SVM
Accuracy	99.78%	98.95%	95.12%
F1-Score (DDoS)	98.30%	97.20%	93.45%
Recall (DDoS)	99.65%	96.80%	88.50%
Training Time	52.40 sec	68.90 sec	120.30 sec
False Positive Rate	0.2%	0.8%	1.5%

The use of the random forest classifier with thresholding mechanism, achieved 99.78% accuracy in differentiating DDoS attacks. With a processing time of 52.4 seconds for 225,711 samples and a false positive rate of 0.2%, the model is computationally efficient and minimizes service disruptions. Its interpretable feature importance scores will enable network administrators to prioritize traffic analysis and resource allocation. The integration of SMOTE mitigated bias toward majority classes, improving recall for DDoS attacks and ensuring robust detection.

CONCLUSION

This study addresses the challenge of detecting DDoS attacks hidden within Flash Crowds through a machine learning-driven framework, achieving multi-classification methodology, practical solutions for class imbalance and potential for real-world deployment. The Random Forest classifier, combined with feature-based thresholds, demonstrated exceptional performance on the datasets. The model surpassed the benchmark study in accuracy, effectively distinguishing between these attacks and legitimate traffic. It also introduced the capability for multi-classification, improving upon previous models that were limited to binary classification. By leveraging domain knowledge to define feature-based thresholds and employing advanced techniques such as SMOTE for class balancing, the model achieved a near-perfect detection accuracy (average 99.78%) across the different traffic types and high precision, ensuring minimal false positives. By reducing reliance on static thresholds and manual tuning, the proposed model framework's scalability and accuracy makes it directly applicable to enterprise networks and advances adaptive cybersecurity systems capable of evolving with emerging threats.

Recommendations

Future research can be focused on real-time implementation, which can be done by

DOI: 10.64290/bimagombe.v9i2A.1061

developing a deployment pipeline for live network environments, integrating the model with network monitoring tools, and evaluating latency and computational overhead. Feature engineering can be improved by exploring additional features like protocol-specific indicators. Furthermore, expanding dataset diversity by incorporating data from various network environments (IoT, cloud, etc.) and simulating mixed traffic patterns will broaden solutions to cyber threats. Finally, explainability and user trust can be prioritized, extending the use of tools like SHAP to create interactive dashboards for network administrators and provide actionable insights alongside predictions**.**

REFERENCES

Biruk, A. M. (2020). Application Layer DDoS Attack Detection in the Presence of Flash Crowds. *EEA*.

Dhingra, A. (2018). DDoS detection and discrimination from flash events: a compendious review. *First International Conference on Secure Cyber Computing and Communication(ICSCCC)*, (pp. 518-522).

Gera, J. B. (2018). Detection of Spoofed and Non-Spoofed DDoS Attacks and Discriminating them from Flash Crowds. *EURASIP Journal on Information Security*.

Jisa David, Ciza Thomas. (2021). Discriminating Flash Crowds Algorithm using Efficient Thresholding Algorithm. *Journal of Parallel and Distributed Computing*, 152(79-87).

K. Naik, & V. Bhosale, &. D. (2023). Malicious User Detection using Honeyword and IP Tracking. *Computer Projects, IEEE.*

Kalkan et al. (2021). A Survey of DDoS Attack Detection and Mitigation Techniques. Computer Networks.

Marinova, S. V. (2020). End-to-End Network Slicing for Flash Crowds. *IEEE Communications Magazine, vol.58*, (pp. 31-37).

Praseed, A. P. (2018). DDoS Attacks at the Application Layer: Challenges and Research Perspective for Safeguarding Web Applications. *Communication Survey Tutor 21(1)*, 661-685.

R. Doshi., N. Apthorpe., & N., Feamster. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. *IEEE Security and Privacy Workshops*, (pp. 29-35). San Francisco, CA, USA.

Razumov, P. O. (2020). Developing of Algorithm for HTTP Flood DDoS Protection. *3rd International Conference on Computer Applications & Information Security ICCAIS* (pp. 1-6). IEEE.

Salah, H. Y.-S. (2023). Q-Learning Based DDoS Detection. *International Journal of Electrical and Computer Engineering (IJECE)*.

Selvakani, K. V. (2021). Attack in SDN-Based DDoS. *Asian Journal of Engineering & Applied Tech.*, 38-44.

Tandon, R. A. (2019). Defending Web Servers against Flash Crowd Attacks. *27th International Conference on Network Protocols* (pp. 1-4). IEEE.

Turke Althobaiti, Y. S. (2023). Securing Cloud Computing from Flash Crowd Attack Using Ensemble. *Computer Systems Science and Engineering (CSSE)*, 454-469.