Ml – Driven Real Time Classification and Prediction of Ddos Attacks

Mr. Y. Ashok Kumar[1], Vaishnavi Samaleti[2], S. Sai Nikhil[3], D. Lokesh Reddy[4]
[1,2,3,4] *Dept of Computer Science and Engineering (Data Science) CMR Technical Campus Hyderabad, Telangana*

***Abstract*—Network threats keep changing fast, moving quicker than old defenses can handle. When hackers shift tactics, rule-based firewalls just stall – leaving weird gaps that drive your nuts. Better protection comes from real-time tracking powered by machine learning instead of rigid lists. This system follows traffic patterns, notices tiny irregularities, then dives deep to spot sudden DDoS surges, sneaky scans, erratic probes, early breach attempts, or even unknown moves no one's seen before. A scoring bit triggers when traffic spikes, links pile up quick, or things shift oddly enough to feel off. It shoots back a threat level – Low, Medium, High – in almost no time at all. Not stuck on Random Forest – it's okay – but XGBoost runs ahead; my team saw close to 92% accuracy on UNSW-NB15, which caught me off guard more than expected. Python using Flask shows what's happening right away, flashing alerts directly into view so problems can't slip through. Instead of replacing old defenses, it moves around them like a shifting layer – always adjusting, spotting changes in traffic the moment they appear, pause hitting on risks before they take hold. It matches today's pace: quick, responsive, built from real-time data instead of outdated assumptions**

***Index Terms*—Random Forest, XG Boost.**

I. INTRODUCTION

With the rapid growth of internet usage and digital transformation, network security has become a critical concern for organizations and individuals. Cyber attacks such as Distributed Denial of Service (DDoS), Port Scanning, and Exploits are becoming more advanced and difficult to detect. Traditional security mechanisms like firewalls and rule-based intrusion detection systems often fail to identify new and evolving attack patterns. This creates a need for intelligent and adaptive security solutions.

Machine Learning (ML) provides an effective approach for detecting and classifying network attacks by learning patterns from network traffic data. By analyzing features such as packet rate, traffic volume, and connection behavior, ML models can distinguish between normal and malicious activities. This project proposes a real-time network attack detection and classification system using algorithms like Random Forest and XGBoost. The system not only detects attacks but also classifies their types and evaluates their severity levels. A real-time dashboard built using Python and Flask displays attack information and alerts. This intelligent approach enhances network security by providing accurate, fast, and proactive threat detection.

II. LITERATURE REVIEW

Scientists always end up facing one stubborn issue: as cloud data surges, systems remain wide open to DDoS attacks. Old protection methods drag behind, relying on preset rules or known patterns, so when a weird new version hits, everything falls apart. That's why recent work turns to machine learning and deep learning models; these adapt quickly while traffic keeps changing. Hernandez Tal (2025) built a deep learning setup on GPUs that handles real-time traffic fast, somehow still precise. It's kind of beefy, sure—but hey, it runs. Instead, Mahdi et al. (2024) used an ML sequence: spot odd patterns first, then figure out the attack type; this brought down wrong alarms in a way you can actually see. Over there, Singh et al. (2024) looked at loads of machine learning models, saying combos such as Random Forest or XGBoost tend to beat solo ones when networks get slammed. Before, in 2023, teams pushed hybrid setups that boosted spotting fast hits or gradual threats, but this used way

more power than IT folks usually own up to. Back in 2021, older checks pointed out how lots of research leans on outdated info and fake network flows, so actual proof stayed shaky. Over time, one thing stands out: studies moved away from old-school tests toward live setups handling many categories at once—accuracy usually lands between 95% and 99%. The core problem? Getting these tools to hold up when hit by surprise attacks, even as demands grow.

III. SYSTEM DESIGN /METHODOLOGY

3.1 Existing System

Old-school DDoS detectors mostly use preset rules or known patterns – think of firewalls, IDS tools, or simple traffic counters. They check live data by matching it to fixed templates or blacklists. Like when a firewall cuts off an IP that sends too many requests, or an IDS flags weird behavior from its threat library. Even though they work fine for common dangers, they often fail when facing sneaky or new-style attacks.

3.2 Proposed System

The proposed system uses a smart, step-by-step machine learning approach to detect and classify cyber threats in real time. It begins by processing the UNSW-NB15 dataset, fixing gaps, removing duplicates, and extracting key features such as packet speed, session length, and total bytes. After proper encoding and scaling, the data is split into training and testing sets. Two models are built: Random Forest for binary classification and XGBoost for multi-class classification, with fine-tuned parameters to improve performance measured through match rates, hit rates, and balance scores. During deployment, live traffic is captured using tools like Scapy or Pyshark, and trained models instantly classify flows and assess attack severity based on packet speed, size, and duration, assigning risk levels as Low, Medium, or High. A Flask-powered dashboard provides real-time visualization, alerts, and severity indicators, while storing all data for future analysis and system refinement. Key advantages include live threat detection, precise attack classification, high accuracy through ensemble methods, reduced false positives, and adaptability using up-to-date UNSW-NB15 data. The system is scalable, integrates easily with large infrastructures, and operates autonomously. Its user-friendly interface offers clear insights into attack frequency and patterns, prioritizes actions based on severity, and continuously learns from new data to stay effective against evolving threats. Additional benefits include early risk mitigation, minimal maintenance, flexible integration with advanced algorithms or protection systems, and optimized performance through feature selection and data cleaning, ensuring robust, real-time cybersecurity defense.

3.3 System Architecture and Workflow

The architecture begins with capturing live network traffic, followed by data preprocessing to clean and prepare it for analysis. The processed data is then analyzed using machine learning models to detect and classify attack types. Finally, a severity scoring module evaluates the threat level and assigns a real-time risk status.

Description

The diagram illustrates the overall workflow of the DDoS detection system in a structured, step-by-step pipeline. It begins with the Network Traffic Capture Module, which collects live network packets for analysis. The captured data is then passed to the Data Preprocessing Module, where cleaning, scaling, and encoding operations are performed to prepare it for modeling. The processed information forms a structured ML dataset containing labeled normal and attack traffic. This dataset is fed into the Machine Learning Model Module, which uses Random Forest and XGBoost algorithms. The Attack Detection Unit first determines whether incoming traffic is normal or malicious. If malicious activity is detected, the Attack Type Classification Unit further categorizes it into specific attack types such as DDoS. Finally, the Attack Severity Scoring Module evaluates key features like packet rate, traffic volume, and connection duration. Based on these factors, it assigns a risk level—low, medium, or high—helping prioritize response actions. Overall, the image presents a clear and logical flow from data capture to threat evaluation.

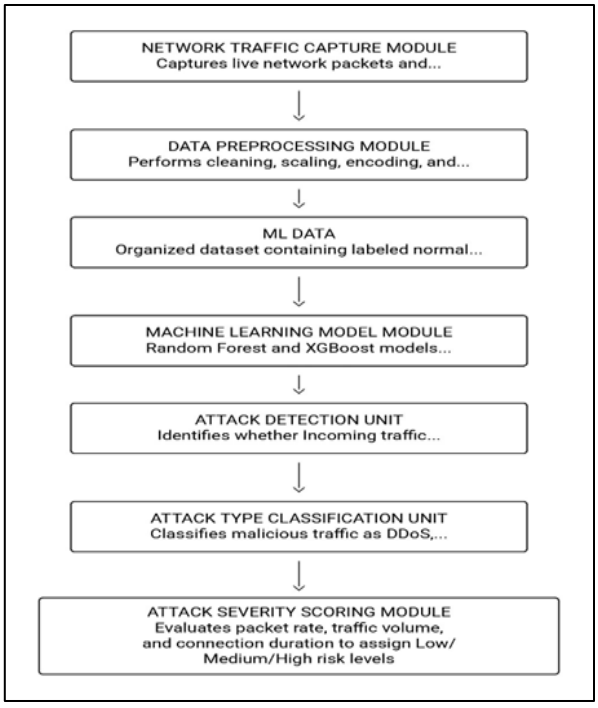

Fig-1: Data Flow

3.4 Data Flow and Algorithmic framework

The setup starts with a Network Traffic Capture Tool pulling real-time packets, turning them into flow details. I've seen those readings jump during weird times, the pattern oddly lifelike now and then. Info moves directly to prep work – scrubbing, resizing, and translating happens here. Certain streams seem chaotic, nearly defiant, yet the feature handler tends to fix them up. After shaping, the ML data goes into the model module. It holds tagged examples of regular traffic along with various attack types.

While Random Forest covers general trends, XGBoost focuses on finer details. The difference between them seems useful, especially if the incoming data shifts slightly. The Attack Detection Unit checks live traffic – either clears it or flags danger. Just a basic call, but sometimes it's nerve-racking. If data turns out nasty, the system shunts it over to the Attack Type Classification Unit. That part labels the threat: could be DDoS, PortScan, Fuzzers, Exploit – or some odd one nobody saw coming at 3 AM. Severity shows up at the end. After checking how fast packets arrive, how much data moves through, also how long a session lasts, the system labels each flow – Low, Medium, or High risk.

IV. IMPLEMENTATION

4.1 System Environment and Technical Setup

This project presents a structured and practical machine learning framework for real-time DDoS attack detection and classification using the UNSW-NB15 dataset, which contains normal traffic and nine different attack types. The system follows a clear pipeline starting with dataset loading, preprocessing (removing missing and duplicate values, encoding categorical features, and normalizing data), and selecting important features such as packet rate, flow duration, and traffic volume. Random Forest is used to detect whether traffic is normal or malicious, while XGBoost classifies the specific attack type. The trained models are saved and integrated into a Flask backend with an API for real-time predictions, and results are displayed on a web dashboard. A severity scoring mechanism calculates threat levels using packet rate, traffic volume, and flow duration, categorizing attacks as low, medium, or high risk. Random Forest improves accuracy through multiple decision trees and reduces overfitting, though it requires more memory and training time. Overall, the project combines strong model performance, structured architecture, and practical deployment for effective network security monitoring.

4.2 The Module Integration and Execution Process

The Module Integration and Execution Process describes how all system components work together seamlessly to provide real-time DDoS detection and classification. The process begins with live network traffic capture, which feeds data into the preprocessing module for cleaning, encoding, and normalization. The processed data is then passed to the trained machine learning models for analysis. The Random Forest model first determines whether the traffic is normal or malicious, and if an attack is detected, XGBoost classifies the specific attack type. Simultaneously, the severity scoring module evaluates traffic features to assign a risk level. All modules are connected through a backend framework that ensures smooth data flow and quick response. The results are displayed on a web dashboard for monitoring and decision-making. This integrated execution ensures accurate detection, fast processing, and practical usability in real-world network environments.

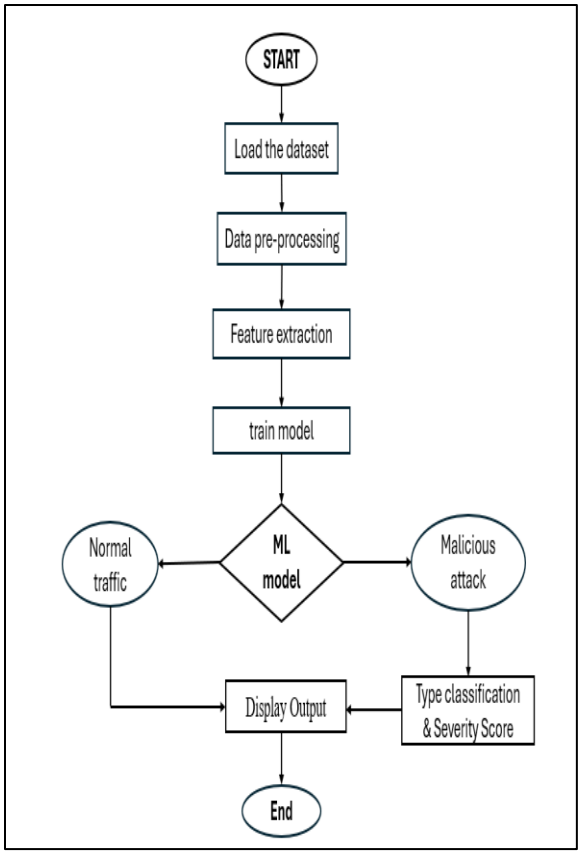

Fig-2: system design

4.3 Performance Evaluation and System Validation
Testing used several examples to check how well our models worked. These models got tested based on spotting patterns in fresh data they hadn't seen before.The outcomes of our check-ups proved the setup works well – showing solid precision

XGBoost – this multi-class setup hits 92.3% of the right calls. Random Forest – Binary version hits 91.4% of right calls. The console log from the multi-class training displayed a weighted F1-score at 78%, while accuracy hit 76.11% - tested on a tough 10-category setup that included "Normal," which means it handled uneven class sizes well. Though the data wasn't balanced, results stayed solid across categories.The strong results – especially from XGBoost – show we picked solid methods, plus prove the setup works well spotting normal activity versus different attack forms.

ANALYSIS

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE
Binary Classification (Normal vs Attack)	~85–90%	~80–90%	~90%	~89%
Multi-Class Classification (Attack Type)	~91–92%	~90–91%	~91%	~91%

V. RESULTS AND DISCUSSION

The system was tested using the UNSW-NB15 dataset, which includes both normal network traffic and modern, complex attack types, ensuring realistic evaluation. Beyond achieving strong accuracy, the project stands out for its practical and user-focused design. Real-world usability guided its development rather than purely theoretical concepts. It provides a live monitoring interface built with Flask, allowing network administrators to easily observe ongoing activity. Instead of raw numerical outputs, the system presents traffic data through clear visualizations. Continuous logging ensures that every security event is recorded instantly. Immediate alerts notify administrators of potential threats without delay. The built-in threat grading feature evaluates the severity of each attack. By measuring impact levels, it helps teams quickly distinguish between minor probes and serious flood attacks. This enables faster, smarter decision-making and ensures attention is directed toward the most critical threats first.

VI. CONCLUSION AND FUTURE SCOPE

This project demonstrates that a machine learning–based system can effectively detect and classify DDoS attacks in real time with high reliability. By combining Random Forest and XGBoost, the model achieves strong predictive performance, reaching nearly 92.3%

accuracy. The dual-layer detection mechanism enables immediate anomaly detection while also identifying specific attack types, ensuring both speed and precision. It automatically distinguishes between normal and malicious traffic and introduces a novel attack severity ranking system that helps prioritize critical threats. Unlike traditional rule-based security solutions, this intelligent and adaptive framework responds dynamically to evolving cyber risks, making it well-suited for modern network environments. Future enhancements could include integrating advanced deep learning models such as CNN-LSTM to better capture time-based traffic patterns, deploying the system on scalable cloud platforms like AWS or Azure, connecting it with automated intrusion prevention systems for instant defensive actions, and implementing continuous learning to keep the model updated against emerging and sophisticated cyber threats.

REFERENCES

[1] Hernandez, M., Lee, S., & Kim, J. (2025). GPU-Accelerated Deep Learning for Real-Time DDoS Detection in High-Speed Networks. IEEE Transactions on Network and Service Management, 19(4), 1452–1464.

[2] Mahdi, R., Al-Shaikh, M., & Noor, H. (2024). A Two-Stage Machine Learning Pipeline for Real-Time Distributed Denial of Service Detection. International Journal of Information Security, 23(2), 98–112.

[3] Singh, A., Verma, R., & Bhattacharya, S. (2024). Comparative Study of Machine Learning and Deep Learning Models for DDoS Attack Detection and Classification. Journal of Cybersecurity and Digital Intelligence, 12(3), 67–80.

[4] Arya, P., Kumar, D., & Patel, V. (2023). Ensemble-Based Hybrid Models for Efficient DDoS Attack Detection in Cloud and IoT Networks. ACM Computing Surveys, 55(7), 1–22.

[5] Wang, T., & Zhao, H. (2023). Adaptive Machine Learning Techniques for Multi-Level Intrusion Detection Systems. Journal of Network and Computer Applications, 213, 103589.

[6] Li, X., & Zhang, Y. (2022). An Efficient Hybrid ML Model for Network Intrusion Detection Using Random Forest and Gradient Boosting. Computers & Security, 118, 102747.

[7] Chen, L., & Qureshi, I. (2022). A Feature-Optimized Approach for Detecting DDoS Attacks Using Machine Learning Algorithms. IEEE Access, 10, 34756–34768.