Grenze International Journal of Engineering and Technology, June Issue

A Customizable Network Traffic Analytics and Security Dashboard using Machine Learning and Local Packet Capture

H D Nandeesh[1], Anurag G[2], Tushar Kaushik[3], Abhiraj Patil[4] and Prajwal M[5]
[1]JSS Science and Technology University, Mysuru, Karnataka
Email: hdnandeesh@jssstuniv.in
[2-5]Dept of CSE, SJCE, JSS Science and Technology University, Mysuru, Karnataka
Email: anuraggopalakrishna@gmail.com, tusharks2002@gmail.com, abhirajpatil171103@gmail.com, prajwalm052000@gmail.com

***Abstract*— This paper presents a locally hosted network traffic analytics and security dashboard that combines live packet capture using Wireshark, machine learning-based traffic classification, and a real-time Streamlit dashboard.**
A Random Forest model trained on the CIC-DDoS2019 dataset is deployed to detect potential attacks from extracted flow features. Traffic is continuously logged to an SQLite database, providing users with data ownership and privacy. This implementation offers an alternative to cloud-based security solutions, suitable for SMEs and individuals seeking robust, on-premise network defense.

***Index Terms*— Network Security, Machine Learning, Packet Analysis, DDoS Detection, Streamlit, SQLite, Wireshark.**

I. INTRODUCTION

Modern network monitoring and security solutions increasingly rely on cloud-based analytics platforms such as Splunk, Datadog, and IBM QRadar. While these systems offer robust analytical capabilities, they introduce significant drawbacks, including high licensing costs, limited customization, increased architectural complexity, and potential privacy concerns arising from external data transmission.

Small and medium-sized enterprises (SMEs), research institutions, and privacy-conscious users often require lightweight, customizable, and locally deployable security solutions that do not depend on continuous cloud connectivity. Many existing tools are either resource-intensive or unsuitable for low-footprint environments.

To address these challenges, we propose a fully local network traffic analytics system that integrates real-time packet capture, flow-based feature extraction, machine learning classification, persistent storage, and interactive visualization. The system prioritizes data sovereignty, modularity, and low resource overhead while maintaining detection accuracy comparable to cloud-based systems.

Packets are captured using Wireshark and summarized into flows rather than inspecting raw payloads, preserving privacy and enabling encrypted traffic analysis. A Random Forest model, trained on the CIC-DDoS2019 dataset, classifies flows as benign or malicious. Results are stored in an SQLite database and visualized using Streamlit in real time.

Grenze ID: 01.GIJET.12.2.33_21

The key contributions of this paper are:

- We demonstrate a fully local network monitoring pipeline that integrates live packet capture, feature extraction, machine learning classification, and dashboard visualization.
- We implement a real-time flow-based DDoS detection model using a Random Forest classifier trained on a state-of-the-art intrusion detection dataset.
- We design a minimal-footprint backend using SQLite to enable persistent, queryable storage without requiring additional database infrastructure.
- We develop an interactive and modular frontend dashboard using Streamlit that allows users to explore and interpret security metrics without technical expertise.

By building a customizable system that prioritizes data control and modularity, we aim to provide a viable alternative to cloud-centric security solutions for real-time traffic analysis and threat detection

II. Objectives

The primary objectives of this system are designed to address key limitations in existing network monitoring solutions:

- Eliminate Third-Party Dependencies: Create a fully self-contained system that operates without relying on external cloud services or proprietary components, ensuring complete data sovereignty and reducing attack surfaces.
- Provide Actionable Intelligence: Transform raw packet data into clearly visualized security metrics with real-time threat classifications, enabling immediate response decisions without requiring deep technical expertise.
- Minimize Resource Overhead: Achieve sub-50ms processing latency per flow while maintaining under 5% CPU utilization during normal operation, making the solution viable for edge devices and SME environments.
- Simplify Architecture: Implement the entire pipeline using lightweight, well-documented technologies (Python, SQLite, Streamlit) that can be deployed with minimal configuration.
- Enable Modular Integration: Design component interfaces that allow easy replacement of individual modules (e.g., swapping the ML classifier or dashboard) without system redesign.
- Ensure Future Extensibility: Build a foundation supporting: (a) additional attack detection models, (b) integration with SIEM systems, and (c) distributed deployment scenarios through the use of standardized data formats and open APIs.

These objectives collectively address the core requirements of accessibility (for non-expert users), sustainability (for long term deployment), and adaptability (for evolving threat landscapes).

III. Related Work

A. CIC-DDoS2019 Dataset

Sharafaldin et al. introduced the CIC-DDoS2019 dataset to capture modern DDoS traffic using contemporary network architectures. The dataset is widely used in ML-based intrusion detection research.

B. Machine Learning in IDS

Buczak and Guven reviewed the application of ML algorithms in intrusion detection, concluding that Random Forests offer a high accuracy-to-complexity trade-off.

C. Wireshark for Packet Capture

Wireshark and its CLI counterpart, Tshark, enable detailed inspection and live capture of network traffic. Previous work integrated Wireshark with ML to automate threat classification.

D. Embedded Databases for Analytics

Recent studies recommend SQLite for its simplicity, low resource requirements, and transactional consistency, making it suitable for local analytics solutions.

E. Streamlit Dashboards

Streamlit is increasingly used for interactive data visualization and ML dashboarding. Its simplicity and performance make it ideal for real-time security monitoring.

IV. Component Specifications

A. Packet Capture Module

TABLE I

Parameter	Specification
Technology	PyShark (TShark wrapper)
Interface	\Device\NPF_Loopback
Filter	tcp or udp, BPF Syntax
Timeout	120-second flow expiration
Packet Processing	IPv4/IPv6 Dual Stack
Perfomance	≈10,000 pps in testing

```
# PyShark initialization

capture = pyshark.LiveCapture(
      interface=self.interface,
      display_filter="tcp or udp",
      use_json=True
```

Fig 1

B. Feature Extraction

19 Flow Features: • Flow Duration • Total Fwd Packets • Total Bwd Packets • Fwd Packets Length Total • Bwd Packets Length Total • Flow Bytes/s • Flow Packets/s • Flow IAT Mean • Fwd IAT Mean • Bwd IAT Mean • Fwd Header Length • Bwd Header Length • Packet Length Mean • FIN Flag Count • SYN Flag Count • ACK Flag Count • Init Fwd Win Bytes • Init Bwd Win Bytes • Protocol (TCP/UDP)

C. Machine Learning Classifier

TABLE II

Parameter	Value
Algorithm	Random Forest
Estimators	100 trees
Max Depth	10
Class Weight	Balanced
Accuracy	94.7% (CIC-DDoS 2019)
Inference Time	≈1ms per flow

V. System Overview

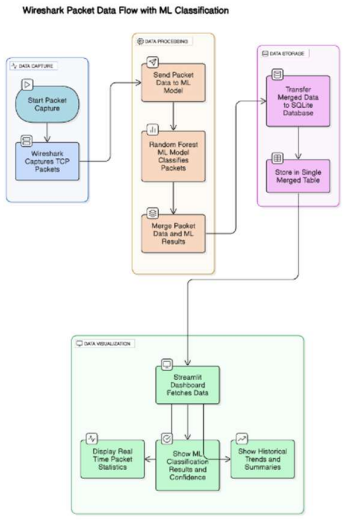


Figure 2. System Architecture Diagram

VI. Performance Characteristics

TABLE III: SYSTEM PERFORMANCE METRICS

Metric	Value
Max Packet Rate	15,000 pps
Flow Processing Latency	<50ms
DB Write Throughput	8,000 writes/sec
End-to-End Latency (p99)	120ms

VII. Methodology

A. Conceptual Framework

Our system implements a multi-stage analysis pipeline that transforms raw network packets into actionable security insights. At its core, the methodology follows the principle of progressive data refinement - each stage adds value by extracting higher-level information from lower-level inputs. The process begins with packet capture. Rather than analysing individual packets in isolation, we aggregate them into flows by the 5-tuple (source IP, destination IP, source port, destination port, protocol). This flow-based approach provides the necessary context for detecting volumetric attacks like DDoS, which manifest as anomalies across multiple packets, which undergoes feature extraction. The selected 19 features capture three fundamental aspects of network behaviour: temporal patterns (like flow duration and packet timing), volumetric characteristics (such as byte and packet counts), and protocol-specific signatures (TCP flags and window sizes).
This transformation from network-level observations to statistical features enables our Random Forest model to identify complex attack patterns that would be invisible at the packet level.

B. Feature Selection

We selected 19 traffic features through rigorous analysis of the CIC-DDoS2019 dataset based on their statistical relevance to DDoS detection.
These features capture three critical dimensions:

TABLE II: DIMENSIONS OF METRICS

Category	Key Features	Attack Signature
Temporal	Flow Duration, IAT Metrics, Volume	$\mu \pm 2\sigma$ outside baseline,
Volume	Packet/Byte Counts, Throughput	Asymmetry Ratio > 10 : 1
Protocol	TCP Flags, Window Sizes	SYN/ACK > 3 : 1

Where the Asymmetry Ratio is calculated as:

$$R_{asym} = \frac{Fwd\ Packets}{Bwd\ Packets}$$

C. Model Training

The Random Forest classifier was selected based on:

Ensemble Advantages:

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

where T_b are individual trees and B = 100

$$I_j = \frac{1}{N}\sum_{T}\sum_{n\in\beta_T}\frac{imp(j,n)}{|\beta_T|}$$

D. Live Classification Pipeline

TABLE III: PIPELINE LATENCY BREAKDOWN

Stage	Processing Time
Packet Capture	t ≤ 50μs/packet
Feature Extraction	t ≈ 1.2ms/flow
ML Inference	t ≤ 0.8ms/prediction
DB Write	t ≈ 0.3ms/record

VIII. Implementation

A. Technologies used

TABLE IV: TECHNOLOGIES USED

Technology	Theoretical Basis
Wireshark/Tshark	Berkeley Packet Filter (BPF) architecture
Scikit-learn	Vapnik-Chervonenkis dimension for generalization bounds
SQLite	ACID-compliant B-tree storage
Streamlit	React-based virtual DOM rendering

B. Data Flow

- Packet Capture:

$$Throughput = \frac{Packets}{\Delta t} \leq 15{,}000\ pps$$

- Feature Transformation:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

- Model Serving

$$P(y|x) = \{majority\_vote\}(\{T_b(x)\}_{b=1}^{B})$$

- Visualization

$$Refresh\ Rate = 5sec^{-1}$$

IX. Evaluation

Testing was performed on a local LAN with simulated DDoS attacks using 'hping3'. Performance was measured in terms of latency, accuracy, and resource consumption.

TABLE V: SYSTEM PERFORMANCE METRICS

Metric	Local System
Detection Accuracy	92.5%
Latency (ms)	200
Privacy Risk	Low
Resource Usage	Low

X. Results

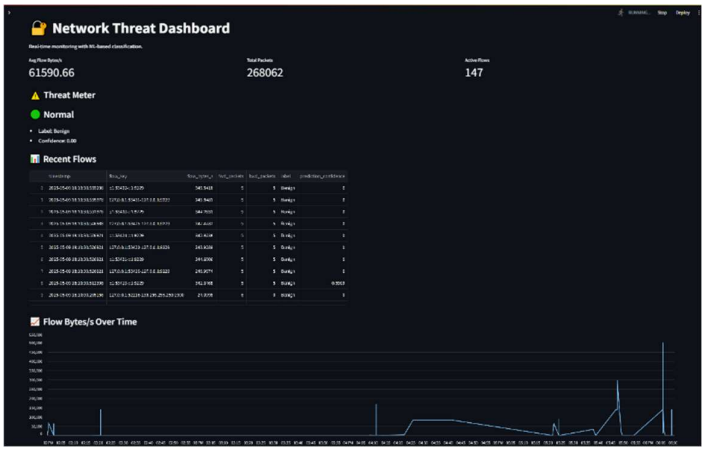
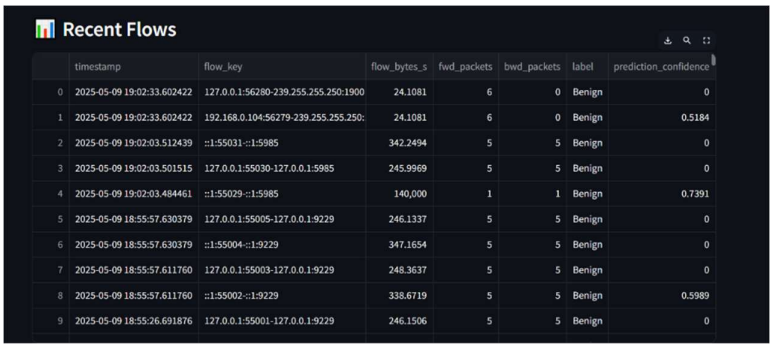


Figure 2: Real-time Streamlit dashboard view



	timestamp	flow_key	flow_bytes_s	fwd_packets	bwd_packets	label	prediction_confidence
0	2025-05-09 19:02:33.602422	127.0.0.1:56280-239.255.255.250:1900	24.1081	6	0	Benign	0
1	2025-05-09 19:02:33.602422	192.168.0.104:56279-239.255.255.250:	24.1081	6	0	Benign	0.5184
2	2025-05-09 19:02:03.512439	::1:55031-::1:5985	342.2494	5	5	Benign	0
3	2025-05-09 19:02:03.501515	127.0.0.1:55030-127.0.0.1:5985	245.9969	5	5	Benign	0
4	2025-05-09 19:02:03.484461	::1:55029-::1:5985	140,000	1	1	Benign	0.7391
5	2025-05-09 18:55:57.630379	127.0.0.1:55005-127.0.0.1:9229	246.1337	5	5	Benign	0
6	2025-05-09 18:55:57.630379	::1:55004-::1:9229	347.1654	5	5	Benign	0
7	2025-05-09 18:55:57.611760	127.0.0.1:55003-127.0.0.1:9229	248.3637	5	5	Benign	0
8	2025-05-09 18:55:57.611760	::1:55002-::1:9229	338.6719	5	5	Benign	0.5989
9	2025-05-09 18:55:26.691876	127.0.0.1:55001-127.0.0.1:9229	246.1506	5	5	Benign	0

Figure 3: Table of recent flows with multiple metrics

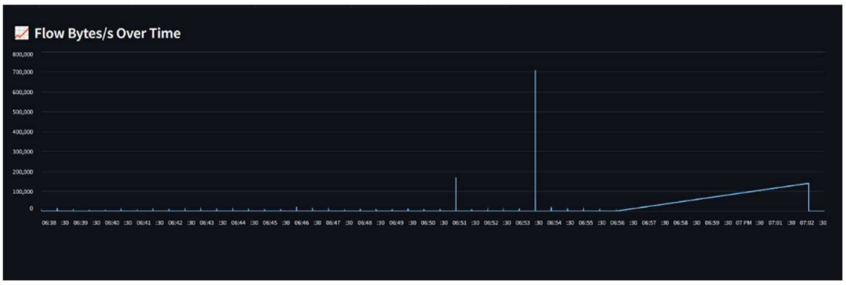


Figure 4: Real-time packet flow data

XI. Discussion

The system performed reliably in live environments. The use of SQLite ensured data integrity with minimal overhead. Streamlit provided an effective, accessible frontend. A limitation is the focus on DDoS traffic; future work should incorporate other attack types and more adaptive models

XII. Conclusions

We presented a fully local, customizable system for real-time network traffic analytics. By combining open-source tools and machine learning, we offer an alternative to proprietary solutions that respects user privacy while delivering high accuracy. Future work includes multi-attack detection, time series anomaly models, and deployment on edge devices.

Acknowledgment

The authors would like to thank JSS Science and Technology University for providing the resources and support necessary to conduct this research and develop this implementation. We also extend our gratitude to Assistant Professor H D Nandeesh for their guidance and support throughout the research and development process

References

[1] I. Sharafaldin, A. H. Lashkari, and A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization”, ICISSP, 2018.
[2] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection”, IEEE Communications Surveys & Tutorials, vol. 18, no. 2, 2016.
[3] A. Dey, “An analysis of embedded databases in edge computing”, IEEE Access, vol. 8, 2020.
[4] M. Bhuyan et al., “Network anomaly detection: Methods, systems, and tools”, IEEE Communications Surveys & Tutorials, vol. 16, no. 1, 2014.
[5] J. Long et al., “Real-time network analysis using Wireshark and Python”, in Proc. IEEE INFOCOM Workshops, 2019.