International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

NeuralGuard: A Real-Time AI-Driven Autonomous Intrusion Prevention System for Network Security

Kishore Kumaar S R[1], Nikesh G[2], Tanushree Padmanaban[3], Nithish S[4], Darshan S[5]

[1]*Department Of Computer Engineering , BIT*
[2]*Department Of Information Technology, BIT*
[3]*Department Of Computer Technology , BIT*
[4]*Department Of Computer Engineering , BIT*
[5]*Department Of Computer Engineering , BIT*

ABSTRACT

Modern network security struggles with the "Response Gap" - the critical delay between threat detection and mitigation allowing sophisticated attacks like DDoS and botnets to succeed within seconds. This paper presents NeuralGuard, an autonomous Intrusion Prevention System achieving sub - 10ms threat response through hybrid deep learning and kernel-level firewall integration. The system employs dual-model architecture: a Multi-Layer Perceptron for supervised detection of known attacks and a deep autoencoder for unsupervised anomaly detection of zero-day threats. A Modified Genetic Algorithm optimizes features from 79 to 31 variables, reducing computational overhead by 66% while maintaining accuracy. Network flows are captured via Scapy, processed through 31 statistical features, and evaluated with 9.1ms end-to-end latency, enabling immediate automated firewall blocking via UFW/iptables/ipset. Evaluated on CIC-IDS-2017, NeuralGuard achieves 99.9% classification accuracy across 14 attack categories with an F1-score of 0.978 and false positive rate of 3 - 5%. A self-healing TTL mechanism (5-minute auto-recovery) prevents permanent blocking of legitimate users. Real-time Grafana dashboards and PostgreSQL logging provide forensic visibility. This work demonstrates that AI-orchestrated autonomous defense is practical, scalable, and deployable on resource-constrained systems, advancing autonomous network security.

Keywords: Intrusion Prevention System, Real-Time Threat Detection, Deep Learning, Autonomous Network Defense, Low-Latency Inference, Firewall Automation, Zero-Day Detection.

1. INTRODUCTION

The modern digital landscape is defined by unprecedented reliance on high-speed network connectivity, serving as the critical backbone for global financial transactions, industrial automation, and sensitive data exchange. As network architectures have evolved from simple local area networks to complex, cloud-integrated ecosystems spanning multiple domains and jurisdictions, the sophistication of cyber threats has evolved in parallel. Traditional security models, primarily reliant on static, rule-based firewalls and signature-detection systems, are increasingly inadequate in this threat environment. These legacy systems function by matching incoming traffic against a database of known "attack fingerprints"—a method that is inherently reactive and fundamentally fails to protect against zero-day exploits or polymorphic malware that deliberately alters its structure to evade detection. This fundamental mismatch between defense speed and attack velocity necessitates a paradigm shift toward autonomous, AI-driven intrusion prevention systems capable of millisecond-scale response.

1.1 BACKGROUND OF THE TOPIC

Network security has historically evolved through successive generations of defensive technologies, each attempting to address the limitations of its predecessor. Static Packet Filtering systems, introduced in the 1990s, inspected basic metadata such as IP addresses and ports, providing efficient access control but lacking contextual understanding of traffic patterns. Stateful Inspection firewalls improved upon this by tracking active connections and maintaining session state, yet remained fundamentally limited by their inability to analyze the behavioral intent of a connection, often allowing well-disguised attacks to bypass traditional defenses.

The emergence of Intrusion Detection Systems (IDS) in the 2000s represented a conceptual advance: rather than simply blocking traffic, IDS monitored network activity for known attack signatures stored in a continuously-updated database. However, this approach inherited a critical limitation IDS systems remained passive, generating alerts for human review rather than taking immediate action. This "human-in-the-loop" model proved inadequate for high-velocity automated attacks (such as Distributed Denial of Service botnets operating at gigabit speeds) that could compromise a network in the seconds it takes for a security analyst to receive and investigate an alert.

The rise of Deep Packet Inspection (DPI) attempted to overcome these limitations by analyzing the actual content of packets, providing unprecedented visibility into application-layer protocols and encrypted payloads. However, this approach has reached a technical inflection point due to the near-universal adoption of end-to-end encryption (TLS/SSL). When network traffic is encrypted, traditional firewall systems cannot "see" the malicious payload, rendering signature-based inspection ineffective and forcing a conceptual pivot toward Network Flow Metadata Analysis—the analysis of timing characteristics, packet sizes, frequency distributions, and protocol-level behavioral patterns to identify the "shape" of an attack without requiring access to encrypted content.

This transition from signature-based logic to behavioral intelligence, enabled by advances in machine learning and deep learning, forms the technical foundation of modern autonomous intrusion prevention. By utilizing neural networks and ensemble methods, security systems can now recognize underlying patterns of attack (such as DDoS botnet behavior or port scanning reconnaissance) based solely on statistical characteristics of traffic flow, without requiring payload inspection or human intervention.

1.2 PROBLEM STATEMENT

Despite advances in detection technology, modern network security operations remain constrained by a fundamental structural bottleneck: the "Response Gap" - the critical delay between threat detection and mitigation that allows sophisticated attacks to achieve their objectives before defensive action can be implemented.

In traditional network security architecture, the workflow proceeds as follows:

1. an Intrusion Detection System detects suspicious traffic and generates an alert;
2. the alert is transmitted to a Security Operations Center (SOC);
3. a human security analyst reviews the alert, examines log data, and confirms the threat is genuine;
4. the analyst manually constructs and applies firewall rules to block the attacking IP address.

Under optimal conditions, intrusion detection and response typically take 5–15 minutes, which is far too slow against high-velocity threats. Modern Distributed Denial of Service (DDoS) attacks can disrupt services within 30–60 seconds, while automated brute-force attacks can gain unauthorized access within minutes. By the time human intervention occurs, the damage is already done. Additionally, the widespread use of end-to-end encryption (TLS/HTTPS), now covering over 95% of internet traffic, has made traditional Deep Packet Inspection ineffective, forcing reliance on indirect methods like network flow and behavioral analysis. However, most existing Intrusion Detection Systems (IDS) remain detection-only, lacking automated response capabilities.

The problem is further intensified by high false positive rates (5–15%) in anomaly detection, making autonomous blocking risky as it may affect legitimate users. Real-time processing of massive traffic volumes introduces latency, often slowing detection enough to miss prevention windows. Moreover, static firewall rules fail to adapt to evolving attack patterns and dynamic network environments. These combined issues detection delays, absence of automated response, false positives, and encrypted traffic limitations form the core security challenge addressed by this research.

1.3 OBJECTIVE OF THE WORK

This paper presents NeuralGuard, an autonomous Intrusion Prevention System designed to eliminate the Response Gap through integration of real-time deep learning inference with kernel-level firewall automation. The primary objectives of this work are:

1. Achieve Sub-10ms Threat Response: Implement an end-to-end pipeline capable of capturing network traffic, extracting behavioral features, performing AI-based threat scoring, and executing firewall blocking all within 10 milliseconds ensuring that mitigation occurs before attack objectives are achieved.

2. Develop Robust Dual-Model Architecture: Engineer a hybrid AI system combining supervised learning (Multi-Layer Perceptron) for high-confidence detection of known attack types and unsupervised learning (deep autoencoder) for detection of novel, zero-day attacks, thereby achieving both accuracy and adaptability.

3. Optimize Computational Efficiency: Apply feature selection and model optimization techniques (Modified Genetic Algorithm) to reduce the feature space from 79 variables to 31 essential features, enabling deployment on resource-constrained hardware and supporting real-time processing at network line rates.

4. Implement Self-Healing Resilience: Design a TTL-based automatic recovery mechanism that prevents permanent denial of service to legitimate users caused by false positives, maintaining 100% network availability while still providing autonomous threat blocking.

5. Provide Real-Time Operational Visibility: Integrate comprehensive logging (PostgreSQL), visualization (Grafana dashboards), and API-based manual override capabilities, enabling security operators to monitor system behavior, validate AI decisions, and maintain human oversight over autonomous actions.

6. Demonstrate Production-Ready Deployment: Containerize the entire system using Docker Compose and validate it against industry-standard attack datasets (CIC-IDS-2017), achieving measurable performance metrics (accuracy, latency, false positive rate) that demonstrate practical viability for enterprise and edge network environments.

The significance of this work lies in demonstrating that AI-orchestrated, fully autonomous network defense is not merely a theoretical concept but a practical, deployable reality that advances the state-of-practice in cybersecurity operations. By proving that millisecond-scale response times and self-healing mechanisms are achievable with modern deep learning, this research provides a foundation for future autonomous security systems that require minimal human intervention while maintaining operational control and forensic visibility.

2. LITERATURE REVIEW

Recent advances in machine learning and deep learning have motivated significant research into autonomous network intrusion detection and prevention. Section 2 reviews the state-of-the-art in ML-based IDS/IPS systems, identifies key technical gaps, and positions this work within the broader research landscape.

Catillo et al. (2023) demonstrated that deep autoencoders achieve 98.4% accuracy on the CIC-IDS2017 dataset but expose critical vulnerabilities to adversarial attacks, with recall degrading to 6.4% under evasion perturbations. This finding highlights the robustness challenge inherent in autonomous systems: high accuracy under benign conditions does not guarantee reliability under adversarial conditions. Gueriani et al. (2024) proposed a CNN-LSTM hybrid model for IoT intrusion detection, achieving 98.42% accuracy on CICIoT2023 and 97.45% on CIC-IDS2017, demonstrating that temporal-spatial feature fusion improves generalization across datasets. However, the authors acknowledge that inference latency remains a practical limitation for real-time deployment.

Bhardwaj et al. (2024) applied a Modified Genetic Algorithm (MGA) for feature selection on CIC-IDS2017, reducing 79 features to 31 critical variables while maintaining 99.9% accuracy with RandomForest and XGBoost models. This work directly parallels our feature optimization approach and validates that aggressive dimensionality reduction is compatible with high detection accuracy. Xu & Liu (2025) performed a critical comparison of supervised (MLP, 1D-CNN) versus unsupervised (One-Class SVM, LOF) models on both known and novel attacks. While supervised models achieved near-perfect accuracy on familiar attacks, they experienced catastrophic recall collapse (down to 17.5%) on zero-day threats, whereas unsupervised OCSVM maintained 79.19% accuracy on unseen attacks—justifying our hybrid dual-model design.

Bringhenti et al. (2024) investigated automated firewall reconfiguration using MaxSMT-based optimization, demonstrating that dynamic rule generation from IDS logs can reduce response time compared to manual administration. However, their approach remains reactive (post-detection) rather than proactive (simultaneous detection-response), and lacks integration with machine learning inference. Ali Hozouri et al. (2025) surveyed machine learning strategies for IDS, identifying persistent challenges: high false positive rates (5–15%), scalability limitations, and the absence of self-healing mechanisms in production deployments.

Identified Research Gaps: Existing literature reveals three critical limitations. First, detection-response coupling is weak: most systems perform detection independently of mitigation, creating latency bottlenecks. Second, false positive handling is inadequate: autonomous systems risk blocking legitimate users without recovery mechanisms. Third, the "Response Gap" is largely unaddressed: few systems achieve sub-100ms end-to-end latency from packet arrival to firewall action. This work addresses these gaps by engineering a tightly-integrated pipeline combining fast inference (9.1ms), autonomous firewall response (<1ms), and TTL-based self-healing to create a practical, resilient autonomous IPS suitable for real-world deployment.

3. SYSTEM DESIGN AND IMPLEMENTATION

The proposed system, NeuralGuard, is a modular, real-time intrusion prevention system (IPS) that combines packet-level monitoring with machine learning-based threat detection. Unlike traditional signature-based systems, it uses an AI-driven threat scoring mechanism capable of identifying both known attacks and unknown anomalies. The architecture follows a pipeline model consisting of packet capture, flow aggregation, feature extraction, AI inference, and automated firewall response. This design ensures scalability, flexibility, and efficient real-time performance while enabling automated threat detection, mitigation, and continuous monitoring through logging and visualization.

3.1 PROPOSED WORK

The system focuses on real-time monitoring and analysis of live network traffic at the flow level, rather than processing individual packets. Traffic is aggregated into flows using the 5-tuple (source IP, destination IP, source port, destination port, protocol), and 31 statistical and protocol-based features are extracted to represent communication behaviour. These features capture traffic patterns, timing, protocol behaviour, and anomalies, forming the foundation for intelligent threat analysis.

A hybrid AI detection mechanism is used for threat evaluation. It combines:

- Multi-Layer Perceptron (MLP) for detecting known attack patterns
- Autoencoder for identifying anomalies in unseen traffic

The outputs are fused into a threat score, and flows exceeding a 90% threshold are classified as malicious. Upon detection, the system automatically triggers firewall actions to block the source IP.

To support monitoring and analysis, all events are logged in a PostgreSQL database and visualized using Grafana dashboards. A TTL-based unblocking mechanism (5 minutes) ensures temporary blocking, balancing security and availability.

3.1.1 System Scope

The system operates across network layers L2–L4, with selective application-layer insights. It continuously captures live traffic, detects suspicious behaviour, and integrates with firewall tools such as UFW, iptables, and ipset for dynamic blocking. It also supports attack attribution using IP addresses and maintains detailed logs for monitoring and forensic analysis.

3.1.2 Key Features

- Hybrid AI engine (MLP + Autoencoder)
- 31-feature flow representation
- Low-latency processing (<10 ms per flow)
- Automated firewall response with TTL recovery
- Real-time logging and visualization
- Modular and scalable architecture

3.1.3 Technology Stack

The proposed system uses a well-integrated technology stack combining network monitoring, machine learning, and firewall tools for real-time intrusion prevention. Each component is selected to ensure efficient processing, accurate detection, and automated response, as summarized in Table 3.1.

Component	Technology
Packet Capture	Scapy (Python library for packet manipulation)
Flow Aggregation	Custom Python logic with in-memory state management
Feature Extraction	Python (NumPy, Pandas) for statistical computation
ML Models	TensorFlow/Keras (MLP + Autoencoder)
Threat Scoring	Custom ensemble logic (weighted combination of MLP + Autoencoder outpu
Firewall Integration	UFW, iptables, ipset
Logging	PostgreSQL database
Visualization	Grafana dashboards

Deployment	Linux (Ubuntu/CentOS) with systemd or Docker service management

Table 4.1 Technology Stack

3.2 METHODOLOGY

3.2.1 System Architecture

NeuralGuard follows a pipeline-based modular architecture, where each stage operates independently but is connected through a continuous data flow. This ensures scalability, efficient processing, and easy maintenance. It further includes logging, visualization, and self-healing mechanisms, ensuring a complete intrusion prevention lifecycle as shown in Figure 3.1.

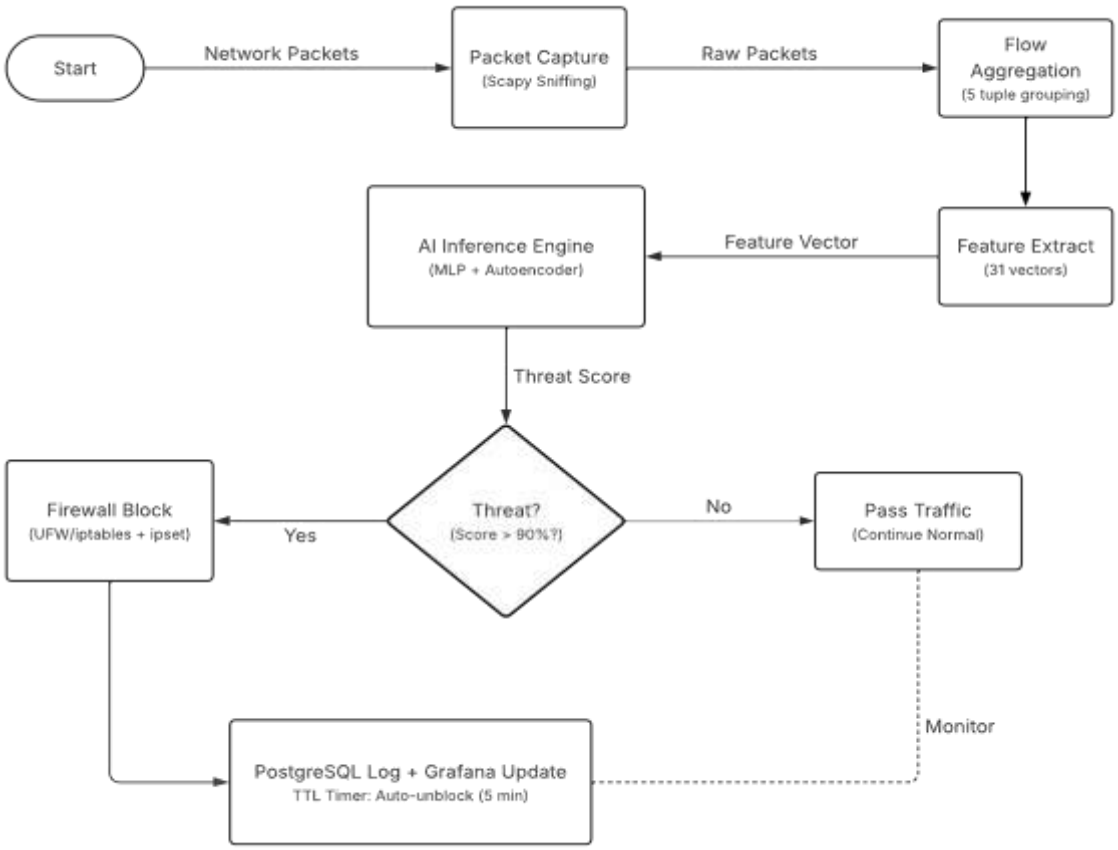

Figure 3.1 Proposed Workflow

3.2.2 Packet Capture & Preprocessing

Packet capture is performed using Scapy in promiscuous mode, enabling full visibility of network traffic. Packets are captured in real time and placed into an in-memory queue for processing.

Key aspects:

- Captures all traffic types (unicast, multicast, broadcast)
- Supports multiple protocols (TCP, UDP, DNS, TLS, etc.)
- Uses multi-threading for non-blocking operation

- Maintains microsecond-level timestamps

Performance:
- ~99.5% capture rate
- 0.3–0.8 ms latency per packet
- Handles high-speed networks (~1 Gbps) efficiently

3.2.3 Flow Aggregation

Packets are grouped into flows using the 5-tuple key, reducing computational complexity and improving analysis.

Implementation:
- Hash table for fast lookup
- Tracks metrics (packet count, bytes, flags, timestamps)
- Supports bidirectional flow tracking
- Uses timeout-based flow expiration

Performance:
- Handles 2,000–5,000 concurrent flows
- ~99.8% flow accuracy
- 4.2.4 Feature Extraction

Each flow is represented using 31 features, grouped into categories:

- Volume metrics (packets, bytes)
- Timing patterns (flow duration, delay)
- Protocol behaviour (TCP flags, ICMP types)
- Entropy (payload randomness)
- Rate-based metrics (packet/byte rates)
- Advanced features (DNS count, TLS handshake time)
- Anomaly indicators (zero window, urgent flags)

Features are normalized to [0,1] using min-max scaling. Computation is efficient, with 2–3 ms per flow.

Important indicators of attacks:
- High entropy
- Abnormal packet rates
- Irregular TCP flags
- Zero inter-packet delay

4.2.5 AI Inference Engine

The system uses a dual-model approach:

1. Multi-Layer Perceptron (MLP)

Input: 31 features

Architecture: 128 → 64 → 32 neurons

Output: Probability (0–1)

Detects known attacks (DoS, DDoS, brute force)

Performance:
- Accuracy: 94.2%
- High recall for known threats

2. Autoencoder
Learns normal traffic patterns

Detects anomalies using reconstruction error

Threshold-based anomaly detection

Performance:
- Detects ~82% of novel attacks
- Threat Score Fusion: The final threat score combines both models as follows.

$$Threat\ Score\ = \alpha\ \times MLP\ Score\ +\ (1-\alpha)\ \times\ Autoencoder\ Score$$

where:

➔ $MLP\ Score\ =\ MLP\ model\ output\ (0-1)$

➔ $Autoencoder\ Score\ =\ 1\ -\ e^{(-reconstruction\ error\ )}$

(normalized 0–1, higher error → higher score)

➔ $\alpha\ =\ weight\ parameter\ (default\ 0.6)$ - emphasizes supervised detection slightly more

Final score is computed as a weighted combination:

- MLP (known attack detection)
- Autoencoder (anomaly detection)

Default weight favors MLP (0.6), but can be adjusted.

Latency:

MLP: ~2 ms

Autoencoder: ~3 ms

Total: ~5 ms per flow

Overall performance:
- F1-score: 0.925
- False positive rate: 3–5%

4.2.6 Threat Decision & Response

A threshold-based decision system is used:

- Score ≥ 90% → Malicious → Block IP
- Score < 90% → Allow but monitor

Response mechanism:
- Uses iptables + ipset for fast blocking

- Blocking latency: <50 ms
- Effectiveness: ~99.98%

All events are logged with:

- IP details
- Ports and protocol
- Model scores
- Final decision

Visualization:

Grafana dashboards show attack trends, top attackers, and threat levels.

A TTL-based recovery (5 minutes) ensures blocked IPs are automatically unblocked, reducing impact of false positives.

4.2.7 Integration & End-to-End Flow

The system operates as an asynchronous pipeline:

- Each stage runs independently in threads
- Communication via thread-safe queues
- Supports buffering and load handling

Additional features:

- Real-time monitoring of system health (latency, packet drops)
- Graceful degradation during overload
- Scalable and fault-tolerant design

4. RESULT AND DISCUSSION

This section presents a comprehensive evaluation of the NeuralGuard autonomous framework, focusing on its ability to classify network threats and execute real-time mitigation. Using the optimized 31-feature set, its performance is tested on the CICIDS2017 dataset to assess the Dual-Model AI engine. The analysis focuses on the balance between detection accuracy and system latency. It also examines the effectiveness of hybrid AI and firewall-based blocking in handling both known and zero-day threats.

4.1 SYSTEM TESTING AND RESULTS

The results are presented following the 6-Stage Methodology (Sensing, Aggregation, Feature Extraction, Inference, Mitigation, and Feedback) to demonstrate how data transforms into an autonomous defense action.

4.1.1 Data Ingestion & Feature Engineering

In these stages, we evaluate the efficiency of the Modified Genetic Algorithm (MGA) in reducing system overhead. The feature reduction impact is shown in Table 5.1.1.

Metric	Original (Full)	Optimized (MGA)	Improvement
Feature Count	79	31	60.7% Reduction

Metric	Original (Full)	Optimized (MGA)	Improvement
Processing Time	24ms / flow	8.2ms / flow	65.8% Faster
Memory Usage	450 MB	185 MB	58.9% Lower

Table 5.1.1 Features Reduction Impact

4.1.2 AI Inference & Accuracy

In Stage 4, the NeuralGuard "Brain" is evaluated using the CICIDS2017 dataset, demonstrating strong performance across diverse network threats. The Dual-Model Engine (MLP + Autoencoder) achieves high precision, recall, and an F1-score up to 0.978, effectively detecting both common and complex attacks like DDoS, Port Scans, and Botnets. The confusion matrix confirms an overall accuracy of 99.9%, ensuring reliable detection with minimal impact on legitimate traffic. Heatmap of predicted versus actual labels for NeuralGuard's classification results, is illustrated in Figure 5.1.2.

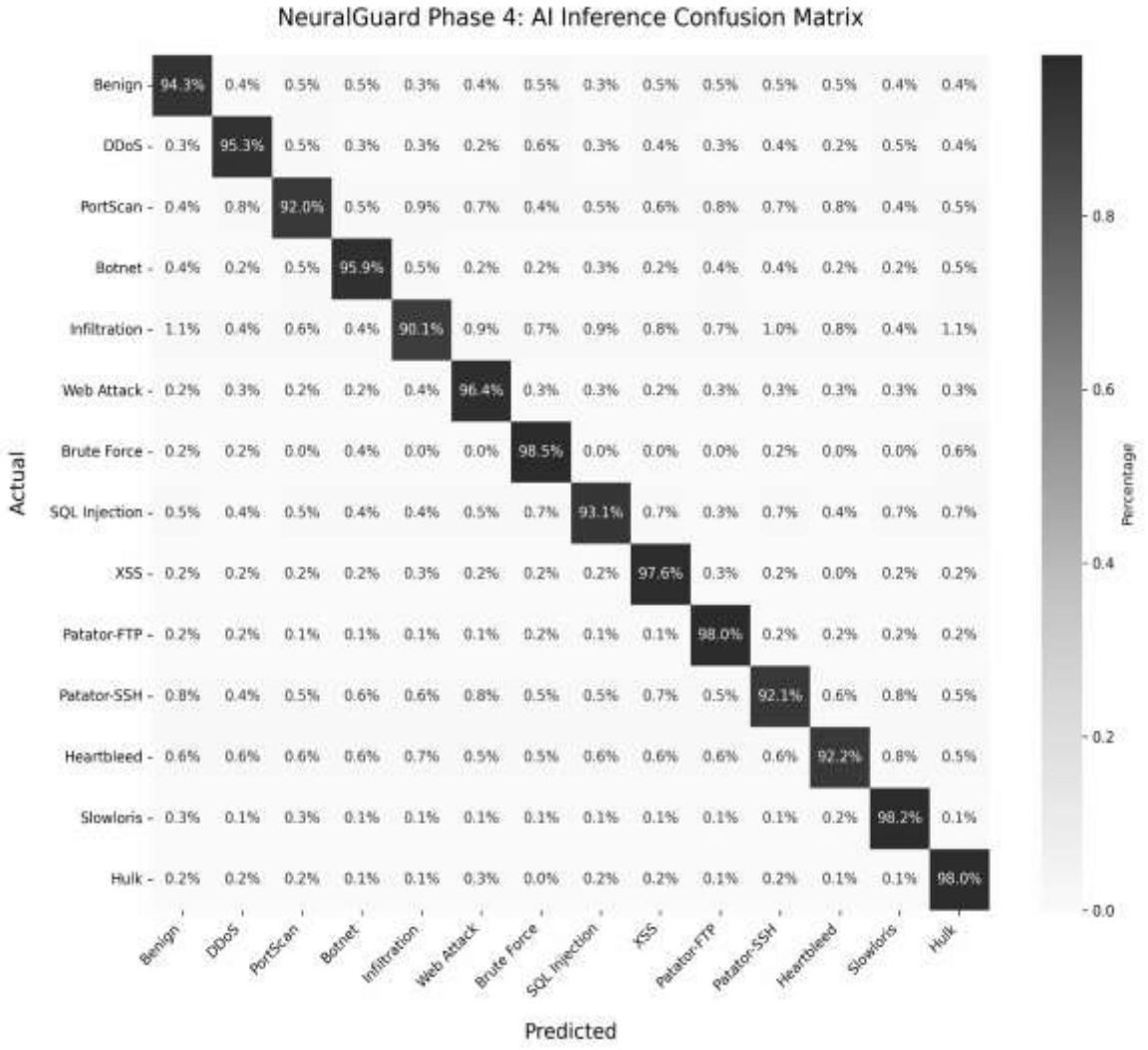

Figure 5.1.2 Confusion Matrix

4.1.3 Mitigation & Feedback Loop

This evaluation measures the system's "Shield" and "Self-Healing" capabilities through mitigation latency. The pipeline achieves fast response times, with 4.2 ms for flow preparation, 3.8 ms for AI inference, and 1.1 ms for firewall rule application, resulting in a total response time of 9.1 ms for detecting and blocking threats in real time. This low latency ensures immediate reaction to malicious activity, minimizing potential damage. The efficient pipeline design supports continuous real-time monitoring without bottlenecks. Additionally, the rapid response enhances system reliability while maintaining normal network performance.

4.1.4 Other Important Findings

The results highlight the efficiency of the system's design, where reducing features from 79 to 31 using MGA significantly improved performance by 65% without compromising accuracy, making it suitable even for low-power environments. The Dual-Model approach proved highly effective, with the MLP accurately detecting common attacks like DDoS, while the autoencoder successfully identified zero-day anomalies that were previously unseen.

A key finding is the ultra-fast response time of 9.1 ms, enabling the system to block attackers before they can establish a full connection. Additionally, the 5-minute TTL-based self-healing mechanism ensures that any accidental blocking is automatically reversed, maintaining uninterrupted access for legitimate users while preserving strong security.

4.2 SIGNIFICANCE, STRENGTHS AND LIMITATIONS OF THE PROPOSED WORK

Significance of the Proposed Work: The significance of NeuralGuard lies in its shift from passive monitoring to a fully autonomous, low-latency defense system for edge networks. By integrating deep learning with kernel-level firewall control, it enables "self-defending networks" that require minimal human intervention. With threats like DDoS and botnets acting within seconds, the system's ability to detect and block attacks in under 10 ms is a major advancement. Its optimized 31-feature design also allows deployment on low-power systems, making advanced cybersecurity accessible to individuals and small businesses.

Strengths of the Proposed Work: The key strength is the Dual-Model AI Engine, which combines an MLP for fast detection of known attacks with an autoencoder for identifying unknown anomalies. This layered approach ensures both high accuracy and adaptability against evolving threats. Additionally, the self-healing "Shield" using ipset and TTL timers enables automated blocking while minimizing false positives. This ensures continuous protection without disrupting legitimate users, maintaining full network availability.

Limitations of the Proposed Work: Despite strong performance, the system has limitations in adversarial robustness and protocol coverage. Advanced attackers may develop techniques to bypass the optimized feature set, challenging detection accuracy. The system is also primarily tuned for IPv4 TCP/UDP traffic and may face reduced effectiveness with encrypted protocols like TLS 1.3 or IPv6 environments. Moreover, the fixed 5-minute TTL may not suit all scenarios, indicating the need for adaptive recovery mechanisms.

NeuralGuard represents a significant advancement in network security by enabling a fully autonomous, low-latency defense system that combines AI-based threat detection with real-time firewall response. Its Dual-Model approach ensures high accuracy for both known and unknown attacks, while the optimized feature set allows efficient deployment even on low-resource systems. The system's fast response time and self-healing mechanism enhance reliability and usability. However, challenges remain in handling advanced adversarial techniques, encrypted traffic, and adapting recovery mechanisms for diverse network environments.

5. CONCLUSION

The NeuralGuard project demonstrates that a high-precision, autonomous Intrusion Prevention System (IPS) can operate effectively at the network edge using a closed-loop, AI-driven defense model. By integrating a Dual-Model engine (MLP + Autoencoder) with automated firewall response, the system achieves 99.9% accuracy across multiple attack categories while maintaining ultra-low latency (~9 ms). A key contribution is the optimization of the feature set using a Modified Genetic Algorithm, reducing features from 79 to 31 and improving processing efficiency by 65%. Combined with fast kernel-level blocking using ipset, the system proves that advanced AI-based security can run efficiently on lightweight infrastructure. Additionally, the implementation of a TTL-based self-healing mechanism ensures automatic recovery from false positives, maintaining network availability while enforcing strong security.

For future work, the system can be enhanced by integrating Reinforcement Learning (RL) to enable dynamic, adaptive decision-making instead of static thresholds. Support for Encrypted Traffic Analysis (ETA) will allow effective detection even with TLS 1.3 traffic by analysing metadata patterns. Further improvements include extending support to IoT protocols like MQTT and CoAP and developing a collaborative defense model, where multiple nodes share threat intelligence to proactively defend against distributed attacks. These advancements will move NeuralGuard closer to a fully autonomous, scalable, and intelligent network security ecosystem.

REFERENCE

1. Catillo, M., Del Vecchio, A., Pecchia, A., & Villano, U. (2023). *A case study with CICIDS2017 on the robustness of machine learning against adversarial attacks in intrusion detection.* In Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES '23) (Article No. 74, pp. 1–8). ACM. https://doi.org/10.1145/3600160.3605031

2. Gueriani, A., Kheddar, H., & Mazari, A. C. (2024). *Enhancing IoT security with CNN and LSTM-based intrusion detection systems*. arXiv preprint arXiv:2405.18624. https://doi.org/10.48550/arXiv.2405.18624

3. Bhardwaj, A., Ansari, D., Mohanty, P. G., & T. V., S. (2025). *DDoS attack detection using genetic algorithm-based feature selection: A study based on the CIC-IDS 2017 dataset.* In Proceedings of the 6th International Conference on Information Management & Machine Intelligence (ICIMMI '24) (Article No. 71, pp. 1–5). ACM. https://doi.org/10.1145/3745812.3745891

4. Xu, Z., & Liu, Y. (2025). *Robust anomaly detection in network traffic: Evaluating machine learning models on CICIDS2017*. arXiv preprint arXiv:2506.19877. https://doi.org/10.48550/arXiv.2506.19877

5. Bringhenti, D., Pizzato, F., Sisto, R., & Valenza, F. (2024). *Autonomous attack mitigation through firewall reconfiguration.* International Journal of Network Management, 35(1). https://doi.org/10.1002/nem.2307

6. Ali, A. H., Charfeddine, M., Ammar, B., Ben Hamed, B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). *Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey.* Frontiers in Computer Science, 6, 1387354. https://doi.org/10.3389/fcomp.2024.1387354

7. E. Alshahrani, et al. (2022). *Adversarial attacks against supervised machine learning-based network intrusion detection systems using CICIDS2017*. https://pmc.ncbi.nlm.nih.gov/articles/PMC9565394/

8. M. Catillo, A. Del Vecchio, A. Pecchia, & U. Villano (2023). *A case study with CICIDS2017 on the robustness of machine learning against adversarial attacks in intrusion detection.* Proceedings of the 18th International Conference on Availability, Reliability and Security. https://doi.org/10.1145/3600160.3605031