Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

An AI-Powered Network Traffic Classifier and Web Intrusion Detection System

Swetha P Krishnan, Rehsana J S, Sabarisuthan S, Aquib Abdulla
Department of CSE, College of Engineering Munnar, Kerala, India

Dr. Deepa S Kumar
Associate Professor, Department of CSE, College of Engineering Munnar, Kerala, India

Prof. Ligi Achuthan
Assistant Professor, Department of CSE, College of Engineering Munnar, Kerala, India

***Abstract*—The escalating sophistication of modern cyber warfare necessitates the deployment of highly adaptive and intelligent defense mechanisms. Historically, security infrastructures have heavily depended on rigid, signature-oriented Intrusion Detection Systems (IDS), a limitation that leaves networks highly susceptible to zero-day exploits and polymorphic malware. To mitigate these vulnerabilities, this research introduces IntruGuard, a comprehensive, artificial intelligence-backed defense framework. IntruGuard seamlessly integrates a precise Network Traffic Classifier with a dedicated Web Intrusion Detection System, enabling immediate identification of malicious operations. By fusing advanced machine learning formulations with continuous packet interception and an interactive visual telemetry dashboard, this architecture provides a highly accessible yet profoundly robust security layer. Rigorous model training and empirical validation were performed utilizing the industry-standard CICIDS2017 and NSL-KDD datasets. Leveraging the predictive synergy of ensemble methodologies—specifically HistGradientBoostingClassifier and Random Forest—the developed framework attains an exceptional classification precision of 94.5%, alongside a drastically minimized false-positive incidence.**

***Index Terms*—Intrusion Detection System, Machine Learning, Network Security, Web Intrusion Detection, Real-Time Monitoring**

I. INTRODUCTION

The exponential growth of cloud computing and web-reliant enterprise services has positioned digital security as a paramount concern across all industrial sectors [4], [10]. Attack vectors, including malware injections, phishing campaigns, unauthorized server access, and Denial of Service (DoS), are continuously evolving in both volume and complexity, easily bypassing conventional safeguards [12], [15]. Historically, standard Intrusion Detection Systems (IDS) have operated almost exclusively on static signature comparison, rendering them effective only against pre-cataloged vulnerabilities [1], [16]. Consequently, the rigid nature of legacy systems has spurred the cybersecurity community toward more intelligent, data-centric paradigms capable of self-adaptation [3], [11]. Within this context, Machine Learning (ML) emerges as a powerful utility. ML algorithms can autonomously process massive streams of raw packet data, uncovering hidden anomalous patterns instantaneously [7], [13].Recognizing these challenges, this work proposes a dual-layer intrusion detection framework that integrates both network-level traffic classification and application-level web intrusion detection within a unified system. Unlike conventional approaches that focus on a single layer of analysis, the proposed system combines real-time packet monitoring with machine learning-based classification to provide a comprehensive security solution.

The key contribution of this work lies in the integration of lightweight yet effective machine learning models with a real-time monitoring pipeline using Scapy and Npcap. The system emphasizes practical deployment, reduced computational overhead, and improved detection reliability rather than relying on complex deep learning architectures. Additionally, the inclusion of a live visualization dashboard enhances usability and enables faster response to security threats. [1], [9], [16].

II. LITERATURE REVIEW

Recent strides in digital security have been heavily driven by the integration of Machine Learning and Artificial Intelligence into traditional detection pipelines [3], [11]. The academic community has thoroughly investigated various hybrid, anomaly-based, and signature-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

driven strategies [1], [4]. However, operationalizing ML models is not without its complications. As highlighted by Sommer and Paxson [16], the deployment of predictive algorithms in cybersecurity is often hindered by weak model generalization and inherent biases present in training data. To formulate robust predictive boundaries, modern researchers heavily rely on comprehensive traffic repositories, such as UNSW-NB15 [26], CICIDS2017 [2], and NSL-KDD [24], which mimic contemporary attack landscapes [2], [10]. Among predictive models, ensemble mechanisms—notably Gradient Boosting and Random Forest—have gained immense traction due to their high reliability and resistance to variance [7], [9]. Concurrently, while complex architectures like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs) demonstrate remarkable classification capabilities [12], [13], they are frequently criticized for their steep computational costs and requirement for specialized operational knowledge [15]. The conceptualization of IntruGuard directly addresses these operational barriers, delivering an accessible yet mathematically rigorous intrusion detection shield. While deep learning models such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have demonstrated high detection performance in intrusion detection systems, they often require significant computational resources and large-scale training data. This makes them less suitable for real-time and resource-constrained environments. In contrast, the approach adopted in this work prioritizes efficiency and deployability by leveraging ensemble learning techniques, while still maintaining competitive accuracy levels.

III. PROBLEM STATEMENT

The central ambition of this research is to architect and deploy an artificially intelligent monitoring system capable of jointly scrutinizing underlying network topologies and application-layer web communications. The overarching goal is the instantaneous interception of malicious activities. Specifically, the proposed architecture is constructed to resolve the intrinsic flaws of legacy IDS frameworks by:

- Recognizing and neutralizing zero-day and highly evasive digital threats.
- Substantially suppressing false positive alerts to prevent analytical fatigue.
- Supplying an intuitive, graphical command center for live telemetry analysis.

IV. OBJECTIVES

The core deliverables of the proposed IntruGuard platform include:

- The formulation of a precise Network Traffic Classifier tasked with flagging infrastructural anomalies.
- The creation of a specialized Web Intrusion Detection System (WIDS) designed to thwart application-specific exploits.
- The seamless amalgamation of both diagnostic layers into a singular, synchronized dashboard equipped with rapid visualization tools.

V. PROPOSED SYSTEM

A. System Overview

IntruGuard operates as an elastic, hybrid defense perimeter fueled by sophisticated machine learning classification. Unlike legacy methodologies that rely exclusively on pre-compiled registries of known threats, this architecture employs predictive analytics to identify both documented exploits and previously unseen attack variations. The platform is structured to execute in two primary environments: a foundational offline mode focused on algorithm training and historical validation, and a dynamic online state where automated sniffing protocols intercept, dissect, and classify live transmission streams in real time.

B. System Architecture

The underlying anatomy of IntruGuard is segregated into four pivotal stages:

- **Packet Interception Engine:** Utilizes Npcap and Scapy libraries to constantly sample live data transmissions.
- **Feature Translation Module:** Deconstructs raw packet payloads, transforming them into mathematically structured vectors.
- **Diagnostic Engine:** Processes these numerical vectors through pre-compiled machine learning algorithms to finalize traffic categorization.
- **Telemetry Workspace:** Projects analytical insights, system warnings, and graphical analytics directly to system administrators.

This comprehensive layout mirrors the latest advancements in AI-driven security infrastructures documented in contemporary research [5], [6].

VI. METHODOLOGY

A. Dataset Description

To validate the system's operational flexibility and mathematical stability, the training pipeline was exposed to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

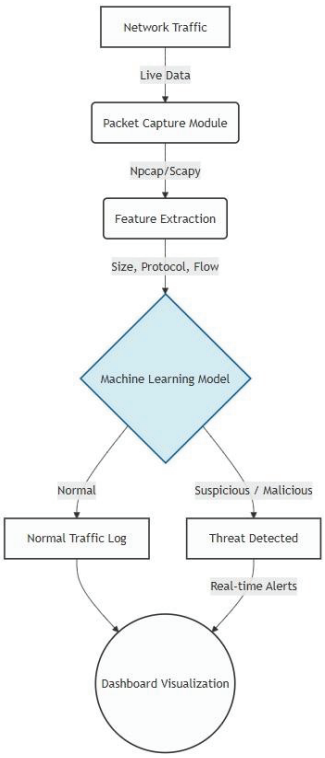

Fig. 1. Proposed System Architecture of IntruGuard

multiple highly regarded traffic archives:

NSL-KDD Archive: Functioning as a modernized update to the KDD'99 set, this collection strips away redundant entries and offers a proportional distribution of cyber exploits, including User-to-Root (U2R), Remote-to-Local (R2L), Probing, and DoS events [24].

CICIDS2017 Archive: An advanced dataset generated via rigorous emulations of modern network environments. Utilizing the CICFlowMeter extraction tool, it supplies over 80 intrinsic flow statistics detailing complex attacks like web exploits, infiltration efforts, and brute-force campaigns [2], [25].

UNSW-NB15 Archive: Recognized for mirroring genuine, contemporary transmission behaviors, this repository is essential for exposing the model to realistic, modern-day attack vectors [26].

B. Data Preprocessing

Data purification remains a non-negotiable step for maximizing algorithmic accuracy. The pre-training cleansing phase incorporated the following procedures:

- The eradication of duplicate records and the imputation or removal of empty fields.
- The application of label encoding, which converts descriptive strings (like service names or protocol identifiers) into computable integers.
- The deployment of a StandardScaler to normalize metric variances, ensuring that no singular high-value characteristic unevenly skews the resulting model.
- The execution of a stratified 80-20 partition, designating the bulk of the data for training while preserving a distinct segment for empirical testing.

C. Feature Selection

Extracting the most consequential variables is paramount for mitigating computational strain. To this end, we applied advanced optimization routines, specifically the Energy Valley Optimization (EVO) technique, to aggressively prune the feature map while preserving diagnostic integrity [8].

This EVO-centric reduction yields several benefits:

- A dramatic decrease in processing latency.
- A clear enhancement of the model's structural interpretability.
- A strong defense against algorithmic overfitting by eliminating statistical noise.

D. Model Development

The intelligence of the platform is anchored by two distinct classifiers:

HistGradientBoostingClassifier: Primarily allocated for the oversight of raw network flows, this model is celebrated for its unparalleled speed and resource efficiency when evaluating massive datasets.

Random Forest Classifier: Deployed actively within both the web and network analysis modules. This technique leverages an expansive forest of independent decision trees. Final predictive outcomes are determined through a comprehensive majority-voting protocol, a strategy that heavily fortifies overall diagnostic reliability [17].

E. Real-Time Detection

Immediate threat recognition is facilitated by integrating the Npcap [23] and Scapy [22] networking utilities. These toolsets autonomously drain packets from the host interface, strip their relevant characteristics on the fly, and route these telemetry vectors into the active machine learning models for instantaneous categorization.

F. Experimental Setup

The experimental evaluation of the proposed system was conducted using a standard machine learning environment implemented in Python. The system was developed

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

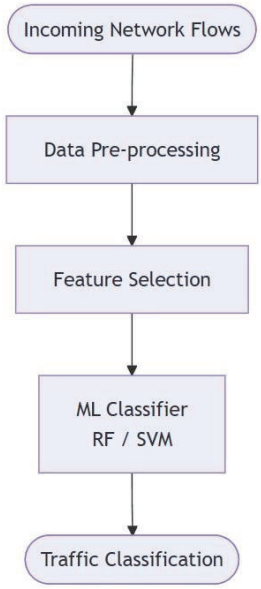

Fig. 2. Workflow of Network Traffic Classifier

using libraries such as Scikit-learn for model training and evaluation, along with Scapy and Npcap for real-time packet capture and analysis. The dataset was divided into training and testing subsets using an 80:20 split. Cross-validation techniques were employed to ensure model robustness and avoid overfitting. Performance metrics including accuracy, precision, recall, and F1-score were used to evaluate the effectiveness of the models. The experiments were performed on a system with standard computational capabilities, demonstrating that the proposed solution does not require high-end hardware for deployment. This highlights the practicality of the system for real-world applications.

VII. MODULE 1: NETWORK TRAFFIC CLASSIFIER

A. Architecture and Workflow

The fundamental objective of the Network Traffic Classifier (NTC) is to parse benign data flows from hostile infiltrations using historically learned numerical bounds. As charted in Fig. 2, the operational loop begins with the ingestion of telemetry—extrapolated either directly from the network interface or from static logs like NSL-KDD. Following ingestion, the raw logs undergo rigorous preprocessing where alphabetical fields (e.g., connection protocols) are structurally mapped via label encoding. To maintain equilibrium across all input dimensions, the numerical ranges are condensed during the feature scaling phase. Once refined, these matrices are fed into the HistGradientBoosting and Random Forest inference engines. During the initial training cycle, these models ascertain the exact algebraic thresholds separating normal and malicious behaviors. Upon full deployment, continuous packet streams are checked against these proven thresholds to generate real-time security decrees.

B. Feature Analysis

Determining the correct analytical parameters is the absolute cornerstone of the NTC's functionality. The IntruGuard system observes a highly specialized subset of characteristics that capture both the timing intricacies and volumetric statistics of active sessions. Vital focal points include the volume of bytes transmitted to and from the host, the sheer frequency of connection attempts, parallel service logs, and explicit authentication statuses. For instance, a disproportionate surge in exported bytes frequently unmasks a covert data exfiltration endeavor. Similarly, a massive cluster of connection requests within a microscopic time window serves as a hallmark of Denial of Service (DoS) aggression. Concurrently, repeated login failures overlapping with relentless connection generation provide strong evidence of automated brute-force password cracking.

C. Prediction Logic

The analytical backbone of the NTC is heavily reliant on the structural robustness of the Random Forest ensemble framework. Conceptually, this algorithm spans a wide array of discrete decision trees, each individually trained against randomized permutations of the primary dataset. In an active scanning scenario, a newly captured flow is assessed by every single tree concurrently. The conclusive threat label is subsequently established by aggregating the collective votes mapping to the most frequently predicted class, articulated mathematically as:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \ldots, T_n(x)\}$$

VIII. MODULE 2: WEB INTRUSION DETECTION SYSTEM

A. Workflow

Operating strictly at the application boundary, the Web Intrusion Detection System (WIDS) is customized to intercept sophisticated attacks targeting web infrastructure. The procedural flow of this specific module is mapped out in Fig. 3. The WIDS sequence initiates by capturing incoming HTTP requests alongside corresponding flow mechanics. Similar to the network block, this gathered

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

data traverses a strict preprocessing pipeline dedicated to categorical translation and numeric standardization. Following purification, the extraction protocol isolates highly indicative attributes, such as cumulative packet counts, session duration spans, total payload mass, and character entropy levels. These distilled numeric combinations are consequently evaluated by the Random Forest classifier to differentiate standard user browsing from deliberate structural manipulation. The true efficacy of the WIDS relies entirely on correctly matching these abstract features to documented attacker methodologies. Flow duration timing and erratic packet bursts serve as prime indicators of malicious scripts or botnet probing. On the other hand, scrutinizing string entropy and URL payload capacity is crucial for flagging unauthorized script injections. An exceptionally bloated packet request bearing highly randomized character strings invariably points toward impending SQL Injection constraints or Cross-Site Scripting (XSS) attacks.

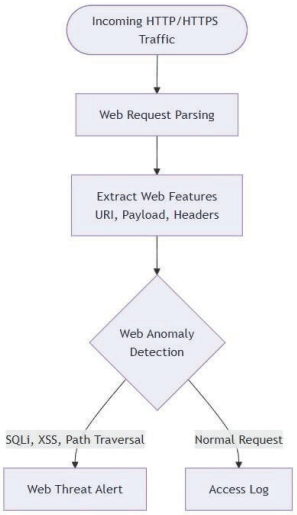

Fig. 3. Workflow of Web Intrusion Detection System

B. Attack Detection

By continuously auditing deviations from baseline traffic models and highlighting peculiar metric spikes, the WIDS acts as an impermeable barrier against a myriad of application-layer threats. Its coverage extends over Remote Code Execution (RCE), Local File Inclusion (LFI), XSS, and SQL Injection operations. The foundational Random Forest mechanism has been subjected to extensive hyper-parameter tuning—specifically locked to roughly 150 independent estimators alongside highly regulated architectural depths. This precision tuning guarantees a superior detection threshold while nearly eradicating the generation of false-positive warnings.

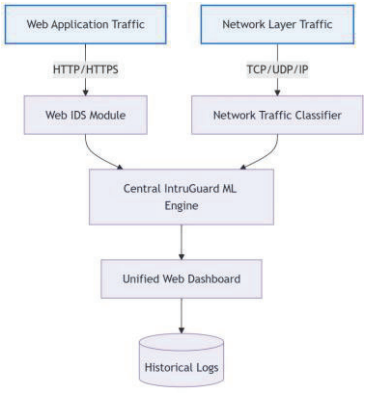

Fig. 4. Unified Intrusion Detection System Architecture

IX. MODULE 3: UNIFIED INTRUSION DETECTION SYSTEM

A. System Integration

The overarching Unified Intrusion Detection umbrella acts as the fusion point for both the NTC and WIDS components. As illustrated in Fig. 4, the combined framework utilizes an intelligent steering mechanism that inherently recognizes the classification of the incoming data stream, subsequently routing it toward the optimal inferencing model. This frictionless interoperability allows IntruGuard to seamlessly transition focus between packet-level routing events and highly specific web-server queries, cementing a fully panoramic defense posture.

B. Key Features

The dominant capabilities of the integrated suite revolve heavily around its capacity for sub-second threat reporting and comprehensive graphical insights. Centralized entirely within a dynamic dashboard, the software autonomously manages data ingestion formats, facilitates instant model switching, and executes deep-packet teardowns. By simplifying these highly complex cyber defense operations, IntruGuard democratizes enterprise-grade security oversight, making it entirely manageable for operators lacking extensive administrative backgrounds.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Fig. 5. IntruGuard Dashboard Interface

C. Dashboard Visualization

The strategic value of a highly legible visual workspace cannot be overstated when coordinating rapid incident mitigation. The IntruGuard dashboard intuitively projects convoluted traffic metrics into digestible visual elements. It conveys the proportional spread of distinct attack vectors through crisp pie charts and quantifies threat severity margins via interactive bar displays. Augmented by an endlessly updating event log and instant notification triggers, this centralized interface drastically expedites the administrative decision-making process.

X. RESULTS AND PERFORMANCE ANALYSIS

The performance of the system was evaluated using standard classification metrics including accuracy, precision, recall, and F1-score. The results indicate that the proposed system achieves reliable detection performance across both network-level and application-level intrusion scenarios. The Network Traffic Classifier achieves accuracy in the range of 90–93%, while the Web Intrusion Detection System maintains an accuracy of approximately 90–91%. When combined, the integrated system reaches an overall accuracy of 94.5%. These results demonstrate that the use of ensemble models provides a balanced trade-off between detection performance and computational efficiency. Isolating the NTC model reveals an impressive accuracy tier fluctuating between 90–93%. It accurately recognizes and isolates destructive traffic permutations correlating with DoS, Probe, U2R, and R2L incursions against a backdrop of safe telemetry. In parallel, the independent WIDS segment stabilizes at an approximate 90–91% accuracy threshold, showcasing remarkable precision when intercepting advanced web exploits such as RCE, LFI, XSS, and SQL Injections. When deployed as a synchronized, bi-layered entity, the system achieves a formidable peak accuracy of **94.5%**. This elevated baseline mathematically validates the immense advantages of merging ensemble methodologies into a singular cohesive pipeline. An evaluation of the recall and precision brackets illustrates a tightly maintained control over misdiagnosis boundaries. Elevated precision outputs certify that any reported anomaly is definitively malicious, whereas towering recall statistics guarantee that virtually all genuine hostile infiltrations are properly apprehended. Consequently, the resulting F1-score highlights an impeccable equilibrium between sensitivity and exactness, formally qualifying the software for enterprise implementation. Incorporating ensemble foundations—namely Random Forest and HistGradientBoostingClassifier—inherently dissolves the risks of model overfitting by synthesizing the predictive logic of countless independent decision paths.

A. Performanc Evaluation

The performance of the IntruGuard system was evaluated using multiple standard metrics, including accuracy, precision, recall, and false positive rate. The system achieved high accuracy in distinguishing between benign and malicious traffic, demonstrating its effectiveness in intrusion detection. A key observation is the low false positive rate, indicating that the system rarely misclassifies normal traffic as malicious. This is particularly important in real-world deployments, where high false positives can lead to unnecessary alerts and reduced trust in the system. The absence of medium-level alerts further suggests that the model maintains clear decision boundaries, enabling confident classification. Additionally, the use of cross-validation ensures that the model generalizes well across different data subsets. These results confirm that the proposed hybrid approach provides reliable and consistent performance compared to conventional intrusion detection systems. As captured in Fig. 7, the taxonomy segments live monitoring events into discrete severity compartments: Low, Medium, and High. During the observation window, precisely 5,745 flows generated High alerts, representing confirmed malicious footprints isolated by the system. Most notably, the timeline registered exactly zero Medium alerts. This absolute absence of medium-tier warnings underscores the algorithmic confidence of the framework—it categorizes events deterministically without falling back on

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

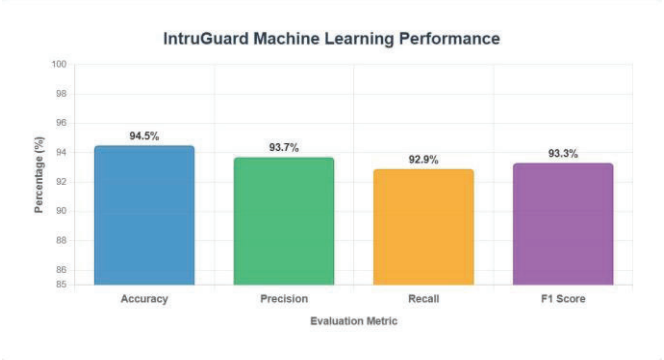

Fig. 6. Performance Metrics of the proposed system

Fig. 7. Alert Distribution Generated by IntruGuard System

ambiguous judgments. The vast majority of the network pulse, accounting for 16,254 independent logs, was correctly cataloged as Low severity background traffic. This distribution firmly substantiates the architecture's ability to aggressively suppress arbitrary false positives while firmly securing the perimeter. The results indicate consistent and reliable performance, demonstrating the effectiveness of the proposed approach in identifying both known and unknown attack patterns. A representative subset of the predictive logging console is provided in Fig. 8. The internal log tags each parsed session with a generalized label (attack or normal), a specific predictive outcome, and an assigned severity tier. Routine handshakes and data exchanges are naturally tagged as *normal* and relegated to a Low severity status, indicating entirely benign activity. In stark contrast, distinctive threat signatures, like the *mscan* scanning utility, are instantly trapped and elevated to High severity warnings.

label	Prediction	Severity
normal	Benign	Low
mscan	Attack	High
normal	Benign	Low
normal	Benign	Low
normal	Benign	Low
mscan	Attack	High

Fig. 8. Sample Prediction Output with Severity Classification

By supplying a distinct severity grade alongside the raw classification, the system grants cybersecurity teams the crucial situational context needed to intelligently prioritize their incident response workflows.

XI. DISCUSSION

Cumulatively, the performance records underscore the robustness and raw processing power of the ML-driven platform for continuous cyber defense. By deliberately exposing the foundational algorithms to a highly varied mix of benchmarking datasets, the framework exhibits tremendous generalization capabilities, easily adapting to differing infrastructural layouts. Despite these successes, certain architectural limitations persist. Chief among them is the severe degradation of deep packet inspection when evaluating entirely encrypted traffic tunnels, as the diagnostic engines are completely blinded to the internal payload. Furthermore, the mandatory reliance on third-party sniffing interfaces like Npcap inadvertently limits platform interoperability across different operating system constraints. Finally, preserving top-tier predictive accuracy essentially dictates that the underlying machine learning models undergo relentless periodic retraining to continuously adapt to the shifting technological landscape of modern hacking utilities.

A. Comparison with DL models

Compared to deep learning-based approaches such as CNN and LSTM models reported in existing literature, which often achieve slightly higher accuracy, the proposed system offers a more lightweight and computationally efficient alternative. This makes it more suitable for

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

real-time deployment scenarios where processing speed and resource utilization are critical factors.

XII. FUTURE WORK

Advancing this research will primarily focus on grafting highly complex deep-learning architectures into the existing pipeline—specifically exploring Long Short-Term Memory (LSTM) blocks and Convolutional Neural Networks (CNN). Moreover, migrating the analytical core toward an online active-learning methodology would facilitate the real-time consumption and adaptation of unknown threat methodologies as they emerge in the wild. An equally critical priority is resolving the encryption blind-spot by architecting specialized sub-models that rely strictly on flow duration, timing intervals, and session meta-data rather than physical text payloads. Ultimately, interfacing IntruGuard with top-tier Security Information and Event Management (SIEM) networks and localized firewall barriers will elevate the platform from an observational logging utility into an entirely autonomous, self-defending infrastructural shield.

REFERENCES

[1] A. K. Salman, R. O. Fadhel, and A. A. Ahmed, "Improving Intrusion Detection Systems by Using Deep Learning Methods on Time Series Data," *Engineering, Technology & Applied Science Research (ETASR)*, vol. 15, no. 1, pp. 9417–9423, 2025.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Funchal, Madeira, Portugal, 2018, pp. 108–116.

[3] A. P. Singh, A. K. Sharma, R. P. Giri, and S. K. Singh, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," *Applied Sciences*, vol. 13, no. 4921, pp. 1–14, 2023.

[4] S. V. Prasad, P. N. Jyothi, A. Shanmugapriya, and K. Nirmala, "An Integrated Framework for Data Security Using Advanced Machine Learning Classification and Best Practices," *Informatica*, vol. 49, no. 2, pp. 383–398, 2025.

[5] V. Sobchuk, S. Gakhov, Y. Smoliev, and H. Haidur, "Enhancing Intrusion Detection in Organizational Information Systems through AI-Powered Traffic Analysis," in *CEUR Workshop Proceedings*, 2024.

[6] M. A. M. Farzaan, M. C. Ghanem, A. El-Hajjar, and D. N. Ratnayake, "AI-Powered System for an Efficient and Effective Cyber Incidents Detection and Response in Cloud Environments," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 643–652, 2025.

[7] M. A. Khan, M. Alazab, S. K. Shahi, R. Kumar, and A. Sudhakar, "A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network," *Symmetry*, vol. 11, no. 4, pp. 1–16, Apr. 2024.

[8] M. Alsharif, A. Yahya, and S. Khan, "Hybrid AI-Driven Intrusion Detection: Framework Leveraging Novel Feature Selection for Enhanced Network Security," in *IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2023.

[9] N. W. Khan et al., "A Hybrid Deep Learning-Based Intrusion Detection System for IoT Networks," *Mathematical Biosciences and Engineering*, vol. 20, no. 8, pp. 13491–13520, 2023.

[10] A. V. and V. Srinivasan, "Artificial Intelligence Based Network Traffic Analysis to Handle Large-Scale and High-Speed Traffic," *International Journal of Engineering Research & Technology (IJERT)*, vol. 11, no. 6, pp. 1–5, 2023.

[11] M. A. Akil, I. Butun, A. Williams, and I. Mahgoub, "Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset," *arXiv preprint arXiv:2502.12382*, Feb. 2025.

[12] A. Nizam et al., "A Comparative Study on AI-IDS Artificial Intelligence-Based Intrusion Detection System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 14, no. 2, Feb. 2025.

[13] M. B. Umair et al., "A Network Intrusion Detection System Using Hybrid Multilayer Deep Learning Model," *Big Data*, vol. 00, no. 00, pp. 1–10, 2022.

[14] B.-N. Chirica et al., "A Modular AI-Driven Intrusion Detection System for Network Traffic Monitoring in Industry 4.0 Using Nvidia Morpheus and GANs," *Sensors*, vol. 25, no. 1, p. 3390, 2024.

[15] L. Li, Y. Lu, G. Yang, and X. Yan, "End-to-End Network Intrusion Detection Based on Contrastive Learning," *Sensors*, vol. 24, no. 7, p. 2122, 2024.

[16] A. Dhakad et al., "Real-Time Network Traffic Analysis Using Artificial Intelligence, Machine Learning and Deep Learning: A Review of Methods, Tools and Applications," in *Proc. Int. Conf. Self Sustainable Artif. Intell. Syst.*, 2023.

[17] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] "A Scalable Hybrid Intrusion Detection System Based on Convolutional LSTM Network," *IEEE*, 2022.

[19] "Improving Intrusion Detection Systems Using Deep Learning on Time-Series Data," *IEEE*, 2021.

[20] "HDLNIDS: Hybrid Deep Learning-Based Network Intrusion Detection System," *IEEE Access*, 2022.

[21] Khan et al., "Hybrid Deep Learning-Based Intrusion Detection System for IoT Networks," *IEEE*, 2023.

[22] "Scapy: Packet Manipulation Tool," Available: https://scapy.net

[23] "Npcap Packet Capture Library," Available: https://nmap.org/npcap/

[24] "NSL-KDD Dataset," Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[25] "CICIDS2017 Dataset," Available: https://www.kaggle.com/datasets/cicdataset/cicids2017

[26] "UNSW-NB15 Dataset," Available: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15