SEBHA UNIVERSITY CONFERENCE PROCEEDINGS, VOL.5 NO. 4 2026
DOI: 10.51984/MF3GBX61

وقائع مؤتمرات جامعة سبها
Sebha University Conference Proceedings

Conference Proceeding homepage: http://www.sebhau.edu.ly/journal/CAS

Towards an Intelligent Hybrid Intrusion Detection System Using Machine Learning and Large Language Models (LLM)

Mohamed Hawedi[1], Ragab Ihnissi[2]

[1]Department of Software Engineering, Regdalin High Institute of Science and Technology, Regdalin, Libya;
[2] Department of Software Engineering, Sabratha University, Sabratha, Libya

Keywords:
Intrusion Detection System
Machine Learning
Large Language Models,
Network Security
Hybrid Detection

ABSTRACT

Intrusion Detection Systems (IDSs) play a critical role in modern cybersecurity by continuously monitoring and protecting computer networks against increasingly sophisticated and evolving cyber threats. Despite significant advances in intrusion detection, conventional IDS approaches, including signature-based and machine learning (ML)-based methods, often struggle to balance detection accuracy, adaptability, and real-time performance in dynamic network environments. This paper proposes an intelligent hybrid IDS framework that integrates a lightweight machine learning detector with Large Language Model (LLM)-based contextual reasoning. The proposed framework adopts a two-stage detection strategy. In the first stage, a computationally efficient ML model rapidly classifies network traffic, allowing high-confidence flows to be processed immediately while forwarding suspicious or low-confidence flows to the second stage for further analysis. In the second stage, the LLM incorporates and analyzes contextual information to validate and refine the initial ML predictions. Finally, a confidence-guided decision mechanism combines the outputs of both stages to produce the final intrusion detection decision.A preliminary evaluation demonstrates that the proposed framework improves recall by reducing false negatives while maintaining high overall detection accuracy and computational efficiency.

نحو نظام ذكي هجين لكشف التسلل باستخدام التعلّم الآلي والنماذج اللغوية الكبيرة (LLM)

محمد هويدي[1]، رجب اهنيسي[2]

[1] قسم هندسة البرمجيات، المعهد العالي للعلوم والتكنولوجيا، رغدالين، ليبيا؛
[2]قسم هندسة البرمجيات، جامعة صبراتة، صبراتة، ليبيا

الكلمات المفتاحية:
نظام كشف التسلل
التعلم الآلي
نماذج اللغة الضخمة
أمن الشبكات
الكشف الهجين

الملخص

تلعب أنظمة كشف التسلل (IDS) دورًا محوريًا في الأمن السيبراني الحديث، إذ تراقب وتحمي شبكات الحاسوب باستمرار من التهديدات السيبرانية المتطورة والمتزايدة التعقيد. ورغم التقدم الكبير في مجال كشف التسلل، فإن أساليب أنظمة كشف التسلل التقليدية، بما فيها الأساليب القائمة على التوقيعات والأساليب القائمة على التعلم الآلي، غالبًا ما تواجه صعوبة في تحقيق التوازن بين دقة الكشف، وقابلية التكيف، والأداء الفوري في بيئات الشبكات الديناميكية. تقترح هذه الورقة البحثية إطار عمل هجينًا ذكيًا لأنظمة كشف التسلل، يدمج كاشفًا خفيف الوزن قائمًا على التعلم الآلي مع استدلال سياقي قائم على نموذج اللغة الكبير (LLM). يعتمد إطار العمل المقترح استراتيجية كشف ثنائية المراحل. في المرحلة الأولى، يصنف نموذج التعلم الآلي الفعال حسابيًا حركة مرور الشبكة بسرعة، مما يسمح بمعالجة التدفقات عالية الموثوقية فورًا، بينما يُحيل التدفقات المشبوهة أو منخفضة الموثوقية إلى المرحلة الثانية لمزيد من التحليل. في المرحلة الثانية، يدمج نموذج اللغة الكبير المعلومات السياقية ويحللها للتحقق من صحة تنبؤات التعلم الآلي الأولية وتحسينها. وأخيرًا، تقوم آلية اتخاذ القرار الموجهة بالثقة بدمج مخرجات المرحلتين لإنتاج قرار الكشف النهائي عن الاختراق. ويُظهر تقييم أولي أن الإطار المقترح يحسن الاستدعاء عن طريق تقليل النتائج السلبية الخاطئة مع الحفاظ على دقة الكشف الإجمالية العالية والكفاءة الحسابية.

*Corresponding author:
E-mail addresses: howedi.edu@gmail.com ,(R. Ihnissi) ragab.ihnissi@sabu.edu.ly
Article History : Received 30 May 2026 - Received in revised form 30 September 2026 - Accepted 11 November 2026

1 Introduction

The rapid growth of cloud computing and large-scale distributed systems has significantly increased the complexity, scale, and heterogeneity of modern network infrastructures. While these environments enable flexible resource provisioning and cost-efficient computing, they also expand the attack surface and expose systems to increasingly sophisticated cyber threats, including malware, ransomware, denial-of-service (DoS) attacks, web application attacks, insider threats, and zero-day exploits. Consequently, Intrusion Detection Systems (IDSs) have become essential security mechanisms for continuously monitoring network traffic and detecting malicious activities that threaten the confidentiality, integrity, and availability of modern information systems. Traditional IDSs are generally classified into signature-based and anomaly-based approaches. Signature-based IDSs identify attacks by matching predefined signatures and provide high detection accuracy for known threats. However, they are inherently limited in detecting zero-day attacks and previously unseen attack patterns. In contrast, anomaly-based IDSs establish models of normal network behavior and identify deviations that may indicate malicious activity. Although these approaches improve the detection of unknown threats, they often suffer from high false-positive rates and reduced reliability in dynamic network environments [1].

Cloud computing introduces additional challenges for intrusion detection due to its distributed architecture, multitenancy, scalability requirements, and heterogeneous security policies. Conventional IDS solutions are typically designed as generic security mechanisms and often fail to satisfy the diverse security requirements of individual cloud tenants. To address these limitations, Hawedi *et al.* proposed a Multi-Tenant Intrusion Detection System (MTIDS), which integrates signature-based and anomaly-based detection while considering tenant-specific security requirements and resource efficiency [2]. They subsequently extended this concept by introducing Intrusion Detection System as a Service (IDSaaS), enabling flexible and lightweight deployment of intrusion detection capabilities tailored to cloud environments [3]. These studies highlight the importance of scalable and adaptive IDS architectures for modern cloud infrastructures.

Recent advances in machine learning (ML) have significantly improved intrusion detection by enabling data-driven analysis of network traffic. Supervised and unsupervised learning techniques, including Decision Trees, Support Vector Machines, ensemble learning, and deep neural networks, have demonstrated high detection accuracy and scalability compared with conventional rule-based approaches [4–6]. Nevertheless, most ML-based IDSs rely on static training datasets and handcrafted feature representations, limiting their ability to adapt to evolving attack strategies, concept drift, and continuously changing network behaviors.

More recently, Large Language Models (LLMs) have emerged as powerful reasoning models capable of understanding contextual relationships and generating semantically rich interpretations from heterogeneous information sources. Although originally developed for natural language processing, LLMs have demonstrated considerable potential in cybersecurity applications, including threat analysis, security log interpretation, vulnerability assessment, and incident investigation [7]. Their ability to incorporate contextual knowledge and provide human-interpretable reasoning offers new opportunities for improving intrusion detection. However, directly applying LLMs to real-time intrusion detection remains challenging because of their high computational cost, inference latency, and scalability constraints.

Despite these advances, existing IDS solutions still face important limitations. ML-based approaches provide fast and efficient detection but often lack the contextual understanding needed to identify complex attacks. In contrast, LLM-based approaches offer powerful reasoning capabilities but require substantial computational resources, making real-time deployment challenging. Consequently, developing an IDS that achieves high detection accuracy, contextual understanding, computational efficiency, and scalability remains an open research challenge.

Motivated by these challenges, this paper proposes an intelligent hybrid intrusion detection framework that combines the efficiency of machine learning with the contextual reasoning capabilities of Large Language Models. The proposed framework employs a two-stage detection strategy. In the first stage, a lightweight ML model rapidly classifies network traffic, allowing high-confidence traffic to be processed immediately while suspicious or low-confidence flows are selected for further analysis. In the second stage, the selected flows are enriched with contextual information and analyzed by an LLM to validate and refine the initial ML predictions. Finally, a confidence-guided decision mechanism integrates the outputs of the ML detector and the LLM reasoning module to generate the final intrusion detection decision. By combining fast statistical learning with contextual reasoning, the proposed framework improves detection reliability while maintaining computational efficiency, scalability, and suitability for real-time deployment in enterprise and cloud environments.

The main contributions of this paper are summarized as follows:

- A two-stage hybrid intrusion detection framework that combines lightweight ML-based traffic screening with LLM-driven contextual reasoning.
- A selective reasoning mechanism that forwards only suspicious or low-confidence traffic to the LLM, reducing computational overhead while supporting real-time intrusion detection.
- A confidence-guided decision mechanism that integrates ML predictions with LLM-based reasoning to improve the reliability of intrusion detection decisions.
- A comprehensive experimental evaluation demonstrating improved detection performance over standalone ML-based approaches while maintaining high detection accuracy and computational efficiency.

2 Background

This section presents the background of intrusion detection systems and motivates the hybrid ML–LLM approach adopted in this work.

2.1 Intrusion Detection Systems

Intrusion Detection Systems (IDSs) are security mechanisms designed to monitor network or system activities to identify malicious behavior that may compromise confidentiality, integrity, or availability. IDSs are commonly used to detect cyber attacks such as denial-of-service attacks, probing activities, and unauthorized access attempts. Based on their detection strategy, IDSs are generally classified into signature-based and anomaly-based approaches.

- Signature-Based Intrusion Detection:
  Signature-based intrusion detection systems identify malicious activity by comparing observed network traffic or system events against a database of predefined attack signatures. These systems are highly effective in detecting known attacks and typically produce low false alarm rates, which explains their widespread adoption in operational security environments. However, their reliance on predefined signatures limits their ability to detect unknown or zero-day attacks, as new threats can only be identified after corresponding signatures are created and updated [1,3].
- Anomaly-Based Intrusion Detection:
  Anomaly-based IDSs aim to detect malicious activities by modeling normal network or system behavior and identifying deviations from this baseline. Unlike signature-based systems, anomaly-based approaches are capable of detecting previously unseen or emerging attacks. However, they are often affected by high false alarm rates, particularly in dynamic environments where normal behavior evolves over time [1]. In cloud and multi-tenant settings, anomaly-based detection is especially challenging due to traffic heterogeneity, workload variability, and concept drift.

2.2 Machine Learning-Based Intrusion Detection

To enhance anomaly detection capabilities, machine learning (ML) techniques have been widely incorporated into IDS design. ML-based IDSs employ supervised, unsupervised, or ensemble learning algorithms to automatically learn patterns from network traffic data. Techniques such as decision trees, support vector machines, ensemble models, and deep learning architectures have demonstrated improved detection accuracy and scalability compared to traditional rule-based systems [4–6]. Despite these improvements, most ML-based IDSs rely on static training datasets and handcrafted feature

representations, which limits their adaptability to evolving attack behaviors and rapidly changing cloud traffic patterns.

2.3 Large Language Model-Based Intrusion Detection

More recently, Large Language Models (LLMs) have been investigated as advanced reasoning components in intrusion detection systems. LLMs are capable of capturing semantic relationships, contextual dependencies, and higher-level abstractions across heterogeneous data sources, including network flows, logs, and security alerts. Prior studies have shown that LLMs can support contextual reasoning, improve explainability, and enhance adaptability by learning representations that generalize beyond fixed feature sets [7].

However, due to their substantial computational overhead and inference latency, LLMs are not well suited for standalone real-time intrusion detection. Consequently, recent research has increasingly focused on hybrid architectures that combine lightweight ML-based detectors for fast traffic filtering with LLM-driven analysis for deeper contextual interpretation [8].

3 Related Work

Intrusion Detection Systems (IDSs) have been extensively studied as a core security mechanism for protecting networked, cloud-based, and enterprise environments. Early IDS solutions predominantly relied on signature-based detection, which remains effective for identifying known attacks but is inherently limited in detecting previously unseen or evolving threats. To address this limitation, anomaly-based IDSs were introduced to detect deviations from normal traffic behavior. However, these systems often suffer from high false positive rates and limited robustness in highly dynamic environments [1].

To improve detection accuracy and adaptability, machine learning (ML) techniques have been widely adopted in IDS design. Alzahrani and Alenazi proposed ML-IDSDN, a machine learning-based IDS for software-defined networking environments, demonstrating that tree-based classifiers can achieve high detection accuracy using flow-level features [4]. Louk and Tama introduced a dual ensemble approach combining bagging and gradient boosting decision trees to enhance anomaly detection across multiple benchmark datasets. While effective, these approaches rely on static feature representations and conventional ML pipelines, limiting their ability to adapt to evolving attack semantics. More recent deep learning (DL)-based IDSs employ optimized neural architectures and feature fusion strategies to improve generalization [5, 6]. Despite these advances, DL-based IDSs remain dependent on labeled datasets and offline training, which constrains responsiveness to emerging threats.

In cloud and multi-tenant environments, prior studies have emphasized the importance of tenant-aware security mechanisms. Hawedi *et al.* proposed a multitenant intrusion detection system (MTIDS) that integrates signature-based and anomaly-based detection while considering tenant-specific requirements and cost efficiency in public cloud infrastructures [2]. Subsequently, an IDS-asa-Service (IDSaaS) model was introduced to enable cloud tenants to deploy flexible and lightweight IDS solutions tailored to their operational needs [3]. These works highlight the need for scalable and adaptive IDS architectures in cloud settings.

More recently, Large Language Models (LLMs) have been investigated for intrusion detection due to their contextual reasoning and semantic modeling capabilities. Yang *et al.* surveyed the integration of LLMs into Network Intrusion Detection Systems (NIDSs), introducing the concept of Cognitive NIDS, where LLMs act as processors, detectors, explainers, and controllers [7]. Adjewa *et al.* proposed a continuous intrusion detection framework combining transformer-based models for malicious traffic detection with Gaussian Mixture Models for identifying unknown attacks [8]. These studies demonstrate the potential of LLMs to enhance adaptability and interpretability in IDSs.

Several recent works have evaluated LLMs as standalone intrusion detectors, reporting encouraging performance but also revealing important limitations. Hassanin *et al.* [9] evaluated a pre-trained LLM for cyber threat detection using datasets such as UNSW-NB15 and TON-IoT, reporting near-perfect accuracy. However, their approach relied heavily on bias-inducing features such as IP addresses and timestamps, increasing the risk of overfitting. Similarly, Gutierrez-Galeano´ *et al.* [10] fine-tuned a T5-based model for intrusion detection and achieved strong performance across multiple CIC IDS datasets, but still relied on destination ports, raising concerns about generalization. Zhang *et al.* [11] integrated GPT-2 with Suricata and the ELK stack for real-time threat detection, yet their fine-tuning prompts also included timestamps and ports.

More recent studies explored fine-tuned or in-context LLMs across multiple datasets. Zhang *et al.* [12] demonstrated that GPT-4 could achieve F1-scores exceeding 95% using a small number of in-context examples. Nevertheless, the feature selection process relied on LLM-driven importance ranking, again introducing bias and limiting reproducibility. In addition, most of these studies did not include comparisons with strong ML or DL baselines, restricting the robustness of their evaluations.

Comparative studies between LLMs and traditional ML models remain limited and sometimes inconclusive. Guastalla *et al.* [13] reported superior performance of GPTbased models compared to a simple MLP baseline, though the weakness of the baseline reduced the comparative value. Conversely, Houssel *et al.* [14] observed weak performance across both LLMs and ML classifiers, suggesting dataset or modeling challenges. Nwafor *et al.* [15] demonstrated that well-configured ML models could outperform LLMbased approaches, even when bias-inducing features were removed, though not all experimental results were reported, limiting transparency.

Mehavilla *et al.* [16] conducted a comprehensive benchmarking study evaluating LLMs for flow-based intrusion detection in comparison with traditional ML and DL baselines. Their work emphasized fair experimental comparison using identical datasets, features, and evaluation protocols. While pretrained LLMs achieved competitive performance, the study did not propose an integrated detection architecture or address deployment-oriented constraints such as inference latency and scalability.

Overall, existing studies predominantly treat LLMs as monolithic detectors and rarely address practical deployment considerations, including throughput, resource consumption, and integration with existing IDS pipelines. Although LLM-based approaches show promise, their direct deployment in real-time IDSs is constrained by computational cost and scalability limitations. These gaps motivate the present work, which proposes a hybrid ML–LLM intrusion detection framework that combines efficient MLbased filtering with selective LLM-based contextual reasoning, enabling scalable, adaptive, and deployment-ready intrusion detection for modern cloud and enterprise environments.

The remainder of this paper is organized as follows. Section II reviews the related work on machine learning– based intrusion detection and LLM-assisted security analysis. Section III describes the proposed hybrid intrusion detection architecture, including the lightweight ML-based detector (Stage 1) and the LLM-driven contextual reasoning component (Stage 2). Section IV presents the experimental setup, datasets, and evaluation metrics. Section V reports and discusses the experimental results for both stages. Finally, Section VI concludes the paper and outlines directions for future work.

4 Proposed System Architecture

This section presents the proposed hybrid intrusion detection architecture, which integrates a lightweight Machine Learning (ML)-based detector with Large Language Model (LLM)-based contextual reasoning. The framework combines the efficiency of lightweight ML with the semantic reasoning capabilities of LLMs to improve intrusion detection accuracy while supporting scalable real-time deployment. The overall architecture is illustrated in Fig. 1.

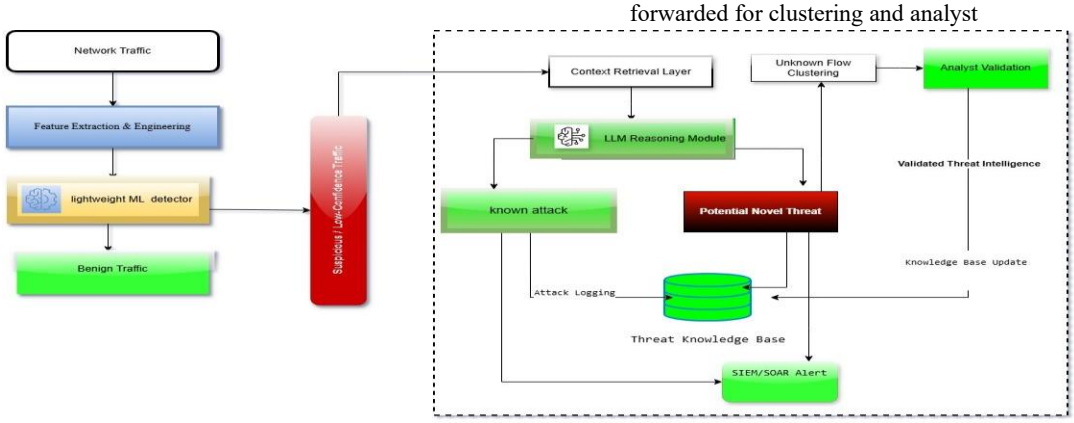


Figure 1: Proposed hybrid intrusion detection architecture combining a lightweight ML-based detector with LLM-driven validation. Validated threat information is subsequently incorporated into the Threat Knowledge Base to support future detection.

4.1 Overview of the Proposed Framework

The proposed framework employs a two-stage detection strategy. In the first stage, network traffic is processed by a lightweight ML detector. High-confidence predictions are accepted directly, while suspicious or low-confidence traffic is forwarded to the LLM stage for further analysis.

In the second stage, the LLM combines engineered network features with contextual information retrieved from the Threat Knowledge Base to refine the initial ML prediction. Known attacks generate SIEM/SOAR alerts, while potential novel threats are forwarded for clustering and analyst

4.2 Network Data Source and Feature Extraction

Network traffic serves as the primary input to the proposed framework. The Feature Extraction and Engineering layer transforms network observations into structured numerical features suitable for ML-based detection. Representative features include flow duration, packet statistics, protocol behavior, traffic volume, connection patterns, and statistical flow attributes. These features provide an efficient representation of network behavior for the first-stage detector.

4.2 Network Data Source and Feature Extraction
Network traffic serves as the primary input to the proposed framework. The Feature Extraction and Engineering layer transforms network observations
into structured numerical features suitable for ML-based detection. Representative features include flow duration, packet statistics, protocol behaviour, traffic volume, connection patterns, and statistical flow attributes. These features provide an efficient representation of network behavior for the first-stage detector.

4.3 Stage 1: Lightweight ML-Based Detection

The first stage employs a lightweight ML detector to rapidly classify network traffic as benign or suspicious. High-confidence benign traffic is accepted without additional processing, while suspicious or low-confidence traffic is forwarded to Stage 2 for deeper analysis. This selective routing reduces unnecessary LLM inference and supports efficient network traffic processing.

4.4 Stage 2: LLM-Based Contextual Reasoning

Stage 2 analyzes suspicious or low-confidence traffic using a Large Language Model (LLM). The LLM combines engineered network features with contextual information retrieved from the Threat Knowledge Base to refine the initial ML prediction and support the analysis of ambiguous or potentially previously unseen threat behaviors.

4.4.1 Context Retrieval Layer

The Context Retrieval Layer retrieves relevant information from the Threat Knowledge Base, including known attack patterns, historical alerts, protocol behavior, threat intelligence, and analyst-validated knowledge. This information complements the network features provided to the LLM.

4.4.2 LLM Reasoning

The LLM analyzes the network features and retrieved context to validate and refine the prediction generated during Stage 1. This reasoning provides additional contextual analysis for traffic that cannot be classified with sufficient confidence by the ML detector.

4.4.3 Known Attack Identification

When the observed behavior matches existing attack knowledge, the LLM classifies the traffic as a known attack. The detection is recorded in the Threat Knowledge Base, and a SIEM/SOAR alert is generated to support security monitoring and incident response.

4.4.4 Potential Novel Threat Detection

If the observed behavior does not sufficiently match existing attack knowledge, it is identified as a potential novel threat and forwarded to the Unknown Flow Clustering module for further investigation.

4.5 Novel Threat Discovery and Knowledge Enrichment

Potential novel threats undergo clustering and analyst validation before validated threat information is incorporated into the Threat Knowledge Base.

4.5.1 Unknown Flow Clustering

The Unknown Flow Clustering module groups similar suspicious flows based on their behavioral characteristics, facilitating the identification and analysis of potentially previously unseen attack patterns.

4.5.2 Analyst Validation

Security analysts review the generated clusters to determine whether they represent genuine threats or benign activities. This validation prevents unverified behaviors from being incorporated into the framework.

4.5.3 Threat Knowledge Base Update

Validated threat information is incorporated into the Threat Knowledge Base, which maintains known attack information, historical detections, protocol knowledge, analyst feedback, and newly identified threat intelligence for future LLM reasoning.

4.5.4 SIEM/SOAR Alert Generation

Confirmed attacks generate SIEM/SOAR alerts to support security monitoring, incident response, forensic investigation, and other Security Operations Center (SOC) activities.

Algorithm 1 Hybrid ML–LLM Intrusion Detection Pipeline

Require: Network traffic flows F
Ensure: Detection results and generated SIEM/SOAR alerts
1: for each network flow $f \in F$ do
2: Extract and engineer network features
3: Apply the Lightweight ML Detector
4: Obtain prediction L_{ML} and confidence C_{ML}
5: if L_{ML} is benign and C_{ML} is high then
6: Accept as benign traffic
7: else

```
        Retrieve contextual information from the Threat Knowledge Base
        Invoke the LLM Reasoning Module
        Analyze retrieved context together with engineered network features
        Obtain refined prediction L_LLM
        if L_LLM corresponds to a known attack then
            Log attack information
            Generate a SIEM/SOAR alert
        else
            Classify as a potential novel threat
            Forward to Unknown Flow Clustering
            Perform Analyst Validation
            Update the Threat Knowledge Base with validated threat intelligence
        end if
    end if
end for
return Detection results and generated SIEM/SOAR alerts
```

4.6 Algorithmic Workflow Representation

Algorithm 1 summarizes the operational workflow of the proposed hybrid intrusion detection framework. The framework combines lightweight ML-based traffic screening with LLM-based contextual reasoning. Only suspicious or lowconfidence traffic is forwarded to the second stage, significantly reducing the computational cost associated with LLM inference while maintaining high detection performance.

Algorithm 1 illustrates the operational workflow of the proposed framework. A lightweight ML detector first performs rapid traffic screening and immediately accepts benign traffic with high confidence. Only suspicious or lowconfidence traffic is forwarded to the LLM-based reasoning stage. The Context Retrieval Layer retrieves relevant threat knowledge to support semantic reasoning by the LLM. Based on the retrieved context, the LLM determines whether the traffic corresponds to a known attack or a potential novel threat. Known attacks generate SIEM/SOAR alerts, whereas potential novel threats are grouped by the Unknown Flow Clustering module and subsequently validated by security analysts. The validated threat intelligence is incorporated into the Threat Knowledge Base, enabling continuous knowledge enrichment and improving the framework's ability to detect future attacks while preserving real-time operational efficiency.

The proposed architecture integrates lightweight ML detection, LLM-based contextual reasoning, threat knowledge retrieval, unknown flow clustering, analyst-guided validation, and continuous knowledge enrichment within a unified intrusion detection framework. By selectively invoking the LLM only for suspicious or low-confidence traffic, the framework achieves an effective balance between detection accuracy, computational efficiency, and adaptability to emerging cyber threats.

5 Experimental Results

This section evaluates the performance of the machinelearning classifiers and the proposed hybrid ML–LLM intrusion detection framework. The evaluation first compares the Decision Tree (DT) and Support Vector Machine (SVM) classifiers as lightweight network traffic detectors. The SVM is then integrated with the LLM stage to investigate the effect of confidence-guided contextual verification. In addition to classification performance, the experiments examine the proportion of traffic routed to the LLM and the computational overhead introduced by LLM inference.

5.1 Experimental Setup

The proposed Hybrid IDS was evaluated using the CICIDS2017 Web Attacks dataset as a binary classification problem. A fixed 80/20 split was used, resulting in 136,292 training flows and 34,074 test flows with 78 network-flow features. The experiments were implemented in Python using Pandas and NumPy for data processing and scikit-learn for the Decision Tree and linear SVM classifiers.

For selective LLM verification, Llama 3.1 was deployed locally through Ollama. The SVM decision margin was used to identify uncertain predictions for additional LLM analysis, allowing the LLM to focus on selected network flows rather than processing the complete test set. This setup was used to evaluate the detection performance and computational characteristics of the proposed Hybrid IDS.

5.2 Baseline ML Detection Performance

The Decision Tree and linear SVM classifiers were evaluated using the same CICIDS2017 Web Attacks test set containing 34,074 network flows. The evaluation considered Table 1: Performance Comparison of the Baseline ML Classifiers

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC	PR-AUC
Decision Tree	99.9795	99.3103	99.0826	99.1963	0.995368	0.984110
Linear SVM	99.4571	73.9962	88.7615	80.7091	0.997731	0.800338

accuracy, precision, recall, F1-score, ROC-AUC, PR-AUC, and the corresponding confusion matrices.

As shown in Table 1, the Decision Tree achieves the strongest overall classification performance, with higher precision, recall, F1-score, and PR-AUC than the linear

SVM. The Decision Tree produces only three false positives and four false negatives, whereas the SVM produces 136 false positives and 49 false negatives.

Although the Decision Tree performs better at the selected classification threshold, the SVM achieves a slightly higher ROC-AUC of 0.997731, compared with 0.995368 for the Decision Tree. This indicates strong ranking capability of the SVM decision scores despite its weaker threshold-based classification performance.

The linear SVM is used in the hybrid evaluation because its decision function provides a continuous margin that can be used to identify network flows close to the classification boundary. These low-margin flows are treated as uncertain and selectively forwarded to the LLM for additional analysis.

5.3 SVM Feature Analysis

To examine the network characteristics influencing the SVM predictions, the absolute coefficients of the linear SVM were analyzed. The 20 features with the largest absolute coefficients are presented in Fig. 2.

As shown in Fig. 2, several inter-arrival-time features have relatively large SVM coefficients, including *Fwd IAT Min*, *Flow IAT Max*, *Fwd IAT Std*, *Flow IAT Min*, and *Fwd IAT Max*.

Other influential features include backward inter-arrivaltime measurements, packet-length statistics, TCP flag information, flow-rate characteristics, and idle-time measurements. These coefficients provide insight into the features that contribute most strongly to the linear SVM decision function. However, a large absolute coefficient indicates that a feature has a stronger influence on the SVM decision boundary; it does not necessarily mean that the feature itself causes malicious network activity.

5.4 Hybrid ML–LLM Detection Performance

The baseline SVM was subsequently integrated with the LLM reasoning stage. Rather than forwarding every network flow to the LLM, the proposed framework applies confidence-guided routing based on the SVM decision margin. Samples located within the predefined uncertainty region are selected for contextual LLM verification, while high-confidence samples retain their original SVM predictions.

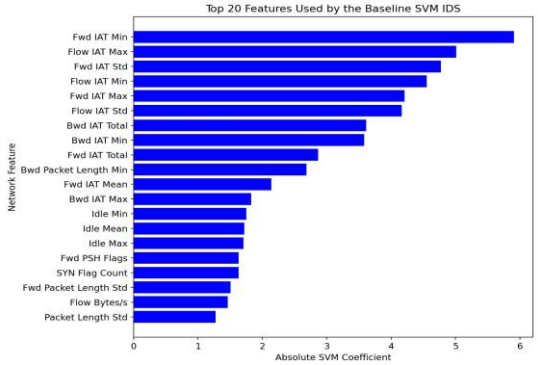


Figure 2: Top 20 network features according to the absolute coefficients of the baseline linear SVM.

Table 2: Performance Comparison of the Baseline SVM IDS and Hybrid IDS

Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	FN	FP
Baseline SVM IDS	99.4571	73.9962	88.7615	80.7091	49	136
Hybrid IDS	99.4629	74.3738	88.5321	80.8377	50	133

This design is intended to preserve the computational efficiency of lightweight ML classification while using LLM reasoning only when additional analysis may be beneficial.

Table 2 compares the classification performance of the baseline SVM IDS and the proposed Hybrid IDS.

The results show that the Hybrid IDS maintains essentially the same overall classification accuracy as the baseline SVM. Accuracy increases slightly from 99.4571% to 99.4629%. Precision increases from 73.9962% to 74.3738%, while the F1-score increases from 80.7091% to 80.8377%. In contrast, recall decreases marginally from 88.7615% to 88.5321%. Therefore, the LLM stage does not uniformly improve all classification metrics. Instead, it modifies a small number of uncertain SVM decisions, resulting in slightly higher precision and F1-score while maintaining approximately the same overall accuracy. Fig. 3 presents a graphical comparison of the principal detection metrics.

The similarity between the two systems is expected because only a very small proportion of the complete test set is selected for LLM analysis. Consequently, most final classifications remain identical to those produced by the baseline
SVM.

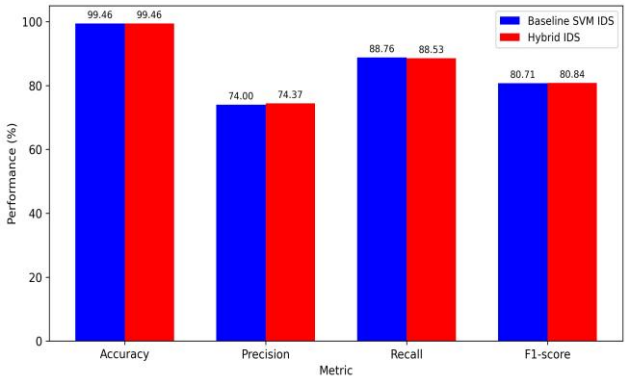


Figure 3: Performance comparison between the baseline SVM IDS and the proposed Hybrid IDS.

5.5 Confusion Matrix Analysis

The effect of selective LLM verification can be examined more directly through the confusion matrices shown in Fig. 4.

For the baseline SVM, 33,502 benign flows are correctly classified as benign and 387 attack flows are correctly identified. The classifier produces 136 false positives and 49 false negatives. Following selective LLM verification, the Hybrid IDS correctly classifies 33,505 benign flows and 386 attack flows. The number of false positives decreases from 136 to 133, while false negatives increase from 49 to 50. Therefore, the Hybrid IDS eliminates three false alarms but introduces one additional missed attack. This explains the slight improvement in precision and F1-score together with the small reduction in recall. These results characterize the LLM primarily as a selective contextual verification component rather than as a replacement for the first-stage machine-learning classifier.

5.6 Margin-Guided LLM Routing

A central objective of the proposed architecture is to avoid applying computationally expensive LLM inference to every observed network flow. To achieve this, margin-guided routing is used to identify uncertain SVM predictions that require additional verification. Specifically, flows with an absolute SVM decision margin below 0.20 are considered uncertain and selected as candidates for LLM analysis.

Table 3 summarizes the distribution of network flows between the SVM processing path and the LLM verification path.

Table 3: Margin-Guided Routing of Network Flows

Method	Count	Percentage
SVM only	34,033	99.8797
LLM candidates	41	0.1203

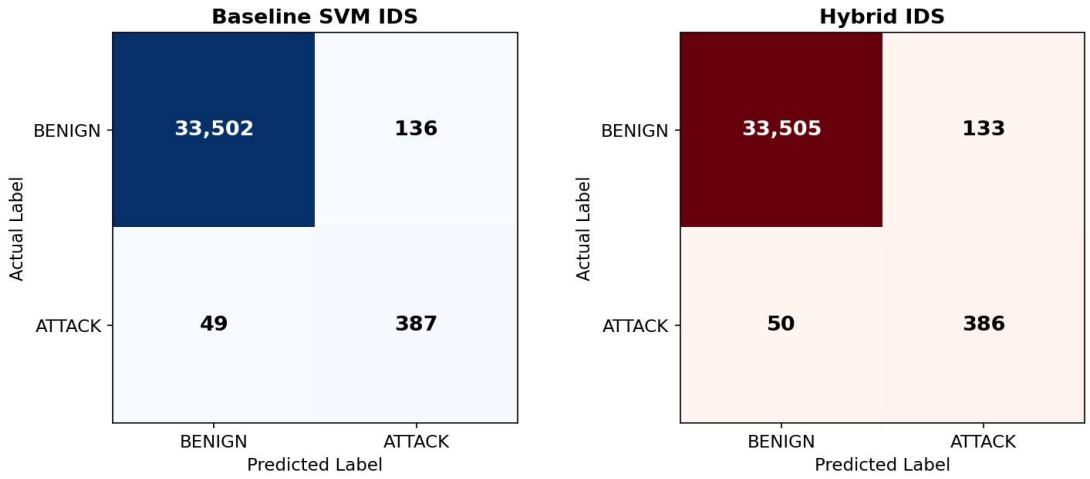


Figure 4: Confusion matrices of the baseline SVM IDS and the proposed Hybrid IDS.

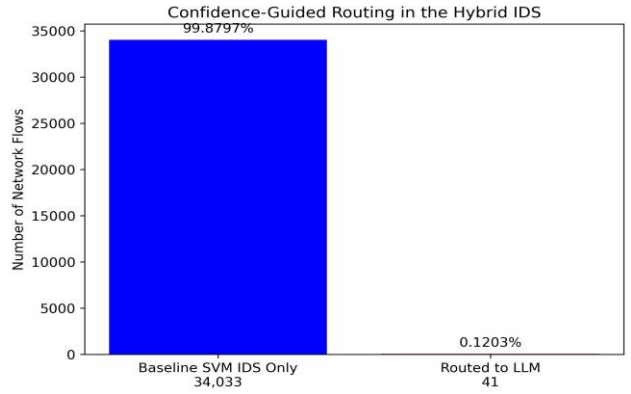


Figure 5: Margin-guided routing of network flows in the proposed Hybrid IDS.

As illustrated in Fig. 5, only 41 of the 34,074 test flows are identified as uncertain using the margin threshold of 0.20. These flows represent only 0.1203% of the test set, while the remaining 34,033 flows (99.8797%) are handled directly by the SVM without requiring LLM inference.

Due to the computational cost of local LLM inference, the prototype evaluation was limited to a 10-minute execution window. Within this period, Llama 3.1 successfully processed 18 of the 41 candidate flows. Therefore, the 41 flows reported in Table 3 represent flows selected as candidates for LLM verification, whereas 18 were actually evaluated by the LLM during the time-limited experiment.

The routing results demonstrate the computational selectivity of the proposed architecture. Rather than applying LLM inference to every network flow, the lightweight SVM performs the initial high-throughput classification and directs only uncertain predictions to the LLM for additional verification. Consequently, expensive LLM inference is restricted to a very small fraction of the observed traffic.

This separation between efficient ML-based screening and selective LLM verification is a key architectural characteristic of the proposed Hybrid IDS. It enables the LLM to complement the first-stage classifier by focusing on uncertain predictions while avoiding unnecessary LLM inference for most network flows.

5.7 LLM Inference Latency

In addition to detection performance, the computational overhead of the LLM stage was evaluated by measuring the inference latency for network flows selected by the confidence-guided routing mechanism. As shown in Fig. 6, the average inference latency was 31.94 seconds per flow, while the median latency was 29.66 seconds. The relatively close average and median values indicate that LLM inference generally required tens of seconds per flow under the evaluated local implementation.

These results highlight the importance of selective LLM routing in the proposed framework. Since LLM inference is substantially more computationally demanding than lightweight ML classification, applying the LLM to all network flows would introduce considerable processing overhead. The confidence-guided routing mechanism therefore limits LLM analysis to uncertain flows, reducing unnecessary LLM invocations. Nevertheless, the measured inference latency remains an important constraint for real-time deployment.

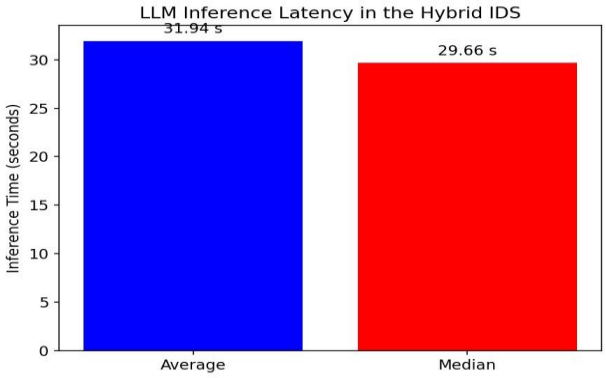


Figure 6: Average and median LLM inference latency in the proposed Hybrid IDS.

5.8 Overall Discussion

The experimental results demonstrate the feasibility of the proposed two-stage ML–LLM intrusion detection approach. Although the Decision Tree achieves the strongest standalone classification performance, the linear SVM provides a suitable decision-margin representation for confidence-guided routing and is therefore used in the Hybrid IDS.

Compared with the baseline SVM, the Hybrid IDS maintains approximately the same overall accuracy while slightly improving precision and F1-score. False positives decrease from 136 to 133, whereas false negatives increase from 49 to 50, resulting in a slight reduction in recall. These results indicate a modest precision–recall trade-off rather than a uniform improvement in detection performance.

The main benefit of the proposed hybrid approach is its selective use of LLM reasoning. Only 41 of the 34,074 test flows (0.1203%) are routed to the LLM, while the lightweight ML stage handles the remaining 99.8797%. This substantially limits the number of computationally expensive LLM invocations while introducing contextual reasoning for uncertain network traffic.

However, the measured LLM inference latency, with an average of 31.94 seconds and a median of 29.66 seconds per flow, remains an important constraint for real-time deployment. Overall, the results support the feasibility of confidence-guided ML–LLM integration while highlighting LLM inference efficiency as an important area for future optimization.

6 Conclusion and Future Work

This paper proposed a hybrid intrusion detection framework that combines lightweight machine learning with selective Large Language Model (LLM) reasoning. The experimental results show that the Decision Tree provides strong standalone classification performance, while the SVM enables the hybrid configuration through its decision margin, which is used to identify uncertain predictions for additional LLM verification. Overall, the Hybrid IDS maintains detection performance comparable to the baseline SVM while reducing false positives, although a slight reduction in recall is observed. These findings indicate a modest precision–recall trade-off rather than a consistent improvement across all detection metrics. A key advantage of the proposed architecture is the selective use of LLM reasoning. Instead of applying computationally expensive LLM inference to every network flow, the framework directs only uncertain SVM predictions to the LLM for additional verification. This substantially limits the LLM workload while preserving the efficiency of the first-stage ML detector. However, the experimental evaluation also highlights the computational cost of local LLM inference and its current limitations for real-time processing. Future work will focus on reducing LLM inference latency, optimizing the routing threshold, extending the evaluation to multiclass intrusion detection, and validating the framework using additional datasets and real-world network environments.

References

[1] S. Einy, C. Oz, and Y. D. Navaei, “The anomaly-and signature-based ids for network security using hybrid inference systems,” *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 6639714, 2021.

[2] M. Hawedi, C. Talhi, and H. Boucheneb, “Multitenant intrusion detection system for public cloud (mtids),” *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5199–5230, 2018.

[3] ——, “Security as a service for public cloud tenants (saas),” *Procedia computer science*, vol. 130, pp. 1025–1030, 2018.

[4] A. Alzahrani, S. Alqahtani, and A. Alshahrani, “Mlidsdn: Machine learning based intrusion detection system for software defined networks,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 9, 2022.

[5] A. Ayantayo, A. Kaur, A. Kour, X. Schmoor, F. Shah, I. Vickers, P. Kearney, and M. M. Abdelsamea, “Network intrusion detection using feature fusion with deep learning,” *Journal of Big Data*, vol. 10, no. 1, p. 167, 2023.

[6] N. M. Eldakhly, "Optimized intrusion detection with deep learning classification models," *Neural Computing and Applications*, vol. 37, no. 25, pp. 21091– 21119, 2025.

[7] S. Yang, X. Zheng, X. Zhang, J. Xu, J. Li, D. Xie, W. Long, and E. C. H. Ngai, "Large language models for network intrusion detection systems: Foundations, implementations, and future directions," *arXiv preprint arXiv:2507.04752*, 2025.

[8] F. Adjewa, M. Esseghir, L. Merghem-Boulahia, and C. Kacfah, "Llm-based continuous intrusion detection framework for next-gen networks," in *2025 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2025, pp. 1198–1203.

[9] M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, and D. Sharma, "Pllm-cs: Pre-trained large language model (llm) for cyber threat detection in satellite networks," *Ad Hoc Networks*, vol. 166, p. 103645, 2025.

[10] L. Gutierrez-Galeano,´ J.-J. Dom´ınguez-Jimenez,´ J. Schafer, and I. Medina-Bulo, "Llm-based cyberat-¨ tack detection using network flow statistics," *Applied Sciences*, vol. 15, no. 12, p. 6529, 2025.

[11] X. Zhang, T. Chen, J. Wu, and Q. Yu, "Intelligent network threat detection engine based on open source gpt-2 model," in *2023 International Conference on Computer Science and Automation Technology (CSAT)*. IEEE, 2023, pp. 392–397.

[12] H. Zhang, A. B. Sediq, A. Afana, and M. ErolKantarci, "Large language models in wireless application design: In-context learning-enhanced automatic network intrusion detection," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 2479–2484.

[13] M. Guastalla, Y. Li, A. Hekmati, and B. Krishnamachari, "Application of large language models to ddos attack detection," in *International Conference on Security and Privacy in Cyber-Physical Systems and Smart Vehicles*. Springer, 2023, pp. 83–99.

[14] P. R. Houssel, P. Singh, S. Layeghy, and M. Portmann, "Towards explainable network intrusion detection using large language models," in *2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*. IEEE, 2024, pp. 67–72.

[15] E. Nwafor, U. Baskota, M. S. Parwez, J. Blackstone, and H. Olufowobi, "Evaluating large language models for enhanced intrusion detection in internet of things networks," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 3358– 3363.

[16] L. Mehavilla, M. Rodr´ıguez, J. Garc´ıa, and A. Ale-´ sanco, "Evaluating large language models effectiveness for flow-based intrusion detection: a comparative study with ml and dl baselines," *Artificial Intelligence Review*, vol. 59, no. 2, p. 50, 2026.