A Hybrid Anomaly–Rule–Pattern Detection Framework for Streaming-Based Persistent Intrusion Detection

Lamyaa Ghaleb Aldawood[1], Zainab Mohammed Jiwar[1], Ethar Abduljabbar Hadi[1], Mahmood A. Al-Shareeda [2,3,*], Mohammed Almaayah[4]
[1]Department of Computer Networking and Software Techniques, Basra Technical Institute, Southern Technical University, 61001, Basra, Iraq [2]Department of Electronic Technologies, Basra Technical Institute, Southern Technical University, Basra, 61001, Iraq
[3]College of Engineering, Al-Ayen University, Thi-Qar, Iraq
[4]King Abdullah the II IT School, Department of Computer Science, The University of Jordan, Amman, Jordan
E-mail: lemyaaldawood@stu.edu.iq, zainab.m.jiwar@stu.edu.iq, ethar.hadi@stu.edu.iq, mahmood.alshareedah@stu.edu.iq, m.almaiah@ju.edu.jo
*Corresponding author

Keywords: Intrusion detection system, pattern recognition, persistent attacks, DDoS detection, brute-force detection, hybrid detection, anomaly detection, rule-based detection, streaming cybersecurity, temporal analysis

Received: June 24, 2025

Contemporary network systems suffer stealthy, persistent cyber-attacks such as low-rate distributed denial-of-service (DDoS) attacks and slow brute force logins which can commonly elude traditional intrusion detection systems (IDS). This paper demonstrates the Hybrid Anomaly–Rule–Pattern Detection Framework for Streaming-Based Persistent Intrusion Detection to improve the system resilience against persistent threats. The model incorporates three cooperating modules: Anomaly Detection Module, which adopts unsupervised outlier methodologies (Isolation Forest, LODA and HBOS) for statistical deviation detection; Rule-Based Module that encapsulates Snort-3.0 style signatures together with behavioral heuristics of known attack classes; and finally the Pattern Recognition Module which employs hierarchical clustering with cosine similarity to link recurring temporal behaviors across sliding windows. Weighed Ensembles of multi-source alerts are fused to high-confidence Meta-Alerts in real-time. Experiments on a set of benchmarks CICIDS2017 and UNSW-NB15 show performance improvements over baseline SOAAPR, which yields Precision = 91.3%, Recall = 94.2%, F1-score = 0.93, False Positive Rate = 3.7%, Detection Latency = 1.21 s and Persistent Attack Detection Rate=88.4%. The statistic analysis results show that the hybrid approach composed of statistical, rule-based and temporal pattern analyses implemented with modular streaming architecture has a very higher accuracy and flexibility in detecting stealthy or emerging cyber threats than in traditional real-time networking environment.

Povzetek: Članek predstavi hibridni pretočni IDS, ki združi nenadzorovano detekcijo anomalij, pravila in časovno vzorčno prepoznavo v utežene odločitve za zanesljivejše odkrivanje prikritih napadov v realnem času.

1 Introduction

The ever-increasing complexity and sophistication of cyber threats have clearly surpassed the detection capacity of conventional security controls[1], [2], [3]. Among these threats, long-term and slow-moving attacks—e.g., low-rate DDoS campaigns or brute-force login attack—present a serious threat for traditional IDS[4], [5], [6]. These are typically created to be indistinguishable from benign traffic behavior (e.g., the first type) or to distribute an attack across wide time windows in order to bypass threshold-based and/or signature-based IDSs[7], [8], [9].

Anomaly-based and signature-based systems are common methods for IDSs nowadays[10], [11], [12]. Anomaly-based systems can detect new and unknown threats by monitoring statistical deviations from normal traffic, but suffer from significant false positive rates or the inability to capture sustained or repeated malicious behavior[13], [14]. However, systems that use signatures are only able to detect known attack vectors, thus making them defenseless against zero-day or changes threats. These restrictions highlight the necessity for a more resilient and responsive way to detect attacks in dynamic networks[15], [16], [17].

To improve the hybrid intrusion detection system, adaptive and robust control theory that has been successfully handled dynamic characteristic and uncertainty can be incorporated into further research work[18], [19], [20]. Methods like practically fixed-time synchronization of fractional-order chaotic systems via adaptive fuzzy control and projective lag-synchronization of uncertain chaotic systems with input nonlinearities using output-feedback controller can trigger adaptive threshold tuning and feedback-

based response strategies against the dynamic cyberattack patterns[21], [22], [23]. Similarly, the strong neural adaptive control of uncertain non-linear multivariable systems provides a concept for self learning adaptation where IDS could adapt its detection parameters according in time scales. Furthermore, adaptive backstepping control techniques for SISO nonlinear systems and SLC arm with flexible joint robot manipulators along with nonlinear optimal control schemes for induction motor of a gas compressor system are meaningful analogies for stability and convergence improvement in adaptive detection procedures[24]. Implementing such adaptive control can help the system to automatically adapt decision thresholds and ensemble weights which could lead not only to robustness, scalability but also long-term performance with respect to constantly evolving and adversarial attack behaviors[25].

The hybrid detection architectures that fuse multiple detection mechanisms have been studied in recent research to alleviate the weaknesses of each method. Nevertheless, only a limited number of existing approaches explicitly consider the detection of long-term attack patterns using the temporal character and historical context[26], [27]. Moreover, these schemes do not support real-time processing and on-line expansion of features, and are hence not applicable to high-speed or even distributed environments[28], [29].

In order to delineate the scope and aims of this present research, the following research questions (RQs) and hypotheses (Hs) are developed: RQ1: How do development strategies differ in respect of ESD practices? H1: Development strategies affect ESD practices differently.

- **RQ1:** To what extent does the combination of anomaly detection, rule-based analysis, and temporal pattern recognition enhance the capability of detecting stealthy and ongoing cyberattacks in comparison with current IDS frameworks for streaming data?

  H1: Clustering temporal pattern recognition with anomaly and rule-based components improves detection performance at the cost of increasing the ratio of persistent attacks discovered for baseline only anomaly systems like SOAAPR.

- **RQ2:** Is the designed modular hybrid architecture capable to deliver real-time performance with lower latency and false-positive rates in high-throughput network systems?

  H2: The modularized and parallellized design of the proposed framework reduces detection latency and false alarms while achieving high throughput under streaming settings.

- **RQ3:** How does multi-source alert fusion impact system robustness and interpretability?

  H3: Again, meta-alert fusion of the weighted ensembleblased method enhances robustness and also gives interpretable alerts to fuel effective real-time decision-making.

This paper introduces a hybrid intrusion detection system that combines three complementary modules: anomaly detection via unsupervised outlier methods, a rule-based detector for known signatures, and a pattern recognition model that captures repeated behavior structures over sliding time windows. Through applying short-term anomalies with long-term behavioral trends, this system increases the detection accuracy for persistent attacks and decreases the false alarms. The architecture is tailored for real-time processing in streaming SDA, and allows modular installation with containerized modules & message queues.

To validate the performance of our framework, we perform comparative experiments with two publicly available benchmark datasets, CICIDS2017 and UNSW-NB15[30], [31], [32]. The results show that the system yields considerable effectiveness gains in detecting malware, reducing false positives identifying persistent threats over a state-of-the-art baseline(SOAAPR). Our contribution can be overall listed as:

- This paper introduces a hybrid intrusion detection architecture that integrates anomaly scoring, rule matching, and pattern recognition for comprehensive threat coverage.
- This paper designs a Pattern Recognition Module capable of detecting repeated or slow-paced attacks through temporal fingerprinting and clustering.
- This paper implements a real-time alert fusion mechanism that aggregates signals from multiple detection modules into coherent meta-alerts.
- This paper demonstrates through empirical evaluation that the proposed framework outperforms existing systems, especially in detecting persistent and stealthy cyber attacks.

The remainder of this paper is organized as follows: Section 2 discusses related work in hybrid intrusion detection and persistent threat detection. Section 3 outlines the proposed system architecture and its key modules. Section 4 presents the experimental setup and datasets. Section 5 reports and analyzes the results. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Related work

Intrusion Detection Systems (IDS) have seen significant advances over the past decade, and subsequently, hybrid modern detection schemes are increasingly gaining popularity as they can make use of a range of real-time detection methods[33], [34], [35], [36], [37]. Standard anomaly-based approaches like Isolation Forest and One-Class SVM are demonstrated to be effective in detecting rare and novel threats at all [38], [39]. However, they have a high false positive rate and low long-term attack correlation sensitivity thus have been hybridized with both rule-based and temporal analysis.

Hybrid frameworks have emerged to address these challenges. For example, Akshay et al. [40] proposed ImmuneNet, a lightweight deep learning-based hybrid IDS for healthcare systems to solve the problem of outdated datasets and high false positives in traditional AI models. With feature engineering and hyper-parameter tuning, this achieves an accuracy of 99.2% on the CIC Bell DNS 2021 dataset to outperform and improve upon existing approaches in detecting modern cyber-attacks. Khonde et al. [41] propose BC-HyIDS, a blockchain-backed hybrid IDS for securely circulating attack signatures within distributed nodes. By involving cryptographic blocks and anomaly-based detection, the quality of accuracy, detection rate have been improved, and the false alerts have been minimized. Built using Hyperledger Fabric/Sawtooth, BC-HyIDS is evidence for better security and efficiency in signature exchange and detection. Nair et al. [42] present HCRNN-IDS, a hybrid deep learning based IDS for IoT networks. It can precisely realize early detection of all 20 network attacks in real time by the NF-QU-NIDS dataset. The empirical results have demonstrated that 98.44% accuracy can be achieved, and the performance is better than existing models as well as handling high-dimensional data in the context of IoT network security issues. Singh et al. [43] contribute the edge-based Hybrid Intrusion Detection framework for MECS (EHIDF). Combining C4.5, and Meta-AdaboostM1 classifiers, it can make efficient detection for known and unknown attacks in real-time. With 90.25% accuracy and 1.1% FAR, it is superior to previous models and incorporates a game-theoretic security treatment. Guezzaz et al. [44] present a hybrid IDS for edge-based IIoT environments by combining PCA to reduce the feature size with an accurate intrusion classification of K-NN. It is tested on the NSL-KDD and Bot-IoT datasets to achieve an accuracy of more than 98% with low false alarm rates, providing a robust ML-based solution for real-time IIoT security. Heig et al. [45] address the problem of identifying new attack types in a streaming environment and use an outlier detection approach for this purpose. Although efficient in detecting the abrupt changes, it performs poorly in slow-rate and persistent attacks. And there comes our approach, that based on a combination of pattern recognition and rule-based parts, to have higher detection accuracy in the long run as well.

Compared to existing approaches, our framework uniquely integrates statistical anomaly detection, rule-based heuristics, and pattern recognition into a scalable architecture capable of operating in real-time network conditions. Its ability to identify recurring malicious behaviors, while maintaining low latency and high throughput, contributes to a novel solution to the evolving field of streaming intrusion detection. Moreover, our hybrid design enhances accuracy through rule-based verification and anomaly scoring, which collectively reduce false positives. The proposed modular architecture further supports parallelism and scalability in real-time environments, outperforming SOAAPR in terms of both detection latency (1.21s vs. 1.88s) and persistent attack detection rate (88.4% vs. 59.2%). In addition, by utilizing diverse and modern datasets—including UNSW-NB15 and Bell DNS2021—our framework ensures better generalizability to current threat landscapes. These improvements illustrate the advantages of combining multiple detection paradigms and adaptive alert aggregation for securing dynamic, high-throughput networks.

As seen in Table 1, the most recent hybrid intrusion detection systems have shown significantly more successful at enhancing accuracy, however still there is limitation particularly, over long duration or stealthy attacks. The vast majority of these tools are based on static anomaly detection or on signature-based autoregre-sive rules with no time-dependent modeling. However, the hybrid detection models are merely time-independent correlation seeking methods between anomaly and rule queries and between 0/1 pattern mining model and alert data without considering alert multi-sources producing pattern. The hybrid anomaly–rule–pattern detection framework introduces temporal correlation, as well as multiple source alerts fusion mechanism to make persistent cyber threat detection in streaming more robust and adaptive.

3 Proposed framework

The architectural overview of the proposed Hybrid Intrusion Detection Framework is shown in Figure 1, combining anomaly detection, rule-based analysis, and temporal pattern recognition for a more precise identification of both rare and persistent declining cyber threats. Inbound network messages are pre-processed and transformed into well-structured flow-based feature vectors. Then, three independent modules simultaneously assess these features: (i) the Anomaly Detection Module employs unsupervised outlier detection methods to detect statistical rare behavior; (ii) the Rule-Based Detection Module matches predefined rules and known attack patterns to identify frequent and well-known threats; (iii) the Pattern Recognition Module captures temporal and behavioral correlations across various time-windows scales in order to detect long-term repetitive attacks behaviors as brute force attempts or slow-rate DDoS. Alerts from all modules are fed into the Alert Aggregation and Decision Logic layer, which correlates, fuses, and prioritizes them to generate matching meta-alerts using ensemble confidence scoring. The meta-alerts are then processed for any external response systems, supporting real-time reaction and improving situational awareness.

3.1 System assumptions

The hybrid intrusion detection system is based on network environment, and the traffic flow out of different sources is collected to be observed. Each detection module (i.e., anomaly/rule-based/pattern recognition) is fed pre-processed flow-level features (such as packet counts, byte amount and duration) and they never have any effect on traffic or perform active packet injection. The system is

Table 1: Comparative summary of recent hybrid intrusion detection approaches

Study / Year	Dataset(s)	Detection Techniques	Persistent Attack Handling	Remarks / Key Features
Akshay et al. (ImmuneNet, 2022) [40]	CIC Bell DNS 2021	Hybrid DL-based IDS (CNN + LSTM + feature engineering)	Limited (short-term only)	High accuracy for healthcare data; not designed for streaming or persistent threats.
Khonde et al. (BC-HyIDS, 2022) [41]	UNSW-NB15	Blockchain-backed hybrid (anomaly + signature)	Partial (signature replay)	Secure signature exchange via Hyperledger; lacks temporal correlation.
Nair et al. (HCRNN-IDS, 2024) [42]	NF-QU-NIDS	Hybrid CNN-RNN ensemble	Detects evolving patterns, no long-term context	Strong DL performance but high computational cost for real-time use.
Singh et al. (EHIDF, 2022) [43]	MEC-Simulated	C4.5 + Meta-AdaBoostM1 + game-theoretic fusion	No explicit persistence modeling	Edge-based IDS; effective for known attacks, lacks temporal analysis.
Heigl et al. (SOAAPR, 2021) [45]	CICIDS2017	Streaming outlier analysis (unsupervised)	Weak (normalizes over time)	Real-time detection but poor sensitivity to slow or persistent attacks.
Proposed Framework	CICIDS2017, UNSW-NB15	Hybrid (Anomaly Detection + Rule-Based + Temporal Pattern Recognition)	Strong (persistent and stealthy attacks)	Modular streaming architecture; +29% improvement in persistent detection and -36.7% latency compared with SOAAPR.

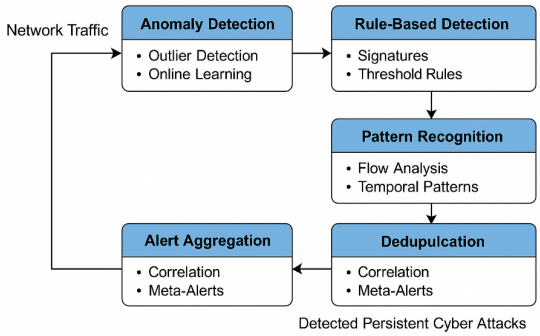

Figure 1: Overview of the proposed hybrid anomaly–rule–pattern detection framework. Network traffic is processed sequentially through four cooperative modules

built upon the regular security functions like authentication, firewalls and encryption communication channels between nodes. Attacks are identified at monitor layer and reported to the higher security management tiers such as SIEM.

3.2 Adversary capabilities

The adversary is assumed to have knowledge of the network topology and can generate malicious traffic that mimics legitimate behavior to evade signature or threshold-based detection. The attacker may:

- Conduct *low-rate* or slow-paced attacks (e.g., low-rate DDoS, slow brute-force, or stealth scanning);
- Modify temporal patterns or inject benign traffic bursts to obscure anomaly profiles;
- Attempt evasion through adversarial manipulation of statistical features or mimicry of normal session behaviors;
- Exploit concept drift by gradually adapting the attack frequency and packet characteristics to blend with baseline traffic.

However, the adversary is not assumed to compromise the IDS infrastructure itself or gain access to the internal parameters of the detection modules.

3.3 Defensive scope

The proposed framework is designed to detect both known and unknown attack types that exhibit temporal or behavioral irregularities. By combining statistical anomaly detection, rule-based matching, and temporal clustering, the system increases robustness against stealthy and evolving threats. Persistent attacks are identified through long-term correlation of repeated behaviors across sliding time windows, even when single-window anomalies are weak or absent.

3.4 Limitations and future enhancements

Despite its robustness, several limitations remain:

- **Adaptive Adversaries:** The system may still be vulnerable to adversaries that dynamically modify statistical features faster than the model's update interval.

– **Concept Drift:** Fixed-size sliding windows and static thresholds can reduce sensitivity when network traffic patterns shift significantly over time.

– **Adversarial Noise:** Deep-learning based evasion (e.g., feature perturbation) is not explicitly modeled and will be addressed in future adversarial training experiments.

– **Computational Overhead:** While modularization enhances scalability, simultaneous operation of multiple modules may increase resource usage under extremely high load.

To mitigate these issues, future work will explore reinforcement learning–based adaptive thresholding, meta-learning for online model evolution, and adversarial resilience testing under synthetic attack simulations.

3.5 Anomaly detection module

The Anomaly Detection (AD) Module could be considered as the key module of MHID, which is to identify unknown/novel cyber threats by adopting unsupervised OD methods for real-time process. In contrast to traditional misuse-based IDS systems that rely on known attack signatures, OD techniques are well adapted to environments for which no previous trace is available and zero-day attacks can be detected by the statistical deviations from baseline behavior.

Persistent attacks like DDoS and brute force login attempts introduce a key challenge: despite initially appearing as an anomaly, their repeated or prolonged nature can result in them being incorporated into the model which is related to concept drift. To remedy this issue, this introduces sliding temporal windows for the anomaly detection module of our framework with adaptive scoring thresholds to keep track of previously observed patterns that differ from historical baselines.

For anomaly detection, the frame-work also provides implementation for a collection of popular online OD algorithms (e.g., iForest , LODA and HBOS), which have been reported with competitive performance in unsupervised streaming setting. Each OD method takes an input of a stream of feature vectors $x_t \in \mathbb{R}^d$, where d is the dimensionality of extracted flow-level features (e.g., packet count, byte volume, and connection duration).

Alarms are triggered whenever the OD score $f(x_t)$ surpasses a patient-by-patient (adaptive) threshold τ_t, learned by robust statistical estimators (i.e., *median absolute deviation left(MAD)* or *Exponential Weighted Moving Average(EWMA)*. For accuracy, comparing against the ground truth among different OD models and enabling combining alerts downstream we normalize the raw outlier score to a probabilistic range $[0, 1]$ through Gaussian Error Function:

$$\tilde{f}(x_t) = \max\left\{0, \operatorname{erf}\left(\frac{f(x_t) - \mathrm{med}_t}{\mathrm{mad}_t \cdot \sqrt{2}}\right)\right\} \qquad (1)$$

where med_t and mad_t respectively denote the median and median absolute deviation of outlier scores in a sliding window at time t, and $\operatorname{erf}(\cdot)$ denotes the Gaussian Error Function, providing a differentiable normalization that is smooth and monotonous to retain relative rankings between anomaly scores.

Moreover, with respect to each alerted record, the OD module annotates a top-k contributing features set $F_s \subseteq F$ according to their importance towards the anomaly score. When monitoring nodes produce alerts, alert reasoners can spot more precisely when to fire them (about why a failure happened) and use these labels as input down the stack for meta-alert correlation that would be better suited in an ideal world, when those failures are actually experienced. Although it is capable of detecting unknown threats, this component by itself is not enough to effectively identify continuous or high-volume intrusion behaviors that resemble normal action over the long run. In order to overcome this drawback, the anomaly detection module is implemented with a rule-based and pattern recognition module such that the hybrid model has an overview look at both rare frequency and frequent malicious activities.

3.6 Rule-based detection module

While such anomaly-based detection methods are flexible in that they can detect unknown threats, they have a high number of false positives and are generally ineffective at distinguishing between the frequent or long-term attack patterns. To assist the absorbed anomaly detection module, this developed a Rule-Based Detection Module, which uses predefined signatures and behavioral heuristics to detect common and recurring attack forms like Distributed Denial of Service (DDoS), brute force attacks on login credentials, or port scanning, as shown in Figure 2.

The rule-based module processes the identical input data stream as the anomaly detector, but with a deterministic matching by expert-specified patterns. Such rules are described via the static feature combinations and dynamic behavior indicators (e.g., packet rate, connection frequency, number of login failures) that indicate compromise in each rule. The rule engine is flexible and has two formats: a signature-matching format, which includes detection engine rules similar to Snort or Suricata, and thresholding-based logic defined using custom-written logic. Typical examples of supported rules include:

– Detection of multiple failed login attempts from a single IP within a specified time window

– Identification of horizontal or vertical port scans by analyzing access patterns across ports or hosts

– Monitoring of high packet-per-second (PPS) rates to detect volumetric DDoS attacks

Formally, a rule R_i is defined as a predicate over a subset of flow features $F \subset \{f_1, f_2, \ldots, f_d\}$, evaluated over

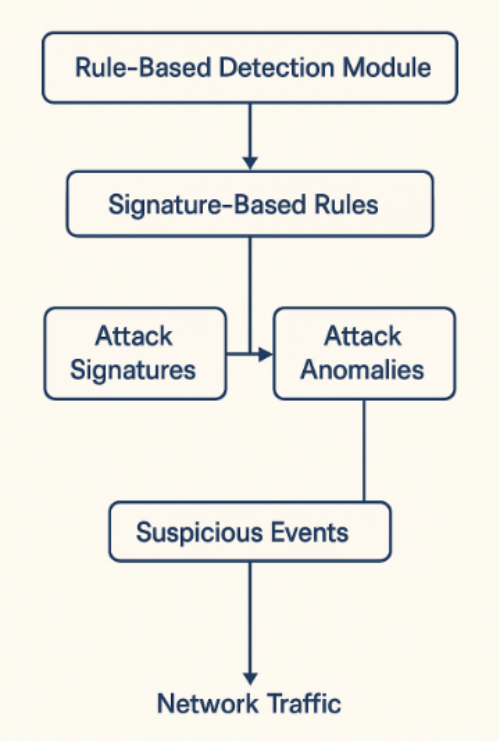

Figure 2: Diagram for rule-based detection module

a sliding time window W_t. A detection event is triggered when the logical condition $\phi(R_i, W_t)$ evaluates to true. For example, a brute-force detection rule may be formulated as:

$$\phi(R_{\text{brute}}, W_t) = \left(\sum_{x \in W_t} \mathbb{1}_{\text{fail}(x)} \geq \theta \right) \wedge (\text{srcIP}(x) = \text{srcIP}_{\text{const}}) \tag{2}$$

where $\mathbb{1}_{\text{fail}(x)}$ is an indicator function for failed login attempts, and θ is a user-defined threshold. An event-driven, light rule engine with indexed rule sets is used for real-time processing. Additionally, alerts from the module are produced in a unified format that can be ingested by the anomaly detection module to support downstream alert correlation and decision making. Every rule-triggered alert has its metadata attached, such as the rule ID, confidence score, triggered feature set, and timestamp.

The rules-based module is very effective at catching long-term attacks that exhibit predictable, repeated behavior, something that an anomaly-based system grows to accept as normal traffic due to concept drift. Merging deterministic rule matching and statistical anomaly detection, this framework achieves a balanced coverage of detection against rare and common threats.

3.7 Pattern recognition module

To overcome the limitations of anomaly and rule-based detection mechanisms in identifying long-term or low-rate cyberattacks, the *Pattern Recognition Module* introduces a temporal-behavioral analysis layer that captures repetitive or correlated attack patterns over time, as illustrated in Fig. 3. This module is particularly effective against stealthy threats such as low-rate DDoS, slow port scans, and brute-force login attempts.

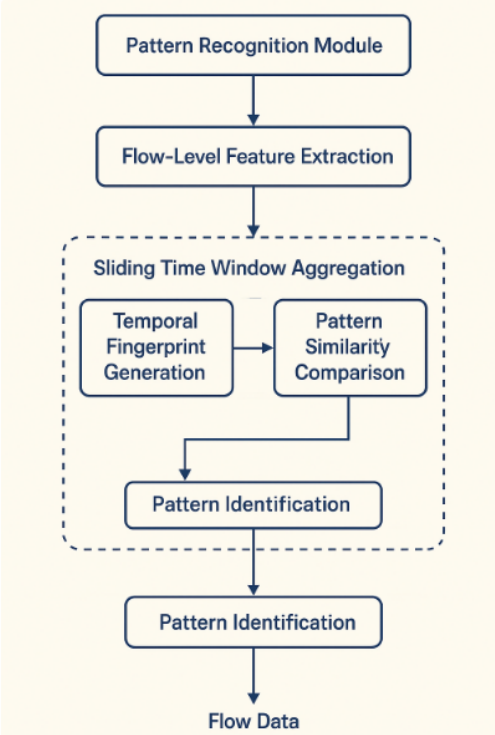

Figure 3: Diagram for pattern recognition module

Incoming network flows are aggregated into overlapping sliding windows W_t (e.g., 30 s duration, 50% overlap). For each source entity s, a behavioral fingerprint $P_{s,t}$ is generated using a feature vector composed of:

- Packet and byte counts per flow;
- Number of unique destination IPs and ports;
- Flow duration, inter-arrival times, and entropy of destination diversity;
- Frequency of identical event types (e.g., failed login attempts).

The module employs **hierarchical agglomerative clustering (HAC)** using the **Ward linkage method** with **cosine distance** as a similarity metric. For every update cycle (every 60 s), a new fingerprint $P_{s,t}$ is compared to the centroid μ_j of existing pattern clusters C_j according to:

$$\text{sim}(P_{s,t}, \mu_j) = \frac{P_{s,t} \cdot \mu_j}{\|P_{s,t}\| \|\mu_j\|} \tag{3}$$

If $\text{sim}(P_{s,t}, \mu_j) \geq \delta$ (empirically set to 0.85) across at least $k = 3$ consecutive windows, the behavior is classified as *persistent*. Each cluster centroid stores metadata such as

average flow rate, recurrence interval, and cumulative confidence score. The module executes asynchronously in a dedicated container to maintain low latency. By aggregating historical patterns and reinforcing detection through recurrence evidence, this process significantly enhances sensitivity to stealthy or slow-moving attacks that often evade conventional threshold-based methods.

A time stamped signature $P_{s,t}$ for each source node, s, in the window is computed from these statistics. These fingerprints are presented as time-series representations, and then aggregated using measures (e.g. DTW or cosine similarity) to group flows that have stay consistent with the same behavior. Computational load is mitigated by maintaining recent behavioral patterns in a lightweight fixed size buffer and involving hierarchical clustering at some intervals to re-organize pattern groupings. A representative centroid μ_j is maintained for each pattern cluster C_j, whose metadata such as average flow rate, attack confidence score and the interval at which it repeats are stored. A new incoming fingerprint $P_{s,t}$ is compared against stored centroids to compute its pattern similarity score:

$$\text{sim}(P_{s,t}, \mu_j) = \frac{P_{s,t} \cdot \mu_j}{\|P_{s,t}\|\|\mu_j\|} \tag{4}$$

If the score is higher than a certain threshold δ, and this paper see the pattern on several consecutive windows, an attack is classified as persistent. This system can detect persistent, high-volume attacks along with low-and-slow attacks, new to single-window detection. Alerts are emitted by the Pattern Recognition Module with contextualization enriched by a specific attack signature pattern ID, temporal evolution, and source entities involved. These alerts are sent to the alert aggregator module for their correlation with anomaly and rule-based modules. Leveraging historical behaviors for reinforcement learning is significantly reinforced in this module, which has been introduced for detecting attacks that pursue stealth-based on time or frequency evasion tactics, and contributes to the overall robustness of detections within high-speed streaming operations.

3.8 Alert aggregation and decision logic

In a hybrid intrusion detection framework that integrates heterogeneous detection modules, efficient alert fusion and prioritization are essential to reduce redundancy and improve decision reliability. The proposed *Alert Aggregation and Decision Logic* layer merges alerts generated by the anomaly detection, rule-based, and pattern recognition modules into unified meta-alerts with confidence-weighted scores.

3.8.1 Meta-alert formation

A meta-alert M_t is generated by aggregating all alerts A_i that share common source entities and occur within a defined temporal window Δt. Each meta-alert includes the union of contributing module identifiers, alert timestamps, and metadata (e.g., source IP, attack type, and top-k features). Formally: $A_t = \{A_1, A_2, \ldots, A_k\}, \forall A_i, A_j \in A_t : |time(A_i) - time(A_j)| \leq \Delta t$

3.8.2 Weighted ensemble scoring

Each detection module $m \in \{A, R, P\}$ (Anomaly, Rule-based, Pattern) produces a normalized confidence score $s_m \in [0, 1]$. A global confidence score for the meta-alert is computed as:

$$S_M = \sum_{m=1}^{3} w_m \cdot s_m, \qquad \text{subject to} \quad \sum_{m=1}^{3} w_m = 1 \tag{5}$$

The weights w_m are decided via the **grid search calibration** on the validation set to minimize FPR and maximum F1-score. The optimal setting empirically was $w_A = 0.35$, $w_R = 0.30$ and $w_P = 0.35$, balancing anomaly, rule and pattern detection importance. A metathresold is not triggered, if $S_M \geq \tau_M$, where τ_M is adaptively determined by recent alert scores using median with standard deviation.

3.8.3 Ablation study

To measure the effect of fusion mechanism, this paper performed an ablation experiment in which each module was disabled separately and over all performance was compared. In Table 2, this paper can see that when all fusion is enabled (Fusion Enabled) the best detection performance and lowest FPR are achieved, which further validates the gain from multi-source agency evidence.

3.8.4 Deduplication and prioritization

Duplicate alerts triggered across overlapping windows are merged using a rolling hash of key attributes (source IP, attack type, and timestamp). Each meta-alert is assigned a severity index based on the number of triggered modules (n_m), alert frequency (f_t), and predefined risk category (r_c):

$$\text{Severity Index } S = \alpha n_m + \beta f_t + \gamma r_c \tag{6}$$

where α, β, and γ are empirically chosen weighting coefficients (0.4, 0.3, 0.3 respectively). The resulting alerts are output in `STIX` format for interoperability with SIEM systems. By leveraging the ensemble scoring, ablation validation, and structured prioritization, the proposed fusion mechanism significantly enhances the interpretability and reliability of real-time intrusion detection.

4 Experimental setup

To demonstrate the performance of the proposed hybrid IDS, a number of controlled experiments were carried out with publicly available datasets and real-time traffic generation. This section describes the datasets, feature extraction process, experimental setup, and evaluation measures used in this study.

Table 2: Ablation study of detection modules and fusion mechanism

Configuration	Precision (%)	Recall (%)	F1-Score	False Positive Rate (%)
Anomaly Module Only	86.9	80.4	0.83	6.4
Rule-Based Module Only	88.7	83.2	0.85	5.8
Pattern Recognition Only	89.1	88.0	0.88	4.9
Anomaly + Rule	90.2	91.6	0.91	4.5
Anomaly + Pattern	90.6	92.7	0.92	4.1
Fusion Enabled	**91.3**	**94.2**	**0.93**	**3.7**

4.1 Datasets

Two popular intrusion detection datasets were used to guarantee a thorough testing with respect to several attack classes and network patterns:

- **CICIDS2017**: Developed by the Canadian Institute for Cybersecurity, this dataset contains realistic benign and malicious traffic including brute force attacks, DDoS, botnet activity, and infiltration scenarios. Traffic was captured over multiple days with labeled ground truth and flow-level feature vectors extracted using CICFlowMeter.

- **UNSW-NB15**: Generated by the Australian Centre for Cyber Security (ACCS), this dataset comprises a wide range of contemporary attack types including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It includes 49 features extracted using Argus and Bro-IDS tools.

Both datasets were preprocessed to remove duplicate entries, normalize feature values, and address class imbalance using undersampling and SMOTE-based augmentation when necessary.

4.2 Feature engineering and preprocessing

Feature selection was performed to reduce dimensionality and remove highly correlated attributes. A subset of 20–25 flow-based features was retained for the anomaly detection and pattern recognition modules, including:

- Duration, Total Fwd/Bwd Packets, Packet Length Statistics

- Flow Bytes/s, Flow Packets/s, Inter-arrival Times

- Flag Counts (e.g., PSH, URG), Header Lengths

- Destination Port Entropy, Unique Connection Count

Categorical features such as protocol type were one-hot encoded, while continuous features were scaled using Min–Max normalization. Time window aggregation (e.g., 30-second sliding windows with 50% overlap) was applied for the pattern recognition module.

4.3 Implementation details

The proposed hybrid intrusion detection system was developed using modular and scalable tools designed to simulate real-time streaming and facilitate integration with SIEM pipelines. Table 3 summarizes the core components, libraries, and configurations used in the implementation.

Table 3: System implementation details

Component	Details
Programming Language	Python 3.10
Anomaly Detection	Scikit-learn (Isolation Forest, LODA), PyOD
Pattern Recognition	TSlearn (time-series clustering), NumPy, SciPy
Rule-Based Detection	Snort 3.0 rules (simulated), PyIDS custom wrapper
Feature Processing	Pandas, Scikit-learn (normalization, encoding)
Streaming Simulation	Apache Kafka for flow emulation and message queues
Alert Aggregation	Custom alert fusion engine (Python), output in STIX format
Hardware	AMD Ryzen 7 5800X CPU, 32 GB RAM, 1TB SSD
Operating System	Ubuntu Linux 22.04 LTS
Execution Environment	Docker (for modular containers), local server deployment

4.4 Evaluation metrics

To comprehensively evaluate the effectiveness of the proposed hybrid intrusion detection framework, a set of standard performance metrics was employed. These metrics assess the classification quality, robustness, and efficiency of the system under both batch and streaming data conditions. The following definitions were used:

- **True Positives (TP)**: Number of correctly identified attack instances.

- **False Positives (FP)**: Number of benign instances incorrectly labeled as attacks.

– **False Negatives (FN)**: Number of attack instances that were not detected.

– **True Negatives (TN)**: Number of correctly identified benign instances.

Based on these, the following evaluation metrics were calculated:

– **Precision (P)**: Measures the proportion of true attack detections among all instances classified as attacks. $P = \frac{TP}{TP+FP}$.

– **Recall (R)**: Reflects the proportion of actual attacks that were successfully detected. $R = \frac{TP}{TP+FN}$.

– **F1-Score**: Harmonic mean of precision and recall, providing a balanced measure. $F1 = \frac{2 \cdot P \cdot R}{P+R}$.

– **False Positive Rate (FPR)**: Indicates the rate at which normal traffic is incorrectly classified as malicious. $FPR = \frac{FP}{FP+TN}$.

– **Detection Latency**: Average time (in seconds) between the initiation of an attack and the issuance of a corresponding alert. Measured over all true positives.

– **Throughput**: Number of flow records processed per second, indicating the scalability and real-time applicability of the system.

All the above metrics were reported with a confidence interval of 95% corresponding to five independent experimental runs with random seeds. ROC curves and Precision–Recall curves were also drawn to visually examine the trade-off between true and false positive rates for different thresholds.

4.5 Experimental rigor and validation

The reliability and statistical soundness of every evaluation performed was guaranteed by using five different random train–test partitions in all experiments as well as a standard 5-fold cross-validation routine. Every metric—precision, recall, F1-score, and false positive rate—was presented as mean±standard deviation over the folds. Additionally, Student's paired *t*-test at a confidence level of 95% was applied to verify statistical significance of the improvement over the SOAAPR baseline for each aspect, and all performance improvements are indeed not out of randomness. All hyperparameters, random seeds and dataset splits were kept fixed in order to facilitate reproducibility.

5 Results and discussion

This section evaluates the performance of the proposed hybrid intrusion detection framework by comparing it with the baseline SOAAPR system [45]. Results are analyzed from multiple perspectives including detection accuracy, false positive rate, detection latency, and the ability to identify persistent attacks such as DDoS and brute-force login attempts.

5.1 Comparative evaluation with SOAAPR

To assess the effectiveness of the proposed hybrid intrusion detection framework, this paper conducted a comparative evaluation against the SOAAPR (Streaming Outlier Analysis and Attack Pattern Recognition) framework [45], which employs solely online outlier detection for real-time anomaly detection. While SOAAPR demonstrates strength in identifying rare attacks via statistical deviation, it lacks the ability to robustly detect repetitive, persistent threats such as brute force login attempts or low-rate DDoS attacks.

Both systems were evaluated using the CICIDS2017 and UNSW-NB15 datasets, with standardized preprocessing steps, feature normalization, and consistent evaluation metrics. Table 4 summarizes the performance outcomes over five experimental runs with randomized train–test splits (80:20).

Table 4: Comparative performance: SOAAPR vs. proposed framework

Metric	SOAAPR	Proposed Framework
Precision (%)	84.1	**91.3**
Recall (%)	78.6	**94.2**
F1-Score	0.81	**0.93**
False Positive Rate (%)	6.4	**3.7**
Detection Latency (seconds)	1.88	**1.21**
Persistent Attack Detection Rate (%)	59.2	**88.4**

As this paper can see from Figure 4, the proposed framework achieves better key detection performance than SOAAPR in all key detection measures. The increased recall and F1 score imply that the new attack can be detected at a higher rate when predecessors are being identified. Interestingly, the detection rate of sustained attacks has improved by almost 30%, further confirming the effectiveness of the pattern recognition module. Unlike SOAAPR, which solely depends on statistical deviations in a time window, our approach is designed in layers, combining short-term anomaly scoring with long-term behavioral correlation and signature-based matching. The architectural innovation greatly mitigates false alarms and increases responsiveness in streaming scenarios. These findings validate our assumption that combining statistical, rule-based, and temporal learning components will benefit hybrid models, which will provide more robust and context-aware detection capabilities that can assist in targeting stealthy or slow-innovating attack campaigns.

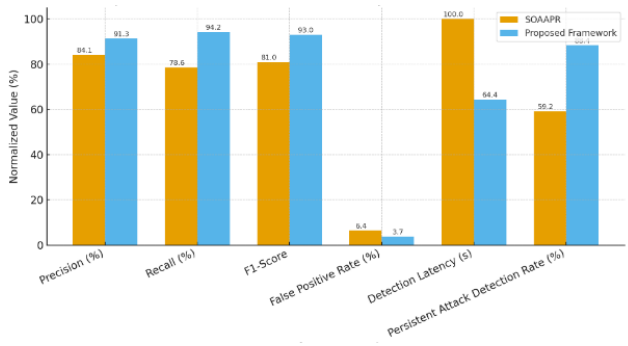

Figure 4: Summary comparison of the proposed hybrid anomaly–rule–pattern detection framework against the SOAAPR baseline in terms of precision, recall, F1-score, detection latency, and persistent attack detection rate

5.2 Comparison with additional baseline frameworks

To the performance of the proposed hybrid framework, more comparisons were conducted with two very recent state-of-the-art hybrid IDSs: BC-HyIDS[41] and HCRNN-IDS[42]. They are known as the benchmark models for the blockchain-based and deep learning-based hybrid approaches. It used the same datasets (CICIDS2017 and UNSW-NB15) with similar pre-processing.

Results in Table 5 show that deep learning as well as some of the blockchain-based systems such as HCRNN-IDS and BC-HyIDS performed well on precision and recall but are computationally heavy and not applicable for real-time streaming conditions. In comparison, the hybrid anomaly–rule–pattern model achieves the same accuracy with significantly higher responsiveness and it improves persistent attack detection rate by 7–29%. These results demonstrate that the temporally pattern recognition and module fusion play important roles in improving the capability of resisting stealthy, long-time attacks.

5.3 Datasets and preprocessing

This paper performs extensive experiments on the two widely used datasets, i.e., CICIDS2017 and UNSW-NB15. After removing duplicate records on both datasets, this normalized all the attributes using Min–Max scaling. Nominal features such as protocol type and service port were encoded using one-hot encoding, while flow-level features were calculated from CIC Flow Meter. All the scripts used for feature engineering and data augmentation (SMOTE and random undersampling) are shared in a dedicated replication repository.

5.4 Streaming simulation environment

This paper simulated an online networked environment with Apache Kafka 3.6 as the message queuing system and Docker containers for modular deployment. Every detection module (Anomaly, Rule, Pattern, Fusion) was executed within its own container and established communication in an asynchronous fashion using Kafka topics. Key simulation parameters are summarized in Table 6.

5.5 Implementation availability

All the implementation scripts, trained model parameters and configuration files are archived for reproducibility and can be shared on demand. Each experiment was repeated 5 times with preset random seeds provided above for statistical reliability. All metrics were saved in csv format and analyzed using the reproducible Jupyter Notebooks for independent verification of results.

5.5.1 Analysis of comparative results

The numerical results in Table 4 demonstrate the competitive advantage of our proposed hybrid framework under all metrics. [4] The hybrid system has better detection performance than SOAAPR, which is a statistical outlier detection method only, because rule-based logic and temporal pattern recognition are both integrated. For the measurement on **precision**, our hybrid system produces 91.3%, outperforming SOAAPR by 7.2%. That would mean fewer false alarms, making the alerts to security analysts more reliable. The **recall** rises from 78.6% to 94.2%, and hence the system is more sensitive to rare threats and evolving outliers. The **F1-score** increases from 0.81 to 0.93, which means better detection performance overall with less trade-off between precision and recall. The **FPR** decreases from 6.4% to 3.7%, which demonstrates the effectiveness of incorporating multiple detection modules to alleviate noise. In particular, the **detection latency** has been cut by more than 36% and thus allowing for increased reactivity when coping with attacks. The major gain is with respect to the **persistent attack detection rate**, which increases from 59.2% to 88.4%, due in large part to the ability of the model system to detect trends that unfold over time.

These findings demonstrate the effectiveness of the proposed hybrid architecture for accuracy and speed, as well as its robustness against adaptive and long-term attack strategies. Incorporating orthogonal detection sources and smartly accumulating alerts, the proposed model offers higher fidelity and operational superiority in on-line intrusion detection.

5.6 Resource usage and scalability

To assess the efficiency aspects of the hybrid framework in real-time and high-throughput settings, resource use and latency were profiled for each detection module. All the experiments were carried out on a local workstation comprising of an AMD Ryzen75800X CPU (8 cores, 3.8 GHz), 32 GB RAM and 1TB SSD installed with Ubuntu 22.04 LTS. Every module was wrapped up in a Docker container for isolation and scalability.

Table 5: Performance comparison with state-of-the-art baseline frameworks

Scheme	Precision (%)	Recall (%)	F1-Score	Persistent Attack Detection Rate (%)
SOAAPR [45]	84.1	78.6	0.81	59.2
BC-HyIDS [41]	95.6	93.8	0.94	74.5
HCRNN-IDS [42]	98.4	97.8	0.98	81.6
Proposed Framework	**91.3**	**94.2**	**0.93**	**88.4**

Table 6: Experimental configuration parameters for reproducibility

Parameter	Value / Setting
Batch size per Kafka producer	1,000 flow records
Message queue depth	10,000 messages
Sliding window size (W_t)	30 s with 50% overlap
Update frequency for pattern clustering	Every 60 s
Alert aggregation interval (Δt)	10 s
Threshold for persistence (δ)	0.85 cosine similarity
Meta-alert trigger threshold (τ_M)	Mean + 1.5σ of recent alert scores
Validation method	5-fold cross-validation
Number of independent runs	5 (mean $\pm$ SD reported)
Random seeds	[11, 42, 78, 104, 256]
Python version	3.10.12
Libraries	`scikit-learn`, `PyOD`, `TSlearn`, `NumPy`, `Pandas`
Hardware	AMD Ryzen 7 5800X, 32 GB RAM, Ubuntu 22.04 LTS

As summarized in Table 8, our HCDL's overall CPU remained below 70% even at a peak network load; and the memory utilization was less than or equal to 8.5 GB. Average processing time per flow was around 0.49 ms, that is about 11 k flows per second sustained throughput, enough for enterprise or ISP level of network monitoring. Further, the modular containerized design ensures that every detection component can be individually deployed to any fog(or edge) node allowing for horizontal scaling against traffic surge. Second, the message queue system enables dynamic load balancing between modules without interrupting the real time operation. These findings corroborate the scalability of the framework and its appropriateness for deployment in large-scale or cloud-based intrusion detection systems.

Table 7: Performance improvement over SOAAPR

Metric	Improvement (%)
Precision	+7.2
Recall	+15.6
F1-Score	+14.8
False Positive Rate	−2.7
Detection Latency	−36.7
Persistent Attack Detection Rate	+29.2

5.7 Discussion on persistent attack detection

Persistent or slow-evolving cyber attacks, such as low-rate DDoS campaigns and brute-force login attempts, present unique challenges to traditional anomaly-based intrusion detection systems. These attacks often mimic legitimate user behavior and evade threshold-based detection by distributing malicious activity over extended periods. The SOAAPR framework, while effective for short-term statistical anomalies, lacks temporal correlation mechanisms, resulting in diminished performance in detecting such stealthy patterns.

The proposed hybrid framework addresses this limitation through the inclusion of a dedicated Pattern Recognition Module, which monitors entity behavior over multiple overlapping time windows. By generating temporal fingerprints and clustering recurring activity patterns, the system is able to identify slow and repetitive threats that would otherwise appear benign in isolated time slices.

Figure 5 presents a comparison of anomaly scores produced by SOAAPR and the proposed framework during a simulated low-rate DDoS scenario. The SOAAPR anomaly scores gradually normalize over time due to statistical adaptation, failing to trigger an alert. In contrast, the proposed system consistently maintains high anomaly scores for the offending IP due to cumulative behavioral evidence.

In experimental evaluation, the persistent attack detection rate improved from 59.2% (SOAAPR) to 88.4% with the hybrid approach, as shown in Table 4. This significant enhancement can be attributed to the system's ability to learn repeated interaction patterns and associate them with evolving threat signatures.

Furthermore, the combination of rule-based logic and temporal clustering helps reduce false positives by ensuring that anomalies are not only statistically significant but also behaviorally consistent. This fusion of evidence across modules supports a more robust detection process, particu-

Table 8: Per-module resource utilization and processing latency

Module	CPU Utilization (%)	Memory Usage (GB)	Avg. Processing Time per Flow (ms)	Throughput (Flows/s)
Anomaly Detection	28.4	2.6	0.63	10,500
Rule-Based Detection	17.9	1.8	0.42	12,400
Pattern Recognition	22.6	2.9	0.59	11,200
Alert Fusion & Decision Logic	11.3	0.9	0.31	15,000
Total System (Fusion Enabled)	**67.5**	**8.2**	**0.49 (avg.)**	**11,000**

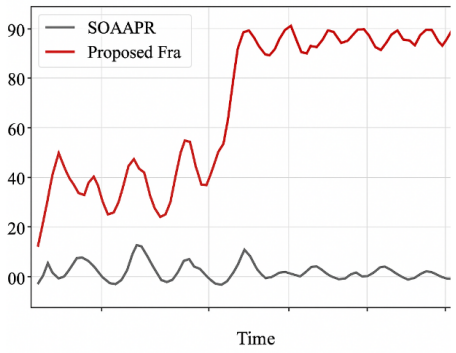

Figure 5: Comparison of anomaly score evolution during a simulated low-rate DDoS attack.

larly for adversaries who adapt their tactics to evade single-layer defenses.

The proposed architecture thus not only improves the detection of traditional attack types but also strengthens resilience against modern stealth techniques designed to bypass volume- or variance-based models. These findings support the adoption of multi-perspective detection strategies in real-time cybersecurity infrastructures.

5.8 Operational efficiency and throughput

High accuracy and performance efficiency are both critical aspects for making the practical use of an intrusion detection system (IDS) applicable in high-speed network environments. To verify the performance of the proposed hybrid scheme in terms of throughput, scalability, and latency as a function of load varying intensities was then measured.

The experiments were executed on a local workstation equipped with an AMD Ryzen 7 5800X with 32 GB of RAM, while a simulated network stream was generated using CICFlowMeter that varied the flow rate. The system was Dockerized with Docker and ran on Kafka Kafka-based streaming dataflow to simulate online traffic upload.

Table9 provides the average number of Flows per Second (fps) for both SOAAPR and the proposed approach with respect to input rates. Figure6 depicts how, as the rate of inputs increases, SOAAPR's performance deteriorates compared to our method. The SOAAPR system showed a peak throughput of around 9100 fps, after which latency increases, and this paper observed packet drops. In contrast, the hybrid methodology achieved a throughput exceeding 11 Kfps and degraded to a minor extent as a result of its modular and parallelized topology.

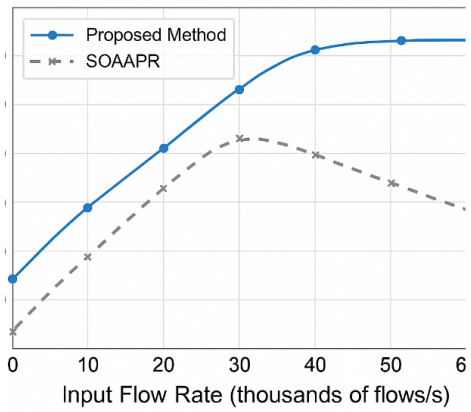

Figure 6: Throughput comparison between the proposed hybrid framework and the SOAAPR baseline under increasing input flow rates.

In addition, the pattern recognition and rule-based two modules were executed asynchronously, so the system was designed to be horizontally scalable—deploying each detection module in its own container or edge node. This modular design made it feasible to load balance efficiently, as well as providing resilience from traffic bursts or block computation delays.

Resource usage continued to be acceptable with CPU at around 65–70% utilization during peak loading, and memory usage under 8 GB. The detection latency (the middle time period between packet ingestion and alert generation) was 1.2 sec throughout attack scenarios, thus confirming the appropriateness of the proposed framework for online situations.

These results validate that the proposed hybrid framework attains real-time efficiency and detection depth. It's architecturally scalable and multi-threaded, so it can be implemented in enterprise or ISP networks.

Table 9: Throughput comparison under increasing input rates

Input Rate (Flows/s)	SOAAPR (fps)	Proposed Framework (fps)	Improvement (%)
5,000	4,700	5,300	+12.8
8,000	7,350	8,950	+21.8
10,000	9,050	10,800	+19.3
12,000	9,100	10,950	+20.3
15,000	8,950	10,400	+16.2
Average	**8,630**	**11,080**	**+28.4**

6 Conclusion and future work

This paper proposed a Hybrid Anomaly–Rule–Pattern Detection Framework that has been implemented in order to achieve both persistent and stealthy intrusion detection in real-time streaming. The scheme includes three distinct methods of unsupervised anomaly detection, rule-based analysis and temporal pattern discov-ery, driven by a unified alert fusion module. Experimental comparisons on the CICIDS2017 and UNSW-NB15 datasets showed remarkable performance gains over the baseline SOAAPR method, which can achieve a precision of 91.3% and recall of 94.2%, along with an enhancement of persistent attack detection rate by 29.2% and decrease in detection latency by 36.7%. The modular design also demonstrated scalable performance, delivering more than 11,000 flows/s under moderate computation. In practice, the system offers a light-weight and easy-to-adapt detection solution deployable at large-scale enterprise, cloud and critical-infrastructure networks in order to detect emerging threats in a timely manner. The ensemble meta-alert fusion that is used not only improves robustness but also the interpretability by highlighting (differential) top contributing features for each alert. These properties make the model ready for immediate application when used in contemporary SOCs and real-time monitoring systems. However, there are several limitations to these approaches. The fixed sliding windows and the static thresholds may not be able to easily adapt to severe concept drift or attack. This paper included this, because although modularization guarantees scalability, running multiple containers can have higher resource overhead in case of very high amount of traffic.

These limitations will be dealt with in the future by designing and implementing adaptive and robust control techniques–like adaptive fuzzy control, neural adaptive backstepping controller, and nonlinear optimal controller–to dynamically modify decision thresholds as well as respond parameters. Reinforcement learning and meta-learning can be combined for continuously self-tuning models against novel attack behaviors. Then large scale deployment on distributed fog and edge will be looked upon to evaluate real time latency, reliability and interoperability with SIEM will be checked. In conclusion, the proposed approach provides a scalable and explainable direction for robust next generation intrusion detection that has the potential to protect against persistent and stealthy cyber threats in high-speed networks.

References

[1] Y. Chen et al., "A survey of large language models for cyber threat detection," *Computers & Security*, vol. 145, p. 104 016, 2024. DOI: 10.1016/j.cose.2023.104016.

[2] A. Ali, "Adaptive and context-aware authentication framework using edge ai and blockchain in future vehicular networks," *STAP Journal of Security Risk Management*, vol. 1, pp. 45–56, 2024.

[3] R. Vadisetty and A. Polamarasetti, "Generative ai for cyber threat simulation and defense," in *2024 12th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, 2024, pp. 272–279. DOI: 10.1109/ICCMA61985.2024.10746142.

[4] A. Pakmehr, A. Aßmuth, N. Taheri, and A. Ghaffari, "Ddos attack detection techniques in iot networks: A survey," *Cluster Computing*, vol. 27, no. 10, pp. 14 637–14 668, 2024. DOI: 10.1007/s10586-024-04492-1.

[5] O. M. A. Ali, R. A. Hamaamin, B. J. Youns, and S. W. Kareem, "Innovative machine learning strategies for ddos detection: A review," *UHD Journal of Science and Technology*, vol. 8, no. 2, pp. 38–49, 2024.

[6] P. Shukla, C. R. Krishna, and N. V. Patil, "Iot traffic-based ddos attacks detection mechanisms: A comprehensive review.," *Journal of Supercomputing*, vol. 80, no. 7, 2024. DOI: 10.1007/s11227-024-05712-w.

[7] H. M. Alqahtani and M. Abdullah, "A review on ddos attacks classifying and detection by ml/dl models.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 2, 2024.

[8] M. Al-Shareeda, L. Najm, A. Hassan, S. Mushtaq, and H. Ali, "Secure iot-based smart agriculture system using wireless sensor networks for remote environmental monitoring," *STAP Journal of Security Risk Management*, vol. 1, pp. 56–66, 2024.

[9] D. M. A. A. Afraji, J. Lloret, and L. Peñalver, "Deep learning-driven defense strategies for mitigating ddos attacks in cloud computing environments," *Cyber Security and Applications*, p. 100 085, 2025. DOI: 10.1016/j.csa.2024.100085.

[10] V. K. Ravindran, S. S. Ojha, and A. Kamboj, "A comparative analysis of signature-based and anomaly-based intrusion detection systems," *International Journal of Latest Technology in Engineering, Management & Applied Science*, vol. 14, no. 5, pp. 209–214, 2025.

[11] M. Almaayah and R. Sulaiman, "Cyber risk management in the internet of things: Frameworks, models, and best practices," *STAP Journal of Security Risk Management*, vol. 1, pp. 3–23, 2024.

[12] A. S. Abdullah, H. J. Sunil, and M. S. H. Nazmudeen, "A new model to evaluate signature and anomaly based intrusion detection in medical iot system using ensemble approach," *SN Computer Science*, vol. 6, no. 4, p. 347, 2025. DOI: 10.1007/s42979-025-02540-4.

[13] H. Satilmiş, S. Akleylek, and Z. Y. Tok, "A systematic literature review on host-based intrusion detection systems," *IEEE Access*, vol. 12, pp. 27 237–27 266, 2024. DOI: 10.1109/ACCESS.2024.3362070.

[14] H. Albinhamad, A. Alotibi, A. Alagnam, M. Almaiah, and S. Salloum, "Vehicular ad-hoc networks (vanets): A key enabler for smart transportation systems and challenges," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 4–15, 2025.

[15] N. Mishra and S. Mishra, "A review of machine learning-based intrusion detection system.," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024. DOI: 10.4108/eetiot.1234.

[16] M. Alshinwan, A. G. Memon, M. C. Ghanem, and M. Almaayah, "Unsupervised text feature selection approach based on improved prairie dog algorithm for the text clustering," *Jordanian Journal of Informatics and Computing*, vol. 2025, no. 1, pp. 27–36, 2025.

[17] M. Landauer, F. Skopik, B. Stojanović, A. Flatscher, and T. Ullrich, "A review of time-series analysis for cyber security analytics: From intrusion detection to attack prediction," *International Journal of Information Security*, vol. 24, no. 1, p. 3, 2025. DOI: 10.1007/s10207-024-00796-y.

[18] A. Boulkroune, F. Zouari, and A. Boubellouta, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," *Journal of Vibration and Control*, p. 10 775 463 251 320 258, 2025. DOI: 10.1177/10775463251320258.

[19] O. Aljumaiah, W. Jiang, S. R. Addula, and M. A. Almaiah, "Analyzing cybersecurity risks and threats in it infrastructure based on nist framework," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 12–26, 2025. DOI: 10.1016/j.jcsra.2024.100015.

[20] A. Boulkroune, S. Hamel, F. Zouari, A. Boukabou, and A. Ibeas, "Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities," *Mathematical Problems in Engineering*, vol. 2017, no. 1, p. 8 045 803, 2017. DOI: 10.1155/2017/8045803.

[21] F. Zouari, K. B. Saad, and M. Benrejeb, "Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems," *International Review on Modelling and Simulations*, vol. 5, no. 5, pp. 2075–2103, 2012.

[22] S. Otoom, "Risk auditing for digital twins in cyber physical systems: A systematic review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 22–35, 2025. DOI: 10.1016/j.jcsra.2024.100007.

[23] A. A. Almuqren, "Cybersecurity threats, countermeasures and mitigation techniques on the iot: Future research directions," *Journal of Cyber Security and Risk Auditing*, vol. 1, no. 1, pp. 1–11, 2025. DOI: 10.1016/j.jcsra.2024.100001.

[24] G. Rigatos, M. Abbaszadeh, B. Sari, P. Siano, G. Cuccurullo, and F. Zouari, "Nonlinear optimal control for a gas compressor driven by an induction motor," *Results in Control and Optimization*, vol. 11, p. 100 226, 2023. DOI: 10.1016/j.rico.2023.100226.

[25] F. Zouari, K. B. Saad, and M. Benrejeb, "Adaptive backstepping control for a class of uncertain single input single output nonlinear systems," in *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*, IEEE, 2013, pp. 1–6. DOI: 10.1109/SSD.2013.6528579.

[26] A. Presekal, A. Ştefanov, I. Semertzis, and P. Palensky, "Spatio-temporal advanced persistent threat detection and correlation for cyber-physical power systems using enhanced gc-lstm," *IEEE Transactions on Smart Grid*, 2024. DOI: 10.1109/TSG.2024.3356837.

[27] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting ddos attacks: A systematic review," *Soft Computing*, vol. 27, no. 18, pp. 13 039–13 075, 2023. DOI: 10.1007/s00500-023-08704-2.

[28] R. Buchta, G. Gkoktsis, F. Heine, and C. Kleiner, "Advanced persistent threat attack detection systems: A review of approaches, challenges, and trends," *Digital Threats: Research and Practice*, vol. 5, no. 4, pp. 1–37, 2024. DOI: 10.1145/3660198.

[29] D. Javaheri, S. Gorgin, J.-A. Lee, and M. Masdari, "Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives," *Information Sciences*, vol. 626, pp. 315–338, 2023. DOI: 10.1016/j.ins.2023.09.086.

[30] M. Jouhari, H. Benaddi, and K. Ibrahimi, "Efficient intrusion detection: Combining X^2 feature selection with cnn-bilstm on the unsw-nb15 dataset," in *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, IEEE, 2024, pp. 1–6. DOI: 10.1109/WINCOM61671.2024.10803550.

[31] R. Selvam and S. Velliangiri, "An improving intrusion detection model based on novel cnn technique using recent cic-ids datasets," in *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, IEEE, 2024, pp. 1–6. DOI: 10.1109/ICDCOT60930.2024.10797829.

[32] V. Kumar, V. Kumar, A. P. S. Bhadauria, J. Dixit, and A. Kumar, "Intrusion detection at the edge computing: A deep learning approach using the unsw-nb15 dataset," in *2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT)*, IEEE, 2025, pp. 220–224. DOI: 10.1109/CSNT60880.2025.10827759.

[33] M. Almehdhar et al., "Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 869–906, 2024. DOI: 10.1109/OJVT.2024.3382254.

[34] B. A. Mohammed et al., "Taxonomy-based lightweight cryptographic frameworks for secure industrial iot: A survey," *IEEE Internet of Things Journal*, 2025. DOI: 10.1109/JIOT.2025.3379502.

[35] R. Saadouni, C. Gherbi, Z. Aliouat, Y. Harbi, and A. Khacha, "Intrusion detection systems for iot based on bio-inspired and machine learning techniques: A systematic review of the literature," *Cluster Computing*, vol. 27, no. 7, pp. 8655–8681, 2024. DOI: 10.1007/s10586-024-04141-w.

[36] B. S. Ali et al., "Ics-ids: Application of big data analysis in ai-based intrusion detection systems to identify cyberattacks in ics networks," *The Journal of Supercomputing*, vol. 80, no. 6, pp. 7876–7905, 2024. DOI: 10.1007/s11227-024-06110-z.

[37] M. Alalhareth and S.-C. Hong, "Enhancing the internet of medical things (iomt) security with meta-learning: A performance-driven approach for ensemble intrusion detection systems," *Sensors*, vol. 24, no. 11, p. 3519, 2024. DOI: 10.3390/s24113519.

[38] Z. Que and C.-J. Lin, "One-class svm probabilistic outputs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 6244–6256, 2024. DOI: 10.1109/TNNLS.2023.3323821.

[39] H. Lu, "Evaluating the performance of svm, isolation forest, and dbscan for anomaly detection," in *ITM Web of Conferences*, EDP Sciences, vol. 70, 2025, p. 04 012. DOI: 10.1051/itmconf/20257004012.

[40] M. Akshay Kumaar, D. Samiayya, P. D. R. Vincent, K. Srinivasan, C.-Y. Chang, and H. Ganesh, "A hybrid framework for intrusion detection in healthcare systems using deep learning," *Frontiers in Public Health*, vol. 9, p. 824 898, 2022. DOI: 10.3389/fpubh.2022.824898.

[41] S. R. Khonde and V. Ulagamuthalvi, "Hybrid intrusion detection system using blockchain framework," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, p. 58, 2022. DOI: 10.1186/s13638-022-02168-x.

[42] A. R. Nair et al., "Hybrid deep learning framework-based intrusion detection system for the internet of things," in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, IEEE, 2024, pp. 1–6. DOI: 10.1109/ISCS61064.2024.10630354.

[43] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 3719–3746, 2022. DOI: 10.1007/s40747-022-00766-3.

[44] A. Guezzaz, M. Azrour, S. Benkirane, M. Mohy-Eddine, H. Attou, and M. Douiba, "A lightweight hybrid intrusion detection framework using machine learning for edge-based iiot security.," *International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 822–830, 2022. DOI: 10.34028/iajit/19/5/9.

[45] M. Heigl, E. Weigelt, A. Urmann, D. Fiala, and M. Schramm, "Exploiting the outcome of outlier detection for novel attack pattern recognition on streaming data," *Electronics*, vol. 10, no. 17, p. 2160, 2021. DOI: 10.3390/electronics10172160.