DOI: https://doi.org/10.15276/aait.09.2026.42
UDC 004.056:004.8

Hybrid rule-based and AI-driven architecture for intelligent cyber event routing and filtering

Artem A. Khalygov[1)]
ORCID: https://orcid.org/0009-0006-5465-4650; khalygovartem@gmail.com
Vladyslav I. Shkolnikov[2)]
ORCID: https://orcid.org/0000-0003-2041-9450; shkolnikov.v.i@navs.edu.ua
Olga A. Dmytriyeva[3), 4)]
ORCID: https://orcid.org/0000-0001-8921-8433; olga.dmytriyeva@iws.uni-stuttgart.de. Scopus Author ID: 57217432857
Bohdan S. Lysov[4)]
ORCID: https://orcid.org/0009-0007-7963-6958; bogukraine@gmail.com
Vira H. Huskova[4)]
ORCID: https://orcid.org/0000-0001-7637-201X; guskovavera2009@gmail.com. Scopus Author ID: 57219533864
Maksym O. Tishkov[4)]
ORCID: https://orcid.org/0009-0006-1741-9977; maksym.tishkov@gmail.com
[1)] Institute of Telecommunications and Global Information Space, 13, Chokolivs'kyi Blvd. Kyiv, 03186, Ukraine
[2)] National Academy of Internal Affairs, 1, Solom'yans'ka Square. Kyiv, 03035, Ukraine
[3)] Institute for modeling hydraulic and environmental systems, 5a, Pfaffenwaldring. Stuttgart, 70569, Germany
[4)] Igor Sikorsky Kyiv Polytechnic Institute, 37, Beresteiskyi Ave. Kyiv, 03056, Ukraine

ABSTRACT

Relevance. Security operations centers receive more alerts than analysts can review, many of them false positives. **The purpose of the article** is to develop and experimentally evaluate a hybrid architecture for cyber event routing that combines rules, machine learning, large language models and analyst validation. **Tasks:** to formalize a selective routing policy for these components and to compare the architecture with each component used alone. **Methods.** A rule-based gate resolves events with high or low rule scores; the remaining events are scored by a one-class support vector machine; a language model is consulted only for events near the decision threshold within an invocation budget; disagreements between layers are escalated to an analyst (simulated in the experiments). **Scientific novelty.** The invocation of each costly component is defined by explicit thresholds, an uncertainty band, a language model budget and analyst capacity, which makes the cost of intelligent processing controllable. **Practical significance.** The routing policy can serve as a configurable filtering stage in security operations centers. **Results.** On kernel process events and distributed file system logs, the architecture matched the best standalone detector on each dataset, although the best detector differed between the datasets (rules for kernel events, machine learning for file system logs). Letting the language model decide on uncertain log sessions significantly lowered the F1-score, while the language model alone performed well on kernel events but poorly on file system logs. **Conclusions.** No statistically significant gain over the best standalone detector was observed; on these data, the language model should support analysts rather than override the other layers.

Keywords: cybersecurity; rule-based systems; machine learning; large language models; anomaly detection; intelligent decision support; hybrid architecture; false positive reduction; human-in-the-loop validation

***For citation*: Khalygov A. A., Shkolnikov V. I., Dmytriyeva O. A., Lysov B. S., Huskova V. H., Tyshkov M. O. "Hybrid rule-based and AI-driven architecture for intelligent cyber event routing and filtering". *Applied Aspects of Information Technology.* 2026; Vol.9 No.4: 648–664. DOI: https://doi.org/10.15276/aait.09.2026.42**

INTRODUCTION

Digital transformation and the growth of network infrastructures have sharply increased the number of cyber events, security logs and alerts generated by information systems, security information and event management (SIEM) platforms, network sensors and monitoring tools. In critical infrastructure, banking and governmental systems, this volume may reach millions of records per day, and a substantial part of these events is noise or false positives that delay the detection of real incidents.

As a result, analysts in Security Operations Centers (SOCs) are overloaded with alerts, which slows incident response, increases investigation time and raises the risk of missing critical threats. This creates a need to automate the filtering, prioritization and contextual analysis of cyber events.

Rule-based systems remain an essential part of cybersecurity due to their speed, deterministic decisions and simple enforcement of security policies, and they are effective against known attacks formalized as IF–THEN rules. However, they adapt poorly to new threats, are difficult to

© Khalygov A., Shkolnikov V., Dmytriyeva O., Lysov B., Huskova V., Tishkov M., 2026

This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/deed.uk)

scale as the number of rules grows, and are ineffective against zero-day attacks, multi-stage scenarios and behavioral anomalies. Machine learning (ML) models enable anomaly detection, and large language models (LLMs) can explain incidents, generate recommendations and assist SOC analysts, but both involve high computational costs, latency, limited interpretability, dependence on data quality and, for LLMs, the risk of hallucinations.

A hybrid architecture can combine these strengths through multi-level filtering: a rule-based gate performs fast initial routing, ML modules assess behavioral risk, and LLM components provide context for complex incidents. **The purpose of this paper** is to develop and experimentally evaluate a hybrid architecture for cyber event routing and filtering that combines rule-based mechanisms, ML, LLMs and Human-in-the-Loop validation, with explicit control over when each costly component is invoked in SOC/SIEM environments.

LITERATURE REVIEW AND PROBLEM STATEMENT

Modern cybersecurity systems employ a wide range of approaches for cyber threat monitoring, detection, and response, including SIEM platforms, IDS/IPS systems, SOAR solutions, rule-based event analysis mechanisms, anomaly detection methods, and AI-driven SOC support systems [1], [3], [5], [19], [20], [22]. In recent years, considerable attention has also been devoted to the use of LLMs for cyber event analysis, explanation generation, and response automation [6], [11], [15].

SIEM systems provide centralized collection, aggregation, and correlation of security events; however, they often generate a large number of alerts and false positives, which overload SOC analysts [5], [14]. Security Orchestration, Automation and Response (SOAR) platforms automate incident response and orchestration processes, although their effectiveness largely depends on the quality of predefined playbooks and integrations [20]. IDS/IPS systems are effective in detecting known signature-based attacks but have limited capabilities for identifying novel and zero-day threats [19].

Rule-based approaches offer high processing speed, transparency, and ease of implementation, yet they demonstrate limited adaptability to emerging threats and complex behavioral anomalies [16]. In contrast, machine learning methods enable the detection of anomalous behavior and hidden patterns using statistical models and neural networks, but they require substantial computational resources and are affected by the data drift problem [1], [2], [4], [7], [8], [21].

A promising direction is the development of AI-based SOC solutions, where artificial intelligence methods are applied to threat prioritization, risk assessment, and decision support [14], [22]. LLMs play a particular role by providing contextual analysis of security logs, threat intelligence data, and incident reports, as well as generating explanations and response recommendations [6], [12], [15]. Nevertheless, their use is associated with high computational costs, processing latency, and the risk of hallucinations [10].

The analysis shows that none of the reviewed approaches alone is sufficiently effective for high-volume SOC/SIEM environments. This calls for hybrid architectures that combine the speed and determinism of rule-based mechanisms, the adaptability of ML models, and the contextual capabilities of LLMs for multi-level cyber event filtering.

Table 1. **Comparative characteristics of existing approaches**

Approach	Advantages	Limitations
SIEM	Centralized monitoring, event aggregation, data correlation	Large number of alerts, high false-positive rate, SOC analyst overload
SOAR	Automated incident response, orchestration workflows, tool integration	Dependence on predefined playbooks and integrations
IDS/IPS	Rapid detection of known attacks, real-time threat detection	Limited adaptability to novel and zero-day attacks
Rule-Based Detection	Decision transparency, high processing speed, compliance support	Inability to detect complex behavioral anomalies
Anomaly Detection / ML	Detection of anomalous behavior, behavioral analysis	High computational costs, data drift, limited interpretability
AI-based SOC	Intelligent threat prioritization and automated analysis	Explainability challenges and dependence on data quality
LLM for Cybersecurity	Contextual analysis, generation of explanations and recommendations	Hallucinations, high latency, and significant computational resource requirements

Source: **compiled by the authors**

In modern SOC/SIEM environments, cybersecurity systems process large volumes of heterogeneous events originating from network sensors, authentication systems, IDS/IPS solutions, endpoint protection platforms, operating system logs, and other telemetry sources. Cyber event streams are characterized by high velocity, heterogeneous data structures, and significant levels of noise, which complicate the timely detection of critical incidents and increase the workload of SOC analysts. Let the system event stream be represented as a set:

$$E = \{e_1, e_2, .., e_n\} \quad (1)$$

where e_i is an individual cyber event, and n is the number of events in the stream. Each cyber event may contain a set of characteristics describing network activity, user authentication parameters, behavioral features, temporal dependencies, and contextual attributes. In addition, events may include textual security log messages, information about system processes, network interactions, and indicators of compromise obtained from threat intelligence sources. The main task of the proposed system is the intelligent filtering, routing, and prioritization of cyber events using a hybrid orchestration architecture that combines rule-based mechanisms, ML-based anomaly detection modules, and LLM-based contextual analysis components.

For each event e_i, it is necessary to determine the optimal processing route:

$$R(e_i) \rightarrow \{Rule, ML, LLM, Monitor\} \quad (2)$$

where *Rule* denotes processing by a rule-based mechanism; *ML* denotes forwarding an event to the anomaly detection module; *LLM* represents contextual analysis using a Large Language Model; and *Monitor* corresponds to event logging and passive monitoring.

Event routing is performed based on the following factors:

– the rule-based score;

– the anomaly score;

– the preliminary risk and its distance from the decision threshold (uncertainty band);

– the LLM invocation budget;

– disagreement between the rule-based and ML layers.

To assess the criticality of cyber events, the system introduces an integrated risk indicator:

$$R(e_i) = \alpha \cdot S_{rule} + \beta \cdot S_{ml} + \gamma \cdot S_{context} \quad (3)$$

where S_{rule} is the score generated by the rule-based mechanism, S_{ml} is the anomaly score produced by the ML model, $S_{context}$ is the contextual score generated by the LLM module, and α, β, γ are weighting coefficients that determine the contribution of each component to the overall risk assessment. The rule-based score is calculated using predefined security policies, IOC matching results, and security threshold violations. The ML anomaly score reflects the degree of abnormality in user behavior, process execution, or network activity based on behavioral indicators and statistical characteristics. The contextual LLM score represents the contextual severity of an event derived from the analysis of incident descriptions, security logs, and threat intelligence feeds.

The event assessment and routing process can be represented as a multi-layer function:

$$\begin{aligned} F(E) = &\ Detection(i) + \\ &+ RiskAssessment(i) + \\ &+ Routing(i) + Rsponse(i) \end{aligned} \quad (4)$$

where $Detection(i)$ denotes the detection of suspicious or anomalous events, $RiskAssessment(i)$ represents risk evaluation, $Routing(i)$ determines the optimal processing module, and $Rsponse(i)$ corresponds to recommendation generation or automated response actions.

The objective is to detect cyber incidents in a timely manner while keeping the cost of intelligent processing under control. Let $\theta = (\theta H, \theta L, \alpha)$ denote the routing parameters (Eqs. (12)–(13)), B the maximum share of events forwarded to the LLM, and K the number of events per 1,000 that an analyst can review. The task is to find θ that maximizes the F1-score on validation data, provided that the shares of events processed by the LLM and by the analyst do not exceed B and K/1,000. False positives are thus reflected in the F1-score, while computational cost and analyst workload are bounded explicitly (Eq. (16)).

At the same time, the system should provide:

– adaptability to emerging attack types;

– system scalability;

– decision explainability;

– real-time event processing capabilities;

– effective integration with SIEM and SOAR platforms.

The research problem is to develop a hybrid intelligent architecture for cyber event routing and filtering that combines the processing speed of rule-based systems, the adaptability of ML models, and the contextual capabilities of LLMs to improve the effectiveness of modern SOC/SIEM environments.

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

RESEARCH AIM AND OBJECTIVES

The aim of this research is to develop a hybrid architecture for intelligent cyber event routing and filtering that combines rule-based mechanisms, machine learning models, LLMs and Human-in-the-Loop (HITL) validation under an explicit selective routing policy, and to verify experimentally whether such an architecture achieves detection quality at least equal to the best standalone component while keeping the use of resource-intensive components and the analyst workload under control.

To achieve this aim, the following objectives are set:

1. To analyze existing approaches to cyber event monitoring, detection, prioritization and response in SOC/SIEM environments (SIEM, SOAR, IDS/IPS, rule-based systems, ML methods and LLM-based solutions) and to identify their limitations with respect to adaptability, explainability, computational cost and false-positive rate.

2. To develop a hybrid architecture that integrates rule-based filtering, ML-based anomaly detection, LLM-based contextual analysis and HITL validation, and to formalize the selective routing policy: the thresholds of the Rule-Based Gate, the uncertainty band in which the LLM is invoked, the LLM invocation budget and the analyst capacity.

3. To evaluate the architecture experimentally on two open datasets with different event sources and to test the following hypotheses: (a) the hybrid architecture matches or exceeds the detection quality (Precision, Recall, F1-score, FPR) of the rule-based, ML-only and LLM-only variants evaluated under identical conditions; (b) the routing policy limits the share of events forwarded to the LLM and to the analyst to the specified budget and capacity; (c) analyst validation of escalated events reduces the false positive rate.

4. To assess the contribution and the computational cost of the LLM layer and to determine, on the basis of the results, the role in which the LLM should be used in the architecture.

MATERIALS AND METHODS

The proposed hybrid architecture for intelligent cyber event filtering is focused on multi-level processing of security events, taking into account their criticality, type, context, and risk level. The main idea of the architecture is the selective use of rule-based mechanisms, machine learning models, and LLM components depending on the complexity of each event. This makes it possible to avoid processing the entire event stream with resource-intensive AI modules and to ensure fast filtering of typical cases. In general, the architecture consists of the following main layers: Data Acquisition Layer, Rule-Based Layer, ML/LLM Layer, HITL Layer, Decision Layer, and Update and Feedback Layer. This structure enables the consistent organization of data collection, initial filtering, intelligent analysis, expert validation, decision generation, and further improvement of rules and models.

For a visual representation of the proposed architecture, it is appropriate to use the scheme shown in Fig. 1.

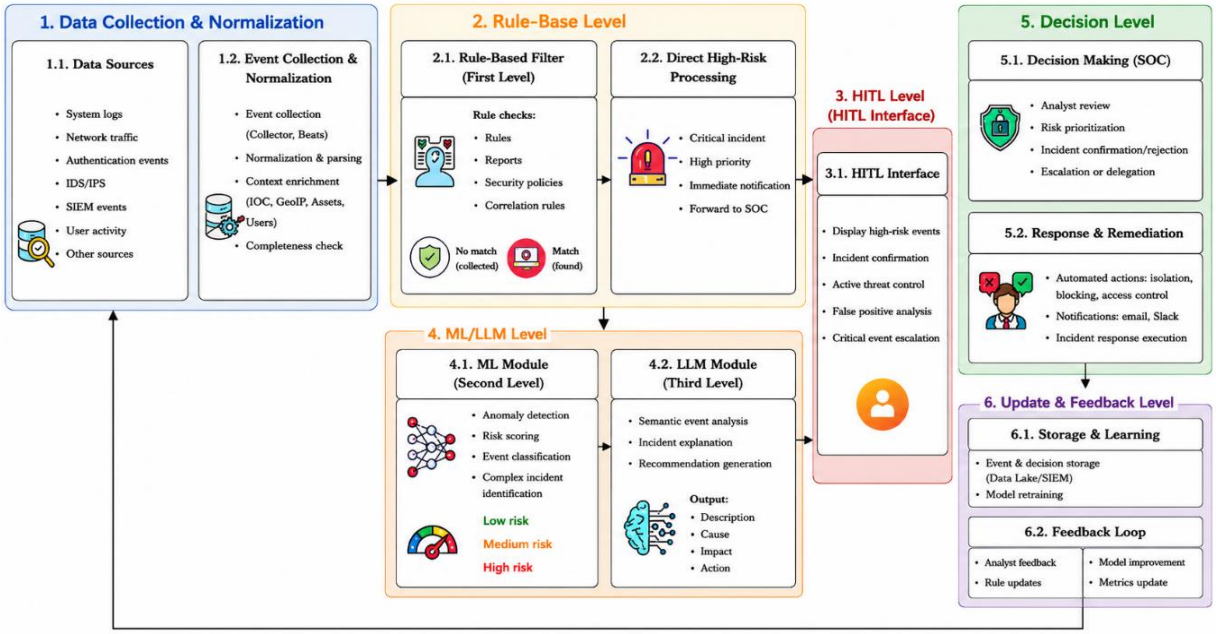


Fig. 1. **Generalized Architecture of the Hybrid Intelligent Cyber Event Filtering System**

Source: **compiled by the authors**

The proposed architecture enables the implementation of the selective AI usage principle, in which resource-intensive ML and LLM components are applied not to the entire event stream, but only to cases that require in-depth analysis. This reduces computational costs, shortens response time, improves decision explainability, and decreases the workload of SOC analysts.

Rule-Based Gate. The Rule-Based Gate is the basic layer of the proposed hybrid architecture and performs the functions of initial filtering, routing, and prioritization of cyber events. The main purpose of this module is to rapidly detect known threats, verify compliance with predefined security policies, and reduce the load on resource-intensive ML and LLM components by selecting only potentially critical or ambiguous events for further analysis. Unlike AI-oriented approaches, the Rule-Based Gate uses deterministic decision logic based on IF-THEN rules. This ensures system transparency, decision explainability, and precise control over response logic in accordance with organizational security policies [16]. Each rule can be explicitly described, verified, and modified by cybersecurity experts without the need to retrain models.

One of the key advantages of the Rule-Based Gate is its ability to process large event streams in real time. Checking predefined conditions and threshold values is significantly faster than ML- or LLM-based analysis, which makes it possible to process millions of events without substantial computational costs. This is especially important for SOC/SIEM environments, where delays in incident detection may lead to critical consequences.

An important component of the Rule-Based Gate is IOC matching, which verifies events against known Indicators of Compromise. Such indicators may include suspicious IP addresses, malicious domains, hash signatures, blocklisted processes, indicators from threat intelligence feeds, and patterns of known attacks [19].

When a match is detected, the system can automatically increase the event risk level or initiate immediate response actions.

In addition to IOC matching, the Rule-Based Gate implements threshold policies for controlling behavioral characteristics. These rules make it possible to detect suspicious activity based on predefined thresholds, such as the number of failed login attempts, sudden increases in network traffic, abnormal process activity, access attempts at unusual times, or suspicious user behavior.

In general, a rule can be represented as follows:

IF condition1 AND condition2 THEN action

For example, a brute-force detection rule may be defined as:

IF failed_logins > 10 AND time_window <= 5 minutes THEN mark_event = "Suspicious"

or rule for detecting suspicious access may be represented as:

IF new_device = true AND unusual_login_time = true THEN send_to = ML_Module

To assess event criticality, the Rule-Based Gate may generate an initial risk indicator:

$$S_{rule} = \sum w_i \cdot r_i \quad (5)$$

where r_i is the result of an individual rule evaluation, w_i is the weight coefficient representing the importance of the rule, and S_{rule} is the aggregated rule-based score.

The obtained score is subsequently used for routing events to the ML and LLM components of the system.

The Rule-Based Gate also plays an important role in ensuring compliance with information security requirements and enforcing access control policies. The use of predefined rules enables security policy enforcement, user activity auditing, compliance with regulatory requirements, and transparent documentation of the system's response logic. The Rule-Based Gate acts as a fast and controlled orchestration layer that provides deterministic event routing, low-latency filtering, IOC matching, and initial risk assessment. This approach optimizes the utilization of AI components, reduces the workload of SOC analysts, and enables efficient processing of high-volume cyber event streams.

ML-Based Anomaly Detection Module. The ML-based anomaly detection module is one of the key components of the proposed hybrid architecture. It is designed to identify atypical behavior, hidden patterns, and complex multi-stage attack scenarios that cannot be effectively detected by traditional rule-based mechanisms. The main tasks of this layer include behavioral analysis, anomaly detection, and risk scoring based on temporal dependencies, user behavior characteristics, and system activity.

Unlike rule-based systems based on deterministic logic, the ML module applies statistical and behavioral models capable of adapting to changes in data structure and detecting atypical scenarios even in the absence of predefined signatures or known indicators of compromise. This is particularly important for detecting zero-day

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

attacks, insider threats, privilege escalation, and complex multi-stage cyberattacks.

The operation of the ML module is based on behavioral analysis, within which the system forms a behavioral profile of a user, process, or endpoint device. For behavioral analysis and anomaly detection, the system uses a set of temporal and behavioral features that characterize user activity, process behavior, and network interactions. The main features used in the ML module are presented in Table 2.

Table 2. **Features used in the ML-based anomaly detection module**

Feature	Description
Failed logins	Number of failed login attempts within a defined time interval
Unusual login time	User login performed at an atypical time
Traffic spike	Sudden increase in network traffic volume or number of requests
Process frequency	Frequency of process launches or system command executions
Process entropy	Level of abnormality or uncertainty in process behavior
Network connections count	Number of network connections or active sessions
Mean inter-event time	Average time interval between events
Behavioral deviation	Deviation of current behavior from the typical user or system profile

Source: **compiled by the authors**

In general, the behavioral feature vector can be represented as:

$$X = \{x_1, x_2, \ldots, x_n\}, \quad (6)$$

where x_1 denotes failed logins, x_2 denotes unusual login time, x_3 denotes traffic spike, x_4 denotes process entropy, and x_5 denotes other behavioral characteristics.

To detect anomalies, the system applies anomaly detection algorithms that estimate the degree of deviation of current behavior from the normal system state.

The following methods can be used in this study:

– Isolation Forest,
– One-Class SVM,
– Autoencoder,
– LSTM Autoencoder,
– clustering-based methods,
– and statistical anomaly detection [1, 2, 4].

One promising approach is the use of unsupervised learning, where the model is trained only on normal behavior scenarios without requiring complete labeling of attack events. This makes it possible to detect new and previously unknown types of threats without the need to construct large labeled datasets [7], [8].

The output of the ML module is an anomaly score, which represents an integrated indicator of the abnormality level of an event or behavioral scenario.

In general, the anomaly score can be represented as:

$$AnomalyScore = 1 - \Pi(1 - s_i), \quad (7)$$

where s_i is the local anomaly score of an individual feature or event, and $AnomalyScore$ is the integrated abnormality level.

This approach makes it possible to account for the accumulation of several weak anomalies that may not individually exceed critical thresholds but collectively form a high risk of a cyber incident.

To assess the risk level, the ML module may also use the following function:

$$S_{ml} = f(x_1, x_2, \ldots, x_n), \quad (8)$$

where $x_1, \ldots, x_n$ denotes the same as in previous formula and S_{ml} represents the anomaly score or ML-based risk score.

Based on the obtained value, the system determines the event risk level and forwards the results to the Decision Support Layer or the LLM Context Layer for further contextual analysis and recommendation generation.

The ML module provides adaptive real-time analysis of temporal dependencies and behavioral patterns, enabling the detection of atypical action sequences, anomalous user activity, suspicious changes in network or system behavior, as well as signs of privilege escalation or lateral movement within the infrastructure. At the same time, the use of ML models is associated with high computational costs, dependence on data quality, changes in data structure over time, and the risk of false positives. Therefore, in the proposed architecture, the ML module is applied selectively only to events that cannot be clearly classified by the rule-based layer and require in-depth behavioral analysis.

LLM Module. The LLM module, or Large Language Model Context Layer, is a component of the proposed hybrid architecture designed to perform contextual analysis of complex cyber

events, interpret the outputs of ML modules, and generate explanations and recommendations for SOC analysts. A key feature of this layer is its ability to process unstructured or semi-structured data that are difficult to analyze using traditional rule-based or classical ML approaches.

Unlike deterministic systems, Large Language Models enable contextual analysis of security events by taking into account incident descriptions, event logs, SIEM alerts, SOC reports, threat intelligence feeds, and other cybersecurity context sources. This makes it possible to identify complex relationships between events, analyze multi-stage attack scenarios, and interpret behavioral anomalies detected by the ML module.

One of the main tasks of the LLM module is incident explanation, namely the generation of explanations regarding the causes of suspicious activity or a potential cyber incident. Based on telemetry data, anomaly scores, and contextual indicators, the model can generate a textual description of an incident, identify a possible attack type, assess its criticality level, and describe potential consequences for the information system.

For example, based on detected behavioral anomalies, the system may generate the following explanation:

"Potential brute-force attack with privilege escalation attempt detected based on repeated failed logins, unusual access time, and abnormal process activity."

The use of LLMs makes it possible to transform a set of technical events and telemetry logs into a context understandable to SOC analysts, thereby simplifying incident analysis and reducing response time. Another important application of Large Language Models is the automatic generation of cyber incident response recommendations. Based on the results of rule-based and behavioral analysis, the LLM module may suggest response actions such as blocking user accounts, isolating devices, launching automated response playbooks, performing additional user verification, creating SOC tasks, escalating events to CERT/CSIRT teams, or initiating manual analyst review.

In general, the operation of the LLM module can be represented as:

$$LLM_{output} = ContextAnalysis + \\ +Explanation + Recommendation, \quad (9)$$

where $ContextAnalysis$ denotes event context analysis, $Explanation$ represents the textual explanation of the incident, and $Recommendation$ corresponds to the recommended response action.

An important advantage of the LLM component is SOC assistance in processing large numbers of incidents. The model can summarize incident information, generate concise event descriptions, explain the reasons for increased risk levels, and support cyber event prioritization. In addition, the LLM component can analyze threat intelligence data [12] and generate response recommendations for SOC analysts.

This approach reduces the workload of SOC analysts and accelerates decision-making in high-volume SOC/SIEM environments. At the same time, the use of LLMs is associated with risks of incorrect or incomplete explanations, inaccurate incident prioritization, significant computational costs, processing delays, and potential leakage of sensitive data [10], [11]. Therefore, in the proposed architecture, the LLM module is applied selectively only to complex, ambiguous, or text-based cyber events, while its outputs may be additionally validated by a SOC analyst at the HITL layer.

HITL (Human-in-the-Loop). One of the key features of the proposed hybrid architecture is the use of the Human-in-the-Loop (HITL) concept, which involves integrating a SOC analyst into the decision-making process for critical or ambiguous cyber events. Unlike fully automated AI systems, the HITL approach combines the speed and scalability of intelligent algorithms with expert assessment by a cybersecurity specialist.

The need for HITL is determined by the fact that even modern ML and LLM models may generate false positives, incorrectly interpret complex attack scenarios, or produce inaccurate recommendations due to hallucination risks and limited model explainability [13, 17]. In critical environments, including SOC/SIEM systems, critical infrastructure facilities, banking systems, and governmental information platforms, incorrect automated response may lead to blocking legitimate users, disrupting business processes, or reducing service availability.

The main task of the HITL layer is analyst validation, which involves verifying the outputs of rule-based, ML, and LLM modules before critical actions are executed. At this layer, the SOC analyst reviews system outputs, confirms or rejects incidents, adjusts event risk levels and priorities, and may initiate additional analysis or approve or cancel automated response actions.

In general, the HITL validation process can be represented as:

$$Decision_{final} = AI_{decision} + Human_{validation}, \quad (10)$$

where $AI_{decision}$ is the result of rule-based, ML-based, or LLM-based analysis, $Human_{validation}$ represents the confirmation or correction of the decision by a SOC analyst, and $Decision_{final}$ is the final system decision.

One of the most important advantages of HITL is false positive reduction. In modern SOC environments, a significant proportion of alerts may be related to legitimate user activity or specific business processes that ML models may interpret as anomalies.

For example, the system may classify an unusual user login as potentially compromised access:

IF anomaly_score > threshold AND new_country = true
THEN generate_high_risk_alert

However, a SOC analyst may verify that the user is on a business trip, after which the incident will be classified as a false positive.

Another important function of the HITL layer is the supervised confirmation of critical decisions. In the proposed architecture, automated response actions may be allowed only for low-risk or well-known attack scenarios, whereas high-risk events require mandatory analyst approval before execution.

The conditions under which an event is escalated to the analyst are formalized in Eq. (15). Another important function of HITL is the adaptive learning loop, which uses SOC analyst decisions to further improve the system. The results of human validation can be stored and used for updating rule-based policies, retraining ML models, adjusting risk scoring, optimizing routing logic, and improving prompt engineering for the LLM module. In general, the adaptive learning loop can be represented as:

$$Model(t+1) = Model(t) + Analyst_{feedback}, \quad (11)$$

where $Model(t)$ is the current state of the model, $Analyst_{feedback}$ represents the results of SOC analyst validation, and $Model(t+1)$denotes the updated model. This approach enables the system to adapt to changes in user behavior, emerging attack types, and the specific characteristics of a particular SOC/SIEM environment. Thus, the HITL layer performs the functions of analyst validation, false positive reduction, supervised confirmation, and adaptive learning loop implementation. It ensures the controlled and explainable use of AI components in cyber incident monitoring and response processes. The integration of Human-in-the-Loop increases the reliability of the hybrid architecture, reduces the risks of incorrect decisions, and provides a balance between automation and expert control.

Selective routing policy. All component scores are normalized to [0, 1]: S_{rule} is computed by Eq. (5), S_{ml} by Eq. (8), and S_{context} is the confidence returned by the LLM module (Eq. 9). The route of an event e in Eq. (2) is defined as

$$\begin{gathered} Route(e) = Rule, if\ S_{rule} \geq \theta_H; \\ Monitor, If\ S_{rule} < \theta_L; \\ LLM, if\ \theta_L \leq S_{rule} < \theta_H \wedge R' - \theta_R < \delta; \text{ML}, \\ \text{otherwise} \end{gathered} \quad (12)$$

where θ_H and θ_L are the upper and lower thresholds of the Rule-Based Gate, and R' is the preliminary risk computed from the deterministic and ML components, i.e. Eq. (3) with $\gamma = 0$:

$$R'(e) = \frac{\alpha S_{rule} + \beta S_{ml}}{\alpha + \beta}. \quad (13)$$

An event routed to ML is labelled positive if $R' \geq \theta_R$. The LLM is invoked only inside the uncertainty band $|R' - \theta_R| < \delta$, whose width is bounded by the LLM invocation budget B:

$$\delta = max\{\delta' : \frac{1}{N_{val}} \Sigma e[R'(e) - \theta_R \vee \delta'] \leq B\}, \quad (14)$$

so that at most a share B of validation events is forwarded to the LLM. An event is escalated to the analyst if the rule-based and ML layers disagree or if it was forwarded to the LLM:

$$\begin{gathered} H(e) = 1[(1[S_{rule} \geq 0.5] \neq 1[S_{ml} \geq \theta_A]) \wedge \\ \wedge\ Route(e) \neq Monitor] \vee \\ \vee\ 1[Route(e) = LLM], \end{gathered} \quad (15)$$

where S_{ml} is defined only for events that reached the ML module. Escalated events are ordered by $|R' - \theta_R|$ and limited to K per 1,000 events (analyst capacity). The parameters $\theta = (\theta H, \theta L, \alpha)$, $\beta = 1 - \alpha$, are selected on the validation subset E_{val} of size N_{val} by solving the constrained problem

$$\begin{gathered} \theta^* = argmax\{\theta \in \Theta\} \\ F1_{val}(\theta), subject\ to\ \rho_{LLM}(K) \leq B, \\ \rho_{HITL}(\theta) \leq K/1000 \end{gathered} \quad (16)$$

where θ is the parameter grid (see Experimental Setup), $\rho_{LLM}(K) = (1/N_{val})\Sigma_{e \in E_{val}} 1[Route(e) = LLM]$ is the share of events forwarded to the LLM,

and $\rho_{HITL}(\theta)$ is the share of events reviewed by the analyst. The first constraint is satisfied by construction through the choice of δ in Eq. (14), and the second through limiting the analyst queue K events per 1,000. The thresholds θR and θA are the F1-optimal thresholds of R' and S_{ml} on the validation subset.

The ML module is thus invoked only between θ_L and θ_H, the LLM only within the budget B, and the analyst only within the capacity K.

Algorithm 1. **Selective routing of a cyber event**

Input: event *e*; parameters θ_H, θ_L, θ_R, θ_A, δ, α, β, K
Output: label *y*, risk *R*, explanation, recommended action

```
compute S_rule(e) - Eq. (5)
if S_rule ≥ θ_H then
    y ← 1; R ← S_rule; route ← Rule - ML is not invoked
else if S_rule < θ_L then
    y ← 0; R ← S_rule; route ← Monitor - event is logged
else
    compute S_ml(e) - Eq. (8)
    R′ ← (α·S_rule + β·S_ml)/(α + β) - Eq. (13)
    y ← 1[R′ ≥ θ_R]; R ← R′; route ← ML
    if |R′ − θ_R| < δ then − uncertainty band, Eq. (14)
        (y_LLM, S_context, explanation, action) ← LLM(e, S_rule, S_ml) - Eq. (9)
        y ← y_LLM; R ← S_context; route ← LLM
if H(e) = 1 then - Eq. (15)
    add e to the analyst queue with priority |R′ − θ_R|
analyst reviews top K queued events per 1,000: y ← decision - Eq. (10)
risk level ← High if y = 1; Medium if e is queued and not yet reviewed; Low otherwise
return y, R, explanation, action
```

Algorithm 1 implements the selective AI usage principle: the Rule-Based Gate resolves events with high or low rule scores without invoking intelligent components, the ML module processes only events between θ_L and θ_H, the LLM is called only for events in the uncertainty band within the budget *B*, and the analyst reviews only disagreements and LLM-processed events within the capacity *K*. The parameter values used in the experiments are given in Table 3.

EXPERIMENTAL SETUP

The experimental evaluation compared the proposed hybrid architecture with standalone rule-based and machine learning approaches under identical conditions and quantified how the selective routing policy distributes events between the layers of the architecture. All experiments were implemented in Python 3.12 using scikit-learn 1.8, pandas 3.0 and SciPy 1.17 and executed on a single virtual CPU core (Intel Xeon, 2.8 GHz) with 3 GB of RAM.

Datasets. Two open datasets [18] representing two types of event sources were used. The BETH dataset [23], [25] contains kernel-level process events collected from honeypot hosts. The version used in this study provides seven numerical fields for each event (process, thread and parent process identifiers, user identifier, mount namespace, number of arguments and return value) and the binary label *sus*, which marks suspicious activity. The HDFS log dataset [9], [24] was used in the form of its publicly available sample of 100,000 log messages, which contains 7,940 block sessions, 313 of which are labelled as anomalous at the block level. The log messages are parsed into 29 event templates; each block session was represented by the vector of template counts, and the raw message sequence of each session was retained as textual input for the LLM layer. The unit of evaluation was a single event for BETH and a block session for HDFS, and results are reported separately for each dataset.

Data splits. For BETH, the predefined training (763,144 events, of which 1,269 are suspicious), validation (188,967; 786) and test (188,967; 171,459, i.e. 90.7%) subsets were used without modification. HDFS block sessions were ordered by the timestamp of their first message and split chronologically into training (60%), validation (20%) and test (20%) subsets of 4,764, 1,588 and 1,588 sessions containing 154, 68 and 91 anomalies, respectively; the chronological split prevents information leakage from future events. Anomaly detectors were trained only on normal events of the training subsets. The validation subsets were used exclusively for selecting the ML model, thresholds and weighting coefficients, and all reported metrics were computed on the test subsets (Table 3).

Features. For BETH, the features from Table 2 were derived from the available fields using the standard preprocessing for this dataset. They include

binary indicators for a system process and a system parent process (IDs 0-2), a system user (user ID below 1,000), the host mount namespace, and negative (failed operation) and positive return values, as well as the number of arguments. Because this version of the dataset has no timestamps, the temporal features from Table 2 (unusual login time and mean inter-event time) could not be computed. For HDFS, each session was represented by its log-scaled template count vector.

Rule-Based Gate configuration. The rules were defined in advance from domain knowledge and were not fitted to the labels. For BETH, two rules were used: R1, activity of a non-system user (user identifier ≥ 1,000; weight 0.6), and R2, a failed system operation (negative return value; weight 0.4). For HDFS, three rules were used: R1, exception or interruption events in the block write pipeline (weight 0.4); R2, abnormal transfer events, namely an empty packet, a change of the block file offset or a redundant block registration (weight 0.2); and R3, an incomplete write pipeline, i.e. fewer received blocks than initiated transfers or fewer block map updates than received blocks (weight 0.4). The score S_{rule} was computed by Eq. (5). IOC matching was not applied, because neither dataset contains indicator fields. In the rule-based only variant, an event was classified as positive if $S_{\text{rule}} \geq \theta_{\text{RB}}$, where θ_{RB} was selected on the validation subset.

ML module configuration. Two unsupervised detectors were considered as candidates: Isolation Forest (200 trees, 256 samples per tree) and One-Class SVM (RBF kernel, $\nu = 0.01$, γ = scale). Each was trained on at most 20,000 normal training events. The candidate with the higher F1-score on the validation subset was selected; for both datasets this was One-Class SVM (validation F1-score 0.728 against 0.092 for Isolation Forest on BETH, and 0.750 against 0.105 on HDFS). The raw detector output s was mapped to the anomaly score $S_{\text{ml}} = 1 - \exp(-\max(0, s - m)/\sigma)$, where m and σ are the median and standard deviation of the outputs on normal training events. In the ML only variant, an event was classified as positive if $S_{\text{ml}} \geq \theta_{\text{A}}$, with θ_{A} selected on the validation subset.

Hybrid routing configuration. The hybrid variant followed Eqs. (12)-(16) and Algorithm 1 with the LLM budget B = 10 %. The grid Θ in Eq. (16) covered $\theta H \in \{0.6, 0.8, 1.0, >1\}$, $\theta L \in \{0, 0.2, 0.4, 0.6\}$ and $\alpha \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$. The selected values are given in Table 3.

Human-in-the-loop simulation. Because no SOC analysts took part in the experiments, analyst validation was simulated. Events were escalated according to Eq. (15) with the capacity K = 10 per 1,000 events. The simulated analyst returned the true label with probability $1 - \varepsilon$, where $\varepsilon \in \{0.05, 0.10\}$ models analyst error. To avoid overestimating the human contribution, results are reported both with and without the HITL layer.

LLM layer. The LLM layer was implemented with the model claude-sonnet-4-6 (Anthropic), accessed through the Anthropic Messages API from a browser-based execution environment (claude.ai artifact runtime) in September–October 2026, with the reasoning effort set to low. In this environment, the provider's default sampling settings are applied and the temperature cannot be fixed. For each event, the model received the raw sequence of log messages of the block session (HDFS) or the seven numerical fields of the kernel event described in the Datasets paragraph (BETH). Events were submitted in batches of 15 HDFS sessions or 50 BETH events per request. A fixed prompt instructed the model to return, for each event, a binary label, a confidence in [0, 1] used as Scontext, a short explanation and a recommended action in JSON format. In the hybrid variant, the prompt additionally contained Srule and Sml, and the label returned by the LLM replaced the decision based on R′; the weight γ in Eq. (3) was therefore not tuned. The LLM-only variant and, for comparison, the other variants were evaluated in a single run on stratified samples of 300 test events per dataset (17 anomalous HDFS sessions and 272 suspicious BETH events). The prompt, the identifiers of the sampled events and the raw model responses are available from the corresponding author upon request.

Evaluation protocol. Classification quality was assessed with Precision, Recall, F1-score, False Positive Rate (FPR) and ROC-AUC; ROC-AUC was computed from the continuous scores S_{rule}, S_{ml} and R' of the respective variants. Every configuration was run with five random seeds, which affect the sub-sampling of training data, the Isolation Forest and the analyst simulation, and results are reported as mean ± standard deviation. Differences between the hybrid architecture and the best standalone variant were tested with McNemar's test (α = 0.05), and 95 % confidence intervals of the F1-score were estimated by bootstrap resampling of the test subset (1,000 resamples). Latency was measured as the wall-clock time of processing a single event on 2,000 test events, and the median and the 95th percentile are reported.

Table 3. **Experimental datasets and selected parameters**

Parameter	BETH	HDFS
Unit of evaluation	Kernel event	Block session
Training/ validation / test size	763,144 / 188,967 / 188,967	4,764 / 1,588 / 1,588
Positive events in test subset	171,459 (90.7%)	91 (5.7%)
Normal events for ML training	20,000 (sampled)	4,610
Rules (weights)	R1 0.6; R2 0.4	R1 0.4; R2 0.2; R3 0.4
θ_{RB} (rule-based only)	0.6	0.2
ML model	One-Class SVM	One-Class SVM
θ_A (ML only)	0.991–0.996	0.999
θ_H / θ_L	0.6 / 0.6	0.6 / 0
α / β	0.2 / 0.8	0.2 / 0.8
θ_R	– (no events reach ML)	0.80
LLM budget B / actual share	10% / 0%	10% / 2.3%
HITL capacity	10 per 1,000 events	10 per 1,000 events

***Source*: compiled by the authors**

RESEARCH RESULTS

Table 4 reports the results on the test subsets. On BETH, the Rule-Based Gate alone achieved a Precision of 1.000, a Recall of 0.941 and an F1-score of 0.969 without false positives. This result is largely explained by the structure of the *sus* label: every event generated by a non-system user is labelled as suspicious, so rule R1 captures most positive events. One-Class SVM reached a ROC-AUC of 0.962, but its thresholded performance was unstable across runs (F1-score 0.788 ± 0.375), because its threshold was selected on a validation subset with only 0.4 % positive events. The hybrid architecture reproduced the rule-based result. The grid search selected $\theta_L = \theta_H = 0.6$, so 85.3 % of the test events were labelled High by the gate, the remaining 14.7% were logged, and no event reached the ML, LLM or HITL layers (Table 5).

On HDFS, the rules detected only 46.2 % of anomalous sessions (F1-score 0.632), because most anomalies are not accompanied by exception events. The ML only variant achieved an F1-score of 0.748 (Precision 0.982, Recall 0.604). The hybrid architecture routed 98.7 % of sessions through the ML module and reproduced the ML result (F1-score 0.748,95 % CI 0.667-0.821). Simulated validation of 0.9 % of sessions by an analyst eliminated the remaining false positives (FPR 0.000) at the cost of a small decrease in Recall (0.596 at $\varepsilon = 0.05$ and 0.589 at $\varepsilon = 0.10$). McNemar's test found no discordant predictions between the hybrid architecture and the best standalone variant on either dataset ($p = 1.0$).

Table 4. **Comparison of cyber event processing variants on the test subsets**

Metric	Rule-Based	ML	Hybrid	Hybrid + HITL ($\varepsilon = 0.05$)
BETH				
Precision	1.000	0.998	1.000	1.000
Recall	0.941	0.744*	0.941	0.941
F1-score	0.969	0.788*	0.969	0.969
FPR	0.000	0.004	0.000	0.000
ROC-AUC	0.970	0.962	0.970	0.970
HDFS				
Precision	1.000	0.982	0.982	1.000
Recall	0.462	0.604	0.604	0.596
F1-score	0.632	0.748	0.748	0.747
FPR	0.000	0.001	0.001	0.000
ROC-AUC	0.731	0.808	0.808	0.808

Means over five runs; standard deviations do not exceed 0.01, except * (Recall ± 0.382, F1-score ± 0.375)

***Source:* compiled by the authors**

Without the LLM layer, the per-event processing time was below 5 ms for all variants (Table 5). The median latency of rule evaluation was 0.33 ms for BETH and 2.48 ms for HDFS, where the rules are evaluated over a whole session and ML scoring took 0.18 and 0.17 ms, respectively. The mean latency of the hybrid architecture was 0.36 ms and 3.07 ms. With the LLM layer, the mean latency of the hybrid architecture on HDFS rose to about 32 ms, because 2.3% of the sessions required an LLM call.

Table 6 reports the results of the LLM layer. On HDFS, 37 sessions (2.3 %) fell into the uncertainty band; 36 of them were anomalous, and the hybrid architecture without the LLM classified 35 of them correctly. The LLM overturned ten of these correct decisions and corrected one error, so adding the LLM layer reduced Recall from 0.604 to 0.505 and the F1-score from 0.748 to 0.667 on the full test

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

Table 5. **Routing of test events and processing latency**

Indicator	BETH	HDFS
Labelled High by Rule-Based Gate	85.3 %	1.3 %
Logged without further analysis	14.7 %	0.0 %
Processed by ML module	0.0 %	98.7 %
Forwarded to LLM layer	0.0 %	2.3 %
Escalated to analyst	0.0 %	0.9 %
Rule evaluation latency, median / p95, ms	0.33 / 0.51	2.48 / 4.16
ML scoring latency, median / p95, ms	0.18 / 0.64	0.17 / 0.30
Hybrid mean latency without LLM, ms	0.36	3.07
LLM time per event, median / p95, s	0.76 / 0.79 (LLM-only)	1.27 / 2.14
Hybrid mean latency with LLM, ms	0.36	32.3

Source: **compiled by the authors**

subset; the difference is significant by McNemar's test ($p = 0.012$). In the LLM-only variant, the LLM reached an F1-score of 0.114 on the HDFS sample (Precision 0.070, FPR 0.233, ROC-AUC 0.673), far below the ML module (0.741), whereas on the BETH sample it reached 0.954 (Precision 1.000, Recall 0.912, ROC-AUC 0.979), close to the rule-based variant (0.976). The median LLM processing time was 1.09-1.27 s per HDFS session and 0.76 s per BETH event, about three orders of magnitude above rule evaluation and ML scoring.

DISCUSSION OF RESULTS

The experiments did not confirm that the hybrid architecture improves detection quality over the best standalone component. The observed property is different: on each dataset, the architecture matched the best standalone variant, although this variant differed between the datasets (rules on BETH, ML on HDFS). The routing policy, tuned only on validation data, thus selected a suitable processing path for each event source without a manual choice of the detection method, while the simulated analyst validation removed the residual false positives on HDFS.

Table 6. **Results of the LLM Layer and of the LLM-Only Variant**

Variant	Precision	Recall	F1-score	FPR
HDFS, full test subset (1,588 sessions)				
Hybrid without LLM	0.982	0.604	0.748	0.001
Hybrid + LLM	0.979	0.505	0.667	0.001
HDFS, sample (300 sessions, 17 anomalous)				
Rule-Based	1.000	0.235	0.381	0.000
ML	1.000	0.588	0.741	0.000
Hybrid	1.000	0.588	0.741	0.000
LLM only	0.070	0.294	0.114	0.233
BETH, sample (300 events, 272 suspicious)				
Rule-Based	1.000	0.952	0.976	0.000
ML	0.996	0.919	0.956	0.036
Hybrid	1.000	0.952	0.976	0.000
LLM only	1.000	0.912	0.954	0.000

Single run with the pipeline of the first random seed; LLM: claude-sonnet-4-6.
Source: **compiled by the authors**

The LLM layer did not improve detection. On HDFS, the uncertainty band contained mainly sessions on which the rules were silent and the ML module flagged an anomaly, and the LLM tended to accept these sessions as normal, because HDFS anomalies are expressed by missing or irregular sequences of events rather than by explicit error messages in the text. On BETH, where suspicious activity is visible in the event fields, the LLM alone performed close to the rules. These results suggest that, at least for log data of this kind, the LLM should explain and enrich decisions for the analyst rather than override the decisions of the deterministic and ML layers, and that the weight γ should be tuned on validation data instead of letting the LLM label decide.

The study has several limitations. First, the BETH label is strongly determined by the user identifier, which makes the rule-based baseline unusually strong and limits the conclusions that can be drawn from this dataset. Second, the validation and test subsets of BETH differ substantially (0.4 % and 90.7 % positive events), so the routing parameters could not learn to use the ML module where the rules fail; on BETH the policy bypassed the ML module entirely. Third, selectivity depended on the dataset: on HDFS almost all sessions passed through the ML module. Fourth, the LLM layer was evaluated with a single model and in a single run on

samples of 300 events. Because the execution environment did not allow the temperature to be fixed, the model outputs may vary between runs, and their stability was not assessed. Events were processed in batches, so the decision for an event may have been influenced by other events in the same request. Token usage was not recorded, and therefore the computational cost of the LLM layer is characterised only by its processing time (Table 5). Finally, the HITL layer was simulated rather than performed by SOC analysts. Future experiments should use the full BETH dataset with temporal fields, the full HDFS dataset, real analyst feedback, several LLMs and a validation-tuned weight γ.

CONCLUSIONS

This paper proposes a hybrid architecture for cyber event filtering that combines rule-based mechanisms, machine learning methods, Large Language Models and analyst-based validation under an explicit selective routing policy with score thresholds, an uncertainty band and an LLM invocation budget. Experiments on the BETH and HDFS datasets showed that the architecture matched the best standalone detector on each dataset (F1-score 0.969 and 0.748), whereas the best standalone detector differed between the datasets, and that simulated analyst validation of 0.9% of HDFS sessions reduced the false positive rate to zero. A statistically significant improvement over the best standalone detector was not observed; letting the LLM decide on uncertain HDFS sessions lowered the F1-score to 0.667, and the LLM alone was competitive only on BETH. The practical significance of the approach lies in a single configurable pipeline that adapts the processing path to heterogeneous event sources in SOC/SIEM environments, CERT/CSIRT centers, banking systems, e-government platforms and critical infrastructure facilities. Future research will focus on using the LLM for explanation rather than decision-making, tuning its weight on validation data, experiments on full datasets with temporal features and on real SOC/SIEM data, the integration of Explainable AI [13], [17], dynamic risk assessment models, and automatic rule updating based on threat intelligence data.

USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence tools were used in this study in two roles. First, the large language model claude-sonnet-4-6 (Anthropic) was used as the LLM context layer of the proposed architecture and as the LLM-only baseline; its configuration and outputs are described in the Experimental setup section and reported in Table 6. Second, Claude Opus 5.5 (Anthropic) was used as an assistant for preparing and checking the experimental code, the data processing and statistical analysis scripts, and for editing the manuscript text. All experimental results were obtained by executing the published code on open datasets, and all results, computations, interpretations and the text of the manuscript were checked and proofread by the authors, who take full responsibility for the content of the article.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Financing: Research was carried out without financial support

Data Availability: The study uses two publicly available datasets. The BETH dataset [23] is available at https://www.kaggle.com/datasets/katehighnam/beth-dataset. The HDFS log dataset (HDFS_v1) is distributed as part of the Loghub collection [24] at https://github.com/logpai/loghub, with the full raw logs archived at Zenodo (https://zenodo.org/records/8196385). The source code is available via the link https://github.com/artemkhalygov/hrb

Use of artificial intelligence tools: The use of artificial intelligence tools in this study is described in the section “Use of Artificial Intelligence”: the model claude-sonnet-4-6 was used as a component of the proposed architecture, and Claude Opus 5.5 was used to assist with code preparation and text editing. All results were verified by the authors

REFERENCES

1. Lunardi, W. T., Lopez, M. A. & Giacalone, J.-P. “ARCADE: Adversarially regularized convolutional autoencoder for network anomaly detection”. *IEEE Transactions on Network and Service Management.* 2023; 20 (2): 1305–1318, https://www.scopus.com/pages/publications/85144793913. DOI: https://doi.org/10.1109/TNSM.2022.3229706.

2. Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M. & Portmann, M. “FlowTransformer: A transformer framework for flow-based network intrusion detection systems”. *Expert Systems with Applications.* 2024; 241: 122564, https://www.scopus.com/pages/publications/85183574703. DOI: https://doi.org/10.1016/j.eswa.2023.122564.

3. He, K. & Kim, D. D., Asghar M. R. “Adversarial machine learning for network intrusion detection systems: A comprehensive survey”. *IEEE Communications Surveys & Tutorials.* 2023; 25(1): 538–566, https://www.scopus.com/pages/publications/85147206191. DOI: https://doi.org/10.1109/COMST.2022.3233793.

4. Schummer, P., del Rio, A., Serrano, J., Jimenez, D., Sánchez, G. & Llorente, Á. “Machine learning-based network anomaly detection: Design, implementation, and evaluation”. *AI.* 2024; 5(4): 2967–2983. DOI: https://doi.org/10.3390/ai5040143.

5. Ban, T., Takahashi, T., Ndichu, S. & Inoue, D. “Breaking alert fatigue: AI-assisted SIEM framework for effective incident response”. *Applied Sciences.* 2023; 13 (11): 6610, https://www.scopus.com/pages/publications/85163079208. DOI: https://doi.org/10.3390/app13116610.

6. Zhang, J., Bu, H., Wen, H., Liu, Y., Fei, H., Xi, R., Li, L., Yang, Y., Zhu, H. & Meng, D. “When LLMs meet cybersecurity: A systematic literature review”. *Cybersecurity.* 2025; 8: 55. DOI: https://doi.org/10.1186/s42400-025-00361-w.

7. Zavaliy, T. I. & Shakhovska, N. B. “Online anomaly detection in correlated data streams using robust Kalman filtering”. *Applied Aspects of Information Technology.* 2025; 8 (4): 377–385. DOI: https://doi.org/10.15276/aait.08.2025.24.

8. Shyaa, M. A., et al. “Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems”. *Engineering Applications of Artificial Intelligence.* 2024; 137: 109143. DOI: https://doi.org/10.1016/j.engappai.2024.109143.

9. He, S., He, P., Chen, Z., Yang, T., Su, Y. & Lyu, M. R. “A survey on automated log analysis for reliability engineering”. *ACM Computing Surveys.* 2021; 54 (6): 130. DOI: https://doi.org/10.1145/3460345.

10. Ji, Z., Lee, N., Frieske, R., et al. “Survey of hallucination in natural language generation”. *ACM Computing Surveys.* 2023; 55 (12): 248, https://www.scopus.com/pages/publications/85152599648. DOI: https://doi.org/10.1145/3571730.

11. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z. & Zhang, Y. “A survey on Large Language Model (LLM) security and privacy: The good, the bad, and the ugly”. *High-Confidence Computing.* 2024; 4 (2): 100211. DOI: https://doi.org/10.1016/j.hcc.2024.100211.

12. Rahman, M. R., Mahdavi-Hezaveh, R. & Williams, L. “What are the attackers doing now? Automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey”. *ACM Computing Surveys.* 2023; 55 (12). DOI: https://doi.org/10.1145/3571726.

13. Gaspar, D., Silva, P. & Silva, C. “Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron”. *IEEE Access.* 2024; 12: 30164–30175, https://www.scopus.com/pages/publications/85186107633. DOI: https://doi.org/10.1109/ACCESS.2024.3368377.

14. Tariq, S., Baruwal Chhetri, M., Nepal, S. & Paris, C. “Alert fatigue in security operations centres: Research challenges and opportunities”. *ACM Computing Surveys.* 2025; 57 (9): 1–38, https://www.scopus.com/pages/publications/105008550436. DOI: https://doi.org/10.1145/3723158.

15. Zhou, W., Zhu, X., Han, Q.-L., Li, L., Chen, X., Wen, S. & Xiang, Y. “The security of using Large Language Models: A survey with emphasis on ChatGPT”. *IEEE/CAA Journal of Automatica Sinica.* 2025; 12 (1): 1–26. DOI: https://doi.org/10.1109/JAS.2024.124983.

16. Pascoe, C., Quinn, S. & Scarfone, K. “The NIST Cybersecurity Framework (CSF) 2.0”. *NIST Cybersecurity White Paper (CSWP) 29. Gaithersburg, MD: National Institute of Standards and Technology*. 2024. p. 1–32. DOI: https://doi.org/10.6028/NIST.CSWP.29.

17. Houda, Z. A. E., Brik, B. & Khoukhi, L. “‘Why should I trust your IDS?’: An explainable deep learning framework for intrusion detection systems in internet of things networks”. *IEEE Open Journal of the Communications Society.* 2022; 3: 1164–1176. DOI: https://doi.org/10.1109/OJCOMS.2022.3188750.

18. Ring, M., Wunderlich, S., Scheuring, D., Landes, D. & Hotho, A. “A survey of network-based intrusion detection data sets”. *Computers & Security.* 2019; 86: 147–167. DOI: https://doi.org/10.1016/j.cose.2019.06.005.

19. Li, Z., Fang, W., Zhu, C., Song, G. & Zhang, W. “Toward deep learning based intrusion detection system: A survey”. *Proceedings of the 6th International Conference on Big Data Engineering (BDE ‘24).* 2024. p. 25–32. DOI: https://doi.org/10.1145/3688574.3688578.

20. Nguyen, P., Dautov, R., Song, H., Rego, A., Iturbe, E., Rios, E., Sagasti, D., Nicolas, G., Valdés V. & Mallouli, W. “Towards smarter security orchestration and automatic response for CPS and

IoT". *IEEE International Conference on Cloud Computing Technology and Science (CloudCom).* 2023: 298–302, https://www.scopus.com/pages/publications/85189630699.
DOI: https://doi.org/10.1109/CloudCom59040.2023.00055.

21. Alotaibi, A. & Rassam, M. A. "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense". *Future Internet.* 2023; 15 (2): 62, https://www.scopus.com/pages/publications/85148899504. DOI: https://doi.org/10.3390/fi15020062.

22. Vielberth, M., Böhm, F., Fichtinger, I. & Pernul, G. "Security operations center: A systematic study and open challenges". *IEEE Access.* 2020; 8: 227756–227779, https://www.scopus.com/pages/publications/85184700667.
DOI: https://doi.org/10.1109/ACCESS.2020.3045514.

23. Highnam, K., Arulkumaran, K., Hanif, Z. & Jennings, N. R. "BETH Dataset: Real cybersecurity data for anomaly detection research". *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.* 2021. – Available from: http://www.gatsby.ucl.ac.uk/~balaji/udl2021/accepted-papers/UDL2021-paper-033.pdf. – [Accessed: 10.09.2024]

24. Zhu, J., He, S., He, P., Liu, J. & Lyu, M. R. "Loghub: a large collection of system log datasets for AI-driven log analytics". *IEEE 34th International Symposium on Software Reliability Engineering (ISSRE).* 2023. p. 355–366. DOI: https://doi.org/10.1109/ISSRE59848.2023.00071.

25. Doğanay, H. A., Orman, A. & Dener, M. "Big data visualization for cyber security: BETH Dataset". *El-Cezerî Journal of Science and Engineering.* 2022; 9 (4): 1572–1582. DOI: https://doi.org/10.31202/ecjse.1209586.

Received 04.09.2026
Received after revision 26.10.2026
Accepted 30.10.2026

DOI: https://doi.org/10.15276/aait.09.2026.42
УДК 004.056:004.8

Гібридна архітектура інтелектуальної маршрутизації та фільтрації кіберподій на основі rule-based та AI-компонентів

Халигов Артем Азимович[1)]
ORCID: https://orcid.org/0009-0006-5465-4650; khalygovartem@gmail.com
Школьніков Владислав Ігорович[2)]
ORCID: https://orcid.org/0000-0003-2041-9450; shkolnikov.v.i@navs.edu.ua
Дмитрієва Ольга Анатоліївна[3), 4)]
ORCID: https://orcid.org/0000-0001-8921-8433; olga.dmytriyeva@iws.uni-stuttgart.de; Scopus Author ID: 57217432857
Лисов Богдан Сергійович[4)]
ORCID: https://orcid.org/0009-0007-7963-6958; bogukraine@gmail.com
Гуськова Віра Геннадіївна[4)]
ORCID: https://orcid.org/0000-0001-7637-201X; guskovavera2009@gmail.com. Scopus Author ID: 57219533864
Тішков Максим Олегович[4)]
ORCID: https://orcid.org/0009-0006-1741-9977; maksym.tishkov@gmail.com
1) Інститут телекомунікацій і глобального інформаційного простору НАН України, 13, Чоколівський бульвар Київ, 03186, Україна
[2)] Національна академія внутрішніх справ, Солом'янська площа, 1, Київ, 03035, Україна
[3)] Інститут моделювання гідравлічних та екологічних систем, Пфаффенвальдрінг, 5а. Штутгарт, 70569, Німеччіна
[4)] Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського" проспект Берестейський, 37. Київ, 03056, Україна

АНОТАЦІЯ

Актуальність. Центри операцій безпеки отримують більше сповіщень, ніж аналітики здатні опрацювати, і значна частина з них є хибними. **Мета статті** – розробити та експериментально оцінити гібридну архітектуру маршрутизації кіберподій, що поєднує правила, машинне навчання, великі мовні моделі та валідацію аналітиком. **Завдання:** формалізувати політику вибіркової маршрутизації між цими компонентами та порівняти архітектуру з кожним компонентом, застосованим окремо. **Методи.** Шлюз на основі правил розв'язує події з високою або низькою оцінкою правил; решту подій оцінює

одноклasовий метод опорних векторів; мовну модель залучають лише для подій поблизу порогу рішення в межах бюджету звернень; розбіжності між рівнями передають аналітику (в експериментах його симульовано). **Наукова новизна.** Залучення кожного ресурсоємного компонента визначено явними порогами, зоною невизначеності, бюджетом мовної моделі та пропускною здатністю аналітика, що робить вартість інтелектуальної обробки керованою. **Практична значущість.** Політику маршрутизації можна використовувати як налаштовуваний етап фільтрації в центрах операцій безпеки. **Результати.** На подіях процесів ядра та журналах розподіленої файлової системи архітектура на кожному наборі даних досягла результату найкращого окремого детектора, хоча найкращий детектор на наборах даних був різним (правила для подій ядра, машинне навчання для журналів файлової системи). Передавання мовній моделі права ухвалювати рішення щодо невизначених сесій журналів суттєво знизило F1-міру. Мовна модель, застосована окремо, показала високі результати на подіях ядра, але низькі на журналах файлової системи. **Висновки.** Статистично значущого виграшу порівняно з найкращим окремим детектором не виявлено; на цих даних мовна модель має підтримувати аналітиків, а не переглядати рішення інших рівнів.

Ключові слова: кібербезпека; системи на основі правил; машинне навчання; великі мовні моделі; виявлення аномалій; інтелектуальна підтримка прийняття рішень; гібридна архітектура; зниження хибнопозитивних спрацювань; експертна валідація рішень

ABOUT THE AUTHORS



Artem A. Khalygov - Postgraduate Student, Institute of Telecommunications and Global Information Space, National Academy of Sciences of Ukraine, 13, Chokolivs'kyi Blvd. Kyiv, 03186, Ukraine
ORCID: https://orcid.org/0009-0006-5465-4650; khalygovartem@gmail.com
Research field: Time series; cybersecurity; critical infrastructure protection; intelligent decision support systems; machine learning; cyber event filtering and routing

Халигов Артем Азимович - аспірант, Інститут телекомунікацій і глобального інформаційного простору НАН України, Чоколівський бульвар, 13. Київ, 03186, Україна
Галузь досліджень: часові ряди; кібербезпека; захист критичної інфраструктури; інтелектуальні системи підтримки прийняття рішень; машинне навчання; фільтрація та маршрутизація кіберподій



Vladyslav I. Shkolnikov - PhD, Associate Professor, National Academy of Internal Affairs, 1, Solom'yans'ka Square, Kyiv, 03035, Ukraine
ORCID: https://orcid.org/0000-0003-2041-9450; shkolnikov.v.i@navs.edu.ua

Research field: Cybersecurity; information security; cybercrime investigation; digital forensics; intelligent systems.
Школьніков Владислав Ігорович – кандидат юридичних наук, доцент. Національна академія внутрішніх справ, Солом'янська площа, 1. Київ, 03035, Україна
Галузь досліджень: кібербезпека; інформаційна безпека; розслідування кіберзлочинів; цифрова криміналістика; інтелектуальні системи



Olga A. Dmytriyeva - Doctor of Engineering Sciences, Professor, Laureate of the State Prize of Ukraine in Science and Technology; Scientific Assistant, Institute for Modelling Hydraulic and Environmental Systems, Department of Stochastic Simulation and Safety Research for Hydrosystems, Institute for modelling hydraulic and environmental systems, 5a, Pfaffenwaldring. Stuttgart, 70569, Germany
ORCID: 0000-0001-8921-8433; olga.dmytriyeva@iws.uni-stuttgart.de. Scopus Author ID: 57217432857
Research field: Mathematical modelling; stochastic simulation; numerical methods; computational mathematics; modelling of complex systems; safety and reliability analysis of hydrosystems

Дмитрієва Ольга Анатоліївна - доктор технічних наук, професор, лауреат Державної премії України в галузі науки і техніки; науковий співробітник Інституту моделювання гідравлічних та екологічних систем, відділ Стохастичного моделювання та дослідження безпеки гідросистем. Штутгарт, 70569, Німеччина
Галузь досліджень: математичне моделювання; стохастичне моделювання; чисельні методи; обчислювальна математика; моделювання складних систем; аналіз безпеки та надійності гідросистем



Bohdan S. Lysov - Postgraduate Student. Igor Sikorsky Kyiv Polytechnic Institute, 37, Beresteiskyi Ave., Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0007-7963-6958; bogukraine@gmail.com
Research field: Time series; cybersecurity; critical infrastructure protection; intelligent decision support systems; machine learning; cyber event filtering and routing

Лисов Богдан Сергійович - аспірант, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», проспект Берестейський, 37. Київ, 03056, Україна
Галузь досліджень: часові ряди; кібербезпека; захист критичної інфраструктури; інтелектуальні системи підтримки прийняття рішень; машинне навчання; фільтрація та маршрутизація кіберподій



Vira H. Huskova - PhD, Associate Professor, Artificial Intelligence Department. Igor Sikorsky Kyiv Polytechnic Institute, 37, Beresteiskyi Ave. Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0000-0001-7637-201X; guskovavera2009@gmail.com. Scopus Author ID: 57219533864
Research field: Artificial intelligence; machine learning; intelligent decision support systems; time series analysis; risk assessment

Гуськова Віра Геннадіївна - кандидат технічних наук, доцент кафедри Штучного інтелекту. Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", пр. Берестейський, 37, Київ, 03056, Україна.
Галузь досліджень: штучний інтелект; машинне навчання; інтелектуальні системи підтримки прийняття рішень; аналіз часових рядів; оцінювання ризиків



Maksym O. Tishkov - PhD Student, Educational and Research Institute for Applied System Analysis. Igor Sikorsky Kyiv Polytechnic Institute, 37, Beresteiskyi Ave., Kyiv, 03056, Ukraine
ORCID: https://orcid.org/0009-0006-1741-9977;maksym.tishkov@gmail.com
Research field: Mathematical modeling; time series forecasting; machine learning; intelligent decision support systems

Тішков Максим Олегович - аспірант, Навчально-науковий інститут Прикладного системного аналізу. Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", пр. Берестейський, 37, Київ, 03056, Україна.
Галузь досліджень: математичне моделювання; прогнозування часових рядів; машинне навчання; інтелектуальні системи підтримки прийняття рішень

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)