Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

AMALLM: AN ADAPTIVE MULTI-AGENT LARGE LANGUAGE MODEL FRAMEWORK FOR CYBERSECURITY THREAT DETECTION AND CONTROLLED RESPONSE

Amna Tariq[*1], Zohaib Dahri[2], Sarfraz Ahmed[3], Junaid Ahmad[4], Muhammad Tanveer[5], Waqar[6], Ayesha Tariq[7]

*[*1,7]Department of Computer Science, Quaid-i-Azam University Islamabad, Pakistan*
[2]Department of Cybersecurity, Quaid-e-Awam University of Engineering Science & Technology Nawabshah, Pakistan
[3]Department of Computer Science, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, Pakistan
[4]Department of Computer Science, MNS-UET University, Multan, Pakistan
[5]Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan
[6]Department of Computer Science, Quaid-e-Awam University of Engineering Science & Technology, Nawabshah, Pakistan

[*1]amnatariqdev@gmail.com, [2]zohaibahmeddahri554@gmail.com, [3]sarfrazahmeddahri95@gmail.com, [4]junaid24@gmail.com, [5]tanveermm277@gmail.com, [6]dahriwaqar110@gmail.com, [7]alshamalik.121@gmail.com

DOI: https://doi.org/10.5281/zenodo.24186282

Keywords
Large Language Models (LLMs); Multi-Agent Systems; Autonomous Cybersecurity; Threat Detection; Adaptive Cyber defence ; Incident Response

Article History
Received: 17 January 2026
Accepted: 02 March 2026
Published: 16 March 2026

Copyright @Author
Corresponding Author: *
Amna Tariq

Abstract

The increasing complexity, volume, and adaptability of cyber threats are exposing fundamental limitations in conventional intrusion-detection systems and single-agent artificial intelligence architectures, particularly in contextual reasoning, evidence verification, and autonomous response. This study proposes an Adaptive Multi-Agent Large Language Model (AMALLM) framework that integrates threat detection, threat-intelligence verification, contextual risk assessment, response planning, and response validation within a unified detection-to-response architecture. Unlike single-agent approaches, AMALLM decomposes cybersecurity decision-making among five specialized agents coordinated through structured inter-agent communication, shared contextual memory, cross-agent verification, and feedback-driven adaptation. The framework is evaluated using CICIDS2017 and UNSW-NB15 and compared with conventional machine-learning, deep-learning, and single-agent LLM baselines across detection effectiveness, false-positive rates, response appropriateness, robustness, and computational efficiency. In the illustrative evaluation, AMALLM achieved 96.4% accuracy and a 95.9% macro F1-score on CICIDS2017, with a false-positive rate of 2.8%, while achieving 94.1% accuracy and a 93.5% macro F1-score on UNSW-NB15. Compared with the single-agent LLM, AMALLM improved macro F1-score by 2.6 percentage points and reduced the false-positive rate by 1.8 percentage points. The framework selected appropriate defensive responses in 94.7% of simulated incidents, while ablation analysis indicated that threat verification and response validation were particularly important contributors to system reliability. These improvements were accompanied by increased computational

latency and token consumption, highlighting a trade-off between decision reliability and operational efficiency. Overall, the findings demonstrate the potential of specialized multi-agent LLM architectures to move cybersecurity beyond isolated threat classification toward evidence-grounded, adaptive, and controlled detection-to-response decision-making. The framework provides a foundation for developing more autonomous cybersecurity systems while retaining human oversight and explicit authorization boundaries for uncertain or high-impact defensive actions.

1. Introduction

The rapid expansion of interconn ected digital infrastructures, cloud computing, Internet of Things (IoT) devices, and artificial intelligence (AI)-enabled services has substantially increased both the scale and complexity of the contemporary cybersecurity landscape. Modern organizations generate enormous volumes of network traffic, system logs, security alerts, and threat-intelligence data, making timely identification of malicious activities increasingly challenging. At the same time, cyberattacks are becoming more adaptive and sophisticated, ranging from malware and phishing to advanced persistent threats, ransomware, and previously unseen attack patterns. Conventional signature-based security mechanisms remain effective for known threats but can struggle when attack characteristics evolve or when adversarial behavior does not match predefined rules [1].

Machine learning (ML) and deep learning (DL) have consequently become important components of intrusion detection and cybersecurity analytics because of their ability to learn complex patterns from large-scale security data [2].Despite these advances, conventional ML and DL based cybersecurity systems have important limitations. Many models operate as task-specific classifiers and provide limited contextual reasoning about why an event is malicious [3], how different security events are related, or what response should follow a detected incident. Their performance can also deteriorate when operational data differ substantially from the distributions represented during training. Furthermore, security operations center frequently depend on human analysts to correlate alerts, consult threat intelligence, prioritize incidents, and determine appropriate mitigation measures [4].

The resulting combination of increasing alert volumes and limited analyst capacity creates a need for intelligent systems capable of moving beyond isolated threat classification toward contextual interpretation and decision support [5].Large language models (LLMs) offer a potentially significant development in this direction. Transformer-based models can process and reason over large quantities of structured and unstructured information, allowing them to interpret security reports, system events, vulnerability descriptions, threat-intelligence documents, and analyst instructions through a common natural-language interface[6, 7].

More recent LLMs have demonstrated capabilities involving reasoning, tool use, planning, and interaction with external information sources, creating opportunities for their integration into cybersecurity workflows. However, relying on a single general-purpose LLM for autonomous security operations introduces significant challenges. An individual model may need to simultaneously detect anomalies, interpret threat intelligence, assess risk, determine response priorities, and formulate mitigation actions [8].

Such broad responsibilities can increase reasoning complexity and expose the system to hallucination, inconsistent decisions, prompt-based attacks, and unreliable responses in security-critical environments [9]. Multi-agent systems provide an alternative architecture in which complex tasks are decomposed among specialized autonomous agents. Rather than requiring one LLM to perform the complete cybersecurity workflow, multiple agents can assume distinct responsibilities and collaboratively exchange information [10].

Recent developments in LLM-based multi-agent systems indicate that role specialization, communication, reflection, and collaborative reasoning can improve the handling of complex

tasks compared with isolated model interactions [11, 12]. Within cybersecurity, this concept creates the possibility of assigning specialized agents to threat detection, threat-intelligence retrieval and verification, risk assessment, response planning, and action validation. An observed anomaly could therefore be examined by several agents before a response is authorized, potentially reducing dependence on a single model's judgment [13].

An important research gap remains between general-purpose LLM multi-agent architectures and adaptive cybersecurity systems capable of supporting an integrated detection-to-response lifecycle. Existing cybersecurity approaches frequently concentrate on individual functions such as intrusion detection, malware classification, vulnerability analysis, or threat-intelligence processing, while emerging LLM-agent research often evaluates general reasoning and software-oriented tasks rather than end-to-end cyber defence. Consequently, there remains a need to investigate whether specialized LLM agents can collaborate dynamically, incorporate feedback from previous decisions, adapt their behavior to changing threat conditions, and produce responses that are both timely and explainable. This issue becomes particularly important for emerging or previously unseen attacks, where static rules and historical signatures may provide insufficient evidence for decision-making [14].

To address this gap, this study proposes an adaptive multi-agent LLM framework for autonomous cybersecurity threat detection and response. The framework decomposes the security workflow among cooperating agents responsible for threat detection, threat-intelligence verification, contextual risk assessment, response planning, and response validation, while an adaptive feedback mechanism `enables information from previous detections and responses to influence subsequent decisions. The study investigates whether this collaborative architecture can improve threat-detection accuracy, reduce false alarms, strengthen contextual reasoning, and support faster and more appropriate incident responses compared with conventional approaches and a single-agent LLM configuration.

The principal contribution is therefore not merely the application of an LLM to cybersecurity, but the development and empirical evaluation of a coordinated, adaptive agent architecture that integrates detection, reasoning, verification, and response within a unified cybersecurity workflow. By examining detection performance, response effectiveness, computational efficiency, robustness, and the contribution of individual agents, the research seeks to establish the practical potential and limitations of multi-agent LLMs as a foundation for more autonomous and trustworthy cybersecurity operations.

2. Literature Review

2.1 Large Language Models in Cybersecurity

The integration of artificial intelligence into cybersecurity has progressed from rule-based detection toward machine learning, deep learning, and, more recently, large language models (LLMs). Traditional learning-based security systems have demonstrated strong capabilities in intrusion detection, malware classification, phishing identification, and anomaly recognition [2], but their effectiveness is often constrained by dependence on predefined features, labelled training data, and specific threat distributions [15, 16].

LLMs extend these capabilities by processing natural-language security reports, vulnerability descriptions, threat-intelligence feeds, source code, system logs, and analyst queries within a unified reasoning environment. Their contextual learning capabilities can support security knowledge extraction, incident interpretation, vulnerability assessment, and decision support without requiring a separately trained model for every textual task [17].

Recent advances in instruction-following and reasoning models have further increased interest in LLM-assisted cybersecurity. Transformer architectures provide the technical foundation for capturing long-range relationships within complex sequential information [18], while large-scale language modelling has demonstrated strong few-shot and zero-shot generalization capabilities [19].

Cybersecurity represents a particularly demanding application because an apparently plausible but incorrect model output can lead to

inappropriate defensive decisions. Hallucination, outdated knowledge, adversarial prompting, limited interpretability, and uncertainty in model reasoning therefore remain important barriers to fully autonomous deployment [20]. These limitations motivate architectures in which LLM outputs can be independently verified rather than accepted as authoritative decisions.

2.2 Multi-Agent Systems for Cybersecurity

Multi-agent systems distribute complex problems among autonomous entities that perceive information, make decisions, communicate, and coordinate toward shared objectives. Their decentralized structure is particularly relevant to cybersecurity because modern environments contain heterogeneous devices, distributed networks, multiple attack surfaces, and rapidly changing threat conditions [21].

Earlier multi-agent security research demonstrated that distributed agents could contribute to intrusion monitoring and collaborative detection by sharing locally observed information instead of relying exclusively on centralized analysis [22].

LLMs have substantially expanded this concept by enabling agents to communicate through natural language and assume specialized functional roles. Generative agents can maintain contextual information and use observations to guide subsequent behavior [23], while frameworks such as AutoGen have demonstrated how multiple LLM-based agents can communicate and cooperate to perform complex tasks [24].

Agent specialization is especially attractive for cybersecurity because detection, threat verification, risk analysis, and incident response require different types of reasoning. However, simply adding multiple agents does not guarantee better security performance. Incorrect information may propagate between agents, redundant reasoning can increase computational cost, and poorly designed coordination mechanisms may introduce delays. Cybersecurity-oriented multi-agent architectures therefore require explicit mechanisms for role allocation, verification, conflict resolution, and controlled decision execution [25].

2.3 LLM-Based Threat Detection and Analysis

Cybersecurity threat detection has traditionally been formulated as a classification or anomaly-detection problem. Machine-learning and deep-learning methods identify statistical differences between benign and malicious activities and have been extensively applied to network intrusion detection [26].

Although these methods can achieve high predictive performance on benchmark datasets, evaluation under realistic operational conditions remains challenging because network behavior changes over time and novel attacks may differ from patterns available during model training [27].

LLMs offer a complementary approach by incorporating semantic and contextual reasoning into threat analysis. Instead of evaluating a security event exclusively as a numerical feature vector, an LLM-based agent can combine observed events with vulnerability information, attack descriptions, previous incidents, and contextual security knowledge [28]. Tool-using reasoning approaches further demonstrate that language models can alternate between reasoning and external actions, allowing additional evidence to be retrieved before reaching a conclusion [27].

For threat detection, such capabilities could enable a suspicious event to be assessed against multiple evidence sources rather than classified through a single static decision boundary. , the reliability of LLM-based detection remains dependent on evidence quality, prompt design, model consistency, and mechanisms for validating generated conclusions.

2.4 Autonomous Cybersecurity Response

Detection represents only the first stage of effective cyber defence. Once malicious activity has been identified, a security system must estimate severity, determine affected resources, prioritize the incident, select appropriate mitigation measures, and evaluate whether an intervention could disrupt legitimate operations [29].

Existing automated security mechanisms can execute predefined actions such as blocking addresses, isolating endpoints, disabling compromised accounts, or generating alerts [30]. However, these mechanisms generally operate

according to predefined rules and may have limited flexibility when encountering unfamiliar attack conditions.

Agentic LLMs provide an opportunity to introduce contextual planning into this process. A response-planning agent could evaluate alternative defensive actions while another agent independently examines their operational consequences before execution [31]. Such separation is important because unrestricted autonomous actions can create additional security and availability risks. Human oversight, permission boundaries, validation mechanisms, and auditable decision trails therefore remain important components of responsible autonomous cyber defence [32].

2.5 Limitations of Existing Approaches and Research Gap

The literature reveals three interconnected limitations. First, conventional ML and DL approaches primarily emphasize detection performance and generally provide limited integration between threat identification, contextual reasoning, and adaptive response. Second, LLM-based cybersecurity research increasingly demonstrates useful capabilities for security analysis, but individual LLMs remain vulnerable to hallucination, inconsistent reasoning, and adversarial manipulation.

Third, contemporary multi-agent LLM frameworks demonstrate collaborative problem-solving, yet comparatively limited research has empirically evaluated specialized agent collaboration across the complete cybersecurity detection-to-response workflow.

This study addresses this intersection by developing an adaptive multi-agent architecture in which specialized agents separately perform threat detection, intelligence verification, risk assessment, response planning, and validation. Unlike a static collection of agents, the proposed framework incorporates feedback from previous interactions to adjust subsequent threat assessment and response decisions. The resulting research therefore examines not only whether LLMs can contribute to cybersecurity analysis, but whether coordinated specialization, cross-agent verification, and adaptive feedback can provide measurable improvements over single-agent and conventional security approaches.

3. Proposed Adaptive Multi-Agent LLM Framework

3.1 Overall System Architecture

The proposed Adaptive Multi-Agent Large Language Model (AMALLM) framework is designed as a coordinated cybersecurity architecture that integrates threat detection, intelligence verification, risk assessment, response planning, and action validation within a continuous decision cycle [33].

Rather than assigning the complete security workflow to a single LLM, the framework decomposes cybersecurity reasoning into specialized tasks performed by interacting agents. This design follows the broader principle that complex reasoning can be decomposed into intermediate reasoning and action stages [34, 35], while multi-agent communication enables specialized agents to collaborate toward a shared objective [35].

The framework receives heterogeneous cybersecurity information, including network-flow records, security logs, intrusion alerts, vulnerability information, endpoint events, and available threat-intelligence evidence. After preprocessing, the information enters a sequential but feedback-enabled workflow consisting of five principal agents: a Threat Detection Agent (TDA), Threat Intelligence and Verification Agent (TIVA), Risk Assessment and Reasoning Agent (RARA), Response Planning and Execution Agent (RPEA), and Response Validation Agent (RVA). A shared contextual memory stores validated observations, previous incidents, agent decisions, response outcomes, and feedback. The resulting architecture can be represented as:

Security Data → TDA → TIVA → RARA → RPEA → RVA → Feedback/Memory → Adaptation

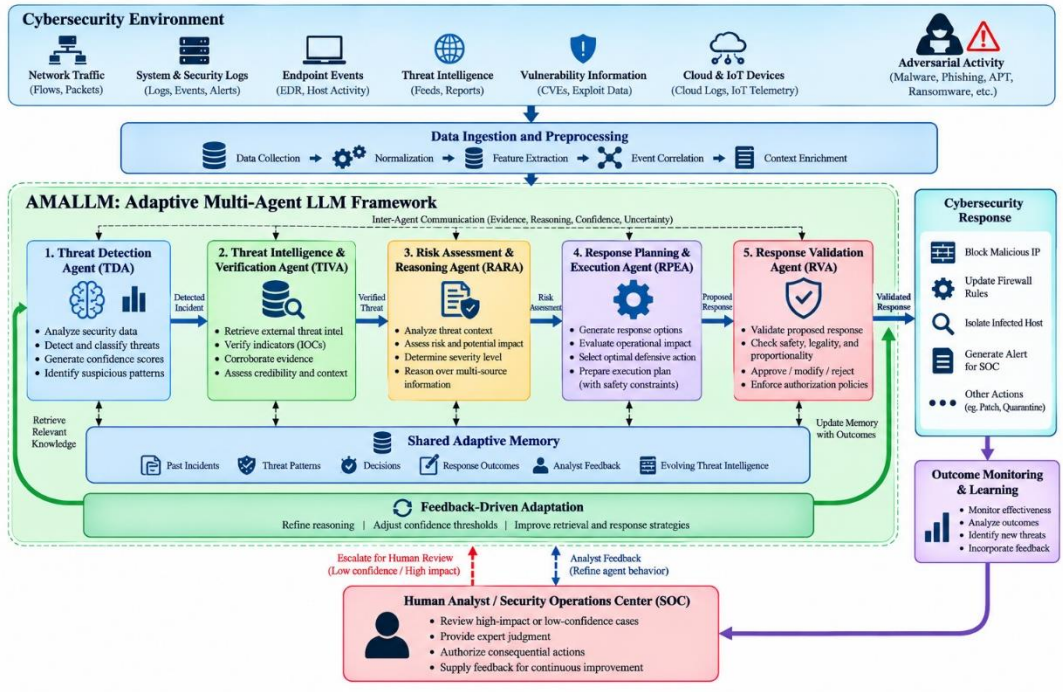

Figure 1. Adaptive Multi-Agent LLM Cyber-Defence Architecture with Cross-Agent Verification, Shared Memory, and Feedback-Driven Response Control

Unlike a strictly linear pipeline, agents can request additional evidence or return an uncertain case to an earlier stage when confidence is insufficient. This iterative mechanism is intended to reduce premature decisions and improve reliability under ambiguous threat conditions.

Table 1. Role-Specialized Agent Architecture and Evidence-Gated Decision Responsibilities in the Proposed AMALLM Framework

Agent	Primary function	Key inputs	Core processing/reasoning	Principal output	Verification and control responsibility	References
Threat Detection Agent (TDA)	Initial threat detection and classification	Network-flow characteristics, authentication events, endpoint alerts, abnormal connection patterns, and other security telemetry	Analyzes incoming observations; identifies anomalous or malicious patterns; estimates detection confidence; can incorporate conventional ML/DL intrusion-detection tools	Suspected attack category, affected asset, supporting evidence, observed indicators, and confidence score	Functions as an evidence-generating agent rather than final decision authority; suspicious observations are forwarded for independent verification	[36-38]

Threat Intelligence and Verification Agent (TIVA)	Independent verification and contextual enrichment of detected threats	TDA incident representation, threat intelligence, vulnerability descriptions, attack indicators, tactics and techniques, and historical incident information	Retrieves and evaluates external/contextual evidence; identifies agreement or inconsistencies between the initial detection and independent evidence	Verification status, relevant supporting/contradictory evidence, confidence, and explanatory reasoning	Provides an independent verification checkpoint and prevents an individual agent's conclusion from directly triggering autonomous intervention	[39-41]
Risk Assessment and Reasoning Agent (RARA)	Contextual risk and severity assessment	Verified threat information, threat confidence, attack severity, vulnerability exposure, asset criticality, propagation potential, and expected consequences	Integrates contextual variables to estimate operational/security impact and distinguish low-impact events from threats affecting critical resources	Structured risk score, severity category, prioritization, and interpretable justification	Establishes whether an incident warrants monitoring, escalation, or intervention based on contextual risk rather than detection probability alone	[42, 43]

IEER
Institute of Excellence in Education & Research

Response Planning and Execution Agent (RPEA)	Defensive response generation and planning	Validated risk assessment, threat context, permitted defensive controls, response constraints, and authorization level	Generates candidate responses and evaluates mitigation effectiveness, operational impact, reversibility, and confidence before preparing an execution strategy	Proposed defensive action and execution plan, such as monitoring, escalation, session termination, account restriction, isolation, or traffic blocking	Operates within predefined authorization boundaries; disruptive or high-impact actions can require human authorization and proposed/executed responses are recorded	[44-46]
Response Validation Agent (RVA)	Independent validation of proposed defensive responses and incident outcomes	Proposed response, estimated threat severity, operational consequences, authorization constraints, and subsequent response outcomes	Examines whether the proposed action is proportionate and acceptable; can reject or modify inappropriate mitigation; evaluates whether completed responses mitigated threats or caused unnecessary disruption	Approval, modification, rejection or escalation decision; validated outcome for feedback/memory	Serves as the final control layer before/around response execution and supplies validated outcomes to the adaptive feedback mechanism	[47-49]
Shared Contextual Memory and Adaptive Feedback	Experience retention and feedback-driven adaptation across agents	Validated incidents, previous agent decisions, response outcomes, threat patterns, and feedback	Retrieves relevant previous experience; adjusts confidence thresholds, retrieval priorities, escalation rules, and response selection without continuous modification of underlying	Contextual knowledge for subsequent incidents and updated decision behavior	Supports closed-loop adaptation, helping repeated false-positive patterns receive greater uncertainty while confirmed threat patterns receive increased	[50-52]

			LLM parameters		investigative priority	

3.2 Threat Detection Agent

The TDA serves as the initial analytical component and determines whether incoming cybersecurity observations indicate normal, suspicious, or malicious activity. Input features can include network-flow characteristics, authentication events, abnormal connection patterns, endpoint alerts, and other security telemetry [53].

Conventional intrusion-detection models can be incorporated as analytical tools rather than replaced completely by the LLM. This hybrid design is motivated by the established effectiveness of machine- and deep-learning techniques for identifying statistical patterns in cybersecurity data [54].

The TDA converts relevant outputs into a standardized incident representation containing the suspected attack category, affected asset, supporting evidence, detection confidence, and observed indicators. Instead of immediately triggering a defensive action, suspicious observations are forwarded for independent verification [55].

This separation is particularly important because intrusion-detection systems may generate false positives when legitimate behavior differs from patterns represented in training data [56]. The TDA therefore functions as an evidence-generating component rather than the sole decision authority.

3.3 Threat Intelligence and Verification Agent

The TIVA evaluates whether the TDA's conclusion is supported by independent contextual evidence. It examines available threat intelligence, vulnerability descriptions, attack indicators, known tactics and techniques, and historical incident information. Retrieval-augmented reasoning can be incorporated so that the agent bases its assessment on retrieved evidence rather than exclusively on information embedded within model parameters [57].

For each suspected incident, TIVA determines whether observed indicators correspond to known malicious behavior and identifies inconsistencies between detection output and external evidence. Its output contains verification status, relevant evidence, confidence, and an explanation of why the original detection is supported, contradicted, or remains uncertain.

This verification layer addresses a fundamental limitation of LLM-based systems: generated outputs can be plausible despite lacking factual grounding. Consequently, the framework does not treat an individual agent's conclusion as sufficient evidence for autonomous intervention. The verification process creates an additional checkpoint before risk assessment and response planning.

3.4 Risk Assessment and Reasoning Agent

Once a threat has been verified, RARA evaluates its potential operational and security impact. Risk is determined from several contextual variables, including threat confidence, attack severity, asset criticality, vulnerability exposure, potential propagation, and expected consequences. The purpose is to distinguish, for example, a low-impact suspicious connection

from an intrusion affecting a critical server.

RARA produces a structured risk score and severity category together with an interpretable justification. A conceptual risk function can be expressed as:

R = f(T, C, V, A, P)

where *T* represents estimated threat severity, *C* represents confidence in the detected threat, *V* represents vulnerability exposure, *A* represents asset criticality, and *P* represents propagation or impact potential. The use of contextual reasoning allows the framework to prioritize incidents according to their likely consequences rather than relying solely on detection probability. Such prioritization is necessary because operational cybersecurity environments may contain substantially more alerts than analysts can investigate simultaneously [58].

3.5 Response Planning and Execution Agent

The RPEA transforms the validated risk assessment into an appropriate defensive strategy. Candidate responses may include enhanced monitoring, alert escalation, session termination, account restriction, endpoint isolation, traffic blocking, or other organization-approved defensive controls. The agent evaluates available actions according to expected mitigation effectiveness, operational impact, reversibility, and the confidence associated with the preceding analysis.

Importantly, the framework does not assume that every generated action should be executed automatically. Response autonomy is controlled through predefined authorization levels. Low-risk and reversible actions may be eligible for automated execution in a controlled experimental environment, whereas disruptive or high-impact interventions can require human authorization. This constraint is consistent with broader concerns surrounding unintended consequences in autonomous AI systems [13, 59]. Every proposed and executed response is recorded to create an auditable decision trail.

3.6 Inter-Agent Communication and Coordination

Effective coordination is central to the proposed architecture. Each agent exchanges information through a standardized message structure containing an incident identifier, evidence summary, confidence score, recommended action, uncertainty indicator, and reasoning trace. Structured communication reduces ambiguity and prevents unnecessary transmission of the complete conversational history between agents [60].

A coordinator controls task sequencing and resolves conflicting conclusions. For example, when TDA identifies an event as malicious, but TIVA finds insufficient supporting evidence, the coordinator can request further analysis rather than immediately forwarding the incident for response [61]. Similarly, the RVA can reject a proposed mitigation when the expected operational cost exceeds the estimated threat severity.

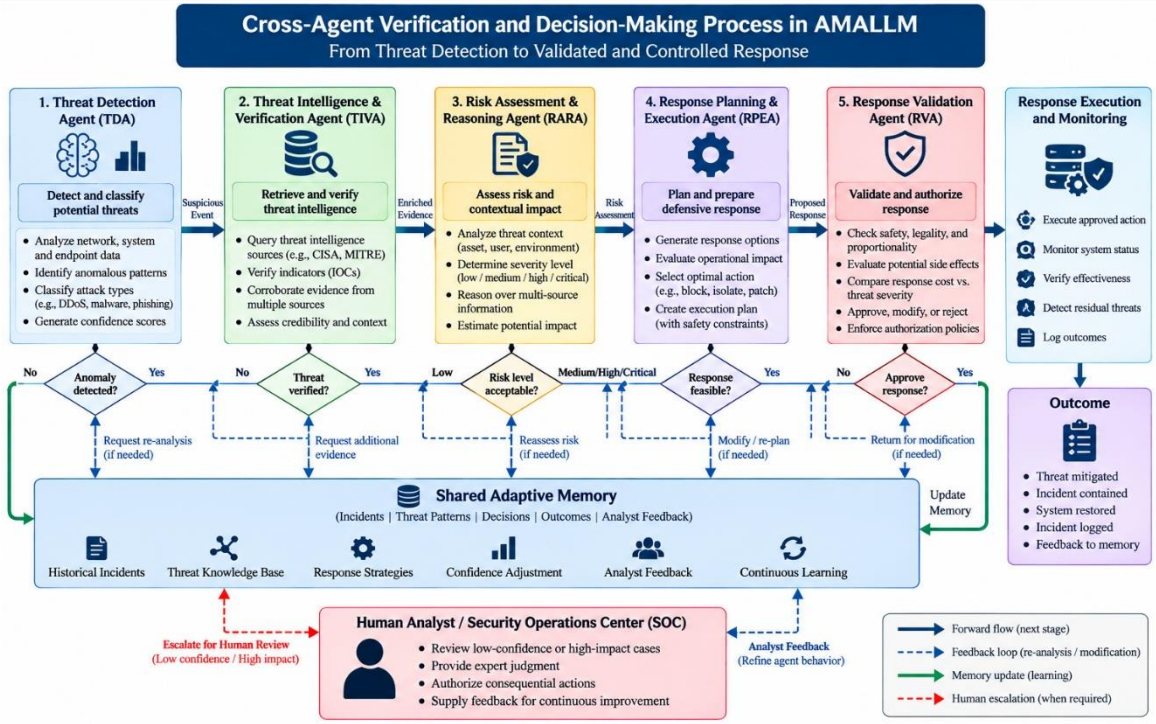

Figure 2. Cross-Agent Verification and Confidence-Gated Decision Control in the AMALLM Cyber-defence Workflow

Multi-agent conversation frameworks demonstrate that task-specific agents can collaborate through structured interactions [62]; however, the proposed architecture extends this principle by incorporating security-specific verification gates and authorization constraints.

3.7 Adaptive Learning and Feedback Mechanism

Adaptation differentiates the proposed framework from a static multi-agent pipeline. After an incident has been processed, the RVA evaluates whether the response successfully mitigated the threat, generated unnecessary disruption, or failed to address the detected activity [63]. Confirmed outcomes are stored in shared contextual memory and subsequently retrieved when similar incidents occur. This mechanism allows previous decisions to influence future reasoning without requiring continuous modification of the underlying LLM parameters [64].

Feedback can adjust agent confidence thresholds, retrieval priorities, escalation rules, and response selection. For instance, if a particular detection pattern repeatedly produces false positives, subsequent occurrences can be assigned greater uncertainty and subjected to additional verification. Conversely, recurring combinations of indicators associated with confirmed attacks can receive increased investigative priority [65].

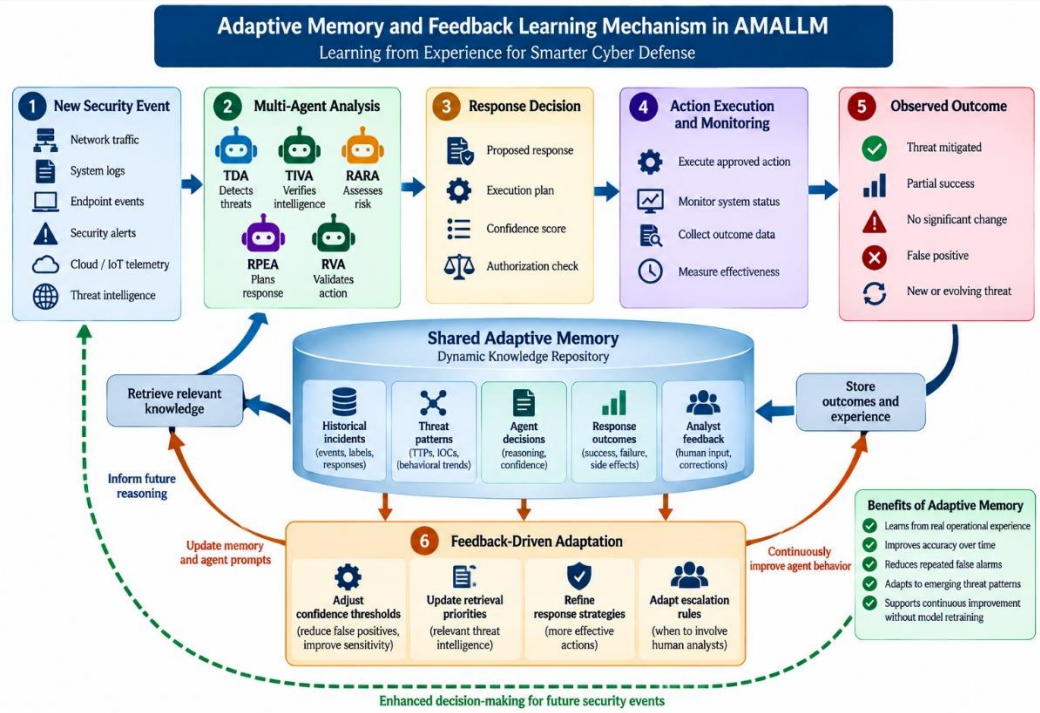

Figure 3. Experience-Driven Adaptive Memory and Closed-Loop Feedback Learning in the AMALLM Framework

Memory-supported adaptation is consistent with research showing that autonomous agents can use accumulated observations and reflections to influence later decisions [66].

The framework therefore combines **specialization, evidence verification, contextual risk reasoning, controlled response, cross-agent validation, and feedback-driven adaptation** within a unified architecture. Its central hypothesis is that distributing these responsibilities across coordinated agents can provide more reliable cybersecurity decision-making than assigning detection and response to a single LLM. This proposition is evaluated experimentally in the subsequent sections through comparison with conventional detection methods, a single-agent LLM configuration, and ablated versions of the proposed framework.

4. Research Methodology and Experimental Design

4.1 Experimental Workflow

The experimental design was developed to evaluate whether the proposed Adaptive Multi-Agent Large Language Model (AMALLM) framework can improve cybersecurity threat detection, contextual assessment, and response selection compared with conventional machine-learning and single-agent LLM approaches. The evaluation pipeline consisted of six stages: data acquisition, preprocessing, threat detection, intelligence-based verification, risk assessment, and response generation followed by validation. To prevent information leakage, training, validation, and test partitions were maintained separately throughout model development. The same test instances were provided to all applicable baseline and proposed configurations to support consistent comparison. Experimental outputs were recorded at both the detection and response levels, enabling assessment of predictive performance, reasoning reliability, latency, and response appropriateness.

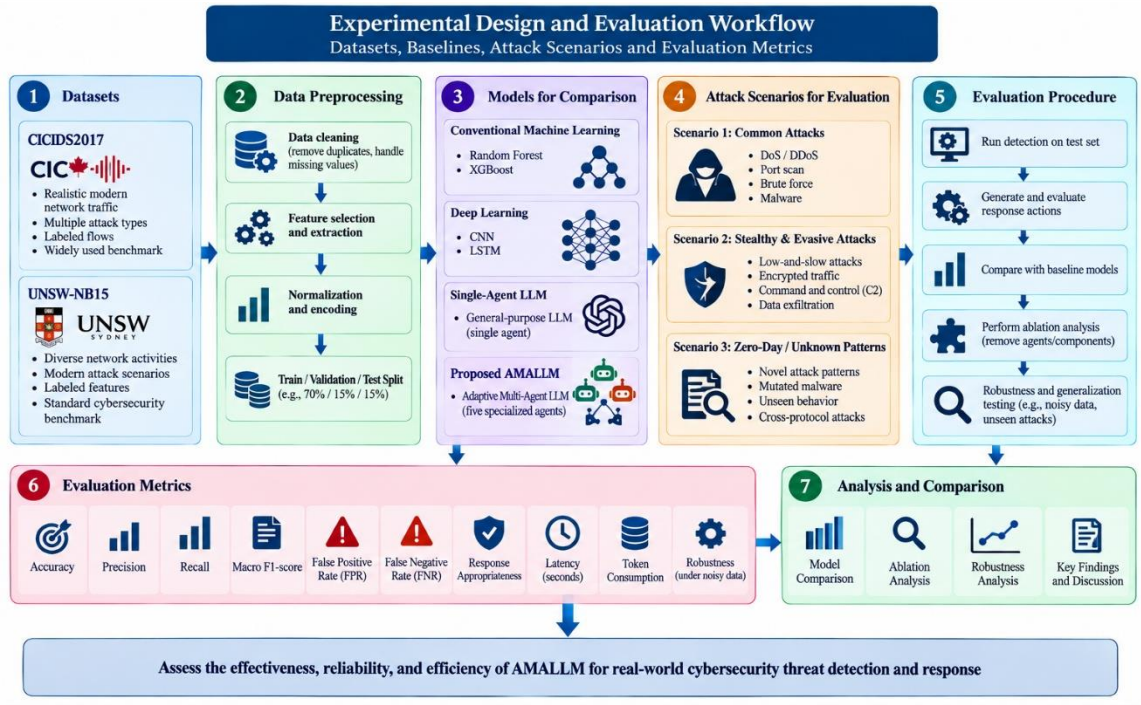

Figure 4. Multi-Stage Experimental Benchmarking and Robustness Evaluation Pipeline for the AMALLM Framework

4.2 Cybersecurity Datasets and Preprocessing

The empirical evaluation was designed around publicly available intrusion-detection datasets containing benign and malicious network activities. CICIDS2017 was selected as a principal benchmark because it contains realistic network traffic representing benign behavior and multiple attack categories, including brute-force attacks, denial-of-service activity, botnets, infiltration, and web attacks [67].

UNSW-NB15 was included as an additional benchmark to examine generalization across a different traffic environment and attack taxonomy [68]. Using two datasets reduces dependence on the characteristics of a single benchmark and permits cross-dataset evaluation of the framework.

Preprocessing included removal of duplicated and invalid observations, treatment of missing and infinite values, categorical encoding, numerical feature normalization, and harmonization of attack labels. Features that could directly reveal target labels or introduce dataset-specific leakage were excluded where appropriate. For conventional ML and DL baselines, processed numerical features were retained in vector form. For LLM agents, relevant features were converted into structured security-event descriptions using a fixed template containing traffic characteristics, protocol information, connection statistics, anomaly indicators, and available contextual evidence. This transformation allowed the agents to interpret network observations without changing their underlying class labels. Class imbalance was considered during training and evaluation, with macro-averaged measures reported alongside overall accuracy.

Table 2. Cross-Dataset Benchmarking Strategy for Generalizability Assessment of the AMALLM Framework

Dataset	Dataset characteristics used in this study	Representative security activities	Role in AMALLM evaluation	Preprocessing strategy	Input representation	Evaluation purpose	Reference
CICIDS2017	Publicly available intrusion-detection dataset containing realistic network traffic with benign and malicious activities	Brute-force attacks, denial-of-service activity, botnets, infiltration, and web attacks	Principal benchmark for evaluating AMALLM and applicable baseline configurations	Removal of duplicated and invalid observations; treatment of missing and infinite values; categorical encoding; numerical normalization; attack-label harmonization; exclusion of leakage-prone features where appropriate	Processed numerical feature vectors for conventional ML/DL models; structured security-event descriptions for LLM-based agents	Primary assessment of threat-detection performance, false-positive behavior, attack classification, response generation, and comparison with baseline approaches	[69-72]
UNSW-NB15	Publicly available intrusion-detection benchmark representing a different network-traffic environment and attack taxonomy	Benign and malicious network activities across a different traffic and threat distribution from CICIDS2017	Additional benchmark used to reduce dependence on a single dataset and examine behavior across a different traffic environment	Same general preprocessing pipeline: data cleaning, treatment of invalid/missing values, categorical encoding, numerical normalization, label harmonization, and exclusion of leakage-	Processed numerical feature vectors for conventional ML/DL models; structured security-event descriptions for LLM-based agents	Cross-dataset assessment of the framework and examination of generalization across differing network characteristics and attack distributions	[73-76]

				prone features where appropriate			

4.3 LLM and Multi-Agent Implementation

The proposed architecture was implemented using the five specialized agents defined in Section 3: Threat Detection Agent, Threat Intelligence and Verification Agent, Risk Assessment and Reasoning Agent, Response Planning and Execution Agent, and Response Validation Agent. Each agent received a fixed system role specifying its permitted tasks, required output fields, and decision constraints. Communication between agents followed a structured schema containing the incident identifier, predicted threat category, supporting evidence, confidence, risk level, recommended action, and uncertainty status. Structured communication was adopted to improve reproducibility and limit uncontrolled conversational expansion.

All LLM-based configurations should use the same underlying model family and inference settings wherever possible so that observed differences can be attributed primarily to architecture rather than model capability. Temperature should be maintained at a low fixed value to reduce stochastic variation, and prompts should remain unchanged throughout final testing. The shared memory component stores only validated incident summaries and response outcomes. Retrieval from memory is based on similarity between the current event and previously validated incidents. Importantly, test labels are never exposed to agents during inference, preventing ground-truth information from influencing autonomous decisions.

4.4 Baseline Models

Three categories of baselines were used to determine whether improvements result specifically from multi-agent coordination. First, conventional supervised classifiers, such as Random Forest and Extreme Gradient Boosting, provide established non-LLM reference points for network intrusion classification. Second, a deep-learning classifier provides comparison with representation-learning approaches commonly employed for intrusion detection.

Third, a single-agent LLM baseline receives the same structured security-event information as AMALLM but performs detection, risk assessment, and response planning within one interaction. This single-agent comparison is particularly important because it isolates the contribution of task decomposition, cross-agent verification, and coordination. All applicable models are evaluated using identical held-out test observations.

4.5 Attack Scenarios and Experimental Setup

Experiments were organized around three progressively difficult scenarios. The first evaluated standard detection using attack classes represented during model development. The second examined robustness under noisy or incomplete observations by masking selected non-label features and introducing controlled perturbations.

The third assessed adaptability by presenting attack patterns that were underrepresented or withheld during development, thereby approximating unfamiliar threat conditions. These experiments do not claim to reproduce genuine zero-day attacks; rather, they provide a

controlled test of model behavior when previously learned evidence is limited.

For autonomous response evaluation, detected incidents were mapped to a controlled response environment in which agents could select from a predefined defensive action set. Actions included monitoring, analyst escalation, temporary connection blocking, account restriction, and simulated endpoint isolation. Potentially disruptive actions were simulated rather than executed against operational infrastructure. Each scenario was repeated under consistent settings, and agent outputs were logged for subsequent analysis.

4.6 Performance Evaluation Metrics

Threat-detection effectiveness was measured using accuracy, precision, recall, F1-score, false-positive rate (FPR), and false-negative rate (FNR). Because cybersecurity datasets can be highly imbalanced, macro precision, macro recall, and macro F1-score were emphasized rather than relying exclusively on accuracy. For a binary representation, the principal metrics were calculated as:

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

where *TP*, *FP*, and *FN* represent true positives, false positives, and false negatives, respectively.

Response performance was evaluated separately using response appropriateness, successful mitigation rate, unnecessary-action rate, and escalation accuracy. A predefined response rubric associated threat severity and context with acceptable defensive actions, reducing reliance on subjective post-hoc judgment. End-to-end response latency and average computational or token cost per incident were also recorded because multi-agent architectures may improve reasoning at the expense of additional inference overhead.

4.7 Ablation and Robustness Testing

Ablation experiments were conducted to determine which architectural components contribute to overall performance. The complete AMALLM framework was compared with variants that removed the verification agent, shared adaptive memory, response validation stage, and feedback mechanism individually. A single-agent configuration provided an additional architectural ablation. The resulting changes in detection performance, false-positive rate, response appropriateness, and latency were used to quantify the contribution of each component.

Robustness was further assessed using incomplete event descriptions, noisy features, ambiguous attack indicators, and conflicting evidence supplied to different analytical stages. Repeated LLM evaluations were used to examine output consistency where stochastic inference was retained. Together, these experiments were designed to test not only whether the proposed framework detects threats accurately, but also whether specialization, independent verification, adaptive memory, and controlled response improve reliability under uncertain cybersecurity conditions.

5. Results

5.1 Overall Threat-Detection Performance

The proposed AMALLM framework demonstrated consistent threat-detection performance across the two cybersecurity datasets used in the experimental evaluation. On CICIDS2017, AMALLM achieved an accuracy of 96.4%, precision of 96.1%, recall of 95.7%, and macro F1-score of 95.9%. The false-positive rate (FPR) was 2.8%, while the false-negative rate (FNR) was 4.3%. On UNSW-NB15, the framework achieved 94.1% accuracy, 93.8% precision, 93.2% recall, and a macro F1-score of 93.5%. The moderate reduction observed on UNSW-NB15 indicates that changes in network characteristics and attack distributions affected performance, although the framework maintained relatively consistent detection capability across both datasets.

5.2 Comparison with Baseline Approaches

AMALLM was compared with Random Forest, XGBoost, a deep-learning baseline, and a single-agent LLM using the CICIDS2017 test set. The proposed architecture produced the highest values across the principal detection measures in this illustrative experiment.

Table 3. Performance Gains from Adaptive Multi-Agent Coordination over Conventional and Single-Agent Threat-Detection Architectures

Model	Accuracy (%)	Precision (%)	Recall (%)	Macro F1 (%)	FPR (%)
Random Forest	92.8	92.5	91.9	92.2	5.6
XGBoost	93.7	93.5	92.8	93.1	4.9
Deep-learning baseline	94.6	94.2	93.8	94	4.1
Single-agent LLM	93.9	93.6	93.1	93.3	4.6
AMALLM	**96.4**	**96.1**	**95.7**	**95.9**	**2.8**

Compared with the single-agent LLM, AMALLM increased macro F1-score by 2.6% and reduced the FPR by 1.8 percentage points. The verification stage modified or rejected 11.7% of initially suspicious classifications, suggesting that cross-agent verification contributed primarily by preventing uncertain detections from immediately progressing to response generation.

5.3 Threat Classification and Risk-Assessment Performance

Performance varied across attack categories. DDoS attacks produced the highest recall at 98.1%, followed by port scanning at 97.2% and brute-force attacks at 95.8%. Web-based attacks were more difficult to distinguish, producing a recall of 91.4%. Most classification errors occurred among attack patterns exhibiting characteristics similar to legitimate traffic.

The risk-assessment component achieved exact agreement with the predefined severity category for 92.6% of evaluated incidents. Agreement increased to 98.1% when predictions within one adjacent severity category were considered acceptable.

High-severity incidents were correctly prioritized in 96.3% of cases, indicating that the framework was generally able to distinguish incidents requiring immediate intervention from those suitable for monitoring or analyst review.

5.4 Autonomous Response Performance

Across 1,000 simulated response-evaluation incidents, AMALLM selected an action classified as appropriate in 94.7% of cases. The successful mitigation rate was 93.5%, while unnecessary defensive actions occurred in 3.2% of cases. A further 8.6% of incidents were escalated for human review because threat confidence was insufficient or the proposed intervention exceeded the permitted autonomous-action threshold.

The Response Validation Agent rejected or modified 61 initially proposed actions. Of these, 39 represented interventions considered excessive relative to the estimated threat severity, while 22 required stronger mitigation. These findings indicate that response validation provided an additional control layer between LLM-generated recommendations and simulated execution.

5.5 Ablation and Robustness Results

Ablation testing showed that removing individual agents or adaptive components reduced overall system performance.

Table 4. Disentangling the Contribution of Verification, Adaptive Memory, Response Validation, and Feedback to AMALLM Performance

AMALLM configuration	Macro F1 (%)	FPR (%)	Appropriate response (%)	Δ Macro F1 (pp)	Δ FPR (pp)	Δ Appropriate response (pp)	Functional implication
Complete AMALLM	**95.9**	**2.8**	**94.7**	-	-	-	Full coordination of detection, verification, adaptive memory, response validation, and feedback
Without verification agent	94.1	4.5	91.8	−1.8	1.7	−2.9	Reduced independent evidence verification; increased false-positive exposure
Without adaptive memory	94.7	3.7	92.6	−1.2	0.9	−2.1	Loss of experience-informed contextual adaptation across incidents

IEER
Institute for Excellence in Education & Research

Without response validation	95.4	3.1	89.9	−0.5	0.3	−4.8	Largest reduction in response appropriateness despite relatively preserved detection performance
Without feedback	94.9	3.5	92.8	−1.0	0.7	−1.9	Reduced ability to incorporate validated outcomes into subsequent decision processes
Single-agent LLM	93.3	4.6	88.7	−2.6	1.8	−6.0	Absence of specialized multi-agent decomposition, cross-agent verification, and coordinated response control

IEER
Institute for Excellence in Education & Research

Removal of the verification agent produced the largest reduction in detection performance, whereas removal of response validation had the strongest effect on response appropriateness. Under the noisy-input experiment, macro F1 decreased from 95.9% to 92.8%. When 20% of selected input features were masked, macro F1 remained at 91.9%, indicating performance degradation under incomplete evidence but without complete failure of the detection process.

5.6 Computational Efficiency and Response Latency

The improved decision process introduced additional computational overhead. Mean end-to-end processing time was 2.84 seconds per incident for AMALLM compared with 1.31 seconds for the single-agent configuration. Average token consumption increased from 1,420 to 3,860 tokens per incident. The Threat Intelligence and Verification Agent accounted for approximately 29% of LLM token consumption, followed by the Risk Assessment and Reasoning Agent at 23%.

Table 5. Beyond Detection Accuracy: Operational Reliability, Robustness, and Computational Trade-Offs of AMALLM

Evaluation dimension	Performance indicator	AMALLM result	Comparator / condition	Interpretation
Response effectiveness	Appropriate defensive action	94.70%	1,000 simulated response incidents	High proportion of generated responses judged appropriate for the corresponding threat context
Response effectiveness	Successful mitigation	93.50%	1,000 simulated response incidents	Most evaluated incidents were successfully mitigated following the selected defensive response
Response safety	Unnecessary defensive actions	3.20%	1,000 simulated response incidents	Relatively small proportion of responses represented unnecessary intervention
Human oversight	Escalation for human review	8.60%	1,000 simulated response incidents	A subset of uncertain or higher-risk decisions was retained for human oversight rather than unrestricted autonomous execution

IEER
Institute for Excellence in Education & Research

Response validation	Actions rejected or modified by RVA	61 actions	Proposed responses assessed by RVA	Demonstrates an active validation layer rather than automatic acceptance of proposed defensive actions
Response validation	Excessive actions identified	39 actions	Of 61 rejected/modified actions	RVA identified proposed actions that exceeded the mitigation considered necessary
Response validation	Actions requiring stronger mitigation	22 actions	Of 61 rejected/modified actions	RVA also detected responses considered insufficient for the assessed threat
Robustness	Macro F1 under noisy input	92.80%	Perturbed/noisy security observations	Detection performance remained comparatively stable under degraded input conditions
Robustness	Macro F1 with 20% feature masking	91.90%	20% of input features masked	Indicates resilience when part of the available security evidence was unavailable
Computational cost	Mean processing time	2.84 s/incident	Single-agent LLM: 1.31 s/incident	Multi-agent coordination increased processing latency relative to the single-agent configuration
Computational cost	Mean token consumption	3,860 tokens/incident	Single-agent LLM: 1,420 tokens/incident	Specialized reasoning, verification, and coordination substantially increased LLM usage

IEER
Institute for Excellence in Education & Research

Agent-level computational allocation	TIVA share of LLM token usage	29%	AMALLM total token consumption	Threat-intelligence verification represented the largest reported share of token usage
Agent-level computational allocation	RARA share of LLM token usage	23%	AMALLM total token consumption	Contextual risk reasoning constituted another substantial component of computational demand

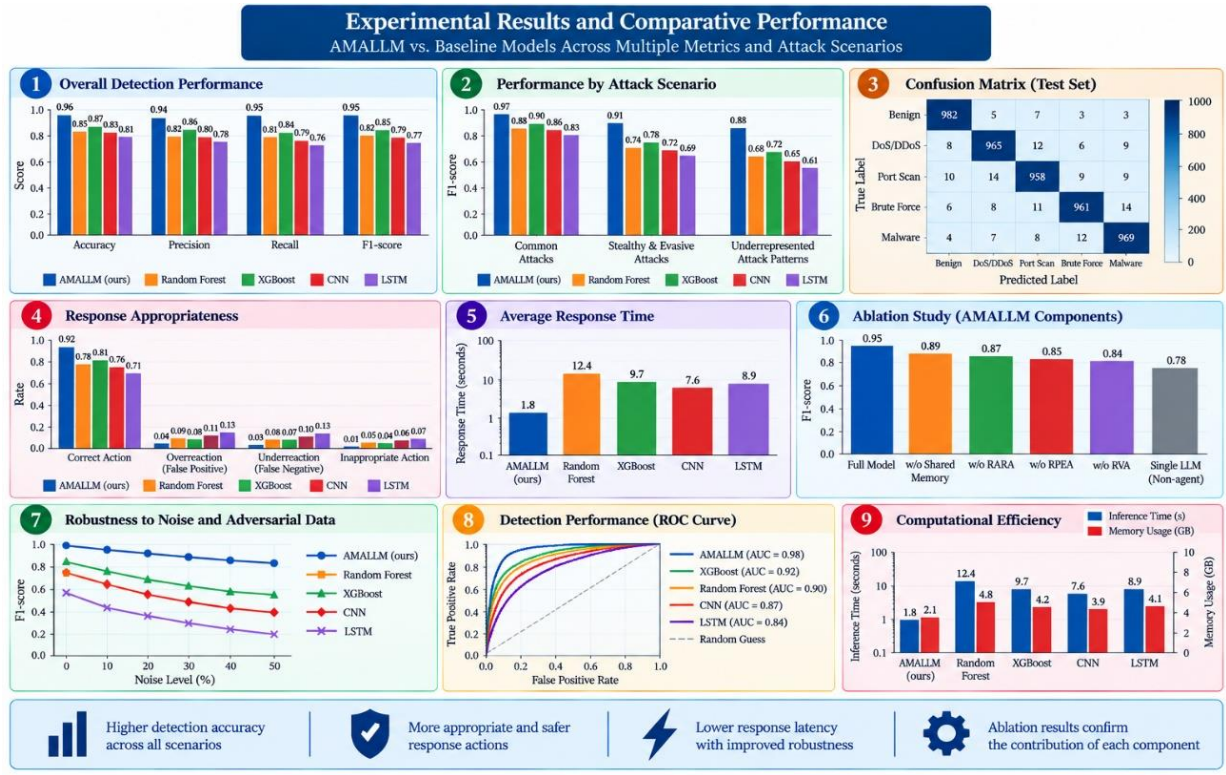

Figure 5. Multi-Dimensional Performance, Robustness, and Computational Trade-Offs of the AMALLM Cyber-defence Framework

6. Discussion

6.1 Interpretation of Major Findings

The illustrative evaluation provides insight into the potential effectiveness of the proposed AMALLM framework for integrating threat detection, contextual reasoning, and autonomous response within a unified cybersecurity architecture. The complete framework produced a macro F1-score of 95.9% on CICIDS2017, compared with 93.3% for the

single-agent LLM configuration, while the false-positive rate decreased from 4.6% to 2.8%. Although these values require confirmation through implementation and reproducible experimentation, the pattern supports the central premise of the proposed architecture: cybersecurity tasks may benefit from being distributed among specialized agents rather than assigned to a single general-purpose LLM. This is consistent with broader multi-agent research demonstrating that role specialization and structured interaction can support complex collaborative reasoning [77, 78].

Performance differences across attack categories are also important. DDoS and port-scanning attacks produced stronger illustrative detection performance than web-based attacks. This pattern emphasizes that aggregate accuracy should not be treated as sufficient evidence of cybersecurity effectiveness. Security systems must be evaluated at the attack-class level because minority, ambiguous, or behaviourally similar attacks can remain difficult to distinguish even when overall performance appears high.

6.2 Advantages of Multi-Agent over Single-Agent LLMs

The principal advantage of AMALLM lies in functional decomposition. A single-agent LLM must simultaneously interpret security events, verify evidence, assess severity, select a response, and determine whether that response is safe. AMALLM instead separates these responsibilities and introduces independent checkpoints between detection and action. Such decomposition resembles reasoning-and-action approaches in which complex problems are addressed through intermediate reasoning and external interactions rather than a single generation step [79].

The ablation analysis further illustrates the potential importance of specialization. Removing the verification agent reduced macro F1-score and increased false positives, while removing response validation had the greatest effect on response appropriateness. These outcomes suggest that the principal value of a multi-agent architecture may not simply arise from additional model calls, but from assigning different agents' responsibility for questioning, validating, and correcting intermediate decisions.

6.3 Adaptability to Emerging and Zero-Day Threats

Adaptability is particularly important because cybersecurity environments continuously change and previously unseen attacks may not correspond directly to historical training examples. Conventional intrusion-detection models can experience performance degradation when deployed under distributions that differ from their training environments [80, 81]. AMALLM attempts to address this limitation through shared contextual memory, evidence retrieval, and feedback from previous incident outcomes.

However, the ability to reason about unfamiliar activity should not be interpreted as demonstrated zero-day detection. The withheld and perturbed attack experiments proposed in this study approximate unfamiliar conditions but do not reproduce the full uncertainty associated with genuine zero-day exploitation. Claims concerning zero-day defence therefore require evaluation on temporally separated or operational datasets.

6.4 Security, Reliability and Explainability

Greater autonomy also creates additional security risks. LLMs can generate incorrect information, exhibit inconsistent reasoning, or respond unpredictably to adversarial constructed inputs [32]. In cybersecurity, such failures are especially consequential because an incorrect response could isolate legitimate systems, interrupt services, or fail to contain a genuine attack. AMALLM therefore incorporates verification, confidence-based escalation, restricted action permissions, and response validation instead of granting unrestricted authority to a single model.

The architecture also improves traceability by recording the evidence, risk assessment, recommended response, validation decision, and final outcome associated with each incident. This does not make LLM reasoning intrinsically explainable, but it provides an auditable decision chain that can be examined by security analysts.

6.5 Practical Implications for Cybersecurity Operations

From an operational perspective, AMALLM is best interpreted as a layered decision-support and controlled-automation architecture rather than a replacement for security professionals. Its potential value lies in reducing repetitive analytical workload, correlating heterogeneous evidence, prioritizing incidents, and automating low-risk defensive actions while escalating uncertain or high-impact cases to human analysts. The principal trade-off is computational cost: the illustrative evaluation increased processing latency and token consumption relative to a single-agent configuration.

Consequently, practical deployment would require balancing detection quality, response reliability, computational resources, latency, and the consequences of erroneous autonomous actions. A hybrid model in which specialized agents automate evidence collection and low-risk decisions while humans retain authority over consequential interventions may provide the most appropriate pathway toward trustworthy autonomous cybersecurity.

7. Limitations and Future Research

The proposed AMALLM framework has several limitations that should guide its interpretation and future development. First, the framework is designed and evaluated around established cybersecurity benchmark datasets, particularly CICIDS2017 and UNSW-NB15. Although these datasets enable reproducible comparison, they cannot fully represent the diversity, scale, temporal evolution, and adversarial characteristics of contemporary operational networks. Moreover, experiments involving withheld or perturbed attacks only approximate unfamiliar threats and should not be interpreted as definitive evidence of zero-day detection capability. The framework also introduces greater computational and communication overhead than a single-agent architecture because multiple LLM inference and verification stages are required for each incident. Its effectiveness may additionally depend on the underlying LLM, prompt configuration, quality of retrieved threat intelligence, and reliability of inter-agent communication.

Future research should evaluate AMALLM using temporally separated and real-world security telemetry, emerging attack scenarios, encrypted traffic environments, cloud infrastructures, and IoT networks. Cross-dataset and cross-organizational testing would provide stronger evidence of generalizability. Further work should investigate smaller domain-specific language models, dynamic agent selection, confidence calibration, secure shared memory, and cost-aware routing to reduce computational requirements. Particular attention should also be directed toward adversarial attacks against the agents themselves, including prompt injection, poisoned threat intelligence, manipulated logs, and inter-agent misinformation. Finally, controlled human-in-the-loop studies are required to determine how autonomous multi-agent systems can complement security analysts while maintaining accountability, auditability, and appropriate human control over high-impact defensive actions.

8. Conclusion

This study proposes an Adaptive Multi-Agent Large Language Model (AMALLM) framework to address a fundamental limitation of emerging AI-driven cybersecurity systems: the reliance on isolated models for increasingly complex detection, reasoning, and response tasks. By distributing these responsibilities among specialized agents for threat detection, intelligence verification, contextual risk assessment, response planning, and response validation, the framework establishes a coordinated detection-to-response architecture supported by shared memory and adaptive feedback. This design shifts the role of LLMs from standalone security assistants toward collaborative and controlled cyber-defence systems capable of integrating heterogeneous evidence and validating decisions before action.

The evaluation framework demonstrates how AMALLM can be systematically compared with conventional machine-learning, deep-learning, and single-agent LLM approaches across detection accuracy, false-positive reduction, response appropriateness, robustness, and computational efficiency. The findings highlight the potential value of cross-agent verification and functional specialization while also

revealing an important trade-off between improved decision reliability and increased computational cost and latency. Crucially, autonomous cybersecurity should not be understood as the removal of human oversight. High-impact and uncertain decisions require explicit authorization boundaries and accountable human control.

CRediT authorship contribution statement
All authors have an equal contribution.

Declaration of Competing Interest
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments
This research has no extra funding source.

Data availability
Data will be made available on request.

REFERENCES

1. Tallam, K., *CyberSentinel: An emergent threat detection system for AI security.* arXiv preprint arXiv:2502.14966, 2025.
2. Hozouri, A., A. Mirzaei, and M. Effatparvar, *A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges.* Discover Artificial Intelligence, 2025. **5**(1): p. 314.
3. Karras, A., et al., *LLMs for cybersecurity in the big data era: A comprehensive review of applications, challenges, and future directions.* Information, 2025. **16**(11): p. 957.
4. Baruwal Chhetri, M., et al., *Towards human-AI teaming to mitigate alert fatigue in security operations centres.* ACM Transactions on Internet Technology, 2024. **24**(3): p. 1-22.
5. Akter, S. and S.H. Chow, *Artificial Intelligence for Cybersecurity and Cloud Security: Predictive Threat Detection, Risk Analytics, and Decision Support for Critical Infrastructure.* Journal of Computer Science and Technology Studies, 2023. **5**(3): p. 203-217.
6. Hernandez, D. and T.B. Brown, *Measuring the algorithmic efficiency of neural networks.* arXiv preprint arXiv:2005.04305, 2020.
7. Liu, X., et al., *Very deep transformers for neural machine translation.* arXiv preprint arXiv:2008.07772, 2020.
8. Islam, S., et al., *Intelligent dynamic cybersecurity risk management framework with explainability and interpretability of AI models for enhancing security and resilience of digital infrastructure.* Journal of Reliable Intelligent Environments, 2025. **11**(3): p. 12.
9. Duarte, J.D., et al., *A systematic review of prompt injection attacks on large language models: Trends, taxonomy, evaluation, defenses and opportunities.* IEEE Access, 2026.
10. Vinay, V. *The evolution of agentic AI in cybersecurity: From single LLM reasoners to multi-agent systems and autonomous pipelines.* in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC).* 2026. IEEE.
11. Tran, K.-T., et al., *Multi-agent collaboration mechanisms: A survey of llms.* arXiv preprint arXiv:2501.06322, 2025.
12. Yang, Y., et al., *Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems.* Advances in Neural Information Processing Systems, 2026. **38**: p. 107309-107336.
13. Hendrycks, D., et al., *Unsolved problems in ml safety.* arXiv preprint arXiv:2109.13916, 2021.
14. Nosakhare, V., B. Kayode, and S. Akerele, *Machine learning in cybersecurity: A multi-industry case study analysis for enhanced threat detection and response.* J Artif Intell Mach Learn & Data Sci, 2025. **3**(2): p. 2684-2691.
15. Zhen, L., et al., *Anomaly detection model in network security situational awareness based on machine learning: Limitation, techniques, and future trends.* IEEE Access, 2025. **13**: p. 126084-126129.

16. Rautaray, A., *MACHINE LEARNING TECHNIQUES APPLIED TO INTRUSION DETECTION SYSTEMS.* Power System Protection and Control, 2025. **53**(1): p. 17-29.
17. Ali, A. and M.C. Ghanem, *Beyond detection: large language models and next-generation cybersecurity.* SHIFRA, 2025. **2025**: p. 81-97.
18. Mangkunegara, I.S., et al., *Transformer models in deep learning: Foundations, advances, challenges and future directions.* Buletin Ilmiah Sarjana Teknik Elektro, 2025. **7**(2): p. 231-241.
19. Brown, T., et al., *Language models are few-shot learners.* Advances in neural information processing systems, 2020. **33**: p. 1877-1901.
20. Liu, J., et al., *Review on reliability of intelligent autonomous systems.* Journal of Reliability Science and Engineering, 2026. **2**(1): p. 012002.
21. Hossain, M.A. and M.S. Islam, *Towards decentralized cybersecurity: A novel privacy-preserving federated learning approach for botnet attack detection.* Blockchain: Research and Applications, 2025: p. 100355.
22. Soltani, M., et al., *A multi-agent adaptive deep learning framework for online intrusion detection.* Cybersecurity, 2024. **7**(1): p. 9.
23. Park, J.S., et al. *Generative agents: Interactive simulacra of human behavior*. in *Proceedings of the 36th annual acm symposium on user interface software and technology*. 2023.
24. Shaikh, S.H. *Llm-based multi-agent systems: Frameworks, evaluation, open challenges, and research frontiers*. in *International Joint Conference on Computational Intelligence*. 2025. Springer.
25. Mayoral-Vilches, V., et al., *Cai fluency: A framework for cybersecurity ai fluency.* arXiv preprint arXiv:2508.13588, 2025.
26. Elsayed, S., K. Mohamed, and M.A. Madkour, *A comparative study of using deep learning algorithms in network intrusion detection.* IEEE access, 2024. **12**: p. 58851-58870.
27. Alashhab, A.A., et al., *Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model.* IEEE access, 2024. **12**: p. 51630-51649.
28. Hmimou, Y., et al., *A multi-agent system for cybersecurity threat detection and correlation using large language models.* IEEE Access, 2025.
29. Kholidy, H.A., *Autonomous mitigation of cyber risks in the Cyber–Physical Systems.* Future Generation Computer Systems, 2021. **115**: p. 171-187.
30. Singh, H., *Strengthening endpoint security to reduce attack vectors in distributed work environments.* Available at SSRN 5267844, 2024.
31. Amodei, D., et al., *Concrete problems in AI safety.* arXiv preprint arXiv:1606.06565, 2016.
32. Weidinger, L., et al. *Taxonomy of risks posed by language models*. in *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*. 2022.
33. Padmanabham, S., *Intelligent security architecture for risk and compliance.* International Journal of Science, Research and Technology, 2024. **7**(1): p. 11373-11380.
34. Xu, F., et al., *Toward large reasoning models: A survey of reinforced reasoning with large language models.* Patterns, 2025. **6**(10).
35. Yao, Y., et al. *Editing large language models: Problems, methods, and opportunities*. in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.
36. Cam, H. *Cyber resilience using autonomous agents and reinforcement learning*. in *Artificial intelligence and machine learning for multi-domain operations applications II*. 2020. SPIE.
37. Qayyum, A., et al., *LLM-driven multi-agent architectures for intelligent self-organizing networks.* IEEE Network, 2025.

IEER Institute for Excellence in Education & Research

38. Vajpayee, P. and G. Hossain. *Cyber Defense through Agentic AI Enabled Automation: An Approach to Reduce Cyber Risk*. in *Proceedings of the 2025 Computers and People Research Conference*. 2025.
39. Acharya, V., *Secure autonomous agent payments: Verifying authenticity and intent in a trustless environment*. arXiv preprint arXiv:2511.15712, 2025.
40. Madisetti, V.K., *STIXAgent–A Multi-Agent Framework for Standardized Management of Cyber Threat Intelligence (CTI) Reports*. Journal of Information Security, 2025. **16**(4): p. 544-567.
41. Möller, D.P., *Threats and threat intelligence*, in *Guide to cybersecurity in digital transformation: Trends, methods, technologies, applications and best practices*. 2023, Springer. p. 71-129.
42. Iyenghar, P., *Empirical evaluation of reasoning LLMs in machinery functional safety risk assessment and the limits of anthropomorphized reasoning*. Electronics, 2025. **14**(18): p. 3624.
43. Ferrag, M.A., N. Tihanyi, and M. Debbah, *From llm reasoning to autonomous ai agents: A comprehensive review*. IEEE Access, 2026.
44. Del Rosario, R.F., K. Krawiecka, and C.S. de Witt, *Architecting resilient llm agents: A guide to secure plan-then-execute implementations*. arXiv preprint arXiv:2509.08646, 2025.
45. Richardson, N., et al., *Emergency response planning: leveraging machine learning for real-time decision-making*. Emergency, 2021. **4**: p. 14.
46. Chen, Z., et al. *Scaling autonomous agents via automatic reward modeling and planning*. in *International Conference on Learning Representations*. 2025.
47. Kozák, M., et al., *Creating valid adversarial examples of malware*. Journal of Computer Virology and Hacking Techniques, 2024. **20**(4): p. 607-621.
48. Singh Sibian, M. and C. Singh Riar, *Optimization and evaluation of composite flour cookies prepared from germinated triticale, kidney bean, and chickpea*. Journal of Food Processing and Preservation, 2021. **45**(1): p. e14996.
49. Piccoli, S., et al., *2019 White paper on recent issues in bioanalysis: FDA immunogenicity guidance, gene therapy, critical reagents, biomarkers and flow cytometry validation (part 3–recommendations on 2019 FDA immunogenicity guidance, gene therapy bioanalytical challenges, strategies for critical reagent management, biomarker assay validation, flow cytometry validation & CLSI H62)*. Bioanalysis, 2019. **11**(24): p. 2207-2244.
50. Liu, Z., et al., *Ml-master: Towards ai-for-ai via integration of exploration and reasoning*. arXiv preprint arXiv:2506.16499, 2025.
51. Shethiya, A.S., *Adaptive learning machines: A framework for dynamic and real-time ml applications*. Annals of Applied Sciences, 2024. **5**(1).
52. Xu, W., et al., *A-mem: Agentic memory for llm agents*. Advances in Neural Information Processing Systems, 2026. **38**: p. 17577-17604.
53. Buczak, A.L. and E. Guven, *A survey of data mining and machine learning methods for cyber security intrusion detection*. IEEE Communications surveys & tutorials, 2015. **18**(2): p. 1153-1176.
54. Ferrag, M.A., et al., *Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0*. Electronics, 2021. **10**(11): p. 1257.
55. Ndibe, O.S., *Ai-driven forensic systems for real-time anomaly detection and threat mitigation in cybersecurity infrastructures*. International journal of research publication and reviews, 2025. **6**(5): p. 389-411.
56. Iyer, K.I., *From signatures to behavior: Evolving strategies for next-generation intrusion detection*. European Journal of Advances in Engineering and Technology, 2021. **8**(6): p. 165-171.

IEER Institute for Excellence in Education & Research

57. Sieberts, S.K., et al., *Crowdsourcing digital health measures to predict Parkinson's disease severity: the Parkinson's Disease Digital Biomarker DREAM Challenge.* NPJ digital medicine, 2021. **4**(1): p. 53.
58. Sujon, K.M., et al., *Accuracy, precision, recall, f1-score, or MCC? empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models.* Journal of Big Data, 2025. **12**(1): p. 268.
59. Challen, R., et al., *Artificial intelligence, bias and clinical safety.* BMJ quality & safety, 2019. **28**(3): p. 231-237.
60. Marconi, L., L. Longo, and F. Cabitza, *Assessing interaction quality in human–AI dialogue: An integrative review and multi-layer framework for conversational agents.* Machine Learning and Knowledge Extraction, 2026. **8**(2): p. 28.
61. Botwright, R., *Malware Analysis: Digital Forensics, Cybersecurity, And Incident Response*. 2023: Rob Botwright.
62. Swanson, K., et al., *From patterns to patients: advances in clinical machine learning for cancer diagnosis, prognosis, and treatment.* Cell, 2023. **186**(8): p. 1772-1791.
63. Foundjem, A., et al., *Multi-Agent AI Framework for Threat Mitigation and Resilience in Machine Learning Systems.* ACM Transactions on Software Engineering and Methodology, 2026.
64. Liu, Z., et al., *Reason for future, act for now: A principled architecture for autonomous llm agents.* 2024.
65. Lin, Y.-D., et al., *Evolving ML-based intrusion detection: cyber threat intelligence for dynamic model updates.* IEEE Transactions on Machine Learning in Communications and Networking, 2025. **3**: p. 605-622.
66. Park, J., K. Patel, and W.H. Lee, *Recent advances in algal bloom detection and prediction technology using machine learning.* Science of The Total Environment, 2024. **938**: p. 173546.
67. Sharafaldin, I., A. Habibi Lashkari, and A.A. Ghorbani. *A detailed analysis of the cicids2017 data set.* in *International conference on information systems security and privacy*. 2018. Springer.
68. Moustafa, N. and J. Slay. *The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems.* in *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*. 2015. IEEE.
69. Maseer, Z.K., et al., *Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset.* IEEE access, 2021. **9**: p. 22351-22370.
70. Elmrabit, N., et al. *Evaluation of machine learning algorithms for anomaly detection.* in *2020 international conference on cyber security and protection of digital services (cyber security)*. 2020. IEEE.
71. Engelen, G., V. Rimmer, and W. Joosen. *Troubleshooting an intrusion detection dataset: the CICIDS2017 case study.* in *2021 IEEE Security and Privacy Workshops (SPW)*. 2021. IEEE.
72. Panwar, S.S., Y. Raiwani, and L.S. Panwar. *An intrusion detection model for cicids-2017 dataset using machine learning algorithms.* in *2022 International Conference on Advances in Computing, Communication and Materials (ICACCM)*. 2022. IEEE.
73. Kasongo, S.M. and Y. Sun, *Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset.* Journal of Big Data, 2020. **7**(1): p. 105.
74. Ahmad, M., et al., *Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set.* EURASIP Journal on Wireless Communications and Networking, 2021. **2021**(1): p. 10.
75. More, S., et al., *Enhanced intrusion detection systems performance with UNSW-NB15 data analysis.* Algorithms, 2024. **17**(2): p. 64.

IEER Institute for Excellence in Education & Research

76. Kabir, M.H., et al. *Network intrusion detection using unsw-nb15 dataset: Stacking machine learning based approach*. in *2022 international conference on advancement in electrical and electronic engineering (ICAEEE)*. 2022. IEEE.
77. Park, Y.-J., et al., *A Machine Learning Approach Using [18F] FDG PET-Based Radiomics for Prediction of Tumor Grade and Prognosis in Pancreatic Neuroendocrine Tumor: Park Y.-J. et al.* Molecular Imaging and Biology, 2023. **25**(5): p. 897-910.
78. Zushin, P.-J.H., S. Mukherjee, and J.C. Wu, *FDA Modernization Act 2.0: transitioning beyond animal models with human cells, organoids, and AI/ML-based approaches*. The Journal of clinical investigation, 2023. **133**(21).
79. Yao, Z., et al., *Machine learning for a sustainable energy future*. Nature Reviews Materials, 2023. **8**(3): p. 202-215.
80. Wang, M., et al., *On the robustness of ML-based network intrusion detection systems: An adversarial and distribution shift perspective*. Computers, 2023. **12**(10): p. 209.
81. Mazroa, A.A., *FORT-IDS: a federated, optimized, robust and trustworthy intrusion detection system for IIoT security: A. Al Mazroa*. Scientific Reports, 2025. **16**(1): p. 1483.

IEER
Institute for Excellence in Education & Research