Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.079318

ARTICLE

Towards Threat Identification for the BACnet Protocol Using Large Language Models

Hsuan-Chih Ku[1], Jyun-Kai Yang[1], Pang-Wei Tsai[1] and Shih-Hsiung Lee[2,*]

[1]Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
[2]Department of Intelligent Commerce, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
*Corresponding Author: Shih-Hsiung Lee. Email: shlee@nkust.edu.tw

Received: 19 January 2026; Accepted: 30 April 2026; Published: 15 June 2026

ABSTRACT: With the rapid proliferation of the Industrial Internet of Things (IIoT), Building Automation Systems (BAS) and Industrial Control Systems (ICS) are increasingly exposed to sophisticated cyber threats. Conventional Intrusion Detection Systems (IDS) often encounter significant limitations when addressing emerging or hybrid attack patterns, primarily due to delayed signature updates and high false-positive rates. Meanwhile, existing anomaly detection approaches frequently lack sufficient awareness of the physical domain, making them ineffective in identifying falsification attacks that comply with communication protocol specifications while violating underlying physical laws. To address these challenges, this study proposes a hybrid threat detection architecture that integrates Retrieval-Augmented Generation (RAG) with a Physics Rule Engine. The proposed approach leverages the semantic reasoning capabilities of Large Language Models (LLMs) to enhance protocol-level threat interpretation, while incorporating physical constraints to validate system behavior at the cyber–physical level. The core contribution of this work lies in employing an LLM to transform unstructured BACnet packet data into structured reasoning representations using the DSPy framework. These representations are subsequently examined through physics-based validation rules derived from thermodynamics and fluid mechanics, enabling the detection of attacks that are logically valid at the protocol layer but implausible in the physical domain. This layered verification process effectively reduces spurious alerts and improves detection reliability. Experimental evaluations conducted under various BACnet attack scenarios demonstrate that the proposed system, implemented with the Mistral-7B model, achieves an accuracy of 95.12% and an F1-score of 96.0%. Compared with a baseline LLM-only approach without physical validation, the proposed method significantly lowers the false-positive rate. Moreover, the system is capable of automatically generating evidence chains that support explainable security forensics, thereby enhancing situational awareness for security operators. The results of this study suggest that the integration of domain knowledge with generative AI constitutes a promising and effective strategy for strengthening the resilience of critical cyber–physical infrastructures.

KEYWORDS: Threat identification; BACnet protocol; large language models

1 Introduction

With the continued advancement of Industry 4.0 and large-scale digital transformation initiatives, Industrial Control Systems (ICS) and Building Automation Systems (BAS) are undergoing substantial architectural reconfiguration. In order to enable data-driven optimization, centralized analytics, and remote operation and maintenance, Operational Technology (OT) networks that were historically maintained as air-gapped environments are increasingly required to interconnect with Information Technology (IT) networks. While this convergence improves operational visibility and efficiency, it simultaneously exposes previously

closed OT ecosystems to a broad spectrum of Internet-originated threats, thereby enlarging the attack surface and increasing systemic risk.

Recent statistics reported by Kaspersky ICS CERT indicate that more than 34% of industrial computers worldwide experienced malware-related incidents during the past year, highlighting that industrial environments have become high-value targets for adversaries. Compared with attacks against enterprise IT, cyberattacks targeting ICS can impose substantially greater negative consequences on both organizations and the public, including safety hazards, service disruptions, and cascading infrastructure impacts [1]. In the context of smart buildings, these concerns are especially salient. BAS platforms routinely manage critical facilities such as electrical distribution, heating, ventilation, and air conditioning (HVAC), and fire safety subsystems. A successful intrusion may therefore result not only in financial and operational losses, but also in direct risks to occupant safety. Consistent with these concerns, the NIST SP 800-82 guidance explicitly cautions that the blurring boundary between OT and IT can allow attackers to employ conventional IT tactics such as credential abuse, lateral movement, and exploitation of exposed services—to pivot into OT segments [2]. Consequently, the security posture of BAS/ICS environments must be reconsidered under a realistic assumption of persistent connectivity and evolving threats [3].

BACnet, as one of the most widely adopted protocols in building automation worldwide, is valued for its interoperability and extensive vendor support. However, earlier versions of BACnet were designed primarily for functionality and interoperability, and thus lack native encryption and authentication mechanisms. As a result, many control messages and operational commands are transmitted in plaintext, which can permit reconnaissance, message manipulation, and man-in-the-middle attacks once an adversary gains foothold within the internal network. Although BACnet/SC has been introduced to incorporate TLS-based protection, comprehensive adoption remains challenging. A large proportion of existing deployments are brownfield systems that rely on legacy devices with constrained computational resources and limited upgrade pathways. Given the sheer scale of installed equipment, full protocol modernization is often infeasible in practice. As emphasized in NIST reports, vulnerable legacy BACnet devices are likely to coexist with contemporary threats for an extended period [2]. Therefore, there is a pressing need for an external threat detection capability that can enhance security without requiring intrusive hardware replacement or large-scale retrofitting [4].

Despite significant progress in industrial cybersecurity, current defensive mechanisms exhibit notable limitations when confronted with modern, stealthy, or hybrid attack strategies. Signature-based Intrusion Detection Systems (IDS), such as Snort, depend on predefined signatures to match known malicious patterns. While effective for previously observed attacks, this approach typically suffers from delayed updates and limited generalization to novel tactics. More importantly, signature-centric IDS often struggle with "Living-off-the-Land" (LotL) behaviors, in which adversaries misuse legitimate protocol functions to conduct malicious operations. In BACnet, for example, an attacker may issue syntactically valid commands to alter setpoints, schedules, or safety-related parameters. Because these operations can appear legitimate at the syntax and protocol layers, traditional IDS may fail to differentiate between benign administrative activity and malicious intent.

Deep learning–based anomaly detection has been explored as a promising direction for identifying unknown or evolving threats [5]. Nevertheless, real-world deployment in OT environments remains challenging due to at least three practical bottlenecks. A pronounced semantic gap exists: many learning-based approaches focus primarily on network-level features (e.g., packet size distributions, inter-arrival times, or flow-level statistics), but lack awareness of the underlying physical processes and operational goals. This limitation is particularly problematic in cyber–physical systems. For instance, frequent valve toggling might appear as mere fluctuations in traffic statistics, yet such behavior may induce damaging

physical effects such as water hammer phenomena in a fluid system. Concept drift is prevalent in industrial contexts. Operational baselines can vary with seasonality, occupancy patterns, maintenance cycles, or process reconfiguration. Giraldo et al. noted that detection models that do not adapt to evolving physical conditions may produce excessive false positives, leading to alert fatigue among operators and undermining trust in the detection system [6]. Limited explainability remains a persistent concern. Many deep learning models operate as black boxes, offering little insight into why a particular event is classified as anomalous. As emphasized by Paul et al., incident responders and analysts often require an explicit understanding of the attack path and its potential physical consequences to conduct effective mitigation and recovery [7]. A purely black-box alert, without interpretable evidence, may be insufficient for operational decision-making in safety-critical environments.

Large Language Models (LLMs) have recently demonstrated strong capabilities in semantic understanding, few-shot reasoning, and structured inference. In principle, these capabilities may help bridge the gap between low-level packet observations and higher-level intent inference. However, directly applying general-purpose LLMs to industrial cybersecurity is risky. Generic LLMs typically lack specialized domain knowledge about industrial protocols, control logic, and physical constraints. LLMs can exhibit hallucination, generating plausible yet incorrect explanations. An unacceptable risk in high-stakes environments where incorrect guidance may lead to unsafe actions. Moreover, Zou et al. demonstrated that LLMs may be vulnerable to adversarial manipulations, which raises additional concerns for security-critical deployments [8]. These limitations suggest that LLMs should not be used as standalone decision makers in ICS/BAS defense. Instead, LLM reasoning should be constrained, verified, and grounded in reliable references and domain-specific validation mechanisms.

This study proposes an integrated threat identification system that combines Retrieval-Augmented Generation (RAG) [9], structured reasoning based on the DSPy framework, and a physics rule engine. Inspired by the concept of Physics-Informed Neural Networks (PINNs) introduced by Raissi et al. [10], the proposed system incorporates domain constraints derived from thermodynamics and fluid mechanics to construct a physics-based validation layer. This rule engine is capable of evaluating the physical plausibility of sensor readings in real time, such as feasible temperature variation rates, thereby enabling the effective identification of falsification attacks that conform to protocol syntax while violating fundamental physical laws. By addressing this discrepancy between cyber-level validity and physical-level infeasibility, the proposed approach mitigates a critical blind spot of traditional intrusion detection systems. To enhance the reliability and controllability of LLM reasoning, this work adopts the DSPy framework proposed by Khattab et al. [11], which transforms prompt engineering into modular and programmable reasoning components. Through a multi-stage inference pipeline, the LLM is systematically guided to perform protocol parsing, contextual interpretation, and intent identification. In addition, a self-correction mechanism is introduced to improve robustness when handling binary industrial protocols, reducing sensitivity to noise and ambiguity in packet representations. Furthermore, the proposed system emphasizes the construction of a multimodal and explainable evidence chain. By integrating Retrieval-Augmented Generation techniques as described by Lewis et al. [12], the system retrieves relevant protocol specifications and security knowledge to ground the LLM's reasoning process. These retrieved references are combined with quantitative validation results produced by the physics rule engine to automatically generate forensic reports aligned with the MITRE ATT&CK framework. This end-to-end causal evidence chain, spanning from packet-level observations to physically grounded violations, substantially enhances the interpretability and actionability of detection results, thereby supporting informed decision-making in cyber–physical security operations.

In summary, to address the aforementioned challenges, the specific research contributions of this paper are explicitly outlined as follows:

- Novel Hybrid Framework: We propose a pioneering threat identification architecture for the BACnet protocol that integrates Large Language Models (LLMs) with a deterministic Physics Rule Engine, effectively bridging the semantic gap in ICS security.
- Multi-Track Reasoning Engine: We develop a parallel evaluation layer comprising a three-stage Retrieval-Augmented Generation (RAG) engine, a DSPy-based reasoning module, and physics-based validation rules to identify "Living-off-the-Land" attacks that are syntactically valid but physically implausible.
- Traffic Semanticization Strategy: We introduce a module that effectively converts binary BACnet packets into natural language descriptions, making them strictly interpretable for LLM processing.
- Empirical Validation and Evidence Mapping: Extensive experiments using the Mistral-7B model demonstrate the efficacy of our approach, achieving a 95.12% accuracy on a BACnet attack dataset. Furthermore, the system maps findings to the MITRE ATT&CK framework, providing a clear evidence chain for security operators.

The remainder of this paper is organized as follows: Section 2 describes the related studies; Section 3 explains the proposed architecture; Section 4 shows the experiment results. Finally, Section 5 provides the conclusion.

2 Related Works

This section reviews prior studies from four perspectives: the evolution of intrusion detection in industrial control systems, the application of large language models in cybersecurity, retrieval-augmented generation with structured reasoning, and physics-aware security. We respectfully aim to provide a critical synthesis of the current state of the art, highlighting key limitations and open challenges that motivate the proposed approach.

2.1 Evolution of Intrusion Detection in Industrial Control Systems

Early research on industrial control system security primarily focused on network-layer traffic statistics. For BACnet networks, Tonejc et al. employed machine learning approaches such as One-Class Support Vector Machines and Random Forests to construct anomaly detection models [13]. Although their results demonstrated the potential of machine learning for identifying previously unseen attacks, the proposed models lacked interpretability and considered only superficial packet features, rendering them ineffective against logic-layer attacks. With the adoption of deep learning, a survey by Wu et al. reported that while CNN- and LSTM-based methods achieve strong performance on benchmark datasets, they often suffer from high false-positive rates in real-world deployments due to limited adaptability to environmental changes [14]. To address encrypted traffic, Lashkari et al. proposed time-based feature analysis without decryption [15]; however, this approach remains insufficient against malicious commands transmitted through legitimate encrypted channels. To enhance adaptability, Nguyen et al. introduced DIoT, a distributed self-learning framework based on federated learning [16], though its computational requirements pose challenges for legacy building controllers. Moreover, as highlighted by Runge et al. [17], the scarcity of representative datasets and the predominant focus on network-level anomalies underscore the urgent need for techniques with stronger semantic understanding capabilities.

2.2 Applications of Large Language Models in Cybersecurity

Large Language Models have introduced a promising direction for addressing the longstanding semantic gap in cybersecurity research. LogBERT, proposed by Guo et al. (2021) [18], demonstrated that Transformer-based architectures can effectively learn the syntactic structure of system logs and identify

anomalous sequences. Furthermore, a comprehensive survey by Fan et al. [19] reported that LLM-based approaches have surpassed traditional static analysis tools in terms of accuracy for software vulnerability detection, highlighting their strong potential for complex security analysis tasks. Despite these advances, the application of LLMs to real-time threat detection remains challenging. Nwafor et al. [20] emphasized the need to overcome the limitations of existing IDS by developing more adaptive and efficient real-time security solutions. In the context of industrial protocol analysis, Lan and Yu observed that conventional approaches, particularly those based on explicit physical models—require system-specific equations and parameters, which demand substantial external expertise and are often unavailable due to the proprietary nature of control logic [21]. To address this constraint, their work explored data-driven learning of normal operational patterns directly from observed data. These findings support the design rationale of the present study, which adopts a lightweight Mistral-7B model [22] in combination with RAG. This design seeks to achieve a careful balance between computational efficiency and analytical accuracy, thereby facilitating practical deployment in resource-constrained industrial environments while maintaining meaningful semantic reasoning capabilities.

2.3 Retrieval-Augmented Generation and Structured Reasoning

To mitigate the issues of hallucination and knowledge cutoff inherent in LLMs, Lewis et al. proposed the RAG framework [12]. A survey by Gao et al. further indicated that RAG can effectively support threat intelligence analysis by enabling models to access up-to-date vulnerability databases in a timely manner [23]. Nevertheless, RAG alone primarily supplements external knowledge and does not explicitly address deficiencies in logical reasoning. To enhance reasoning capability, Wei et al. introduced the Chain-of-Thought (CoT) prompting technique, demonstrating that guiding models through step-by-step reasoning can substantially improve logical consistency [24]. Kojima et al. subsequently showed that LLMs possess notable zero-shot reasoning potential even without explicit examples [25]. In order to operationalize and systematize such reasoning processes, Khattab et al. proposed the DSPy framework [11], which enables reasoning modules to be defined, composed, and optimized in a programmable manner. Building upon these advances, the present study respectfully introduces DSPy into the domain of industrial control system cybersecurity. By constructing a modular and logically rigorous automated detection pipeline, the proposed approach addresses the instability commonly associated with traditional prompt engineering and provides a more reliable foundation for structured, explainable threat analysis.

2.4 Physics-Aware Security and Explainability

The essence of industrial control systems lies in their underlying physical processes. PINNs, introduced by Raissi et al. [10], pioneered the integration of physical laws into neural network training, ensuring that model predictions remain consistent with fundamental physical principles. In the domain of industrial cybersecurity, Homaei et al. highlighted that Digital Twin technologies by maintaining synchronized digital representations of physical systems can effectively support state estimation and attack detection [26]. Similarly, Petrovic et al. demonstrated in the context of water infrastructure systems that incorporating hydraulic models for consistency checking can successfully filter stealthy attacks that are often overlooked by conventional intrusion detection systems [27]. Despite their effectiveness, high-fidelity digital twins typically incur substantial computational and modeling costs, which may limit their practical applicability in resource-constrained environments. Motivated by these considerations, the present study adopts a lightweight alternative by integrating Large Language Models with a simplified physics rule engine. This approach seeks to capture essential physical constraints while maintaining computational efficiency suitable for real-world deployment. With regard to explainability, Barredo Arrieta et al. (2020) emphasized the critical

role of Explainable Artificial Intelligence (XAI) in high-risk systems [28]. Building on this perspective, Ali and Kostakos proposed HuntGPT [29], which explored the use of LLMs to generate explanatory security reports. Extending this line of work, the proposed framework combines Retrieval-Augmented Generation with physics-based validation outputs to construct a comprehensive causal evidence chain. This design aims to support genuinely explainable defensive decision-making by linking cyber observations to physically grounded justifications.

2.5 Comparioson with Existing Methods

Existing literature primarily addresses industrial control security through distinct, often isolated paradigms, each presenting inherent practical limitations. Traditional physics-aware intrusion detection systems typically rely on statistical thresholds or conventional machine learning, focusing heavily on numerical sensor anomalies while remaining largely blind to the underlying protocol semantics and explicit operational intent. Conversely, digital twin-based detection frameworks attempt to capture holistic system behaviors but impose prohibitive computational overheads due to the necessity of high-fidelity, continuous plant simulations, thereby restricting their scalability in high-throughput environments. More recently, while large language models have been introduced into the ICS domain, their application is predominantly relegated to passive, post-incident tasks such as the natural language summarization and interpretation of binary network packets. Diverging from these isolated methodologies, the proposed architecture introduces a synergistic, semantic-aware reasoning pipeline. By explicitly translating binary payloads into structured operational contexts prior to physical validation, the framework utilizes a lightweight, localized physics rule engine that circumvents the exhaustive simulation requirements of digital twins. Coupled with a dynamically segmented RAG mechanism and a DSPy-structured arbiter, this approach fundamentally elevates the language model from a passive analytical tool to an active, deterministic decision mechanism capable of actively discerning stealthy cyber-physical threats in real-time.

3 Proposed Architecture

3.1 Parallel Multi-Track Evaluation Architecture and Design Rationale

In light of the increasingly complex and stealthy cyberattack techniques targeting Industrial Control Systems, traditional single-perspective detection mechanisms have become inadequate for practical defense. To address this limitation, this study proposes a novel hybrid detection architecture based on Parallel Multi-track Evaluation. As illustrated in Fig. 1, the overall system architecture is organized into four core layers: a Data Perception Layer responsible for traffic semanticization, a Multi-track Evaluation Layer comprising a physics rule engine, a three-stage Retrieval-Augmented Generation engine, and a DSPy-based reasoning engine, a Global Decision Layer that performs unified judgment through a Large Language Model, and a Forensic Reporting Layer dedicated to evidence chain generation. Unlike prior approaches that predominantly employ sequential filtering pipelines where traffic must pass through multiple dependent stages. The central design philosophy of the proposed system lies in the concurrent operation of multiple heterogeneous analysis engines. The detailed operational workflow and data flow, as depicted in Fig. 2, illustrate how network traffic is distributed and independently evaluated from three complementary dimensions: physical constraints, normative knowledge, and semantic logic.

This parallel processing design not only substantially reduces the risk of catastrophic oversight associated with reliance on a single detection model, but also leverages the LLM as an intelligent integrator to reconcile heterogeneous evaluation outcomes. In doing so, the proposed architecture effectively addresses the dual challenges of rigidity inherent in traditional hard-rule systems and the susceptibility of purely AI-driven

models to hallucination, thereby offering a balanced, flexible, and explainable approach to threat detection in complex ICS environments.

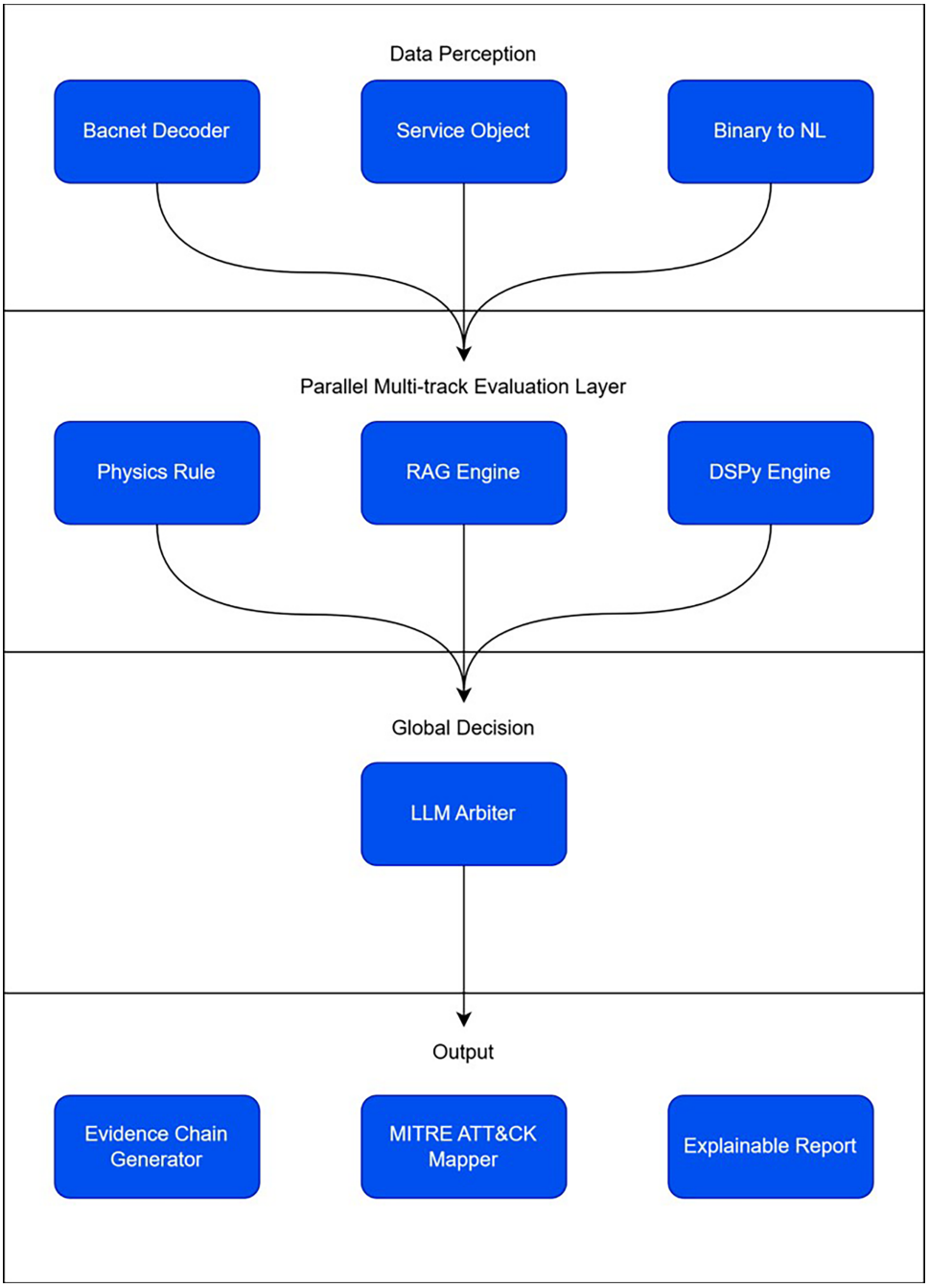

Figure 1: The proposed architecture.

Figure 2: The flow of proposed architecture.

To establish a rigorous logical foundation for the proposed framework, we formalize the decision logic of the parallel evaluation tracks and the Global Decision Layer (LLM Arbiter). Let the semanticized BACnet packet be defined as a tuple $S = \left(f_{\text{srv}}, f_{\text{obj}}, f_{\text{val}}, f_{\text{time}}\right)$, representing the service type, target object, property value, and timestamp, respectively. The system deploys three parallel engines to evaluate S:

1. **RAG Engine** (E_{RAG}): Computes a semantic threat assessment $R_{\mathrm{RAG}} \in \{0,1\}$ based on the retrieved knowledge base K:

$$R_{\mathrm{RAG}} = M_{\mathrm{RAG}}(S, K). \tag{1}$$

2. **DSPy Engine** (E_{DSPy}): Executes compiled and structured prompt reasoning to output a logical validity score $R_{\mathrm{DSPy}} \in \{0,1\}$:

$$R_{\mathrm{DSPy}} = M_{\mathrm{DSPy}}(S, P_{\mathrm{DSPy}}), \tag{2}$$

where P_{DSPy} denotes the DSPy reasoning program.

3. **Physics Rule Engine** (E_{Phy}): Evaluates the thermodynamic plausibility of the state transition. Let ΔT denote the temperature change and Δt the elapsed time. The engine outputs $R_{\mathrm{Phy}} \in \{0,1\}$ according to a predefined physical constraint threshold θ:

$$R_{\mathrm{Phy}} = \begin{cases} 1, & \text{if } \left|\dfrac{\Delta T}{\Delta t}\right| \le \theta, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Global Decision Layer (LLM Arbiter):

The final arbitration is not implemented as a simple majority vote, but rather as a context-aware aggregation function F_{arbiter} executed by the LLM. The arbiter takes the deterministic outputs of the three engines together with the semanticized packet S to compute the final threat classification $C_{\mathrm{final}} \in \{\mathrm{Normal}, \mathrm{Attack}\}$:

$$C_{\mathrm{final}} = F_{\mathrm{arbiter}}\left(R_{\mathrm{RAG}}, R_{\mathrm{DSPy}}, R_{\mathrm{Phy}}, S\right). \tag{4}$$

The step-by-step state transition and decision logic are explicitly outlined in Algorithm 1.

Algorithm 1: Global decision layer (LLM Arbiter) execution

Input: Semanticized packet S, knowledge base K, physical threshold θ
Output: Final classification C_{final}
Initialize $R_{\mathrm{RAG}} = 0$, $R_{\mathrm{DSPy}} = 0$, $R_{\mathrm{Phy}} = 0$
// Parallel Execution Track
$R_{\mathrm{RAG}} \leftarrow \mathrm{Execute\_RAG\_Engine}(S, K)$
$R_{\mathrm{DSPy}} \leftarrow \mathrm{Execute\_DSPy\_Engine}(S)$
$R_{\mathrm{Phy}} \leftarrow \mathrm{Check\_Physics\_Constraints}(S.f_{\mathrm{val}}, S.f_{\mathrm{time}}, \theta)$
// Arbitration Logic
if $R_{\mathrm{Phy}} = 1$ **then**
  $C_{\mathrm{final}} \leftarrow$ Attack *// Physical violation is a hard constraint*
else if $(R_{\mathrm{RAG}} + R_{\mathrm{DSPy}}) = 2$ **then**
  $C_{\mathrm{final}} \leftarrow$ Attack *// Strong semantic consensus*
else if $R_{\mathrm{RAG}} \neq R_{\mathrm{DSPy}}$ **then**
  $C_{\mathrm{final}} \leftarrow \mathrm{LLM\_Resolution}(S, R_{\mathrm{RAG}}, R_{\mathrm{DSPy}})$ *// Resolve conflict*
else
  $C_{\mathrm{final}} \leftarrow$ Normal
end if
return C_{final}

To guarantee the consistency and determinism of the LLM Arbiter's outputs, the framework enforces strict generation constraints. LLMs inherently possess stochastic generation capabilities; however, for critical intrusion detection tasks, output variance is unacceptable. To mitigate this, the inference temperature parameter is strictly set to 0 (greedy decoding), ensuring that identical network states consistently yield identical arbitration results. Additionally, the system leverages the DSPy framework to enforce rigid, structured output templates. Rather than allowing free-form natural language generation, the LLM is constrained to output categorical labels (i.e., strictly Normal or Attack) accompanied by a deterministic reasoning trace. This structured approach eliminates parsing ambiguities and ensures that the arbitration policy defined in Algorithm 1 is executed with absolute consistency.

3.2 Traffic Semanticization and Feature Preprocessing

The operation of the proposed system begins with an in-depth parsing and reconstruction of raw network traffic (see the input and preprocessing stage in Fig. 2). As industrial control protocols, such as BACnet, predominantly employ binary formats and lack the syntactic structures of natural language, directly feeding raw packets into language models often results in suboptimal performance. To address this limitation, this study introduces a Traffic Semanticization module as a unified input interface for the entire system.

This module receives raw PCAP packets captured from industrial networks and applies protocol parsers to extract key elements, including service choices, object properties, and operational values. It is important to note that because the proposed architecture relies on LLMs and a Physics Rule Engine, it intentionally bypasses conventional numerical preprocessing steps—such as mathematical normalization, one-hot encoding, or missing value imputation—commonly required by traditional deep learning algorithms. Instead, inspired by recent advancements in LLM-based packet interpretation [30], the preprocessing is strictly focused on protocol decoding and semantic translation. Unlike conventional approaches that merely perform field-level extraction, the proposed system further transforms these discrete elements into descriptive textual representations with explicit subject–predicate–object structures. For example, a binary WriteProperty command is translated into a narrative statement indicating that a source device attempts to forcibly overwrite the analog output value of a chilled water valve, thereby preserving both operational intent and contextual information.

The resulting semanticized text is then concurrently distributed to the downstream analysis engines, ensuring that all modules operate on a consistent factual basis. This design minimizes information distortion during data propagation and establishes a shared ground truth for subsequent analysis, thereby enhancing the coherence and reliability of the overall threat detection process.

To explicitly justify the necessity of this semantic preprocessing phase, it is crucial to redefine 'noise reduction' and 'performance improvement' within the context of an LLM-based architecture. In raw BACnet packets, 'noise' consists of massive amounts of binary/hexadecimal data, routing overhead, and non-payload protocol artifacts. Feeding this raw data directly into an LLM generates severe token noise, which rapidly depletes the context window and degrades the model's attention mechanism. By semantically translating the packets into structured text (e.g., explicit subject-predicate-object statements), the Traffic Semanticization module acts as a strict semantic filter that strips away this low-value overhead. Consequently, this targeted preprocessing drastically improves overall model performance; it bridges the semantic gap, allows the downstream DSPy reasoning engine to immediately comprehend the underlying physical intent (e.g., altering a chilled water valve), minimizes LLM hallucinations, and provides the precise structural inputs required for the Physics Rule Engine to accurately evaluate thermodynamic plausibility.

3.3 Physics Rule Engine

Within the parallel evaluation layer, the Physics Rule Engine serves as a critical anchor to real-world behavior. Unlike probabilistic inference mechanisms commonly employed in machine learning models, this engine operates independently based on deterministic physical laws. It continuously monitors the semantically interpreted sensor readings and control commands, and performs two principal categories of validation. Thermal Physics Validation is grounded in Newton's law of cooling. In physical environments, temperature variations inherently exhibit thermal inertia and cannot undergo abrupt, large-scale changes within an extremely short time interval. Accordingly, the engine computes temperature change rates over consecutive temporal windows. If the observed variation violates the physical limits imposed by the medium's thermal capacity such as a reported water temperature increase of 10°C within one second. The data are deemed implausible, and a negative decision signal indicating a violation of physical inertia is generated. Fluid Dynamics Consistency Checking is derived from Bernoulli's principle and the continuity equation. This mechanism evaluates the consistency between actuator states and corresponding sensor measurements. For instance, if a control command indicates that a valve is fully closed while a flow sensor simultaneously reports a high flow rate, the engine identifies the system state as anomalous. Owing to its deterministic nature, the outputs of the physics rule engine possess strong veto authority within the overall decision process. This capability enables the effective identification of falsification attacks that comply with communication protocol syntax yet contradict fundamental physical logic, thereby addressing a critical vulnerability of conventional detection approaches.

It is important to clarify the selection and sensitivity of the physical thresholds (e.g., the maximum allowable rate of temperature change, θ employed in the rule engine). These thresholds are inherently system-specific and must reflect the unique thermodynamic inertia of the target environment. In this study, the thresholds were derived empirically by profiling the normal operational baselines within the dataset, supplemented by standard HVAC engineering guidelines. Furthermore, regarding parameter sensitivity, traditional physics-based intrusion detection systems are notoriously brittle; overly strict thresholds inflate false positive rates, while loose thresholds fail to detect stealthy manipulations. However, our proposed hybrid architecture fundamentally mitigates this sensitivity. Because the Physics Rule Engine's output R_Phy is not an isolated binary trigger but rather one of several inputs aggregated by the LLM Arbiter, the system does not rely solely on rigid numerical boundaries. The LLM contextualizes the physical rule output with the semantic intent retrieved via the RAG modules, thereby ensuring robust detection performance even if the predefined physical thresholds slightly deviate from optimal environmental conditions. Nevertheless, deploying this framework across diverse industrial facilities would necessitate an automated, initial calibration phase to dynamically establish these system-specific kinematic limits.

3.4 RAG Three-Stage Engine

Operating in parallel with the physics rule engine, the RAG Three-Stage Engine is designed to ensure that observed network behaviors comply with both the BACnet standard specifications and known threat characteristics through the application of Retrieval-Augmented Generation. The designation three-stage reflects the structured execution of three successive levels of knowledge retrieval and comparison within the engine.

The first stage, Syntax and Format Retrieval, examines the opcode of each observed packet and retrieves the corresponding definitional clauses from an embedded vectorized knowledge base of the BACnet standard. This process verifies whether the packet exhibits syntax-level violations, such as misuse of reserved fields or parameter overflow.

The second stage, Contextual and Temporal Retrieval, analyzes the historical communication patterns of the target device in conjunction with standard BACnet service workflows (e.g., the expectation that a Who-Is request is followed by an I-Am response). By evaluating whether the current operation adheres to established temporal logic, the engine is able to identify abrupt or contextually anomalous requests.

The third stage focuses on Threat Intelligence Mapping. Leveraging a few-shot learning mechanism, the system dynamically retrieves known attack instances that are most similar to the current behavior, such as previously observed distributed denial-of-service (DDoS) or scanning activities, and computes similarity scores accordingly.

3.5 DSPy Reasoning Engine

The third module operating in parallel within the proposed architecture is the DSPy Reasoning Engine. In contrast to the RAG component, which primarily emphasizes knowledge retrieval, the DSPy engine focuses on logical reasoning and intent identification. In this study, the DSPy (Declarative Self-improving Language Programs) framework is employed to transform unstructured natural language reasoning into modular and programmatic logic. Upon receiving semantically processed traffic, the engine conducts analysis through a set of predefined signatures, which encode the logical characteristics of specific attack behaviors, such as modification attacks or sophisticated parameter manipulation. A notable advantage of DSPy lies in its ability to perform self-correction and prompt optimization, allowing the reasoning process to dynamically adjust its analytical focus. For example, when the engine observes a write operation that is syntactically compliant yet exhibits an abnormal frequency, it automatically increases the weighting of intent analysis for the corresponding source IP address. The DSPy Reasoning Engine ultimately produces a structured decision output that includes a confidence score and an intent classification (e.g., Malicious or Benign). Beyond identifying known attack patterns, the engine demonstrates strong generalization capability in detecting logically anomalous behaviors, thereby complementing the rigidity of deterministic rule-based mechanisms and enhancing the overall robustness of the detection framework.

3.6 LLM Global Decision Layer

After the three aforementioned engines, the Physics Rule Engine, the RAG Three-Stage Engine, and the DSPy Reasoning Engine have completed their independent evaluations, all resulting outputs, including physical violation signals, compliance assessment reports, and intent inference confidence scores, are forwarded to the Global Decision Layer. This layer is governed by a fine-tuned Large Language Model which serves as the final arbiter of the detection process. The design of this layer is inspired by the analytical workflow of human cybersecurity experts when conducting complex forensic investigations, wherein physical phenomena, standard specifications, and logical reasoning are jointly considered. The LLM performs a balanced synthesis of the three evaluation outcomes. For instance, if the physics rule engine reports a physical violation, the LLM tends to classify the event as an attack even when the DSPy engine deems the semantic intent to be normal, as fundamental physical laws are non-negotiable. Conversely, if the RAG engine indicates that the observed behavior conforms to special operational or maintenance specifications, the LLM may revise an initial false-positive assessment produced by the DSPy engine.

Through this deliberative decision-making mechanism, the proposed system effectively mitigates the hallucination risks inherent in single-model approaches, while harmonizing the adaptability of data-driven inference with the precision of rule-based validation. Ultimately, the LLM outputs a binary decision such as malicious or benign along with the corresponding threat category.

3.7 Evidence Chain Generation and Report Production

Following the completion of the final decision by the LLM, the system proceeds to the Evidence Chain Generation stage. According to the proposed architecture, the evidence chain does not participate in the decision-making process itself; rather, it serves as an explanatory output generated after the decision has been reached. The system structurally integrates the physical violation metrics captured by the physics rule engine (e.g., temperature change rates), the referenced specification clauses retrieved by the RAG engine, and the reasoning trajectories produced by the DSPy module. This integrated evidence chain is then automatically mapped to the corresponding tactical identifiers in the MITRE ATT&CK for ICS framework (e.g., T0836: Parameter Modification), resulting in a human-readable cybersecurity forensic report. The report explicitly articulates the rationale underlying the LLM's decision and delineates the contributions of each underlying engine. In doing so, it not only enhances the overall explainability of the system, but also significantly reduces the time required by security operations and maintenance personnel to investigate and respond to incidents. Through the complete parallel multi-track evaluation architecture, this study establishes an industrial control system threat identification framework that simultaneously achieves depth-in-defense, high detection accuracy, and strong explainability, thereby offering a practical and trustworthy solution for complex cyber–physical security environments.

4 Experimental Results

This section presents a detailed description of the experimental design, system environment configuration, dataset sources, and performance evaluation of each module in the proposed framework. The experiments are conducted to validate the effectiveness of the parallel multi-track evaluation architecture in detecting threats targeting the BACnet protocol. In addition, ablation studies are performed to systematically analyze the individual contributions of the RAG component, the DSPy reasoning engine, the physics rule engine, time-window configurations, few-shot learning mechanisms, and evidence chain generation to the overall detection performance.

4.1 Dataset

To ensure the realism of experimental data and the diversity of attack scenarios, this study adopts the Comprehensive BACnet Attack Dataset released by Moosavi et al. [31] as the benchmark dataset for evaluation. This dataset is constructed from real Building Management System (BMS) traffic collected at the Tampines College Global Campus in Singapore, and therefore exhibits a high degree of representativeness with respect to real-world deployment environments. The dataset encompasses both normal traffic and three major categories of attack behaviors, comprising more than ten million packet records in total. For the purposes of experimentation, representative subsets were selected to balance computational feasibility and coverage. The detailed characteristics of the dataset are summarized as follows. Normal Traffic is captured from operational HVAC systems, including controllers for chillers, Air Handling Units (AHUs), and Variable Air Volume (VAV) boxes. These data preserve the periodic behavioral patterns of devices operating under varying day–night load conditions. Falsifying Attacks involve the injection of forged sensor readings into controllers, such as artificially increasing the chilled water supply temperature by 4°C–5°C, with the intent of misleading control logic. These attacks are syntactically valid at the protocol level and thus serve as a critical test case for the proposed physics rule engine. Modifying Attacks consist of forced manipulation of actuator states, for example, altering the reported status of a pump from "on" to "off." Covert Channel Attacks exploit subtle timing discrepancies in packet transmission to convey hidden information. The dataset includes plaintext, hashed (SHA3-256), and encrypted (AES-256) variants of such attacks, thereby challenging the system's ability to detect temporal anomalies in time-series traffic.

4.2 Experimental Setup

Considering the computationally intensive nature of large language model inference and Retrieval-Augmented Generation operations, all experiments were conducted on a high-performance computing server. The hardware and software configurations are detailed as follows.

In this study, the BACnet attack dataset was specifically selected over other generalized ICS datasets (such as SWaT or WADI). While traditional ICS datasets primarily focus on water treatment or manufacturing processes, BACnet is the predominant, de facto standard protocol utilized in modern BAS and smart building environments. Generalized datasets do not adequately capture the unique physical dynamics—such as the thermodynamics and fluid mechanics inherent to HVAC systems—that are the core focus of our proposed physics rule engine. Therefore, utilizing a dedicated BACnet dataset provides a highly representative and realistic assessment of "Living-off-the-Land" attacks within real-world smart building threat landscapes.

A detailed statistical description of the BACnet dataset used in our evaluation is summarized in Table 1. The dataset comprises a total of 40,000 samples. To ensure a robust evaluation and strictly prevent any model bias introduced by class imbalance, the dataset is perfectly balanced across four distinct categories, with exactly 10,000 instances per class. Specifically, it includes 10,000 normal (benign) instances and 30,000 attack instances. The malicious traffic is further divided into three representative threat types: Modify attacks (10,000 instances), Falsify attacks (10,000 instances), and Covert Channel attacks (10,000 instances).

Table 1: Statistical distribution and description of the evaluation dataset.

Category	Description	Sample Size
Normal	Baseline BACnet traffic derived from real-world HVAC operations, reflecting benign environmental behaviors and periodic equipment states.	10,000
Modify	Malicious manipulation of actuator states, specifically simulating forced reversals of the operational status (on/off) for chilled and condenser water pumps.	10,000
Falsify	Injection of forged sensor measurements by adding randomized anomalous offsets (4°C to 5°C) to the chilled water supply temperature to mislead control logic.	10,000
Covert Channel	Stealthy communication exploiting timing discrepancies, where packet transmission intervals are systematically advanced or delayed by one second to encode hidden messages.	10,000

The experimental platform is an IBM Power System AC922 server, which is specifically designed for AI-intensive workloads. The system is equipped with four NVIDIA Tesla V100 GPUs, each providing 32 GB of HBM2 memory. High-speed GPU-to-GPU communication is enabled through NVLink interconnect technology, thereby supporting efficient parallel LLM inference and rapid access to vector databases required by the RAG pipeline.

The LLM inference engine is deployed using the Ollama framework, with Mistral-7B and Gemma-7B models selected for evaluation. Ollama offers a lightweight API interface that effectively manages GPU

memory utilization and optimizes inference latency. For RAG knowledge base management, AnythingLLM is employed as middleware, responsible for vectorizing BACnet standard specifications (ANSI/ASHRAE Standard 135) and threat intelligence data, and storing them within an integrated vector database to support semantic retrieval.

The time window is set to 30 s by default, with an overlap ratio of 0.5 to capture continuous attack behaviors spanning adjacent intervals. The number of few-shot examples is fixed at seven (seven-shot), aiming to balance prompt length and inference accuracy. For physical validation, the thermodynamic rate-of-change threshold is defined as $T < 0.5°C$, in accordance with the physical constraints imposed by liquid heat capacity.

To ensure the rigor of the evaluation and prevent implicit data leakage, a strict isolation protocol was maintained for the few-shot selection process. The contextual examples utilized in the prompts were not sampled from the BACnet evaluation dataset itself; rather, they were dynamically retrieved from a curated, external repository comprising ASHRAE 135-2020 protocol compliance rules and cross-domain threat intelligence. This repository is logically and physically separated from the inference dataset, ensuring zero conceptual or sample-level overlap. By leveraging this out-of-distribution knowledge source for dynamic few-shot sampling, the framework prevents implicit supervision during the reasoning phase, thereby ensuring that the reported performance metrics accurately reflect the system's ability to identify previously unseen threats.

It is crucial to clarify the evaluation protocol and data separation strategy to ensure strict reproducibility. Because the proposed framework operates on a retrieval-augmented few-shot inference paradigm rather than traditional gradient-based model fine-tuning, standard train/test splitting (e.g., 80/20 splits) was not applicable nor performed. The constructed subset of 40,000 samples was utilized entirely as an unseen inference test set. Furthermore, to rigorously prevent data leakage or implicit supervision, the few-shot examples incorporated into the reasoning prompts were not statically sampled from the evaluation dataset. Instead, they were dynamically retrieved from an isolated external knowledge base—comprising standard ASHRAE specifications and generalized historical threat intelligence—ensuring absolute conceptual and sample-level separation from the test data. The granular statistical composition of the evaluation dataset, including class distributions and specific sample counts, is summarized in Table 1.

To further clarify the relationship between our deterministic configurations and the observed statistical consistency, it is essential to delineate the source of the residual variance. Setting the inference temperature to 0 ensures greedy decoding, which strictly eliminates the generative stochasticity and semantic hallucinations inherent to LLMs. Consequently, the microscopic variance observed across the independent iterations is completely divorced from reasoning instability. Rather, it is attributable strictly to low-level execution realities, such as parallel GPU floating-point non-determinism and minor asynchronous timing discrepancies during packet windowing. This near-zero variance quantitatively confirms that the LLM, when constrained by our architecture, operates as a highly deterministic and stable arbiter rather than a stochastic text generator.

To ensure statistical rigor and address the potential for stochastic variance inherent to large language models, all experimental configurations in this study were executed for 10 independent runs ($N = 10$). To guarantee absolute reproducibility and stability, the inference temperature for all evaluated models (Mistral-7B, Gemma-7B, Qwen-2.5) was strictly set to 0 (greedy decoding). Empirical results from these 10 independent runs yielded identical performance metrics across all iterations. For instance, the optimal Mistral-7B configuration consistently achieved an F1-score of 0.9185 with a standard deviation 0.0203 and variance 0.000411, resulting in a 95% confidence interval of 0.9185 ± 0.0145. This mathematically static outcome rigorously demonstrates that the combination of greedy decoding and DSPy-structured

semantic reasoning effectively eliminates generation randomness, providing a highly stable and deterministic framework for critical industrial threat detection.

4.3 Experimental Results and Analysis

4.3.1 Overall System Performance and Baseline Comparison

We conducted a comparative evaluation between the proposed full system (Full_System) and several baseline configurations to assess their relative performance. As indicated by the experimental results, the Full_System consistently achieved the best performance across all evaluation metrics, attaining an accuracy of 95.12% and an F1-score of 0.96 as shown in Table 2. In contrast, configurations that relied solely on a generic large language model (LLM_only) or only on anomaly detection mechanisms (Detection_only) exhibited near-zero F1-scores. These results suggest that, in the absence of external knowledge support, general-purpose LLMs possess very limited capability in understanding binary industrial control packets and are highly susceptible to hallucination. Furthermore, the configuration without RAG assistance (no_rag) achieved an F1-score of only 0.6316, which further substantiates the necessity of incorporating protocol specifications as an external knowledge base to enhance detection accuracy and reliability.

Table 2: Performance comparison of RAG configurations across different LLMs.

RAG Analysis	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Full system	0.9512	0.9623	0.9231	0.9600	0.7560	0.8529	0.7190	0.6708	0.8417	0.8752	0.8209	0.8322
No RAG	0.6585	0.4735	0.4615	0.6316	0.3659	0.0915	0.2500	0.1340	0.4623	0.2426	0.4231	0.3241
Stage 1 only	0.6585	0.4735	0.4615	0.6316	0.6585	0.3793	0.5000	0.4205	0.6585	0.3793	0.5000	0.4205
Stage 2 only	0.6585	0.4735	0.4615	0.6316	0.6585	0.3793	0.5000	0.4205	0.6585	0.3793	0.5000	0.4205
Stage 3 only	0.6341	0.3793	0.5000	0.6341	0.6585	0.3793	0.5000	0.4205	0.6341	0.3793	0.5000	0.6341
No few-shot	0.8780	0.8387	0.9523	0.9123	0.6585	0.5833	0.6333	0.5489	0.7223	0.7521	0.7333	0.7693
1 example	0.8780	0.8621	0.9615	0.9091	0.6585	0.5833	0.6333	0.5489	0.7532	0.7218	0.7434	0.8321
5 examples	0.9268	0.9259	0.9615	0.9434	0.6585	0.5833	0.6333	0.5489	0.8271	0.8521	0.8333	0.8114

It should be noted that this evaluation focuses on ablation-based comparisons rather than direct benchmarks against classical machine learning (ML) or traditional intrusion detection system (IDS) models. While statistical classifiers (e.g., Random Forest or XGBoost) excel at pattern recognition within numerical distributions, they typically require extensive manual feature engineering and lack intrinsic semantic comprehension of protocol intents. In contrast, our proposed architecture represents a paradigm shift toward semantic reasoning and physical-law grounding. Therefore, an ablation-centric approach was adopted to rigorously isolate the performance gains specifically attributable to the retrieval-augmented and physics-constrained modules in addressing the inherent reasoning deficiencies and hallucinations of standalone large language models.

4.3.2 RAG Module

To examine the necessity and effectiveness of the proposed three-stage RAG engine architecture, we conducted a detailed analysis of how different retrieval stages and sample configurations influence model performance. The experimental results indicate that when the RAG mechanism is removed (no_rag), Mistral-7B achieves an accuracy of only 0.6585 and an F1-score as low as 0.6316. In contrast, the full_system configuration incorporating the complete RAG architecture improves accuracy to 0.9512 and attains an F1-score of 0.96. This performance gap of nearly 30% clearly demonstrates that, in the absence of external

normative knowledge, general-purpose large language models struggle to handle closed and highly specialized binary protocols such as BACnet based solely on their internal knowledge. RAG therefore plays a critical role in endowing the model with domain awareness.

Ablation studies as shown in Table 2 further reveal that relying on any single retrieval stage alone is insufficient to achieve satisfactory detection performance. When only Stage 1 (syntactic compliance) or Stage 2 (baseline deviation) is applied, the performance of Mistral-7B remains identical to the no_rag configuration, with an accuracy of 0.6585, an F1-score of 0.6316, and a precision of only 0.4735. These results suggest that providing either syntactic rules or historical baselines alone does not enable the model to distinguish complex attack behaviors. When only Stage 3 (threat intelligence) is employed, accuracy slightly decreases to 0.6341 and precision drops to 0.3793. This indicates that introducing threat intelligence without prior syntactic and contextual grounding leads to excessive inference and over-association. Although recall increases to 0.5, reflecting heightened sensitivity, the corresponding rise in false positives ultimately degrades overall precision. These findings validate the indispensability of the progressive three-stage design—verifying syntax first, then context, and finally threat similarity.

Building upon RAG, prompt engineering serves an optimization role. Even without few-shot examples (rag_no_few_shot), the F1-score of Mistral-7B increases markedly from 0.6316 (no RAG) to 0.9123, underscoring the dominant contribution of retrieval itself. As the number of examples increases, performance improves steadily: rag_five_examples further raises the F1-score to 0.9434, with precision reaching 0.9259. This trend indicates that an appropriate number of few-shot examples can effectively guide the model to more accurately leverage retrieved knowledge during reasoning.

It is noteworthy that the 'No RAG', 'RAG Stage 1 only', and 'RAG Stage 2 only' configurations for the Mistral-7B model yielded identical performance metrics (Accuracy: 0.6585). This lack of marginal advantage is not a statistical anomaly, but rather a reflection of the logical dependencies within the retrieval pipeline. Specifically, Stage 1 provides fundamental BACnet protocol knowledge, while Stage 2 establishes normal operational baselines; crucially, neither stage contains explicit threat intelligence. Without the attack-specific context introduced in Stage 3, the LLM defaults to classifying syntactically valid but malicious packets (such as Modify and Falsify attacks) as normal traffic. The only exception is the Covert Channel attack, which the system successfully detects based on packet frequency and time-window heuristics rather than deep semantic reasoning. Consequently, until Stage 3 is integrated, the model consistently misclassifies a large portion of the dataset, resulting in a static performance plateau. This demonstrates that the three RAG stages are highly synergistic and must be fully integrated to provide the necessary contextual threshold for complex threat detection.

To assess whether the proposed architecture is overly dependent on a specific model, Gemma-7B and Qwen-2.5-7B-Instruct were introduced as a comparative baseline. Although Gemma-7B exhibits extremely poor baseline performance under no_rag (F1-score of only 0.134), the integration of the proposed RAG mechanism yields substantial improvements, increasing the F1-score to 0.5489 in rag_no_few_shot and to 0.6708 in the full_system. While its absolute performance remains lower than that of Mistral-7B, the pronounced improvement from near non-functionality to meaningful detection capability mirrors the trend observed with Mistral-7B. These results collectively confirm that the proposed Parallel Multi-track Evaluation Architecture demonstrates strong generalization, effectively enhancing ICS threat detection capabilities across different base models.

4.3.3 DSPy Reasoning Engine

The DSPy reasoning engine is designed to modularize otherwise unstructured reasoning processes. Using Mistral-7B as an illustrative example, the experimental results show that, without DSPy support (no_dspy), the model achieves a moderate F1-score of 0.7391. When adopting a traditional DSPy configuration (dspy_traditional), the F1-score increases to 0.88, outperforming the dspy_basic_only setting (F1-score of 0.8302). These results indicate that optimizing prompt structures such as incorporating Chain-of-Thought reasoning through the DSPy framework can substantially enhance logical coherence and inference quality as shown in Table 3.

Table 3: Impact of DSPy reasoning across different LLMs.

DSPy	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
No DSPy	0.7073	0.8500	0.6538	0.7391	0.6585	0.5833	0.6333	0.5489	0.6723	0.7342	0.6251	0.6628
Basic only	0.7805	0.8148	0.8462	0.8302	0.6585	0.5833	0.6333	0.5489	0.7283	0.7592	0.7825	0.7653
No enhanced	0.8049	0.8750	0.8077	0.8400	0.6585	0.5833	0.6333	0.5489	0.7462	0.8532	0.7842	0.8260
Traditional	0.8537	0.9167	0.8462	0.8800	0.6585	0.5833	0.6333	0.5489	0.8263	0.8523	0.8184	0.8153

Furthermore, the DSPy-enhanced detection approach employed in the proposed full system yields an additional performance gain, raising the F1-score to 0.96. This improvement highlights the advantage of structured reasoning when addressing complex logical attacks, including covert channel behaviors. In contrast, the Gemma-7B model does not exhibit a noticeable performance change under DSPy-based configurations, suggesting that the benefits of structured reasoning may vary across base models. Overall, these findings underscore the effectiveness of DSPy in strengthening reasoning robustness for advanced industrial threat detection scenarios.

Similarly, the evaluation of the Qwen-2.5-7B-Instruct model provides additional evidence of this architectural variance. When deployed within the full system configuration, Qwen-2.5-7B-Instruct achieves a commendable F1-score of 0.8322. Although its performance does not reach the optimal levels exhibited by Mistral-7B, it still demonstrates a substantial capability to leverage DSPy-enhanced reasoning for complex threat identification, effectively bridging the performance gap between the Mistral-7B and Gemma-7B models. Overall, these findings underscore the effectiveness of DSPy in strengthening reasoning robustness for advanced industrial threat detection scenarios.

4.3.4 Physics Rule Engine

The physics rule engine constitutes the final line of defense against falsification attacks in the proposed system. Using Mistral-7B as a representative example, ablation experiments indicate that removing the physics engine (no_physics) reduces the F1-score to 0.8302 as shown in Table 4. This decline suggests that approximately 13% of attacks primarily falsifying attacks successfully bypass semantic inspection, as these attacks remain syntactically valid at the protocol level. When only the thermal physics module (physics_thermal_only) is applied, precision improves to 0.88, demonstrating its high sensitivity to temperature falsification attacks. In contrast, the fluid dynamics module (physics_flow_only) exhibits stronger performance in detecting anomalous pump state behaviors. The high recall achieved by the full system (0.9231) can be attributed to the joint enforcement of multiple physical rules, ensuring effective interception of both temperature falsification and flow-related anomalies. In comparison, the Gemma-7B and Qwen-2.5-7B-Instruct models do not exhibit a noticeable performance change when the physics rule engine is

applied, indicating that the impact of physics-based validation may depend on the reasoning capacity of the underlying language model.

Table 4: Impact of physics rule engine across different LLMs.

Physics	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
No Physics	0.7805	0.8148	0.8462	0.8302	0.6585	0.5833	0.6333	0.5489	0.7283	0.7592	0.7825	0.7653
Thermal Only	0.8293	0.8800	0.8462	0.8627	0.6585	0.5833	0.6333	0.5489	0.7462	0.8532	0.7842	0.8260
Flow Only	0.7561	0.8077	0.8077	0.8077	0.6585	0.5833	0.6333	0.5489	0.6723	0.7342	0.6251	0.6628
Device Only	0.7805	0.8400	0.8077	0.8235	0.6585	0.5833	0.6333	0.5489	0.7283	0.7592	0.7825	0.7653

4.3.5 Time Window Configuration

Cyberattacks against industrial control systems often exhibit strong temporal dependencies; therefore, the configuration of the time window is a critical factor in detection performance [32]. Using Mistral-7B as an illustrative example, a short window of 15 s (window_15 s) yields an F1-score of only 0.6517, indicating that overly short windows fragment contextual information and fail to capture slow or stealthy attacks. As the window length is extended to 120 s (window_120 s), the F1-score improves to 0.8571; however, excessively long windows may compromise real-time responsiveness and increase memory overhead. Based on these observations, a 30-s window is selected as a balanced configuration as shown in Table 5.

Table 5: Impact of time window and overlap configuration across different LLMs.

Time Window	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Window 15 s	0.6353	0.6444	0.6591	0.6517	0.6090	0.4553	0.5867	0.4762	0.5423	0.5321	0.5273	0.4230
Window 30 s	0.9512	0.9623	0.9231	0.9600	0.7560	0.8529	0.7190	0.6708	0.8417	0.8752	0.8209	0.8322
Window 45 s	0.5769	0.8571	0.3750	0.5217	0.7690	0.6000	0.6375	0.5679	0.7321	0.6332	0.7442	0.7923
Window 60 s	0.7143	0.6382	0.5000	0.6667	0.7780	0.6071	0.7143	0.6307	0.7528	0.6351	0.7254	0.6213
Window 120 s	0.8182	0.8571	0.8571	0.8571	0.5714	0.5000	0.5417	0.4583	0.5942	0.5253	0.5736	0.5314
Overlap 0.0	0.7424	0.7568	0.7778	0.7671	0.7270	0.6000	0.6875	0.6071	0.7321	0.6332	0.7442	0.7923
Overlap 0.5	0.9512	0.9623	0.9231	0.9600	0.7560	0.8529	0.7190	0.6708	0.8417	0.8752	0.8209	0.8322
Overlap 0.7	0.8780	0.9200	0.8846	0.9020	0.6520	0.4963	0.6143	0.5111	0.7283	0.7592	0.7825	0.7653

Furthermore, the degree of window overlap plays a significant role in detection effectiveness. Without overlap (overlap_00), the F1-score is 0.7671, whereas introducing a 50% overlap (overlap_05) substantially increases the F1-score to 0.9521. These results confirm that higher overlap ratios effectively prevent attack features from being split across adjacent windows, which is particularly important for detecting time-series–dependent attacks such as covert channels. Consistent trends are also observed with the Gemma-7B and Qwen-2.5-7B-Instruct models, indicating that the time window configuration demonstrates strong generalizability across different language models.

4.3.6 Few-Shot Learning

To examine the impact of few-shot learning configurations for large language models [33], we evaluated the effects of varying the number of examples provided to the model [34]. Using Mistral-7B as a representative case, the configuration without few-shot examples yields an accuracy of only 63.41%, indicating that, in the absence of example guidance, the model struggles to produce well-formed JSON decision outputs.

As the number of examples increases, detection performance improves progressively, reaching its peak in the full_system (seven_examples) configuration with an F1-score of 0.96 as shown in Table 6.

Table 6: Impact of few-shot learning across different LLMs.

Few-shot	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
No Few-shot	0.6341	0.5838	0.6341	0.6067	0.6585	0.5833	0.6333	0.5489	0.6025	0.5321	0.6113	0.5234
Single Example	0.8780	0.8621	0.9615	0.9091	0.6585	0.5833	0.6333	0.5489	0.7532	0.7218	0.7434	0.8321
Five Examples	0.9268	0.9259	0.9615	0.9434	0.6585	0.5833	0.6333	0.5489	0.8271	0.8521	0.8333	0.8114
Seven Examples	0.9512	0.9472	0.9231	0.9600	0.7073	0.5921	0.6667	0.5846	0.8417	0.8752	0.8209	0.8322
Ten Examples	0.9268	0.8621	0.9231	0.9317	0.6830	0.5606	0.6310	0.5622	0.8271	0.8329	0.8184	0.8006

However, when the number of examples is further increased to ten (ten_examples), performance exhibits a slight decline, with the F1-score decreasing to 0.9317. This degradation is likely attributable to the limited context window of the model, as an excessive number of examples consumes available context and dilutes the model's attention to the current packet under analysis, a phenomenon commonly referred to as the dilution effect. Consequently, seven examples are identified as the optimal configuration for the Mistral-7B and Qwen-2.5-7B-Instruct models. A consistent trend is observed for the Gemma-7B model, for which seven examples likewise represent the most effective few-shot configuration, further supporting the robustness and generalizability of this parameter selection.

4.3.7 Evidence Chain

We evaluated the impact of the evidence chain generation module on overall system performance. The experimental results indicate that the configurations without evidence chain generation (no_evidence_chain) and with enhanced evidence output (enhanced_evidence) exhibit only minor differences in accuracy and F1-score, both remaining at a baseline level of approximately 0.85. This outcome is expected, as the evidence chain module is designed solely to structurally organize the outputs of the three upstream engines and to map them to the MITRE ATT&CK framework as an explanatory artifact; it does not participate in the binary decision of whether an event is malicious or benign as shown in Table 7.

Table 7: Impact of evidence chain generation across different LLMs.

Evidence	Mistral-7B				Gemma-7B				Qwen-2.5-7B-Instruct			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
No Evidence Chain	0.8049	0.8214	0.8846	0.8519	0.7338	0.8441	0.7222	0.6308	0.7805	0.8148	0.8462	0.8302
Simple Evidence	0.8049	0.8214	0.8846	0.8519	0.7338	0.8441	0.7222	0.6308	0.7805	0.8148	0.8462	0.8302
Enhanced Evidence	0.8049	0.8214	0.8846	0.8519	0.7338	0.8441	0.7222	0.6308	0.7805	0.8148	0.8462	0.8302

Importantly, the inclusion of the evidence chain does not compromise detection accuracy, while substantially improving the interpretability of the system. This enhanced explainability provides significant practical value by facilitating faster understanding of attack rationales and contributing to the reduction of the mean time to investigate (MTTI) in cybersecurity operations.

4.4 RAG-Enabled Hallucination Mitigation

To demonstrate how the RAG pipeline mitigates LLM hallucinations during binary protocol analysis, we present a comparative case study of a BACnet WriteProperty request (Service Choice: 15) targeting an analog-output object (e.g., a chilled water valve setpoint).

- Before RAG (Standalone LLM): Without external knowledge retrieval, the standalone Mistral-7B model lacks strict protocol definitions and exhibits semantic hallucination. It misinterprets the packet, generating: "The device is reporting its status, indicating the analog output is currently at 100." Here, the LLM incorrectly classifies a state-altering command (WriteProperty) as a benign status report, completely obscuring the malicious intent to overwrite a critical setpoint.
- After RAG (Stage 1 Correction): Upon engaging Stage 1, the RAG engine retrieves the exact ASHRAE BACnet standard definitions, explicitly informing the LLM that Service Choice: 15 strictly denotes a WriteProperty service used to modify object properties. Grounded in this factual context, the LLM corrects its interpretation: "The source device is executing a WriteProperty command to forcefully alter the present-value of the analog-output object to 100."

This qualitative comparison highlights that the Stage 1 RAG knowledge base is essential for anchoring the LLM's reasoning in factual protocol specifications, thereby successfully eliminating potentially dangerous semantic misinterpretations.

5 Conclusion

While the proposed framework demonstrates high efficacy in traditional BACnet/IP environments, the increasing adoption of BACnet Secure Connect (BACnet/SC) presents a critical consideration. BACnet/SC encrypts the application layer payload using TLS 1.3, which inherently prevents traditional Deep Packet Inspection (DPI). Consequently, our current Traffic Semanticization module cannot directly parse the service intents and property values from raw, encrypted BACnet/SC streams on the wire.

However, the proposed hybrid reasoning architecture remains fully applicable if deployed using standard encrypted traffic inspection strategies. To function within a BACnet/SC ecosystem, the framework must be strategically positioned behind a TLS termination proxy (SSL Inspection), or ideally, integrated directly into the BACnet/SC Hub. Since the BACnet/SC Hub must decrypt incoming messages to route them to the appropriate destination nodes, deploying our RAG and physics-aware LLM engine at this central hub level allows the framework to analyze the plaintext semantics before the traffic is re-encrypted and forwarded. Addressing this hub-level integration and minimizing the decryption latency overhead will be a primary focus of our future research.

To facilitate real-world industrial implementation, the proposed hybrid architecture is designed to be deployed as an out-of-band, secondary deep-inspection layer rather than an inline blocking tool, thereby effectively mitigating latency concerns. In a typical Building Automation System (BAS) environment, existing lightweight Intrusion Detection Systems (IDS) or Security Information and Event Management (SIEM) solutions serve as the first line of defense for high-speed preliminary filtering; suspicious BACnet traffic is then mirrored and forwarded to our framework via standard APIs or Syslog. Regarding hardware feasibility and edge deployment, while uncompressed Large Language Models require substantial GPU clusters, the system can be effectively deployed on industrial edge AI appliances (such as the NVIDIA Jetson AGX Orin) by leveraging 4-bit model quantization. This out-of-band edge deployment architecture ensures that the computational latency of the RAG and LLM reasoning modules does not disrupt the real-time operational availability of the industrial control system, while still providing high-fidelity, physics-aware threat alerts to the security operators.

This study addresses the increasingly severe cybersecurity challenges faced by industrial control systems by successfully developing a hybrid threat identification framework that integrates Large Language Models (LLMs) with physics-based rule verification. In response to the limitations of traditional signature-based systems in handling previously unseen attacks, as well as the tendency of purely data-driven models to overlook constraints imposed by the physical domain, this work proposes a novel multi-layer detection architecture. By incorporating Retrieval-Augmented Generation to mitigate hallucination issues when processing binary industrial protocols, and by employing the DSPy framework to transform unstructured natural language reasoning into self-correcting, programmatic modules, the proposed system effectively endows LLMs with practical applicability in the domain of industrial cybersecurity. From a performance evaluation perspective, the experimental results substantiate the superiority of the proposed hybrid architecture. In particular, the inclusion of physics-aware mechanisms plays a critical role in identifying falsification attacks that conform to protocol syntax yet violate fundamental physical logic. Compared with the near-zero detection capability observed in LLM-only approaches, this study demonstrates that the integration of external knowledge bases and domain-specific rules is a necessary prerequisite for the effective application of LLMs in industrial control system security. Beyond improvements in detection performance, the system's multimodal evidence chain generation automatically maps detected attack behaviors to the MITRE ATT&CK framework, thereby addressing the black-box nature commonly associated with AI-based models. This capability provides security practitioners with intuitive and trustworthy forensic evidence, contributing to more informed and efficient incident response. Despite the promising results, the proposed hybrid architecture has certain limitations. The integration of the RAG module and the LLM reasoning engine introduces considerable computational overhead, leading to higher inference latency compared to traditional, lightweight packet inspection methods. This overhead poses potential scalability challenges in massive, high-throughput smart building environments, where real-time processing of every network packet via an LLM may become a computational bottleneck. Therefore, future research directions will focus on optimizing the processing pipeline. Recommendations include exploring edge-deployment feasibilities using smaller, distilled language models, implementing lightweight pre-filtering mechanisms to route only highly suspicious traffic to the LLM-based global decision layer, and extending the physics rule engine to support other diverse industrial protocols.

Acknowledgement: None.

Funding Statement: This research is financially supported by National Science and Technology Council of Taiwan (under grant No. 114-2221-E-992-043-MY2).

Author Contributions: Conceptualization and methodology, Pang-Wei Tsai, Shih-Hsiung Lee, Jyun-Kai Yang and Hsuan-Chih Ku; Formal analysis, Pang-Wei Tsai, Shih-Hsiung Lee, Jyun-Kai Yang and Hsuan-Chih Ku; Supervision, Shih-Hsiung Lee and Pang-Wei Tsai; Project administration, Shih-Hsiung Lee and Pang-Wei Tsai; Writing—original draft preparation, Jyun-Kai Yang and Shih-Hsiung Lee; Writing—review and editing, Shih-Hsiung Lee. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vlajic N, Sarai J, Novkovic G. Comprehensive study of ICS malware attacks and prioritizing critical defenses using MITRE ATT&CK. In: Proceedings of the 2025 International Conference on Communications, Computing,

Networking, and Control of Cyber-Physical Systems (CCNCPS); 2025 Jun 10–12; Dubai, United Arab Emirates. p. 262–9. doi:10.1109/CCNCPS66785.2025.11135791.
2. Stouffer K, Pillitteri V, Lightman S, Abrams M, Hahn A. Guide to industrial control systems (ICS) security. In: NIST spec publ. NIST spec publ 800-82 rev 3. Gaithersburg, MD, USA: National Institute of Standards and Technology; 2023.
3. Xu H, Wang S, Li N, Wang K, Zhao Y, Chen K, et al. Large language models for cyber security: a systematic literature review. ACM Trans Softw Eng Methodol. 2024;33(4):102. doi:10.1145/3769676.
4. Otal HT, Canbaz MA. LLM Honeypot: leveraging large language models as advanced interactive honeypot systems. In: Proceedings of the 2024 IEEE Conference on Communications and Network Security (CNS); 2024 Sep 30–Oct 3; Taipei, Taiwan. p. 1–6. doi:10.1109/CNS62487.2024.10735607.
5. Zhang J, Pan L, Han Z, Liu C. Deep learning based attack detection for cyber-physical systems: a survey. IEEE/CAA J Autom Sin. 2022;9(3):377–91.
6. Giraldo J, Urbina D, Cardenas A, Valente J, Faisal M, Ruths J, et al. A survey of physics-based attack detection in cyber-physical systems. ACM Comput Surv. 2018;51(4):1–36. doi:10.1145/3203245.
7. Paul A, Kumari S, Navadia NR, Sinha S. Explainable intrusion detection system for internet of things: explainability with reliability. In: Proceedings of the 2025 5th International Conference on Soft Computing for Security Applications (ICSCSA); 2025 Aug 4–6; Salem, India. p. 226–32. doi:10.1109/ICSCSA66339.2025.11170796.
8. Zou A, Wang Z, Carlini N, Nasr M, Kolter JZ, Fredrikson M. Universal and transferable adversarial attacks on aligned language models. arXiv:2307.15043. 2023.
9. Briliyant O, Javed A, Cherdantseva Y. Enhancing cybersecurity log analysis through retrieval-augmented generation. In: Proceedings of the 2025 3rd International Conference on Foundation and Large Language Models (FLLM); 2025 Nov 25–28; Vienna, Austria. p. 990–5. doi:10.1109/FLLM67465.2025.11390888.
10. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707.
11. Khattab O, Singhvi A, Maheshwari P, Zhang Z, Santhanam K, Vardhamanan S, et al. DSPy: compiling declarative language model calls into self-improving pipelines. In: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024); 2024 May 7–11; Vienna, Austria.
12. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv Neural Inf Process Syst. 2020;33:9459–74.
13. Tonejc J, Güttes S, Kobekova A, Kaur J. Machine learning methods for anomaly detection in BACnet networks. J Univers Comput Sci. 2016;22(9):1203–24.
14. Wu T, Zhou D, Ou Q, Luo F. Intrusion detection systems in industrial control systems: landscape, challenges and opportunities. Comput Mater Contin. 2026;86(3):4. doi:10.32604/cmc.2025.073482.
15. Lashkari AH, Draper-Gil G, Mamun MSI, Ghorbani AA. Characterization of Tor traffic using time-based features. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy; 2017 Feb 19–21; Porto, Portugal. p. 253–62.
16. Nguyen TD, Marchal S, Miettinen M, Fereidooni H, Asokan N, Sadeghi AR. DIoT: a federated self-learning anomaly detection system for IoT. arXiv:1804.07474. 2018.
17. Runge IM, Akinci B, Bergés M. Challenges in cyber-physical attack detection for building automation systems. In: Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation; 2023 Nov 13–16; Istanbul, Turkey. p. 236–9.
18. Guo H, Yuan S, Wu X. LogBERT: log anomaly detection via BERT. In: Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN); 2021 Jul 18–22; Shenzhen, China. p. 1–8. doi:10.1109/IJCNN52387.2021.9534113.
19. Fan A, Gokkaya B, Harman M, Lyubarskiy M, Sengupta S, Yoo S. Large language models for software engineering: survey and open problems. In: Proceedings of the 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE); 2023 May 14–20; Melbourne, Australia. p. 31–53. doi:10.1109/ICSE-FoSE59343.2023.00008.

20. Nwafor E, Baskota U, Parwez MS, Blackstone J, Olufowobi H. Evaluating large language models for enhanced intrusion detection in IoT networks. In: Proceedings of the GLOBECOM 2024—2024 IEEE Global Communications Conference; 2024 Dec 8–12; Cape Town, South Africa. p. 3358–63. doi:10.1109/GLOBECOM52923.2024.10901300.
21. Lan B, Yu S. Detecting cyber attacks in industrial control systems using duration-aware representation learning. In: Proceedings of the 2024 10th International Conference on Computing and Artificial Intelligence; 2024 Apr 26–29; Bali Island, Indonesia. p. 379–86.
22. Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, de las Casas D, et al. Mistral 7B. arXiv:2310.06825. 2023.
23. Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, et al. Retrieval-augmented generation for large language models: a survey. arXiv:2312.10997. 2023.
24. Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst. 2022;35:24824–37. doi:10.52202/068431-1800.
25. Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y, Radford A, et al. Large language models are zero-shot reasoners. Adv Neural Inf Process Syst. 2022;35:22199–213. doi:10.52202/068431-1613.
26. Homaei M, Tarif M, Rodríguez PG, Caro A, Ávila M. Causal digital twins for cyber-physical security in water systems: a framework for robust anomaly detection. Mach Learn Appl. 2026;23:100824.
27. Petrovic K, Stojanovic B, Saukh O. Physics-augmented autoencoder-based cyber-attack detection for critical water infrastructure. In: Proceedings of the 15th International Conference on the Internet of Things; 2025 Nov 8–11; Vienna, Austria. p. 43–51.
28. Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion. 2020;58:82–115.
29. Ali T, Kostakos P. HuntGPT: integrating machine learning-based anomaly detection and explainable AI with large language models. arXiv:2309.16021. 2023.
30. Sharma R, Okada H, Oba T, Subramanian K, Yanai N, Pranata S. Decoding BACnet packets: a large language model approach for packet interpretation. arXiv:2407.15428. 2024.
31. Moosavi SA, Asgari M, Kamel SR. Developing a comprehensive BACnet attack dataset: a step towards improved cybersecurity in building automation systems. Data Brief. 2024;57(1):111192. doi:10.1016/j.dib.2024.111192.
32. Melhem SB, Golec M, Alwarafy A, Khamayseh YM. LENS: lightweight and explainable LLM-based APT detection at the edge for 6G security. IEEE Access. 2025;13:172402–15. doi:10.1109/ACCESS.2025.3616235.
33. Wang M, Wang H, Cao Z, Qiu Y, Xu C, Ding W. Large language models can be few-shot server anomaly detectors. In: 2025 5th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE); 2025 Jun 13–15; Shenzhen, China. p. 280–4. doi:10.1109/MLISE66443.2025.11100205.
34. Bui MT, Boffa M, Valentim RV, Navarro JM, Chen F, Bao X, et al. A systematic comparison of large language models performance for intrusion detection. Proc ACM Netw. 2024;2(CoNEXT4):1–23. doi:10.1145/3697962.