Article

Asset Discovery in Critical Infrastructures: An LLM-Based Approach

Luigi Coppolino †, Antonio Iannaccone *,†, Roberto Nardone † and Alfredo Petruolo *,†

Engineering Department Centro Direzionale, University of Naples Parthenope, 80143 Naples, Italy;
luigi.coppolino@uniparthenope.it (L.C.); roberto.nardone@uniparthenope.it (R.N.)
* Correspondence: antonio.iannaccone001@studenti.uniparthenope.it (A.I.);
alfredo.petruolo001@studenti.uniparthenope.it (A.P.)
† These authors contributed equally to this work.

Abstract

Asset discovery in critical infrastructures, and in particular within industrial control systems, constitutes a fundamental cybersecurity function. Ensuring accurate and comprehensive asset visibility while maintaining operational continuity represents an ongoing challenge. Existing methodologies rely on deterministic tools that apply fixed fingerprinting strategies and lack the capacity for contextual reasoning. Such approaches often fail to adapt to the heterogeneous architectures and dynamic configurations characteristic of modern critical infrastructures. This work introduces an architecture based on a Mixture of Experts model designed to overcome these limitations. The proposed framework combines multiple specialized modules to perform automated asset discovery, integrating passive and active software probes with physical sensors. This design enables the system to adapt to different operational scenarios and to classify discovered assets according to functional and security-relevant attributes. A proof-of-concept implementation is also presented, along with experimental results that demonstrate the feasibility of the proposed approach. The outcomes indicate that our LLM-based approach can support the development of non-intrusive asset management solutions, strengthening the cybersecurity posture of critical infrastructure systems.

Keywords: industrial control systems; cybersecurity; passive and active probing; mixture of experts; industrial IoT

check for updates

Academic Editor: Subir Halder

Received: 10 July 2025
Revised: 1 August 2025
Accepted: 14 August 2025
Published: 17 August 2025

Citation: Coppolino, L.; Iannaccone, A.; Nardone, R.; Petruolo, A. Asset Discovery in Critical Infrastructures: An LLM-Based Approach. *Electronics* **2025**, *14*, 3267. https://doi.org/10.3390/electronics14163267

1. Rationale and Motivation

The convergence of operational technology (OT) and Information Technology (IT) in critical infrastructures (CIs) has significantly increased the exposure to cyber threats [1,2]. In this context, asset visibility becomes a core prerequisite for identifying vulnerabilities and enforcing effective risk mitigation strategies. With a specific focus on industrial control systems (ICSs), a 2023 CISA report documented over 670 vulnerabilities disclosed in the first half of the year alone, with the manufacturing and energy sectors most heavily impacted [3]. Consequently, robust asset discovery mechanisms are fundamental to the cybersecurity posture of ICS environments. However, conventional asset discovery techniques, originally designed for IT infrastructures, are generally inadequate when applied to ICS environments [4]. These methods fail to accommodate the operational constraints, protocol diversity, and real-time requirements that characterize industrial systems. Moreover, any discovery operation must preserve the integrity of operational continuity, avoiding unintended disruptions to control processes. Within industrial control system environments, asset discovery constitutes not merely a recommended practice but rather a mandatory compliance requirement for utility companies and critical infrastructure organizations, which must adhere to stringent regulatory frameworks and industry standards. The internationally recognized ISO/IEC 27001 [5] standard explicitly delineates asset identification as a fundamental component of comprehensive risk assessment methodologies [6]. Inadequate asset identification and suboptimal asset management practices can precipitate substantial economic vulnerabilities and operational failures, as stated in recent empirical research by Trend Micro [7].

Traditional discovery tools, such as Nmap (https://nmap.org, accessed on 1 June 2025), while effective in standard enterprise networks, introduce considerable risks when used in OT environments. Active scanning in ICS networks can interfere with deterministic control loops or trigger fail-safe mechanisms, potentially causing unplanned downtime or degraded performance [8]. Even OT-oriented commercial platforms like Tenable OT Security (https://it.tenable.com/products/ot-security, accessed on 1 June 2025), Claroty (https://claroty.com, accessed on 1 June 2025), and Armis (https://www.armis.com, accessed on 1 June 2025) face notable challenges, including dependency on modifications to network infrastructure, reduced detection capabilities in segmented or air-gapped environments, and limited coverage due to reliance on passive traffic monitoring, which may overlook inactive or intermittently connected devices. These limitations illustrate the fundamental trade-off between security visibility and operational safety.

To address these challenges, this work proposes a novel architecture for automated asset discovery and identification tailored to ICS environments that advance state of the art through three key technical innovations:

- Semantic Context Awareness through LLM Integration: The framework integrates large language models (LLMs) to introduce semantic context awareness in device identification processes, eliminating the need for a priori data classification or context-specific rule sets that constrain existing solutions. This semantic understanding enables dynamic interpretation of heterogeneous data sources and protocols without predetermined taxonomies.
- Unified Asset Lifecycle Management via Mixture of Experts: The architecture employs a Mixture of Experts (MoE) approach that coordinates specialized, fine-tuned lightweight models for distinct operational phases—asset identification, vulnerability assessment, and network parameter optimization—thereby providing comprehensive asset lifecycle management within a unified framework rather than requiring separate tools for each function.
- Non-Intrusive Multi-Modal Data Fusion: The system implements a non-intrusive data fusion methodology that synthesizes information from passive traffic analysis, protocol-aware active probing, and multi-modal sensor inputs (including acoustic and electromagnetic signatures), fundamentally departing from traditional aggressive scanning approaches that pose operational risks to critical infrastructure.

The remainder of the paper is structured as follows. Section 2 reviews the state of the art in asset discovery, highlighting the limitations of the existing tools and methodologies. Section 3 introduces the necessary background, including the architectural components and operational constraints. Section 4 details the proposed architecture, with a focus on the Mixture of Experts framework and its integration with passive, active, and sensor-based probes. Section 5 describes the implementation of the proof of concept and the design of the industrial-like testbed used for validation and feasibility. Finally, Section 7 concludes the paper and outlines directions for future work.

2. Related Work

Although asset visibility is crucial for securing industrial infrastructure, few existing architectures integrate artificial intelligence, and to our knowledge, none currently use large language models (LLMs) for asset discovery and identification. Moreover, no existing approach combines software-based probes with physical sensing modalities in a coordinated and context-aware manner to support this task within industrial control system (ICS) environments.

Vermeer et al. [9] provide a systematization of knowledge on asset discovery techniques developed primarily for internet-facing IT systems. Their framework classifies methods based on the transformation between asset types (e.g., network identifiers and services) and outlines how measurement techniques can be chained to support tasks, such as external asset inventory and exposure analysis. The systematization is comprehensive but remains focused on external observation, passive DNS, and internet-wide scanning. These techniques do not consider the operational safety constraints or process-level interactions present in ICS environments. Furthermore, the work does not account for the integration of heterogeneous data sources, such as physical-layer sensors. In contrast, our architecture explicitly supports the combination of software probes, physical sensing, and AI models in a modular design that preserves the availability and integrity of ICS processes during asset discovery and identification.

Yang et al. [10] propose a fingerprinting technique based on improved decision trees combined with AdaBoost for identifying industrial devices using semi-active probing. Their approach enhances classification accuracy and noise resistance by extracting features based on signal periodicity and stability. However, it applies a static detection strategy and lacks dynamic adaptation based on system characteristics or safety constraints. It also does not include integration of different sensing layers.

A related study by Park et al. [11] evaluates passive traffic analysis in a smart building environment using Wireshark and port mirroring. While the experiment reveals limitations of conventional IT tools in automation networks, it remains focused on basic network-level analysis and does not provide an extensible or modular architectural approach.

In summary, the existing works either target traditional IT systems or offer partial solutions tailored to specific environments without a unified framework for adaptive, context-aware, and safe asset discovery in ICS networks.

On the other hand, an exemplary work that aligns with our vision of implementing an AI-based architecture with a unified framework for asset identification is presented by Wang et al. [12]. Their FL4IoT framework demonstrates significant methodological advances in device fingerprinting through federated learning, achieving 99% identification accuracy while maintaining lightweight computational requirements specifically tailored for IoT environments. However, while their approach leverages network traffic behavioral analysis, we believe that large language model-based analysis within ICS environments presents an alternative practical methodology for asset identification that can achieve comprehensive system understanding with minimal disruption to operational processes, thereby addressing the unique requirements and constraints inherent to industrial control systems. Our contribution lies in the design of a flexible architecture that integrates AI-driven analysis with the coordinated use of physical-layer sensors and software probes, explicitly addressing the operational constraints of industrial environments.

3. Background

3.1. Asset Discovery in ICS

Asset discovery in industrial control systems (ICSs) involves the systematic identification, classification, and inventorying of devices, systems, and communication channels operating within OT networks. Unlike conventional IT environments, ICS asset discovery must address specific constraints, such as real-time operation, safety-critical processes, and legacy systems often lacking modern network security features. The convergence of historically isolated OT infrastructures with enterprise IT networks has created hybrid environments that present new challenges. ICS environments often include legacy devices operating for 20 to 30 years alongside modern networked systems, resulting in diverse communication patterns and protocols such as Modbus, DNP3, BACnet, EtherNet/IP, and HART.

System availability is a critical requirement in ICS, limiting the adoption of intrusive techniques. Many industrial systems operate continuously, disallowing maintenance windows. Active discovery methods may trigger failures in legacy equipment, interfere with safety systems, or disrupt real-time operations, highlighting the need for non-disruptive approaches. The diversity of communication protocols adds further complexity. Many industrial devices use proprietary protocols tailored for specific applications, which traditional tools are unable to interpret or classify effectively. Moreover, asset discovery in ICSs must include firmware and software versioning, operational roles, and interdependencies, which go beyond the scope of IT-centric scanning solutions.

3.2. Available Tools Review

Current tools for ICS asset discovery include adaptations of IT security platforms and dedicated OT solutions, each with distinct strengths and weaknesses. Table 1 summarizes key characteristics of the representative tools. IT tools like Nmap provide basic capabilities such as port and service detection but lack support for industrial protocols and may pose operational risks when used in OT environments. Shodan offers global visibility into internet-exposed devices but cannot reach air-gapped or segmented networks and lacks detailed asset context.

Table 1. Comparative analysis of industrial control system asset discovery tools and platforms.

Tool/Platform	Discovery Method	Key Capabilities	Primary Limitations
Traditional IT Security Solutions			
Nmap	Active scanning	Port scanning, service detection, network enumeration	No industrial protocol support, operational risks to OT devices, inappropriate for ICSs
Shodan	Passive internet scanning	Internet-exposed device identification, global visibility	Air-gapped network inaccessibility, limited asset context
Commercial ICS Security Platforms			
Tenable OT Security	Passive monitoring	Non-intrusive scanning, vulnerability detection, Nessus-based technology	Infrastructure modifications required, passive monitoring constraints
Claroty	Multi-method	Passive monitoring, active queries, database parsing, data fusion	Complex integration, coordination challenges
Armis	Agentless ML	ML-based analysis, behavioral profiling, device fingerprinting	Limited deep inspection, traffic pattern dependency
Nozomi Guardian	AI multi-protocol	AI device profiling, wireless discovery (Wi-Fi, Bluetooth, Zigbee, LoRaWAN), smart polling	Scalability constraints, strategic sensor placement requirements
Universal Industry Limitations			
Incomplete proprietary protocol coverage. Difficulty distinguishing PLC/RTU/HMI types. Limited firmware detection virtual system identification challenges. Unresolved discovery–safety tension.			

OT-focused commercial platforms have improved ICS visibility. Tenable OT Security leverages passive traffic analysis and Nessus-based vulnerability detection but requires infrastructure changes and is constrained by passive-only operation. Claroty combines passive monitoring, active querying, and database parsing, but the integration process can be complex. Armis uses agentless ML-based fingerprinting and behavioral profiling, though its passive dependency limits deep inspection and full coverage. Nozomi Guardian supports wireless protocols and smart polling but requires sensor deployment planning and suffers from scalability issues.

Despite these advances, universal challenges remain: incomplete protocol support, inability to distinguish similar device types (e.g., PLCs vs. RTUs), insufficient firmware detail, and poor handling of virtualized assets. The central issue is the trade-off between discovery completeness and operational safety, which the existing tools do not fully resolve.

3.3. Why LLMs Are Useful in ICSs

Large language models (LLMs) offer several advantages for ICSs due to their ability to process complex, heterogeneous data and extract contextual meaning from unstructured sources. LLMs support protocol understanding, anomaly identification, and device classification across multiple industrial standards. LLMs can interpret technical manuals, network documentation, and configuration files, enabling enriched context for asset identification.

The LLM4PLC framework exemplifies this potential by generating verifiable code for PLCs [13]. The LLMPot project demonstrates automatic honeypot generation for industrial scenarios, showcasing LLM capabilities in replicating protocol behavior and simulating device functions across use cases such as smart grids and aviation systems [14].

Importantly, LLMs can fuse insights from network traffic, system logs, and operational documents, providing a multi-modal understanding. They also extract patterns from historical incident reports and regulatory documentation, converting expert knowledge into actionable discovery logic.

Emerging specialized models such as IoT-LM indicate a growing interest in applying LLMs to industrial contexts, offering improved performance in asset classification by understanding environmental context and device-specific characteristics [15].

In the context of asset discovery, LLMs contribute to understanding and decoding multi-protocol communications, correlating data from diverse and heterogeneous sources, generating explanations for discovery outcomes, and producing adaptive discovery queries for novel devices. LLMs, when integrated with architectures such as the Mixture of Experts (MoE), can assign specialized modules to distinct discovery tasks, improving adaptability and precision in industrial settings.

4. Architecture Overview

The proposed architecture introduces a modular, multi-layered framework designed to support automated, intelligent asset discovery in industrial environments. It integrates heterogeneous sensing, structured data aggregation, and a context-aware analytical layer powered by a Mixture of Experts (MoE) design. Figure 1 illustrates the system architecture and its layered composition.

At the lowest level, the *sensing layer* is connected to both the IT and OT domains of the industrial environment. It employs a combination of *passive* and *active probes*. The former collects traffic data without introducing network load; these include packet sniffers and flow monitors for identifying communication patterns and protocol usage. The latter performs controlled scans—such as port and service discovery—under operational constraints to uncover non-communicating or undocumented devices. These probes are deployed across network segments, with strict segmentation maintained by a typical *Network Segmentation*

Firewall to ensure the isolation of critical OT components while permitting necessary observability.

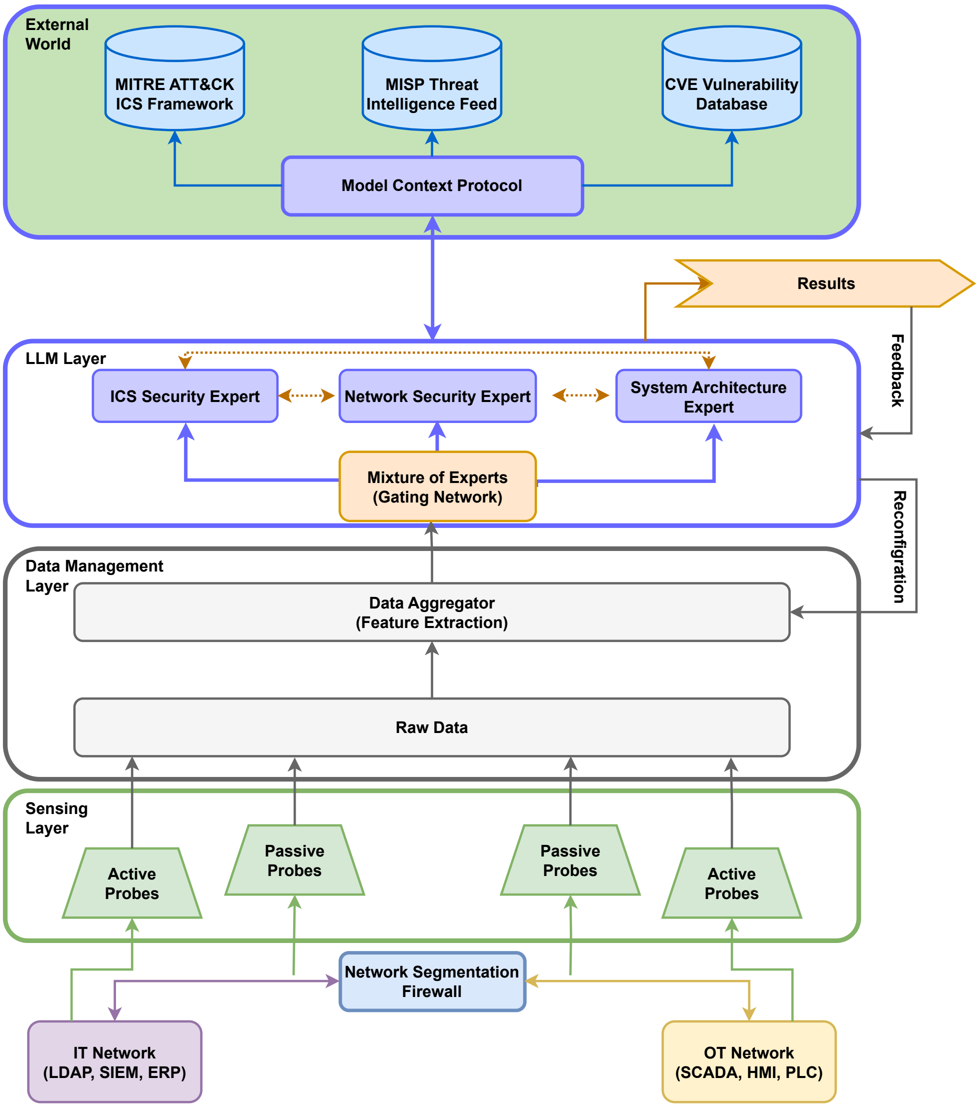

External World
MITRE ATT&CK ICS Framework
MISP Threat Intelligence Feed
CVE Vulnerability Database
Model Context Protocol
Results
Feedback
LLM Layer
ICS Security Expert
Network Security Expert
System Architecture Expert
Mixture of Experts (Gating Network)
Reconfiguration
Data Management Layer
Data Aggregator (Feature Extraction)
Raw Data
Sensing Layer
Active Probes
Passive Probes
Passive Probes
Active Probes
Network Segmentation Firewall
IT Network (LDAP, SIEM, ERP)
OT Network (SCADA, HMI, PLC)

Figure 1. Architecture overview: enabling MoE analysis.

The collected data is then processed in the *data management layer*. *Raw data* from all probes is channeled into a *data aggregator* responsible for feature extraction and preprocessing. This step translates low-level communication traces into structured representations suitable for higher-level analysis. The features include protocol signatures; timing characteristics; device behavior profiles; and metadata, such as firmware versions or known identifiers.

At the core of the system, the *LLM layer* includes a *Mixture of Experts (MoE)* model, which distributes analytical tasks across multiple specialized LLM experts. These experts are routed dynamically via a gating network based on the context of the input data. The *ICS Security Expert* specializes in identifying OT-specific risks and interpreting industrial protocols. The *network security expert* analyzes threats and anomalies within IT and IP-based communication layers. The *System Architecture Expert* reconstructs logical and physical topologies and detects configuration inconsistencies or policy violations. The gating network leverages extracted features and external context to assign the appropriate expert(s) to each discovery task, promoting precision and efficiency.

The architecture integrates threat intelligence from the *external world* through a *model context protocol*, which connects to *MITRE ATT&CK*, *MISP*, and *CVE*. The MITRE ATT&CK supports an ICS framework for adversarial behavior mapping. The MISP threat intelligence feed is for real-time indicator ingestion. The CVE vulnerability database is needed for risk evaluation based on detected asset profiles. These sources inform the MoE layer by refining expert routing decisions and enriching asset risk assessments. The system also supports a feedback and reconfiguration loop, where results generated by the LLM layer are used to provide feedback to the sensing and aggregation layers for improved data collection and to update the gating logic within the MoE network to incorporate recent discovery outcomes.

The architecture's modularity supports deployment in distributed or centralized settings, and its layer separation facilitates updates and extensions. The use of segmented data paths, protocol-aware probes, and intelligent routing ensures compliance with safety and operational constraints found in critical infrastructures.

4.1. MoE Model Description: Activate Only What You Need

The system's core functionalities are supported by the MoE architecture's gating mechanism. The gating network constitutes the fundamental building block responsible for expert selection and activation. The gating network can be formally defined as

$$G(x) = \mathrm{softmax}(W_g \cdot x + b_g)$$

where $x \in \mathbb{R}^d$ represents the input feature vector, $W_g \in \mathbb{R}^{n \times d}$ is the learned gating weight matrix, $b_g \in \mathbb{R}^n$ is the bias vector, and n denotes the number of available experts. The output $G(x) \in \mathbb{R}^n$ represents the activation probability distribution over the expert ensemble.

The fundamental aspect is that the gating network output $G(x) = [g_1, g_2, \ldots, g_n]$ represents the activation scores for the n experts comprising the neural architecture. In our implementation, when input data x arrives from the data aggregator block, it undergoes tokenization and is processed by the gating network to compute affinity scores with three specialized experts: the ICS Security Expert, the network security expert, and the System Architecture Expert.

The expert activation mechanism employs a threshold-based selection strategy where expert i is activated if $g_i > \tau$, where τ is an empirically determined threshold. For multi-expert activation scenarios where $|\{i : g_i > \tau\}| \geq 2$, the system implements a sequential expert consultation protocol. The primary expert (highest activation score) processes the input first, generating output O_1. Subsequently, the secondary expert receives an augmented input comprising the original data x, the primary expert's output O_1, and a contextual system prompt indicating prior expert analysis. This sequential processing enables knowledge distillation and output refinement, where the final response O_{final} represents the enriched synthesis of multiple expert perspectives.

The MoE framework achieves superior computational efficiency by maintaining constant inference cost $\mathcal{O}(k)$ where $k \ll n$ represents the number of activated experts, while equivalent monolithic models require $\mathcal{O}(N)$ operations across all N parameters regardless of input-domain specificity. Expert specialization enables concentrated parameter utilization within focused knowledge domains, circumventing the capacity dilution phenomenon observed in large unified models where parameters must simultaneously encode disparate functional competencies [16]. The gating mechanism facilitates adaptive model complexity, dynamically scaling computational resources proportional to task requirements rather than deploying fixed architectural capacity for all inference operations.

4.2. The Feedback Loop: Analyze–Enrich–Set

Following a CI/CD paradigm, the architecture employs a continuous feedback loop mechanism for iterative model refinement and expert capability enhancement. Each expert maintains bidirectional connectivity to external threat intelligence repositories, enabling real-time access to authoritative cybersecurity data sources, including MITRE ATT&CK framework, MISP (Malware Information Sharing Platform), CVE (Common Vulnerabilities and Exposures) databases, and additional third-party intelligence feeds.

The external information retrieval process operates through a structured API gateway $\mathcal{A} : \mathcal{Q} \to \mathcal{R}$, where $\mathcal{Q}$ represents the expert query space and $\mathcal{R}$ denotes the retrieved information domain. This external data serves two primary architectural functions: lightweight model adaptation through incremental learning protocols and contextual data enrichment for persistent storage within the data management layer.

The lightweight retraining mechanism implements parameter-efficient fine-tuning strategies, specifically Low-Rank Adaptation (LoRA) techniques, enabling expert models to incorporate emerging threat patterns without full model retraining. The enrichment pipeline processes retrieved external intelligence I_{ext} through a semantic fusion operator $\Phi : (I_{local}, I_{ext}) \to I_{enriched}$, where I_{local} represents internally processed data and $I_{enriched}$ constitutes the augmented dataset subsequently stored in the data management infrastructure. This continuous feedback mechanism ensures expert knowledge bases remain synchronized with evolving cybersecurity landscapes while maintaining computational efficiency through selective parameter updates.

This framework operationalizes the Analyze–Enrich–Set paradigm through three distinct computational phases: the analysis phase leverages local inference capabilities of activated experts to process incoming security data, the enrichment phase integrates contextual information through model context protocols and external API data retrieval mechanisms, and the set phase generates actionable output responses that may trigger system configuration modifications or prompt human operator interventions for parameter adjustment. Adherence to the Human-in-the-Loop (HIL) paradigm ensures that all AI-driven recommendations within the CI/CD pipeline undergo mandatory human validation and approval before deployment, maintaining critical oversight over autonomous security decisions.

4.3. Requirements Elicitation and Threat Model

As established in Messe et al., the threat modeling process encompasses four critical activities: Asset Identification, Threat Enumeration, Threat Prioritization, and Mitigation Strategy Development [17]. Enabling automated and enriched asset discovery fundamentally enhances the threat modeling pipeline by automating the asset identification phase, thereby allowing for human expertise to focus on threat analysis and mitigation planning rather than manual asset enumeration.

Inadequately managed asset identification introduces significant security vulnerabilities into industrial control systems. Our architecture addresses the following critical threat vectors:

- **Attack Surface Obfuscation**: Incomplete asset discovery produces fragmented attack surface mappings, creating security blind spots where threat assessment and defensive measures are not implemented. These unmapped segments become high-risk attack vectors.
- **Shadow Device Proliferation**: Unmanaged devices within the operational technology perimeter represent critical security vulnerabilities. These assets, lacking proper patch management, security monitoring, and configuration control, constitute weak nodes susceptible to lateral movement attacks and persistent threats.

- **Protocol Misclassification**: Legacy and proprietary industrial protocols often exhibit non-standard behaviors that confound traditional discovery mechanisms, leading to asset misidentification and inappropriate security policy application.
- **Temporal Asset-State Drift**: Dynamic network topologies and device state changes in ICS environments create temporal inconsistencies in asset inventories, undermining continuous security monitoring and incident response capabilities.

Our architecture addresses these threats through covering six core security requirements: (1) Real-time Asset Visibility, (2) Context-Aware Classification, (3) Non-Intrusive Discovery, (4) Adaptive Threat Intelligence, (5) Multi-Modal Data Fusion Support, and (6) Continuous Asset Lifecycle Management.

The framework establishes Real-time Asset Visibility (1) through continuous passive monitoring and intelligent protocol analysis capabilities that maintain persistent awareness of network-connected devices. Context-Aware Classification (2) is achieved through LLM-based semantic interpretation of device behaviors and network communications, enabling dynamic understanding of asset functionality and risk profiles. The system ensures Non-Intrusive Discovery (3) by implementing operational safety protocols through passive data collection methodologies that avoid disrupting critical control processes. Adaptive Threat Intelligence (4) Integration provides dynamic threat landscape awareness and proactive vulnerability assessment capabilities, enabling the system to evolve with emerging security challenges. Scalable Multi-Modal Data Fusion (5) supports heterogeneous industrial environments through unified processing architectures that synthesize network traffic, protocol data, and physical sensor inputs across diverse operational technology infrastructures. Finally, Continuous Asset Lifecycle Management (6) delivers persistent monitoring capabilities, automated vulnerability correlation, and integrated security posture assessment throughout complete device operational lifecycles.

5. Proof of Concept: AI-Based ICS Asset Discovery

This section presents a proof-of-concept implementation of the proposed architecture to evaluate its feasibility in a realistic industrial setting. The framework was tested in a controlled testbed environment consisting of heterogeneous industrial devices connected in an isolated network. The aim is to replicate conditions similar to those encountered in actual ICS, enabling a practical assessment of the framework's capabilities in asset discovery and classification tasks.

5.1. Testbed Description

The testbed replicates a simplified ICS network composed of representative devices commonly found in modern industrial environments. The network topology is depicted in Figure 2. All devices communicate via a central router and are connected using a combination of Wi-Fi and Ethernet connections. The network is fully isolated from the internet to allow for reproducible experiments without external interference or risk of information leakage.

The selection of devices was based on their relevance in operational settings and their support for remote management interfaces, which are typical of assets targeted in reconnaissance activities.

The testbed includes the following components:

- **The TP-Link Router TL-WR940N:** Serves as the central hub managing internal traffic and segmenting the network from external access. It ensures consistent IP addressing and routing [18].

- **The Robotic Arms (Niryo NED2):** Two robotic manipulators, each based on Raspberry Pi hardware, with 6 degrees of freedom and programmable via API. These are Wi-Fi-connected and reflect devices used in both educational and prototyping scenarios [19].
- **The 3D Printer UltiMaker S7 Pro Bundle:** A Wi-Fi-connected printer representative of auxiliary manufacturing assets commonly integrated into ICS networks [20].
- **The 3D Printer Bambu Lab X1E 3D Printer:** Another Wi-Fi-enabled 3D printer, included to test recognition of similar devices across different manufacturers [21].
- **The Programmable Logic Controller (PLC):** Omron NX1P2, a core component in ICSs responsible for process control. This unit is connected via Ethernet and supports common industrial protocols. Its presence allows for evaluating asset detection in more sensitive segments of the network [22].
- **The Workstation:** A Windows 11 Pro workstation connected via Ethernet. It hosts the asset discovery framework and the local LLM execution engine. This machine is equipped with an Intel(R) Core(TM)i9-14900KF processor and 64 GB of RAM.

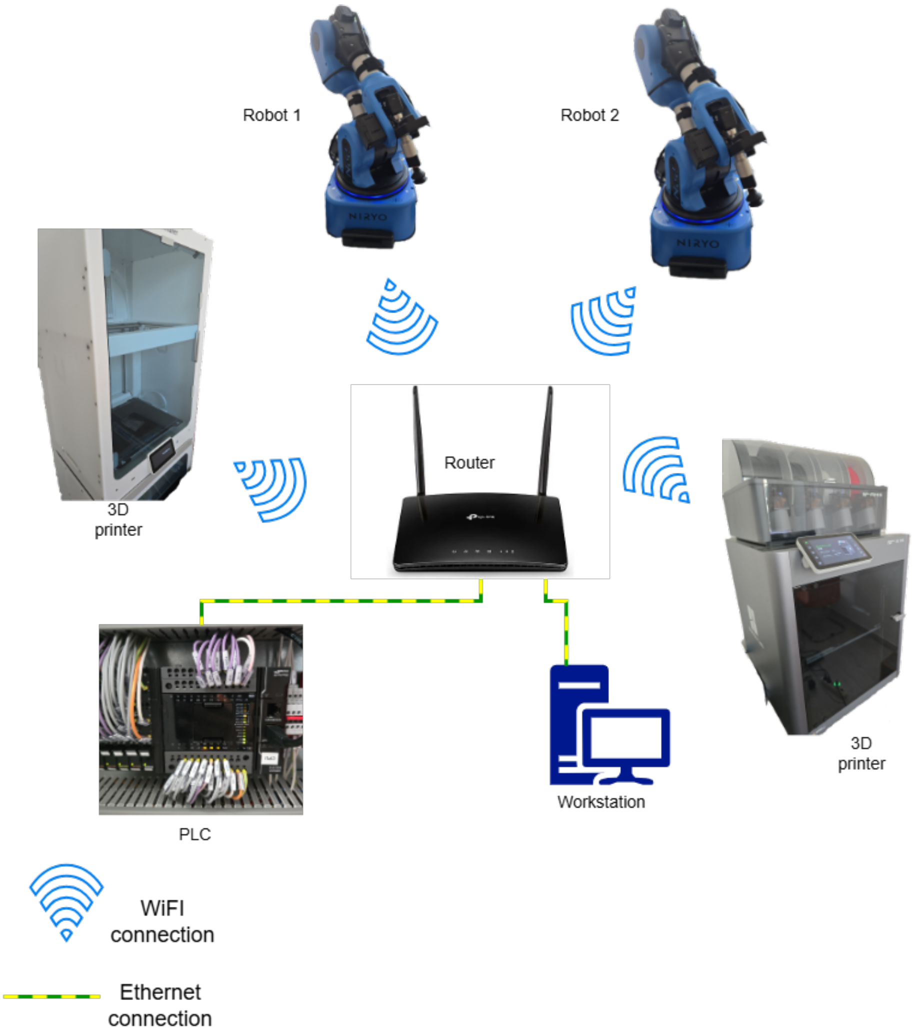


Figure 2. Testbed network topology.

5.2. Tool Selection

All the tools used in this proof of concept are open-source, ensuring full reproducibility. The framework is implemented in Python 3.13 due to its flexibility and wide support for third-party libraries and networking modules [23].

Large language models (LLMs) are central to the framework. We use `Ollama` [24] to deploy and manage local instances of LLMs. The selected model, `gemma3:27b` [25], is executed locally using an NVIDIA GeForce RTX 4090 GPU with 32 GB of VRAM [26], integrated into the workstation described earlier. This setup avoids external API calls and ensures low-latency interaction through Ollama's API interface.

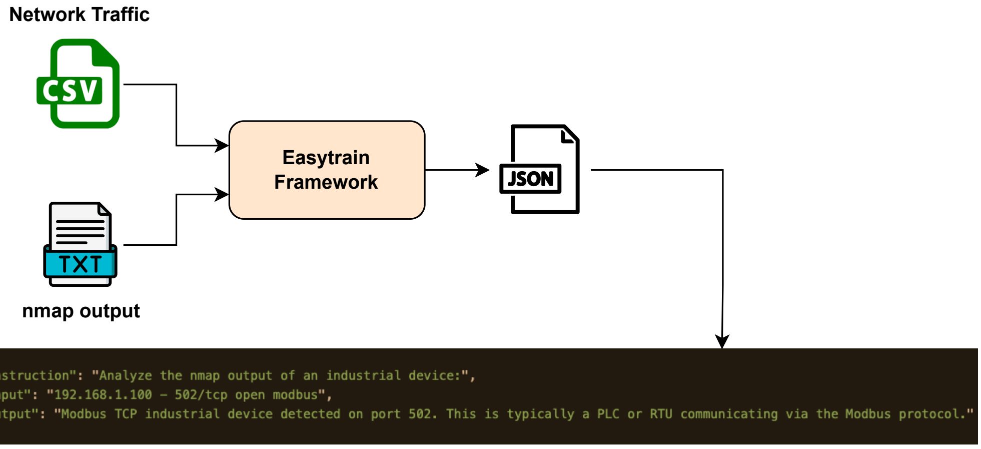
Network scans are performed using `Nmap` [27], which supports various scan types, including SYN scans, OS detection, service version probing, and UDP probing. In addition to textual output, Nmap's .txt output is parsed by the framework and interpreted by the LLM to infer device characteristics, potential roles, and criticality.

To enrich this analysis, any ARP [28] information is captured using the arp command. The tool allows for real-time and offline packet inspection. Captured data is cross-referenced with Nmap results to identify communication patterns, device fingerprints, and protocol-level metadata.

5.3. Fine-Tuning LLM

The fine-tuning process described herein focuses specifically on the development of the **network security expert model**, one of the specialized agents within the multi-agent framework. The same methodological approach is employed for all expert models, with domain-specific knowledge bases substituted according to each agent's specialization requirements.

The fine-tuning process employed Google's Gemma3 model as the foundation architecture, leveraging parameter-efficient fine-tuning (PEFT) through Low-Rank Adaptation (LoRA) to specialize the model for industrial asset discovery and cybersecurity analysis. The training dataset was systematically constructed using a custom-developed framework (EasyTrain framework has been made available by the authors at https://github.com/bobbinetor/easytrain, accessed on 1 June 2025), which processed network traffic captures and Nmap reconnaissance outputs collected from the testbed environment described in the previous section. The general workflow is shown in Figure 3.



```
{
  "instruction": "Analyze the nmap output of an industrial device:",
  "input": "192.168.1.100 - 502/tcp open modbus",
  "output": "Modbus TCP industrial device detected on port 502. This is typically a PLC or RTU communicating via the Modbus protocol."
}
```

Figure 3. Easytrain converts various files into LLM fine-tuning datasets, structured as (instruction, input, and output) tuples.

Dataset preparation involved the extraction of structured knowledge from captured network reconnaissance data, including the port scan results, service fingerprints, and device identification patterns observed across the heterogeneous ICS testbed. The EasyTrain framework employed intelligent chunking algorithms with context-aware segmentation (chunk size: 2000 characters; overlap: 200 characters) to preserve semantic relationships within the network reconnaissance outputs. Each data sample was formatted as instruction–input–output triplets, where instructions prompted asset identification tasks, inputs contained raw Nmap outputs or network signatures, and outputs provided expert-level analysis including device classification, vulnerability assessment, and security recommendations tailored to industrial environments.

An exemplary training sample for the network security expert, with parameters reported in Table 2 demonstrates the structured approach to knowledge extraction:

```
{
  "instruction": "Analyse this Nmap scan output and identify the
                  industrial device type, potential vulnerabilities,
                  and security recommendations",
  "input": "Nmap scan report for 192.168.1.15\nHost is up
           (0.00034s latency).\nPORT    STATE SERVICE VERSION\n
           502/tcp open  modbus  Modbus/TCP\n
           9600/tcp open http    lighttpd 1.4.35",
  "output": "Device Classification: Programmable Logic Controller (PLC)\n
            Protocol Analysis: Modbus/TCP on port 502 indicates industrial
            control system functionality. HTTP service on port 9600 suggests
            web-based management interface.\nVulnerability Assessment:
            Unencrypted Modbus protocol presents data interception risks.
            Older lighttpd version may contain known vulnerabilities.\n
            Security Recommendations: Implement network segmentation,
            deploy protocol-aware firewalls, update web server software,
            and establish secure remote access procedures."
}
```

The LoRA configuration targeted specific attention and feed-forward network components within the Gemma-3 architecture, applying rank decomposition with $r = 8$, $\alpha = 16$, and dropout = 0.1 to the query projection (q_proj), value projection (v_proj), key projection (k_proj), output projection (o_proj), gate projection (gate_proj), up projection (up_proj), and down projection (down_proj) modules. This selective parameter adaptation approach reduced trainable parameters to approximately 0.5% of the total model parameters whilst maintaining the pre-trained knowledge base.

The training was conducted on NVIDIA RTX 4090 hardware with CUDA optimization. The training configuration utilized a batch size of 4 with gradient accumulation steps of 8, yielding an effective batch size of 32. The learning rate was set to 5×10^{-5} with cosine annealing scheduling, and training was limited to 500 steps across 2 epochs to prevent overfitting on the domain-specific dataset. Gradient checkpointing was disabled to maintain compatibility with LoRA parameter updates, whilst the AdamW optimizer with weight decay (0.01) provided stable convergence.

The tokenization process employed Gemma's native tokenizer with sequence truncation at 1024 tokens and custom data collation optimized for variable-length input sequences. During preprocessing, network reconnaissance data was formatted using a structured template that preserved the hierarchical relationship between scan parameters, target specifications, and discovered services.

Post-training model deployment involved merging the LoRA adapters with the base model weights and exporting to a format compatible with the Ollama inference engine. The final model was configured with temperature = 0.7, top-p = 0.9, and top-k = 40 sampling parameters to balance response creativity with technical accuracy.

The complete fine-tuning workflow followed a systematic pipeline:

(1) Network reconnaissance data collection from the testbed environment using Nmap and traffic capture tools;

(2) Automated dataset generation through the EasyTrain framework, converting raw network outputs into structured instruction–input–output triplets;

(3) Data preprocessing and tokenization using Gemma's native tokenizer with custom collation functions;
(4) LoRA-based fine-tuning on CUDA-optimized hardware with mixed-precision training;
(5) Adapter merging and model export for Ollama deployment;
(6) Inference engine configuration with domain-specific system prompts and sampling parameters optimized for technical accuracy in industrial cybersecurity contexts.

It is important to note that whilst this methodology is demonstrated through the network security expert specialization, the identical fine-tuning pipeline is applied to develop all expert models within the multi-agent framework. Each expert model utilizes domain-specific datasets corresponding to their respective areas of expertise (e.g., industrial protocols, device specifications, and compliance frameworks) whilst maintaining consistent architectural configurations and training parameters to ensure interoperability within the collaborative agent ecosystem.

Table 2. Training parameters for the *network security expert* model.

Parameter	Description	Impact
Base model	Gemma-3	Foundation model
Fine-tuning method	LoRA (PEFT)	Reduces training cost while retaining pre-trained knowledge
Adapted modules	q_proj, v_proj, k_proj, o_proj, gate_proj, up_proj, down_proj	Limits fine-tuning to selected model components
LoRA settings	$r = 8$, $\alpha = 16$, dropout = 0.1	Control adaptation range and regularization
Token limit	1024 tokens	Defines max input length per example
Batch size	4 (effective 32)	Number of samples per training step (after gradient accumulation)
Learning rate	5×10^{-5}	Controls weight update size during training
Scheduler	Cosine annealing	Adjusts learning rate dynamically
Epochs	2	Number of full passes through the dataset
Training steps	500	Total training updates applied
Optimizer	AdamW (weight decay 0.01)	Improves convergence while mitigating overfitting
Gradient checkpointing	Disabled	Ensures LoRA compatibility
Tokenizer	Native Gemma tokenizer	Processes input sequences for the model
Chunk size/overlap	2000/200 characters	Maintains context across chunked input
Data format	Instruction, input, output tuples	Structures tasks for supervised fine-tuning
Sampling config	temperature = 0.7, top-p = 0.9, top-k = 40	Balances determinism and response diversity during inference
Hardware	NVIDIA RTX 4090	GPU used for training

5.4. Output Report

The framework, upon completion of the entire analysis based on two inputs (the Nmap output and the `arp -a` output), generates a report in the JSON format (reported in Listing 1) containing enriched information for each host address, including device type identification. The analysis is conducted in three steps. Initially, an analysis is performed on the Nmap output using a large language model (LLM), which produces a preliminary report in JSON format structured as follows:

Listing 1. Formatted JSON output report generated by the framework.

```
{
  "scan_info": {
    "scan_type": "Nmap",
    "timestamp": "unknown"
  },
  "hosts": [
    {
      "ip": "x.x.x.x",
      "hostname": "hostname_if_available",
      "status": "up/down",
      "os_info": "os_info_if_available",
      "open_ports": [
        {
          "port": 80,
          "protocol": "tcp",
          "service": "http",
          "version": "version_if_available"
        }
      ],
      "device_type_hints": [
        "router",
        "server",
        "workstation",
        "iot_device",
        "printer",
        "unknown"
      ]
    }
  ]
}
```

The second step involves the analysis of the `arp -a` output to retrieve manufacturer information based on MAC addresses. As a result, the JSON report is augmented with the manufacturer details. Finally, the last report is generated following an interaction with the LLM, which correlates all available information for each asset and attempts to infer the device type. The result is a complete JSON report provided to the user.

6. Results and Discussion

In this section a preliminary evaluation of the proof of concept is presented, along with the scalability evaluation and current limitations.

6.1. Preliminary Framework Evaluation

This section presents a comprehensive performance analysis of the proposed framework, with particular emphasis on asset classification accuracy. The evaluation employs the same LLMs utilized in the previous throughput analysis to ensure consistency across experimental conditions.

As previously demonstrated, increasing the number of devices in input files correlates with reduced throughput for certain models. To investigate the relationship between prompt length and classification accuracy, we conducted a systematic analysis using identical input files while varying token limits across different LLMs. Accuracy metrics were computed for each experimental run to quantify performance degradation patterns.

Figure 4 illustrates the relationship between token count and classification accuracy across the evaluated LLMs. The results reveal that models generally achieve high accuracy levels at lower token counts. However, as token count increases, accuracy deteriorates across all architectures, albeit with varying degrees of degradation. For this analysis we considered the average accuracy calculated on three runs.

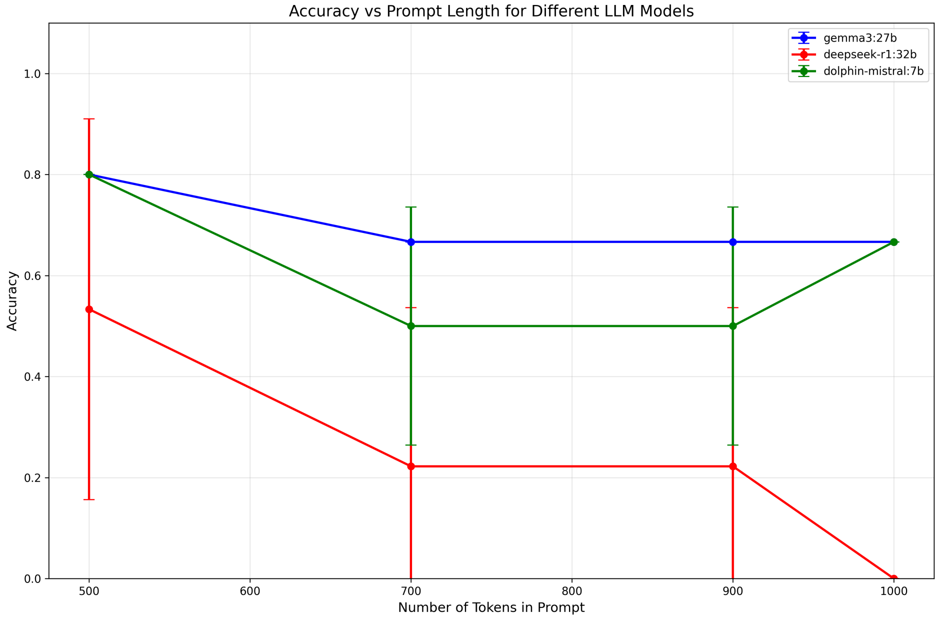


Figure 4. Classification accuracy as a function of token count across different LLM architectures.

Notably, Gemma3:27b demonstrates exceptional stability, maintaining approximately 67% accuracy even as the token count increases substantially. This consistent performance validates Gemma3:27b as a robust choice for the proposed framework, providing reliable accuracy levels across diverse input conditions. In contrast, DeepSeek:32b and Dolphin-Mistral:7b exhibit significant accuracy degradation with increased token counts.

An important observation is that both Gemma3:27b and Dolphin-Mistral:7b show improved accuracy when the token count is reduced through prompt optimization and input file dimension reduction. This finding suggests that performance optimization could be achieved through concise, well-structured prompts and iterative processing approaches, based on batch computation for large networks. However, such optimization techniques fall beyond the scope of the current investigation.

To establish baseline performance metrics for the proposed framework, we evaluated accuracy across different LLMs using unoptimized prompts with standard token allocations.

Table 3 demonstrates that Gemma3:27b achieves superior accuracy performance even without token optimization, reinforcing its suitability for the proposed framework. The experimental results indicate potential for achieving accuracy levels approaching 80% through systematic token optimization and prompt engineering techniques. However, such enhancements represent future research directions beyond the current study's objectives.

Table 3. Comparative accuracy analysis across LLM architectures.

Model	Average Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Gemma3:27b	66.67	46.00	66.00	54.00
DeepSeek-R1:32b	58.30	25.00	50.00	33.33
Dolphin-Mistral:7b	33.00	29.00	33.00	27.00

6.2. Scalability Evaluation

To evaluate the framework’s performance when analyzing a larger number of devices, a scalability assessment based on inference time was conducted. The first analysis was carried out using real data acquired from the testbed, which included six active hosts. The tool was then executed multiple times with the same inputs (Nmap output and `arp -a` output from the testbed) to estimate the mean total inference time. The total inference time is defined as the duration from the initiation of the request to the completion of the final analysis. To estimate this, 50 runs were performed. Additionally, different LLMs were employed to compare the performance of several open-source, fine-tuned models (the experts).

The first analysis was conducted using Gemma3, which has 27 billion parameters (results reported in Figure 5); the second using DeepSeek R1, with 32 billion parameters (results reported in Figure 6); and the final analysis was conducted using Dolphin Mistral, with 7 billion parameters (results reported in Figure 7). Each model was tested using the information from the testbed, specifically focusing on six assets.

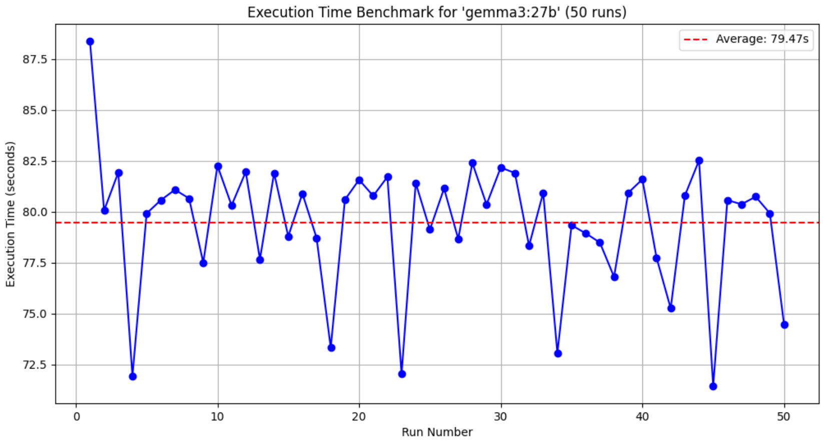


Figure 5. Total inference time for 50 runs using Gemma3:27b.

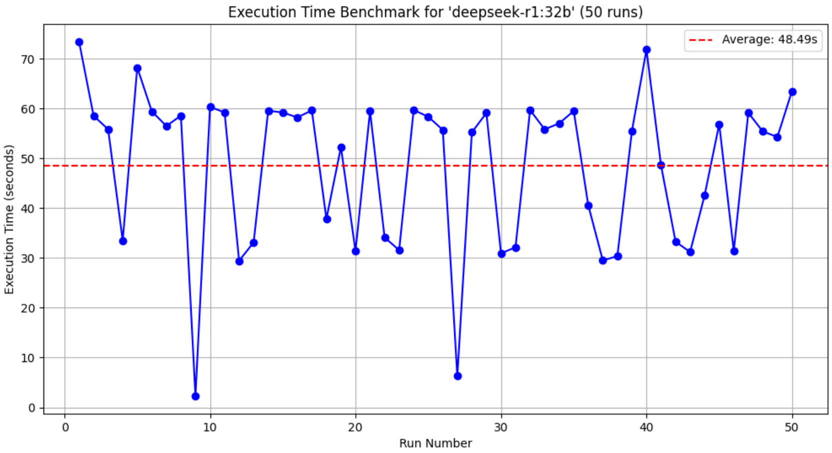


Figure 6. Total inference time for 50 runs using DeepSeek-R1:32b.

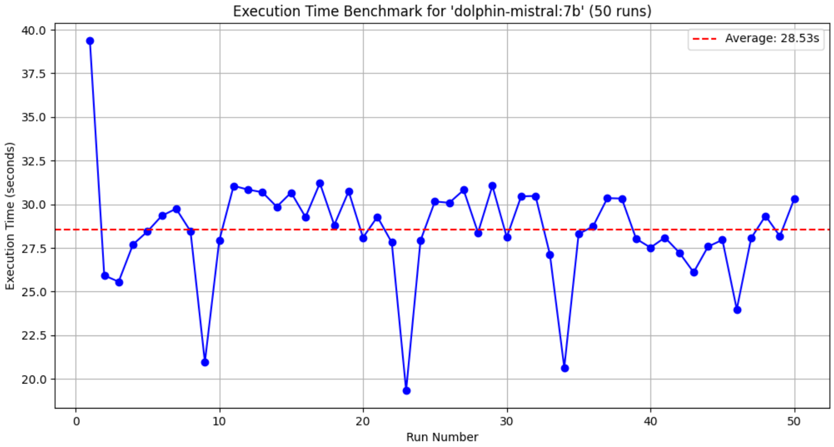


Figure 7. Total inference time for 50 runs using Dolphin-Mistral:7b.

We conducted an empirical analysis to examine how varying the number of network devices affects model performance characteristics. The throughput measurements presented in Figure 8 illustrate the computational behavior of each model when processing different volumes of input data. Our experimental results demonstrate three distinct performance patterns. Dolphin-Mistral 7B achieved the highest processing rates, with throughput values oscillating between 110 and 137 tokens per second. This model's efficiency makes it computationally attractive, yet two limitations emerged: first, the reduced parameter count (7B) potentially constrains classification precision relative to larger architectures; second, substantial performance fluctuations were observed, particularly a pronounced dip at 70 devices before recovering at 100 devices. In contrast, Gemma3:27b exhibited remarkable consistency, delivering steady throughput ranging from 38 to 41 tokens per second regardless of input scale. This uniform behavior pattern suggests robust internal resource management, making it well-suited for environments requiring predictable performance guarantees. The trade-off involves accepting lower absolute throughput in exchange for operational reliability. DeepSeek-R1 32B displayed a more complex performance profile, maintaining stable operation up to 50 devices, experiencing severe degradation at 70 devices (dropping to 1 token/second), and then partially recovering at 100 devices. This non-linear behavior pattern strongly indicates memory management challenges, likely stemming from context buffer limitations when processing intermediate-sized inputs. A significant constraint was discovered during large-scale testing: when presented with 100 devices, both DeepSeek-R1 and Gemma3:27b exhibited incomplete output generation, processing only 15 and 10 devices, respectively. This limitation stems from input token capacity restrictions, where the concatenated Nmap and ARP scan results exceed the models' maximum sequence length. The truncation occurs silently, creating a scalability bottleneck that necessitates input segmentation strategies for enterprise-scale network analysis. These experimental findings reveal that model selection for asset identification requires balancing computational efficiency, output consistency, and scalability constraints based on specific deployment requirements.

As shown in Table 4, the shortest average inference time was obtained using Dolphin-Mistral:7b. However, this model, with only 7 billion parameters, sacrifices accuracy due to its limited parameter size. Notably, a comparison between Gemma3:27b and DeepSeek-R1:32b reveals that, despite DeepSeek-R1 having more parameters, it performs faster. This discrepancy could be attributed to optimization in the GPU code. Gemma3:27b, despite having a larger model, recorded an average inference time of 79.47s.

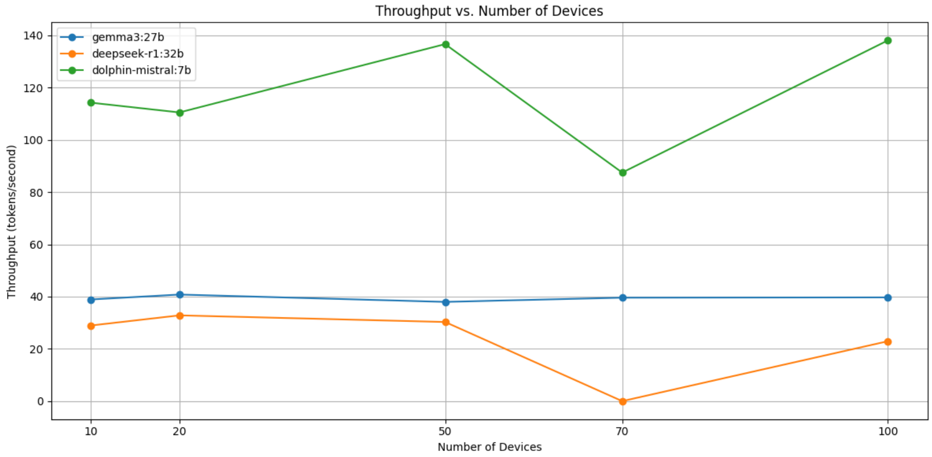


Figure 8. Throughput of different models varying the number of devices.

Table 4. Average time comparison.

Model	Avg. Time
Gemma3:27b	79.47 s
DeepSeek-R1:32b	48.49 s
Dolphin-Mistral:7b	28.53 s

To understand how these different LLM uses the resources, another analysis was performed considering the GPU usage during the inference time. Results of this analysis with 50 runs using the same LLMs, i.e., Gemma3, DeepSeek, and Doplhin Mistral, are reported in Figures 9, 10, and 11, respectively.

Table 5 presents the average GPU utilization across 50 experimental runs for the three evaluated LLMs. The results demonstrate that Dolphin-Mistral achieves the lowest GPU consumption, which is directly attributable to its reduced parameter count of 7 billion parameters. DeepSeek-R1 exhibits superior GPU efficiency compared to Gemma3, maintaining lower resource utilization despite having a larger parameter count. This efficiency gain can be attributed to enhanced GPU code optimization strategies implemented in the DeepSeek-R1 architecture.

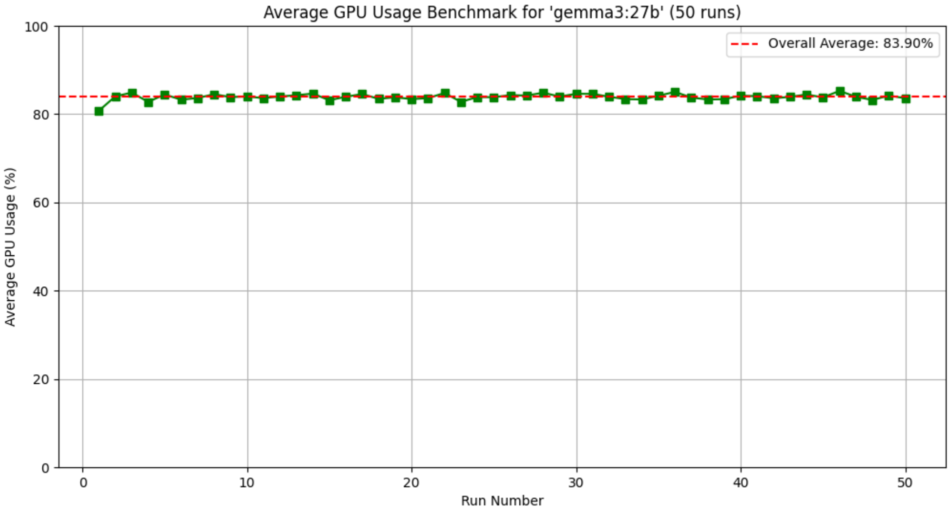


Figure 9. Average GPU usage for Gemma3:27b.

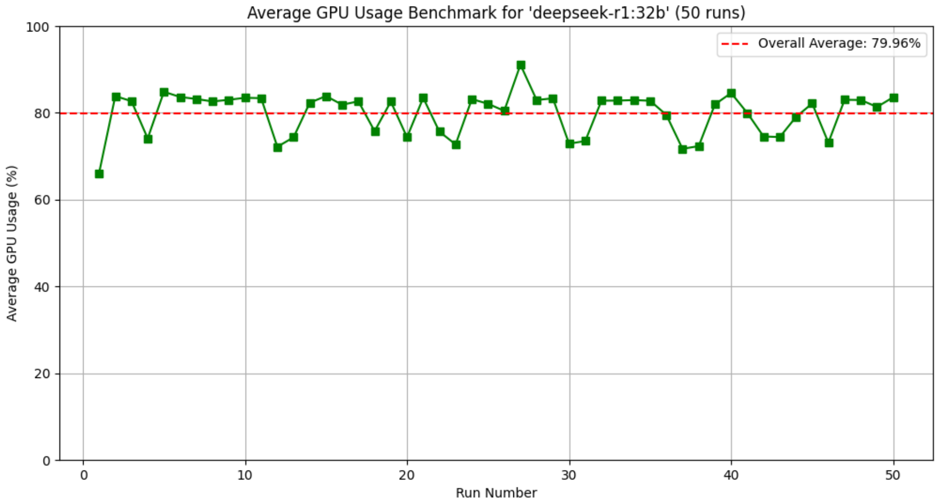


Figure 10. Average GPU usage for DeepSeek-R1:32b.

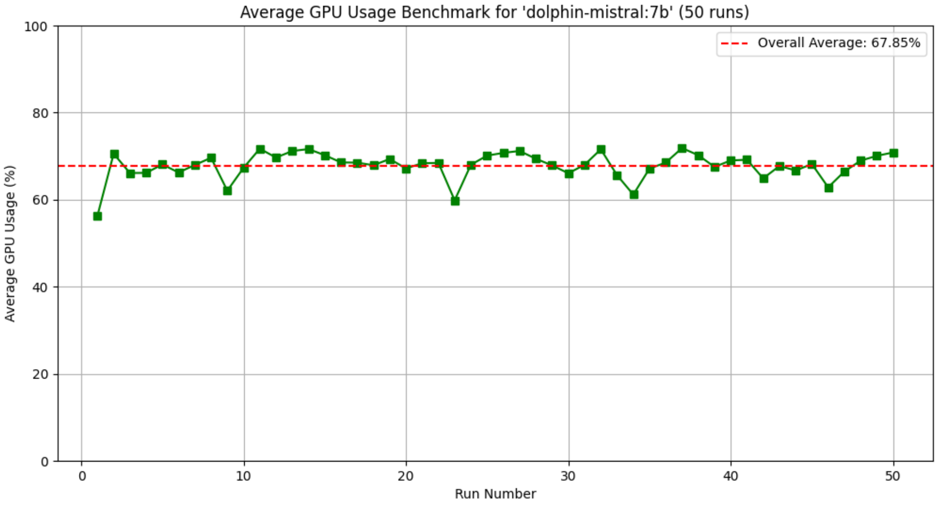


Figure 11. Average GPU usage for Dolphin Mistral:7b.

Table 5. Average time comparison.

Model	Avg. GPU Usage
Gemma3:27b	83.90%
DeepSeek-R1:32b	79.96%
Dolphin-Mistral:7b	67.85%

Although Gemma3 demonstrates the highest GPU utilization levels, reaching approximately 80% usage, it represents the optimal solution for our specific application requirements. This conclusion is supported by several key factors: Gemma3 maintains the most stable throughput performance as illustrated in Figure 8, operates within acceptable GPU utilization ranges, and delivers superior accuracy compared to the alternative models. The combination of stable performance characteristics and high accuracy justifies the increased computational overhead associated with Gemma3's deployment in industrial asset discovery applications.

The objective of these experimental results is to demonstrate the feasibility of the proposed architecture. The conducted experiments validate the applicability of the framework within industrial control system (ICS) environments. The integration of local and open-source models enables data privacy preservation and supports customization through training on proprietary datasets.

The analysis and corresponding plots further confirm the practical deployment of the framework on general-purpose and commercially available machines, without the need for specialized edge servers. This characteristic makes it possible to adopt the framework on-premises with limited computational resources and reduced cost while still maintaining acceptable inference times.

6.3. Current Limitations

The proposed framework demonstrates the feasibility of applying LLM-based approaches to automated asset discovery in ICS environments. Nonetheless, several limitations must be considered when evaluating its applicability to real-world settings.

At present, the proof of concept primarily addresses assets that are directly accessible through standard network scanning tools. Devices operating in isolated network segments remain outside its detection scope. Extending the sensing capabilities to include specialized parsers and out-of-band acquisition techniques would be necessary to achieve more comprehensive coverage.

The current implementation relies solely on software probes. Although the architecture has been designed to support physical-layer sensing modalities—such as electromagnetic or acoustic sensors—these were not included in the experimental validation. As a result, devices that remain silent on the network or exhibit intermittent connectivity may not be adequately captured.

The reliance on large language models introduces further constraints. Even when models are deployed locally, inference introduces non-negligible computational demands. Scalability experiments highlight that large network scans can exceed model token capacity, leading to partial processing of the input. To mitigate this, different strategies can be introduced. One involves automatically partitioning the discovery output into smaller, manageable subsets processed sequentially and later consolidated. A complementary approach is to adopt a multi-pass inference strategy, where an initial pass identifies broad asset categories and subsequent passes refine analysis within smaller subsets of devices. Alternatively, hierarchical decomposition of network topologies based on subnets or operational domains may allow for independent yet context-preserving processing of large infrastructures.

Another limitation concerns the potential for hallucinations in LLM-generated responses, where the model might infer device types or attributes that are not present in the input data. Although preliminary experiments show stable behavior, the risk remains significant in complex or ambiguous scenarios. To mitigate this, we envision a combination of output validation and consistency checking techniques. Strict schema validation can ensure that model outputs follow predefined formats, while cross-verification among specialized experts within the Mixture of Experts architecture can help detect inconsistencies. Rule-based sanity checks that compare model-generated classifications against deterministic scan data would allow for discarding unsupported inferences. Additionally, ensemble inference techniques could be used, where multiple fine-tuned models analyze the same input and results are merged based on confidence thresholds. Retrieval-augmented generation (RAG) mechanisms can further reduce hallucination risks by grounding the model’s reasoning in structured and verified knowledge bases rather than relying solely on probabilistic inference.

Finally, the validation was limited to a controlled test environment with a restricted number of devices and protocols. Industrial infrastructures often present significantly greater protocol diversity, coexistence of legacy and modern systems, and operational constraints that were not fully reproduced. Extensive field testing in live environments could confirm the robustness of the approach and its adaptability to operationally diverse and safety-critical deployments.

7. Conclusions

This investigation establishes foundations for intelligent, adaptive asset discovery in critical infrastructures through the application of large language models and Mixture of Experts architectures. The proposed framework addresses critical limitations in traditional identification methodologies by introducing an architecture that balances asset visibility with the operational safety constraints of industrial environments, advancing the state of the art through three key technical innovations.

First, the integration of large language models introduces **semantic context awareness** that eliminates the need for a priori data classification, enabling dynamic interpretation of heterogeneous industrial protocols and data sources without predetermined taxonomies. Second, the **Mixture of Experts framework** provides unified asset lifecycle management, coordinating specialized lightweight models for identification, vulnerability assessment, and network optimization within a single coherent architecture rather than requiring fragmented toolchains. Third, the system implements **non-intrusive multi-modal data fusion** that fundamentally departs from aggressive scanning methodologies, synthesizing information from passive observation, protocol-aware probing, and sensor inputs while preserving operational safety.

The validation through a testbed implementation confirms the practical feasibility of the proposed method in representative industrial control system (ICS) scenarios. Experimental evaluation supports the objective of this work, with inference performance ranging from 28.53 to 79.47 s in six-asset configurations, depending on model complexity and parameter size. The framework synthesizes multiple data sources—network activity, protocol metadata, and vendor identification—to perform asset classification beyond the capabilities of single-source approaches.

Comprehensive accuracy evaluation demonstrates the framework's classification reliability across different LLM architectures. Critically, the analysis reveals that Gemma3:27b maintains approximately 67% accuracy even with increased token counts, demonstrating exceptional stability for large-scale network deployments. The experimental results indicate potential for achieving accuracy levels approaching 80% through systematic prompt optimization and token management strategies, establishing a clear pathway for performance enhancement in future implementations.

Further experimental analysis demonstrates that the architecture can be deployed on general-purpose and commercially available machines, avoiding the need for specialized edge hardware. This aspect allows for on-premises deployment in environments with limited computational capacity and reduced cost while maintaining acceptable inference latency. The integration of local and open-source models also preserves data privacy and enables training on proprietary datasets, supporting customization for specific industrial contexts.

Despite its capabilities, the framework presents certain limitations. Its focus on network-accessible assets may limit applicability to air-gapped systems. LLM inference requirements impose computational overheads that challenge deployment in resource-constrained settings, and token limits affect scalability in large networks. Additionally, while the accuracy analysis demonstrates stable performance for Gemma3:27b, optimization of prompt engineering and token management remains necessary to achieve optimal classification performance across all model architectures. This study primarily utilized software-based probes for validation; however, the framework's multi-modal architecture is designed to accommodate diverse sensor inputs, and we anticipate significantly enhanced identification accuracy when the full spectrum of sensing modalities is validated in future work.

An additional consideration concerns silent input truncation, which, although not the primary focus of this study, represents a significant operational risk in production environments. From the experimental findings, several strategies naturally emerge from the proposed architecture to mitigate this issue. One option is to automatically segment large-scale network scan inputs into smaller batches that fit within token limits, process them sequentially, and consolidate the results while preserving context between network segments. Furthermore, the Mixture of Experts architecture enables iterative, multi-pass processing, where an initial inference stage performs broad device discovery and subsequent stages focus on targeted analysis of specific clusters, distributing the workload efficiently across experts. Lastly, the architecture can partition extensive networks into hierarchical segments—based on subnets, device categories, or operational domains—which can then be processed independently and merged through the data management layer. These mitigation techniques improve scalability and robustness against silent truncation without altering the core architectural principles.

In summary, this research moves beyond theoretical considerations toward a deployable system that solves three critical practical problems unaddressed by prior work: contextual adaptability without manual configuration, workflow integration across the complete asset lifecycle, and operational safety in safety-critical environments. The demonstrated classification accuracy of up to 66.67% with stability across varying network scales, combined with inference performance suitable for industrial deployment timelines, validates the framework's practical viability. The proposed framework demonstrates that intelligent, AI-driven solutions can operate effectively within industrial cybersecurity contexts, supporting the development of more robust and adaptive defense mechanisms for critical infrastructure protection.

Author Contributions: Conceptualization, L.C., A.I., R.N., and A.P.; Methodology, L.C., A.I., R.N., and A.P.; Software, L.C., A.I., R.N., and A.P.; Validation, L.C., A.I., R.N., and A.P.; Writing—original draft, L.C., A.I., R.N., and A.P.; Writing—review & editing, L.C., A.I., R.N., and A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the Industrial Cyber Shield (ICS) No. C83C22001460001 project funded by INAIL and by the research projects MIRABILE (CUP C67B23000300004)—CDS001108 "Filiera Automotive 4.0"—PNRR M1C2—5.2 Filiere produttive—"Automotive" NextGenerationEU and ISTINTO (CUP C85I23000400004)—CDS001133 "Steel Network 4.0"—PNRR M1C2—5.2 Filiere produttive—"Metallo ed elettromeccanica" NextGenerationEU.

Data Availability Statement: Dataset available on request from the authors.

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

References

1. Santos, S.; Costa, P.; Rocha, A. IT/OT convergence in industry 4.0: Risks and analysis of the problems. In Proceedings of the 2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Aveiro, Portugal, 20–23 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
2. Murray, G.; Johnstone, M.N.; Valli, C. The convergence of IT and OT in critical infrastructure. In Proceedings of the Australian Information Security Management Conference, Perth, Australia, 5–6 December 2017.
3. CISA. *ICS Advisory Report*; Technical Report; Cybersecurity and Infrastructure Security Agency: Washington, DC, USA, 2023.
4. Schrick, N.L.; Lorenzen, C.; Mitchell, C.; Rials, C.; Swartzwelder, R.; Kelley, C.; Sweeney, C.; Nelson, J.; Hendrix, A.; Kalohi, D.; et al. The Growth of Asset Identification in OT Environments and Remaining Challenges. In Proceedings of the 2024 Resilience Week (RWS), Austin, TX, USA, 3–5 December 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.

5. *ISO/IEC 27001:2022*; Information Security, Cybersecurity and Privacy Protection–Information Security Management Systems–Requirements, 3rd edition. International Organization for Standardization: Geneva, Switzerland; International Electrotechnical Commission: Geneva, Switzerland, 2022. Available online: https://www.iso.org/standard/27001 (accessed on 15 June 2025).
6. Angraini; Megawati; Haris, L. Risk Assessment on Information Asset an academic Application Using ISO 27001. In Proceedings of the 2018 6th International Conference on Cyber and IT Service Management (CITSM), Parapat, Indonesia, 7–9 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
7. Trend Micro. New Research Reveals Three Quarters of Cybersecurity Incidents Occur Due to Unmanaged Assets. 2025. Available online: https://newsroom.trendmicro.com/2025-04-29-New-Research-Reveals-Three-Quarters-of-Cybersecurity-Incidents-Occur-Due-to-Unmanaged-Assets (accessed on 30 July 2025).
8. Hanka, T.; Niedermaier, M.; Fischer, F.; Kießling, S.; Knauer, P.; Merli, D. Impact of active scanning tools for device discovery in industrial networks. In Proceedings of the Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2020 International Workshops, Nanjing, China, 18–20 December 2020; Proceedings 13; Springer: Berlin/Heidelberg, Germany, 2021; pp. 557–572.
9. Vermeer, M.; West, J.; Cuevas, A.; Niu, S.; Christin, N.; Van Eeten, M.; Fiebig, T.; Ganán, C.; Moore, T. SoK: A framework for asset discovery: Systematizing advances in network measurements for protecting organizations. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 440–456.
10. Yang, W.; Fang, Y.; Zhou, X.; Shen, Y.; Zhang, W.; Yao, Y. Networked Industrial Control Device Asset Identification Method Based on Improved Decision Tree. *J. Netw. Syst. Manag.* **2024**, *32*, 32.
11. Park, M.; Cho, S.J.; Kim, H. A study on asset identification in smart buildings automation systems. In Proceedings of the 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), Paris, France, 4–7 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 923–925.
12. Wang, H.; Eklund, D.; Oprea, A.; Raza, S. FL4IoT: IoT device fingerprinting and identification using federated learning. *Acm Trans. Internet Things* **2023**, *4*, 1–24.
13. Fakih, M.; Dharmaji, R.; Moghaddas, Y.; Quiros, G.; Ogundare, O.; Al Faruque, M.A. Llm4plc: Harnessing large language models for verifiable programming of plcs in industrial control systems. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice, Lisbon, Portugal, 14–20 April 2024; pp. 192–203.
14. Vasilatos, C.; Mahboobeh, D.J.; Lamri, H.; Alam, M.; Maniatakos, M. Llmpot: Automated llm-based industrial protocol and physical process emulation for ics honeypots. *arXiv* **2024**, arXiv:2405.05999.
15. Mo, S.; Salakhutdinov, R.; Morency, L.P.; Liang, P.P. Iot-lm: Large multisensory language models for the internet of things. *arXiv* **2024**, arXiv:2407.09801.
16. Cai, W.; Jiang, J.; Wang, F.; Tang, J.; Kim, S.; Huang, J. A survey on mixture of experts in large language models. *IEEE Trans. Knowl. Data Eng.* **2025**, *37*, 3896–3915.
17. Messe, N.; Chiprianov, V.; Belloir, N.; El-Hachem, J.; Fleurquin, R.; Sadou, S. Asset-oriented threat modeling. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 491–501.
18. TP-Link Technologies Co., Ltd. TL-WR940N V6 User Guide. Available online: https://www.tp-link.com/us/user-guides/TL-WR940N_V6/ (accessed on 1 June 2025).
19. Niryo SAS. NED2 User Manual v1.0.0; Six-Axis Collaborative Robot; User Manual for NED2 Robotic Arm. Available online: https://static.generation-robots.com/media/manuel-utilisation-ned2-niryo-en.pdf (accessed on 1 June 2025).
20. UltiMaker. UltiMaker S7 Pro Bundle—Technical Specification. Available online: https://ultimaker.com/3d-printers/s-series/ultimaker-s7-pro-bundle/ (accessed on 1 June 2025).
21. Lab, B. Bambu Lab X1E 3D Printer—Technical Specification. Available online: https://eu.store.bambulab.com/products/x1e?srsltid=AfmBOoqdcffZl66_xp9JL-x8gq8jGQTNvtnv9EGNQCfcgbuWmVjZV9GN (accessed on 1 June 2025).
22. Omron Corporation. NX-Series NX1P2 CPU Unit Built-in I/O and Option Board User's Manual. Available online: https://files.omron.eu/downloads/latest/manual/en/w579_nx-series_nx1p2_cpu_unit_built-in_i_o_and_option_board_users_manual_en.pdf?v=1 (accessed on 1 June 2025).
23. Python Software Foundation. Python 3.13.5 Documentation. Available online: https://docs.python.org/it/3/about.html (accessed on 1 June 2025).
24. Ollama Contributors. Ollama Documentation. 2025. Available online: https://github.com/ollama/ollama/tree/main/docs (accessed on 7 July 2025).
25. Google DeepMind. google/gemma-3-27b-it Model Card. Hugging Face Model Repository. 2025. Multimodal Gemma3 Model (27B), 128K Context Window; Access Requires Accepting Google's License. Available online: https://huggingface.co/google/gemma-3-27b-it (accessed on 1 June 2025).

26. NVIDIA Corporation. GeForce RTX 4090 Graphics Card. 2022. Available online: https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4090/ (accessed on 7 July 2025).
27. Nmap Project. Nmap Documentation; Includes Reference Guide, Man Page, Installation Instructions, Scripting Engine Documentation. Available online: https://nmap.org/book/man.html (accessed on 7 July 2025).
28. Atkinson, R.; Bhatti, S.N. Address Resolution Protocol (ARP) for the Identifier-Locator Network Protocol for IPv4 (ILNPv4). RFC 6747. 2012. Available online: https://www.rfc-editor.org/info/rfc6747 (accessed on 7 July 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.