A HYBRID FRAMEWORK FOR UNLABELED NETWORK INTRUSION DETECTION: INTEGRATING LLMS WITH STATISTICAL, HEURISTIC, AND SIGNATURE-BASED METHODS

UM FRAMEWORK HÍBRIDO PARA DETECÇÃO DE INTRUSÃO EM REDES NÃO ROTULADAS: INTEGRANDO LLMS COM MÉTODOS ESTATÍSTICOS, HEURÍSTICOS E BASEADOS EM ASSINATURAS

UN MARCO HÍBRIDO PARA LA DETECCIÓN DE INTRUSIONES EN REDES NO ETIQUETADAS: INTEGRANDO LLM CON MÉTODOS ESTADÍSTICOS, HEURÍSTICOS Y BASADOS EN FIRMAS

DOI: 10.52641/cadcajv11i17.163
Submitted on: 10.16.2026
Accepted on: 11.11.2026
Published on: 11.18.2026

Ruy de Oliveira
Postdoctoral Researcher in Computer Science
Institution: Instituto Federal de Mato Grosso (IFMT)

Mario de Oliveira
Postdoctoral Researcher in Electrical Engineering
Institution: Birmingham City University

Matheus Cândido Teixeira
Ph.D. Candidate in Computer Science
Institution: Instituto Federal de Mato Grosso (IFMT)

Valtemir Emerencio do Nascimento
Postdoctoral Researcher in Computer Science
Institution: Instituto Federal de Mato Grosso (IFMT)

Guilherme Pires Silva de Almeida
Ph.D. in Computer Science
Institution: Instituto Federal de Mato Grosso (IFMT)

ABSTRACT

Computer networks have become essential infrastructures underpinning modern society, enabling personal, corporate, and industrial communications on a global scale. However, as their ubiquity increases, so too does the complexity and attack surface available to malicious actors. Consequently, ensuring the security of these networks constitutes a critical challenge, given that vulnerabilities may be exploited to compromise data integrity, disrupt services, and inflict substantial damage. Against this

backdrop, this paper proposes an approach for network intrusion detection leveraging Large Language Models (LLMs). The proposed system employs LLM models such as LLaMA3 and Qwen3 to interpret unlabeled network packets, supported by statistical analyses, heuristics, YARA signatures, and entropy calculations, with the aim of identifying malicious patterns within network traffic flows. The results highlight the potential of LLMs for the semantic analysis of network traffic, while also demonstrating how complementary techniques can further enhance this capability.

Keywords: LLM; network traffic monitoring; anomaly detection; packet capture analysis; malware detection; intrusion detection system.

RESUMO

As redes de computadores tornaram-se infraestruturas essenciais que sustentam a sociedade moderna, possibilitando comunicações pessoais, corporativas e industriais em escala global. No entanto, à medida que a sua ubiquidade aumenta, também crescem a complexidade e a superfície de ataque disponíveis a atores maliciosos. Consequentemente, garantir a segurança dessas redes constitui um desafio crítico, dado que vulnerabilidades podem ser exploradas para comprometer a integridade dos dados, interromper serviços e causar danos substanciais. Nesse contexto, este artigo propõe uma abordagem para a detecção de intrusões em redes utilizando Grandes Modelos de Linguagem (LLMs). O sistema proposto emprega modelos de LLM, como LLaMA3 e Qwen3, para interpretar pacotes de rede não rotulados, apoiados por análises estatísticas, heurísticas, assinaturas YARA e cálculos de entropia, com o objetivo de identificar padrões maliciosos nos fluxos de tráfego de rede. Os resultados destacam o potencial dos LLMs para a análise semântica do tráfego de rede, ao mesmo tempo que demonstram como técnicas complementares podem aprimorar ainda mais essa capacidade.

Palavras-chave: LLM; Monitoramento de Tráfego de Rede; Detecção de Anomalias; Análise de Captura de Pacotes; detecção de malware; sistema de detecção de intrusões.

RESUMEN

Las redes de computadoras se han convertido en infraestructuras esenciales que sustentan la sociedad moderna, posibilitando comunicaciones personales, corporativas e industriales a escala global. Sin embargo, a medida que aumenta su ubicuidad, también crecen la complejidad y la superficie de ataque disponibles para actores maliciosos. En consecuencia, garantizar la seguridad de estas redes constituye un desafío crítico, dado que las vulnerabilidades pueden ser explotadas para comprometer la integridad de los datos, interrumpir servicios e infligir daños sustanciales. En este contexto, este artículo propone un enfoque para la detección de intrusiones en redes aprovechando los Grandes Modelos de Lenguaje (LLM). El sistema propuesto emplea modelos LLM como LLaMA3 y Qwen3 para interpretar paquetes de red no etiquetados, apoyados por análisis estadísticos, heurísticas, firmas YARA y cálculos de entropía, con el objetivo de identificar patrones maliciosos dentro de los flujos de tráfico de red. Los resultados destacan el potencial de los LLM para el análisis semántico del tráfico de red, al mismo tiempo que demuestran cómo las técnicas complementarias pueden mejorar aún más esta capacidad.

Palabras clave: LLM; Monitoreo de Tráfico de Red; Detección de Anomalías; Análisis de Captura de Paquetes; detección de malware; sistema de detección de intrusiones.

1 INTRODUCTION

The rapid advancement of communication technologies and the Internet has provided unprecedented connectivity among people, devices, and systems. However, this massive expansion has also introduced new challenges for information security. Cyberattacks have become increasingly sophisticated, exploiting vulnerabilities in corporate networks, critical systems, and Internet of Things (IoT) devices (Adam *et al.*, 2024). Network security has evolved from a restricted technical field into a central element of digital governance, requiring intelligent, adaptive solutions capable of operating in real time. In this context, methods based on Deep Learning have stood out due to their ability to recognize complex patterns and autonomously detect intrusions (Abbasi *et al.*, 2021).

With the emergence of Large Language Models (LLMs) (da Silva; Westphall, 2024; Zhang *et al.*, 2025), which are artificial intelligence models trained to understand, generate, and interpret natural language text, new possibilities have arisen for network traffic monitoring and analysis. Popular implementations of these models include GPT, Mistral, Qwen3, DeepSeek, LLaMA (Touvron *et al.*, 2023), and BERT (Devlin *et al.*, 2019). These solutions feature advanced architectures capable of processing large volumes of data and capturing complex patterns. Although originally designed for natural language, LLMs have shown potential for interpreting and contextualizing traffic data, extracting behavioral patterns, and identifying anomalies in packet flows (Kalafatidis *et al.*, 2025; Mahmood *et al.*, 2025; Yang *et al.*, 2025). Their generalization and contextual learning capabilities suggest that LLMs can strengthen cyber defense systems by enabling more accurate and automated responses to emerging threats.

However, despite this potential, the practical deployment of LLMs in network security continues to encounter significant challenges. The majority of existing monitoring solutions remain reliant on fixed rules, known signatures, or classical artificial intelligence models such as hybrid architectures combining neural networks and fuzzy logic (Culling, 2018). The problem is that these solutions exhibit limited efficacy when confronting unknown (zero-day) attacks or obfuscated traffic. Moreover, there is a notable scarcity of datasets that include packet captures (PCAPs) detailing network traffic at a granular level while also providing adequate labels for model training and evaluation (Kumar *et al.*, 2025; Tulczyjew *et al.*, 2024a, 2024b).

Given the growing sophistication of cyber threats and the inherent limitations of approaches based exclusively on rules or fixed signatures, this work proposes a novel approach for anomaly and intrusion detection in network traffic leveraging LLMs. The proposed methodology integrates statistical, heuristic analysis and signature-based detection using YARA rules with the reasoning and contextual correlation capabilities of LLMs, operating directly on packet captures to produce an explainable and actionable risk classification. Accordingly, the main contributions of this model are as follows:

- The proposal of a hybrid detection model that integrates LLMs with conventional statistical, heuristic, and signature-based methods;
- The deployment of locally hosted LLMs, which inherently safeguards sensitive network data, upholds privacy requirements, and guarantees operational autonomy by eliminating reliance on third-party cloud services;
- The development of a purpose-built, labeled dataset derived from wireless network environments, specifically designed to mitigate the persistent scarcity of high-quality packet capture data suitable for training and evaluating intrusion detection models;
- The evaluation of the proposed approach across diverse network scenarios to assess its generalizability and robustness.
- The remainder of this paper is organized as follows: Section 2 reviews selected related work on network security techniques, encompassing anomaly detection and LLM-based agents. Section 3 describes the material and methods in detail. Section 4 discusses the experimental setup and performance evaluation in real-world scenarios. Finally, Section 5 presents the conclusions and outlines future perspectives on the application of LLMs to network security.

2 RELATED WORK

The application of Artificial Intelligence (AI) to strengthen network security is not a new field, but the recent rise of LLMs has opened a promising research frontier. This section explores recent contributions that pave the way for the proposed work, covering reviews and foundations, practical applications, and autonomous agents.

2.1 GENERAL OVERVIEW AND LITERATURE REVIEWS

Zhou *et al.* (Zhou *et al.*, 2025) presented a comprehensive review of the use of LLMs in vulnerability detection and remediation, highlighting both theoretical advances and practical challenges in integrating these models into security pipelines. The study outlines the path toward future applications of LLMs in automated code analysis and vulnerability identification, serving as a foundation for subsequent research.

Silva and Westphall (da Silva; Westphall, 2024) also contributed with a comprehensive analysis of LLM applications in cyber-security, highlighting automated vulnerability analysis approaches and intelligent agents that interact with security tools. This broad perspective reinforces the potential of LLMs in tasks such as threat recognition, incident analysis, and response recommendations, demonstrating how these models can act as active components in cyber defense pipelines.

Complementing this overview, Zhang *et al.* (Zhang *et al.*, 2025) pointed out that Transformer-based models continue to lead the evolution of network intrusion detection. However, significant challenges remain, including the scarcity of representative datasets, high computational costs, and interpretability limitations. Earlier works, such as those by Abbasi *et al.* (Abbasi *et al.*, 2021) and Zhuravchak *et al.* (Zhuravchak *et al.*, 2025), have already indicated the need for more intelligent hybrid approaches to overcome these limitations.

Kim *et al.* (Kim *et al.*, 2025) propose a dual-path payload-aware intrusion detection framework combining a lightweight Xavier-CMAE model and an LLM-CMAE model that leverages pretrained LLM tokenizers. Xavier-CMAE achieves 99.9718% accuracy with a 0.0182% false positive rate without pretraining, while LLM-CMAE attains 99.9696% accuracy at higher computational cost. Evaluated on CIC-IDS2017, the work reveals a trade-off between efficiency and contextual detection capability, suggesting modular architectures for layered IDS deployment.

Adeseye *et al.* (Adeseye *et al.*, 2025) investigate privacy concerns and mitigation strategies in gamified organizational settings using locally hosted LLMs (LLaMA, Gemma, Phi) for qualitative analysis of 82 interview transcripts. They introduce five hallucination mitigation techniques: top-k sampling, temperature tuning, prompt optimization, noise reduction, and small-batch processing. Besides they propose a seven-dimensional evaluation framework for model selection. Results show that LLM-assisted analysis closely aligns with manual NVivo analysis, identifying five

main privacy concerns: data collection, usage, security, user awareness, and fairness/trust.

2.2 REPRESENTATION AND INTERPRETATION OF NETWORK TRAFFIC WITH LLMS

One of the main challenges in applying LLMs to network security is converting network traffic data into formats that language models can understand. In this context, Jiang and Alkhateeb (Jiang; Alkhateeb, 2024) proposed TrafficLLM, a model capable of representing packet flows in a structured textual format, enabling LLMs, originally trained on natural language, to be directly applied to traffic analysis. Guan *et al.* (Guan *et al.*, 2025) advance in this direction with LogLLM, a framework that combines BERT (Devlin *et al.*, 2019) and LLaMA (Touvron *et al.*, 2023) to detect anomalies in logs without relying on traditional parsers (programs that structure raw data into understandable formats).

Complementarily, Kumar *et al.* (Kumar *et al.*, 2025) explored the semantic interpretation of PCAP files, demonstrating how linguistic patterns can emerge from captured traffic. Stein *et al.* (Stein *et al.*, 2024) explored the use of few-shot learning to recognize malicious packets in military communications, showing that LLMs can identify suspicious behaviors even with limited datasets. Nze *et al.* (Nze *et al.*, 2026) combined MITRE ATTCK v14 with GPT-based models (via the OpenAI API) in a LangChain/Streamlit pipeline to auto-generate 400–600-word, industry-specific attack narratives anchored to concrete ATTCK technique IDs. Evaluated by 30 practitioners, the system achieved 85% technique coverage with a 3.2% false-positive rate and 2–8 s latency. The authors acknowledge limitations including dependence on commercial LLM APIs and the absence of validation against real-world red-team execution logs.

2.3 APPLICATIONS IN FAULT AND ANOMALY DETECTION

Tulczyjew *et al.* (Tulczyjew *et al.*, 2024a) proposed LLMcap, an LLM-based model for unsupervised fault detection in PCAP files. The study uses self-supervised learning to identify anomalous behaviors that degrade service quality, demonstrating the ability of LLMs to interpret complex patterns without relying on manual labels. Complementarily, the same authors (Tulczyjew *et al.*, 2024b) presented PCAPVision,

designed for high-volume and high-speed detection, reinforcing the feasibility of LLMs for diagnosis and automation in large-scale network environments.

2.4 LLM-BASED SECURITY FRAMEWORKS AND ARCHITECTURES

The work of Mahmood *et al.* (Mahmood *et al.*, 2025) proposed the LLM-Enhanced Security Framework for IoT Networks, which integrates LLMs with intrusion detection systems and malicious device identification in IoT environments. The approach stands out for its semantic precision in identifying anomalous patterns in heterogeneous traffic. Yang *et al.* (Yang *et al.*, 2025) present an LLM-based cloud traffic monitoring system that uses transfer learning and autoencoders to improve real-time anomaly detection, reinforcing the generalization potential of LLMs in distributed and highly dynamic contexts. Kalafatidis *et al.* (Kalafatidis *et al.*, 2025) complemented this perspective by proposing an LLM-based intrusion detection system for containerized environments, featuring a two-level architecture operating over SDN and Kubernetes.

2.5 LLMS AS AUTONOMOUS CYBERSECURITY AGENTS

Shen *et al.* (Shen *et al.*, 2025) introduced PentestAgent, an LLM-driven framework designed to automate penetration testing through the use of autonomous agents. By orchestrating interactions with security assessment tools, the system can identify vulnerabilities, plan attack strategies, and autonomously exploit discovered weaknesses. This approach represents a significant advancement toward intelligent, agent-based automation in offensive cybersecurity operations.

2.6 SYNTHESIS AND CURRENT CHALLENGES

The reviewed literature reveals a consistent movement toward integrating LLMs at multiple levels of cyber defense, from the semantic translation of packets to the automation of penetration testing (Shen *et al.*, 2025). However, a gap remains in the comparative evaluation of different open-source (Open Source) LLMs applied to network traffic analysis. In this context, our work proposes and implements a network intrusion detection system based on LLMs executed through the Ollama platform

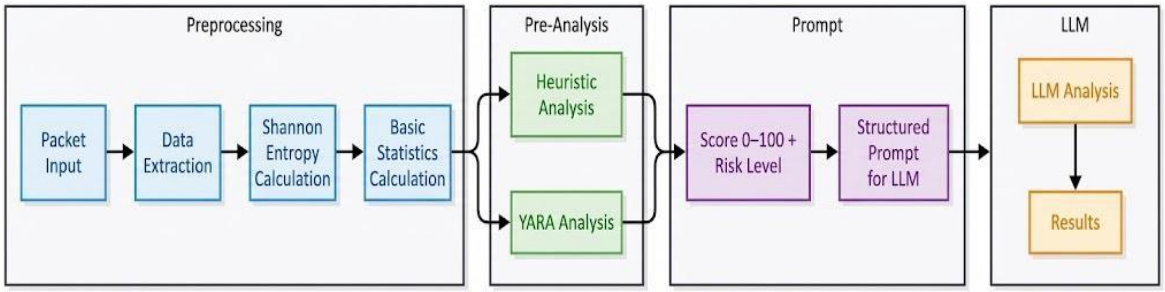
(Gruber; Weber, 2025). Details of the architecture, evaluation methodology, and experimental results are presented, aiming to demonstrate the system's efficiency, interpretability, and applicability for strengthening security in network environments.

3 MATERIAL AND METHODS

This section presents the proposed methodology for network traffic anomaly detection, organized as a multi-stage pipeline that combines traditional packet analysis, security heuristics, signature-based detection, and contextual reasoning provided by LLMs. Each stage builds upon and enriches the output of the previous one, progressively transforming raw packet capture data into structured, interpretable, and context-rich representations. This layered process enables the LLM to perform more informed analyses and effectively identify potential security threats and anomalous behaviors.

The proposed pipeline consists of four main stages: (i) preprocessing, (ii) pre-analysis, (iii) consolidation and prompt generation, and (iv) LLM-based analysis. As shown in Figure 1, the pipeline starts with the extraction and characterization of packets from network traffic, proceeds through heuristic evaluations and YARA signature-based detection (Wilson; Homan, 2021), and culminates in the construction of a structured prompt that integrates the collected evidence for interpretation by the LLM. This process enables the generation of contextualized security assessments based on multiple sources of network intelligence. The operation of each stage is described next.

Figure 1 – General workflow of the proposed system for anomaly detection in network traffic.



Source: :Authors.

3.1 STAGE I: PREPROCESSING

The process begins with the ingestion of data from network traffic and the systematic extraction of packet attributes using the Scapy (Wright, 2008) library in Python. This tool decomposes packets into their constituent fields, including source and destination IP addresses, port numbers, protocols (TCP, UDP, ICMP, DNS, ARP), timestamps, total length, and application-layer information when available.

Next, the Shannon Entropy of each packet is calculated as an indicator of encryption or obfuscation, as indicated in Equation (1):

$$H(X) = -\sum_x \; p(x)\; log_2\; p(x) \qquad (1)$$

where:

$H\,(X\,)$ = entropy of the dataset X
$p(x)$ = probability of occurrence of symbol x within X

This metric quantifies the degree of randomness or unpredictability present in the payload. This indicator is particularly useful because high entropy levels often reflect the use of encryption or obfuscation techniques, frequently observed in Command and Control (C2) channels, botnets, and Advanced Persistent Threats (APTs) (Lyda; Hamrock, 2007; Richer, 2017). This behavior has been repeatedly documented in recent studies on malicious traffic and anomaly detection (Abbasi; Shahraki; Taherkordi, 2021; Kumar *et al.*, 2025; Tulczyjew *et al.*, 2024a).

To reduce the false positive rate, the system employs allowlists of IP addresses and domains corresponding to well-established legitimate services (e.g., Google, Cloudflare, AWS), enabling the pre-filtering of benign traffic before heuristic evaluation. Additionally, adaptive thresholds are applied based on empirical traffic characteristics, specifically flow duration and observed Packets Per Second (PPS).

Complementary to this filtering mechanism, the system computes a set of basic statistical features, namely: protocol distribution, packet type counts, most frequently accessed ports, the cardinality of unique IP addresses, and the labeling of packets exhibiting high entropy. These features collectively form the factual foundation upon which subsequent analyses are conducted.

3.2 STAGE II: PRE-ANALYSIS

Following packet extraction and initial preprocessing, the data undergo two parallel analyses: a structured heuristic evaluation and a complementary inspection using YARA, a rule-based pattern matching system (Culling, 2018; Wilson; Homan, 2021), as depicted in Figure 2. Both processes aim to identify malicious patterns in network traffic and are detailed next.

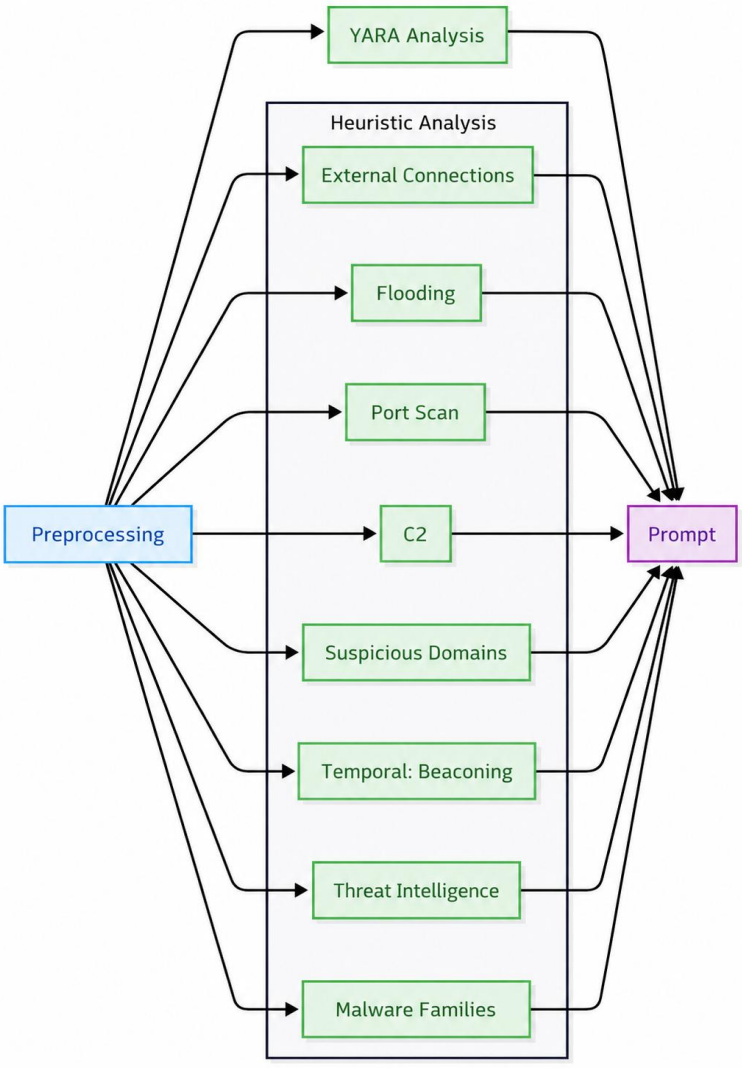
The heuristic analysis synthesizes aggregate traffic statistics and behavioral hallmarks of malicious activity. This module comprises the following constituent components:

- **External connections:** identification of flows established between internal hosts and remote destinations, with emphasis on unusual communications, non-standard ports, or geographically atypical endpoints;
- **Flooding:** detection of anomalous packet transmission rates from a single host, characterizing possible denial-of-service (DoS) attacks or aggressive automated behaviors;
- **Port scan:** correlation of multiple connection attempts to different ports on the same destination, including horizontal patterns (many ports on a single host) and vertical patterns (the same port across multiple hosts);
- **C2 communication:** evaluation of characteristics typical of Command and Control channels, including connection persistence, the use of uncommon ports, and traffic profiles that deviate from the expected behavior of the application;
- **Suspicious domains:** inspection of requests involving newly created domains, domains with questionable reputation, or domains associated with known malicious campaigns;
- **Temporal analysis (beaconing):** calculation of packet interval variance to identify rigid periodicity (often indicative of bots performing regular check-ins with C2 servers);
- **Threat Intelligence (TI):** comparison of observed IPs, domains, and hashes against external sources of Indicators of Compromise (IOCs), strengthening detection with data obtained from updated feeds;

● **Malware families:** preliminary categorization of behavioral patterns compatible with specific malware families, relying on statistical signatures and characteristic traits present in the traffic.

The YARA-based analysis relies on an open-source tool (Culling, 2018) widely used for the identification and classification of malicious code (Wilson; Homan, 2021). Recognized in the security industry as a versatile pattern-matching mechanism, YARA enables the description of malware families through textual or binary patterns (Wilbur, 2024). Each rule consists of a set of strings and a Boolean expression that defines the detection logic. This flexibility allows the inspection of arbitrary data streams, making the tool particularly suitable for Deep Packet Inspection (DPI) in network traffic.

Figure 2 – Heuristic analysis module, integrating multiple types of anomalies and consolidating the result into a risk score.



Source: :Authors.

The YARA module evaluates payloads extracted from packets and identifies previously cataloged patterns of malware or offensive tools (e.g., Emotet, TrickBot, Cobalt Strike) (Wilson; Homan, 2021). In this way, the module complements the heuristic analysis by providing deterministic signature-based detections and reducing uncertainties inherent to purely behavioral methods.

As a result, YARA analysis produces a structured set of match objects that provide not only an indication of the triggered rule but also relevant metadata, including rule identifiers, classification tags, and the precise offsets at which the corresponding patterns were detected. This level of granularity enables more than simple binary detection, allowing the categorization of malware families or attack vectors and providing direct support for response decisions in subsequent stages of the pipeline.

3.3 STAGE III: PROMPT (SCORING AND CONSOLIDATION)

The results from heuristics and YARA analyses are then combined into a single risk score on a 0–100 scale, as shown in the Equation (2).

$$S = min(100, \sum_{i=1}^{n} w_i) \quad (2)$$

where:

S = final score
w_i = score assigned to each piece of evidence or category
n = total number of evidence items (or categories) being summed

The scoring mechanism weights evidence according to specific criteria as follows: critical DDoS attacks (up to 30 points), multiple external connections indicative of botnet activity (up to 25 points), port scanning (up to 20 points), high-entropy C2 communication (up to 20 points), and known malicious domains (up to 10 points), with the total capped at 100 points. Thus, the score serves as an indicator of the degree to which the network is compromised.

To protect the proposed model against prompt injection attacks, all data sent to the LLM undergo a sanitization process before analysis. This process removes control characters and filters potentially malicious or instruction-like content embedded in DNS

domain names. The objective is to ensure that domain names are treated strictly as data, rather than being interpreted by the LLM as commands or instructions.

Although YARA analysis is highly effective at detecting known patterns, it fails to identify novel variants, obfuscated traffic, or emerging malicious behaviors. Heuristic methods, in turn, are prone to false positives, as they rely solely on statistical patterns without semantic understanding. To address the limitations of both approaches, the consolidation stage integrates the two methods with an LLM-based analysis: signatures provide precision, heuristics provide broad coverage, and the LLM supplies contextual interpretation.

The consolidated output, which includes statistical features, detected anomalies, YARA matches, and the final risk score, is transformed into a structured prompt and submitted to a local LLM executed through the Ollama framework (Gruber; Weber, 2025). The primary purpose of this prompt is to provide the LLM with a holistic view of the network context, thus facilitating more resilient and context-aware interpretive reasoning.

3.4 STAGE IV: LLM-BASED ANALYSIS

The final stage consists of contextual analysis performed by the LLM, which correlates heterogeneous evidence, identifies behavioral patterns, and generates a forensic interpretation that includes:

- probable threat category (e.g., C2, exfiltration, automated scanning);
- explanation of the observed behavior;
- justifications based on the received data;
- mitigation recommendations adapted to the scenario.

The final results are consolidated into reports that include quantitative metrics, risk classifications, behavioral patterns, and contextual interpretations generated by the LLM. While the previous stages produce fragmented indicators, the LLM synthesizes this information into an integrated, detailed, and context-aware analysis. Thus, the proposed approach satisfies both the need for technical precision and the demand for interpretable, actionable insights. The combination of heuristics, signatures, and LLMs collectively strengthens the method’s robustness and adaptability.

4 EXPERIMENTAL EVALUATIONS

The four experiments described in this section were designed to assess the effectiveness of the proposed hybrid approach in detecting network threats. To this end, two carefully selected datasets were used to test the limits and specific capabilities of the proposed detection modules. Furthermore, all tests were executed locally, using the Llama3 and Qwen3 (8B) models, so as to ensure full reproducibility of the environment and enable the analysis of interpretive consistency across different LLMs.

4.1 DATASETS

Dataset I features two samples of real malicious traffic from the Malware Traffic Analysis repository (Landers, 2025): Bumblebee and the Neutrino Exploit Kit. Both samples capture the complete infection lifecycle, including:

- Malicious payload download;
- Initial C2 (Command and Control) communication;
- Internal network reconnaissance;
- Browser vulnerability exploitation;
- Non-standard payload encryption.

Dataset II was generated by the authors in a real-world Wi-Fi network environment using a standard access point. The network demonstrated an average latency ranging from 5 to 15 ms and a nominal bandwidth of 300 Mbps, alongside a typical baseline load consisting of devices connected to the router. The test infrastructure consisted of two physical hosts: an attacker machine (Intel Core i5, 12th generation, 16 GB RAM, Linux) and a victim machine (AMD Ryzen 5, 8 GB RAM, Windows). Attack traffic was generated using hping3 and included three distinct vectors: SYN Flood, UDP Flood, and Distributed SYN Flood (simulated). Packet capture was conducted on the victim machine using Wireshark 4.4.9, while traffic merging was implemented in Python 3 with Scapy 2.6.1. Yet, Dataset II contains:

- Benign traffic (10,000 packets);
- SYN Flood: Massive transmission of packets with the SYN flag enabled, aiming to exhaust the victim’s connection table (5,000 packets);

● UDP Flood: Bandwidth saturation through large UDP datagrams (5,000 packets);

● Distributed SYN Flood (Simulated): Use of the IP Spoofing technique (source IP address forgery) to simulate a distributed attack originating from a botnet (10,000 packets).

After collection, the malicious traffic was merged with legitimate traffic in order to challenge the system to distinguish normal usage from denial-of-service attempts based solely on metadata. All datasets used in this work are available in the project's GitHub repository(Jauer *et al.*, 2025).

4.2 EXPERIMENT I: MALWARE

In this experiment, we evaluate the system's ability to correlate YARA signatures with LLM-based contextual interpretation to detect malware whose traffic resembles legitimate applications (e.g., HTTP/HTTPS). The focus was to assess the model's sensitivity to stealthy patterns, obfuscated traffic, and encrypted payloads, even in the absence of volumetric deviations, such as anomalous packet rates or abrupt traffic spikes, that would typically trigger detection by traditional rate-based approaches.

Two malicious traffic datasets were utilized in this experiment: the Bumblebee loader and the Neutrino Exploit Kit. Bumblebee (a malicious program designed to establish initial system access) is characterized by Command and Control (C2) communication and initial reconnaissance routines. The Neutrino Exploit Kit, in turn, employs advanced obfuscation, irregular TLS encapsulation, and encrypted payloads deliberately engineered to evade superficial inspection. Across both datasets, Shannon entropy reliably identified highly random segments on common ports (80/443), while the LLM proved essential for discriminating legitimate connections from the atypical cryptographic patterns that define modern malware.

In the analyzed traffic associated with Bumblebee, multiple connections were observed originating from host 192.168.1.100 toward address 169.254.178.151, using ports above 1024. Although they did not constitute a denial-of-service attack, these connections exhibited unusual characteristics, such as approximately 1,800 packets transmitted in only ten seconds and communication attempts through ports historically

associated with suspicious behavior (139 and 445). Figure 3 shows the consolidated output for this scenario.

Also in Figure 3, it is possible to observe that the scan of the YARA module correctly detected the Bumblebee malware, identifying 5 (five) related files, indicating malicious network communication and vulnerability exploitation, which are the following ones: Bumblebee_Core_Network_Traffic and Bumblebee_LDAP_Advanced_Reconnaissance, pointing to typical loader artifacts involving C2 channels and reconnaissance phases. The combination of high entropy, use of irregular ports, content anomalies, and multiple YARA signatures resulted in a risk score of 79/100 (High). The LLM reinforced the diagnosis by highlighting behavior consistent with the establishment of remote communication, although no additional recommendations were necessary since the detection was conclusively determined by the YARA rules.

Regarding the obfuscated traffic associated with the Neutrino Exploit Kit, as expected, the heuristic engine did not identify relevant volumetric anomalies, a behavior consistent with the stealthy nature of Neutrino. Detection occurred primarily through the correlation between YARA signatures and structural anomalies, including five matches associated with cryptographic patterns and automated exploitation routines. Among them, the rule Neutrino_EK_Encrypted_Payload stood out, suggesting non-standard encrypted encapsulation directed to port 443.

Figure 3 – Result of the analysis performed in Experiment I.

```
### 1. RISK CLASSIFICATION: **Medium**
- **Based on:**  No attack confirmed by the heuristic engine.
- **Key indicators:**
  - **Multiple connections:** 169.254.170.151 (destination IP) and 192.168.1.100 (source IP) show connections on
    non-standard ports (>1024).
  - **Unusual traffic:** 169.254.170.151 has 1,800 packets in 10 seconds, which may indicate port scanning
    or other suspicious activity.
  - **External ports:** Ports such as 139 (SMB) and 445 (SMB) on 169.254.170.151 are associated with
    malicious network activities.

### 2. TRAFFIC YARA CORRELATION:
- **Detections:**
  - **Bumblebee Core Network Traffic (4 files):** Indicates malicious network activity related to C2 communication.
  - **Bumblebee LDAP (1 file):** Suggested use of LDAP protocol for vulnerability exploration.
- **Severity:** **Medium** (suspicious activity, but not confirmed as DOS attacks).

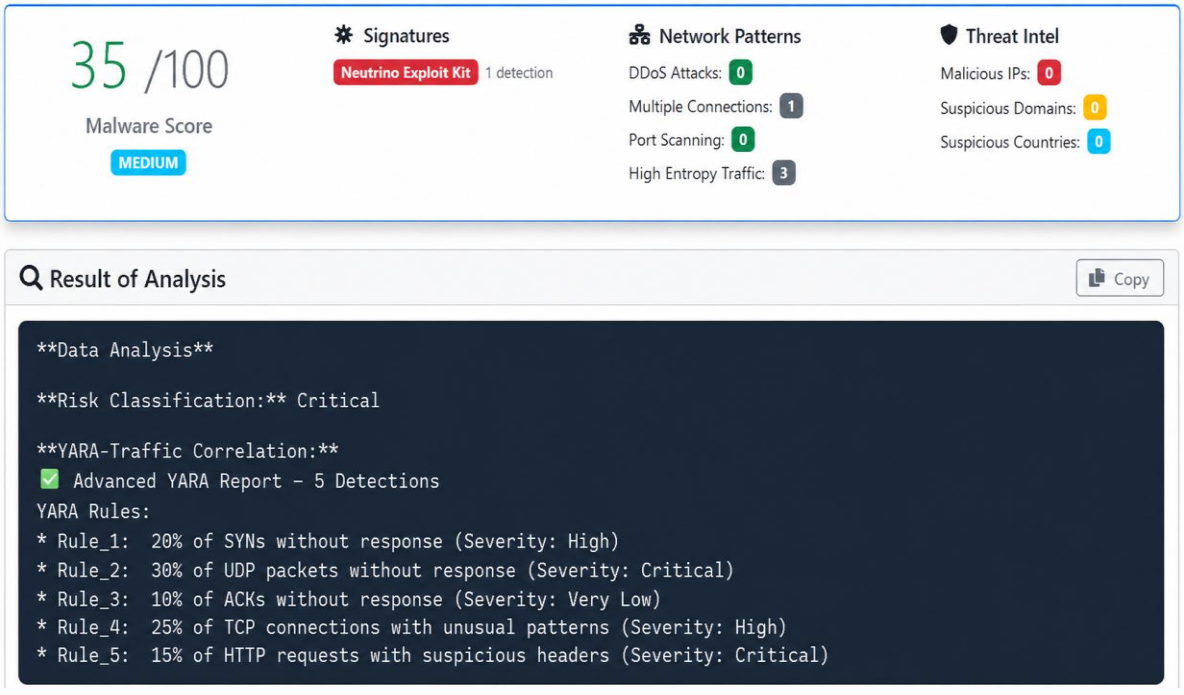
---
```

Source: :Authors.

Despite the absence of volumetric indicators, the structural analysis identified inconsistencies in approximately 25% of TCP flows, in addition to suspicious HTTP headers in 15% of the requests. As shown in Figure 4, which includes a Graphical User interface developed by the authors of this paper, the aggregation of evidence (YARA signatures, high entropy, and protocol irregularities) resulted in a score of 35/100 (Medium/Low). This value reflects the lack of direct impact on network availability, although the traffic still represents a significant threat to the compromised host.

Finally, the response generated by the LLM complemented both analyses by correctly interpreting the context of each threat: in the case of Bumblebee, reinforcing the risk of C2 communication; and in the case of Neutrino, indicating the need for forensic investigation and a review of TLS inspection policies. These results demonstrate that the combination of YARA and LLM-based contextualization is effective even in the presence of stealthy patterns and highly obfuscated traffic.

Figure 4 – Detection of the Neutrino Exploit Kit. The low score (35) reflects the absence of impact on availability, while the YARA alerts and the LLM's contextual interpretation correctly identify the exploitation attempt.



Source: :Authors.

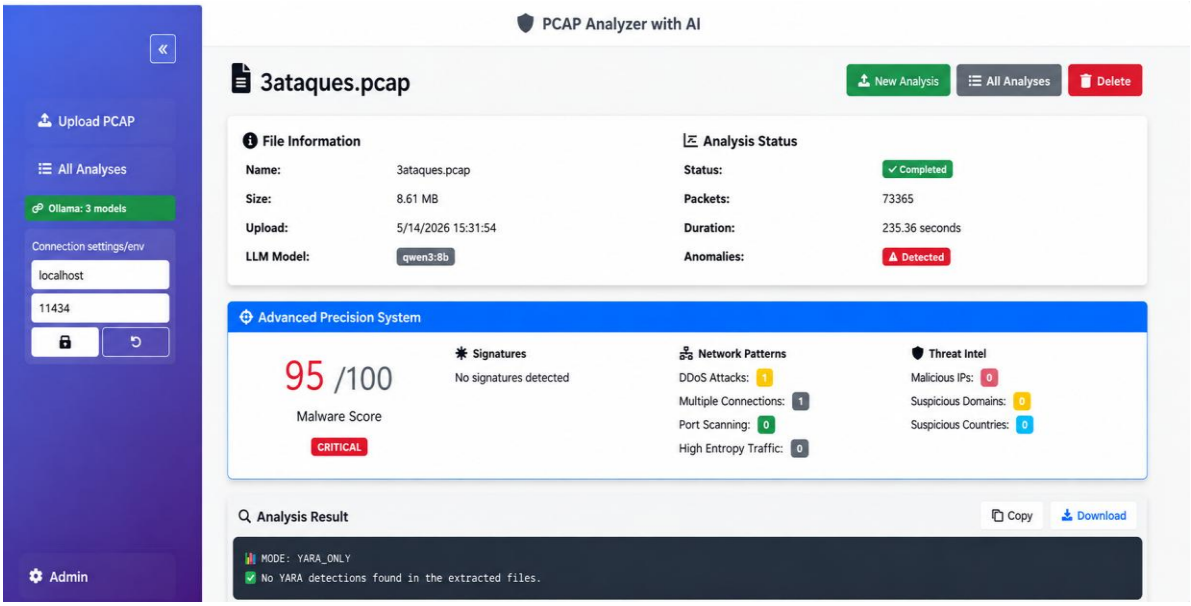
4.3 EXPERIMENT II: VOLUMETRIC ATTACKS (DOS/DDOS)

This experiment evaluates the effectiveness of the heuristic approach in detecting large-scale attacks without employing deep packet inspection (DPI). The evaluation is conducted using Dataset II, which represents mixed traffic containing multiple simultaneous attack vectors.

The heuristic analysis identified extreme asymmetry between requests and responses: the SYN/ACK ratio was only 0.0083, indicating a SYN Flood. In addition, spikes in UDP traffic (20,000 packets) and ICMP traffic (15,000 packets) were detected. The developed GUI shown in Figure 5, highlights that the system assigned a score of 95/100 (Critical) to this evaluated scenario.

The LLM diagnosis proved operationally useful: in addition to correctly identifying the pattern as a distributed DoS attack, it automatically generated mitigation rules (e.g., iptables -A INPUT -s 192.168.1.100 -j DROP), reducing response time.

Figure 5 – Analysis result for Scenario of experiment II, showing a Risk Score of 95/100 and the simultaneous detection of three Denial-of-Service (DoS) attack vectors.



Source: :Authors.

4.4 EXPERIMENT III: ABLATION STUDY

To evaluate the impact of the heuristic modules on the performance of the proposed architecture, an ablation study was conducted using the scenario from

Experiment II, which involves denial-of-service attacks. It is also worth noting that the YARA module does not play a relevant role in this experiment because its purpose is to identify malware signatures. Therefore, only the impact of the heuristic module was evaluated. The system was tested in “Pure LLM” mode (without heuristics and YARA) and compared with the complete hybrid system. The results are presented in Table 1.

Table 1 – Effectiveness Comparison: Pure LLM vs. Complete Hybrid System (Scenario 1).

Analysis Metric	**Pure LLM**	**Complete Hybrid System**
Attack Detection	Failed (Not Detected)	Successful (3 Vectors)
Risk Classification	Low / Normal	Critical (Score 95)
Diagnosis	“Normal client traffic”	“Distributed DoS Attack”
Conclusion	No action required	Immediate IP blocking and WAF deployment

Source: Authors.

The standalone LLM interpreted the massive attack as normal traffic, failing to detect the three attack vectors. In contrast, the complete hybrid model successfully detected the DoS attacks and, in particular, recommended immediate IP blocking and WAF (Web Application Firewall) deployment. These results reinforce that LLMs alone do not possess the statistical inference capability required for real-time analysis and that preprocessing modules are essential.

4.5 EXPERIMENT IV: CROSS-VALIDATION OF MODELS (LLAMA3 VS. QWEN)

The results presented previously were obtained exclusively using the Qwen3 8B LLM model. To evaluate consistency across models, the experiments were also performed using Llama3 LLM model. Both models assigned identical scores and reached the same operational conclusions. However, a qualitative difference was observed: Llama3 produced more descriptive and strategic analyses, whereas Qwen3 stood out for generating technical mitigation commands, making it more suitable for NetOps profiles (operating, monitoring, and maintaining networks on a daily basis).

In summary, although Qwen3 yielded more comprehensive diagnostic outputs, it demanded greater computational resources and exhibited extended response times (by 2–3 minutes).

5 CONCLUSIONS

This article has proposed and evaluated a hybrid detection framework that integrates locally deployed Large Language Models (LLMs) with conventional statistical, heuristic, entropy-based, and YARA signature analyses for network intrusion detection. The experimental evidence substantiates the principal contributions of this study.

The hybrid architecture consistently outperforms LLM-only inference, underscoring the synergistic value of combining traditional indicators with generative reasoning. Moreover, the localized deployment strategy rigorously preserves data privacy and operational autonomy by circumventing third-party cloud processing, a critical requirement for sensitive network environments. To address the enduring lack of

representative wireless traffic datasets, we contribute a newly developed labeled corpus captured from a production Wi-Fi network, expressly designed to facilitate robust model training and evaluation.

The comprehensive assessment of the implemented application confirms its effectiveness and readiness for practical deployment, with results that validate both its detection capabilities and operational efficiency. Finally, to uphold the principles of reproducibility and to support future research, we have made the full experimental suite, including all evaluation scripts and the generated dataset, openly available on GitHub. The repository link is provided in Subsection 4.1.

As future work, the following directions are suggested: increasing the representation of attacks targeting wireless networks, performing real-time analyses, conducting experiments with additional LLM models, applying prompt engineering techniques to improve accuracy, and training the LLM to specialize it for specific network traffic environments.

ACKNOWLEDGEMENTS

The authors express their sincere gratitude to the students Julio Cesar Jauer, Mizael Davi da Silva, and Thiago Ramalho Setúbal for their valuable assistance with the experimental setup and data collection.

REFERENCES

ADAM, M.; HAMMOUDEH, M.; ALRAWASHDEH, R.; ALSULAIMY, B. A Survey on Security, Privacy, Trust, and Architectural Challenges in IoT Systems. **IEEE Access**, [S. l.], v. 12, p. 57128-57149, 2024.

ABBASI, M.; SHAHRAKI, A.; TAHERKORDI, A. Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. **Computer Communications**, [S. l.], v. 170, p. 19-41, 2021.

ADESEYE, A.; ISOAHO, J.; TAHIR, M. Insights on Mitigating Privacy Concerns in Gamification through LLM-Assisted Qualitative Analysis with Minimal Hallucination. **ACM Transactions on Privacy and Security**, [S. l.], 2025.

CULLING, C. **Which YARA rules rule**: basic or advanced? [S. l.]: GIAC (GCIA) Gold Certification and RES, 2018.

DA SILVA, G. J. C.; WESTPHALL, C. B. **A Survey of Large Language Models in Cybersecurity**. [S. l.]: ArXiv preprint, 2024.

DEVLIN, J.; CHANG, M. W.; LEE, K.; TOUTANOVA, K. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. [S. l.]: arXiv preprint arXiv:181004805, 2019.

GRUBER, J. B.; WEBER, M. Ollama: A Quick-Start Guide for Using Large Language Models Locally. **In: Generative AI and Data Science Applications**. Cham: Springer Nature Switzerland, 2025. p. 23-35.

GUAN, W.; CAO, J.; QIAN, S.; GAO, J.; OUYANG, C. **LogLLM: Log-based Anomaly Detection Using Large Language Models**. [S. l.]: ArXiv preprint, 2025.

JAUER, J. C.; SILVA, M. D.; SETÚBAL, T. R. **Segurança de Redes Usando Modelos de LLMs**. 2025. Disponível em: https://github.com/jcjauer/Seguranca-de-Redes-Usando-Modelos-de-LLMs. Acesso em: 10 out. 2026.

JIANG, S.; ALKHATEEB, A. **Digital Twin Aided Massive MIMO: CSI Compression and Feedback**. [S. l.]: ArXiv preprint, 2024.

KALAFATIDIS, S.; PAPAGEORGOPOULOS, N.; KARTAKOULLIS, A.; LEDAKIS, G. LLM-Enhanced Intrusion Detection for Containerized Applications: A Two-Tier Strategy for SDN and Kubernetes Environments. In: AVAILABILITY, RELIABILITY AND SECURITY, LECTURE NOTES IN COMPUTER SCIENCE, 2025. **Proceedings [...]**. [S. l.]: Springer, 2025. p. 55-73.

KIM, Y. C.; LEE, C.; YOON, Y. Payload-Aware Intrusion Detection with CMAE and Large Language Models. **ACM Transactions on Privacy and Security**, [S. l.], v. 29, n. 1, nov. 2025.

KUMAR, A.; GUPTA, G. R.; KUMAR, S.; SAHU, A.; CHAKRABORTY, A. What Does Your Packet Capture Say? In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON DATA SCIENCE AND MANAGEMENT OF DATA (ACM CODS-COMAD), 2025. **Proceedings [...]**. [S. l.]: ACM, 2025. p. 427-431.

LANDERS, B. **Malware Traffic Analysis**. 2025. Disponível em: https://www.malware-traffic-analysis.net. Acesso em: 08 nov. 2026.

LYDA, R.; HAMROCK, J. Using Entropy Analysis to Find Encrypted and Packed Malware. **IEEE Security & Privacy**, [S. l.], v. 5, n. 2, p. 40-45, mar. 2007.

MAHMOOD, M. A. I.; ASHAB, F.; SOHAN, M. S.; CHY, M. H. I.; KADER, M. F. LLM-Enhanced Security Framework for IoT Network: Anomaly Detection and Malicious Devices Identification. **IEEE Access**, [S. l.], v. 13, p. 168405-168419, 2025.

NZE, P. E.; ADEMUWAGUN, A. K.; BELLO, M.; IFEANYI, F. D.; ABDULLAHI, S. M.; TIGHIL, J. Large language model-assisted threat-driven testing system for enhanced cybersecurity readiness. **Journal of Cyber Security**, [S. l.], v. 8, n. 1, p. 469-486, 2026.

RICHER, T. J. Entropy-based detection of botnet command and control. In: PROCEEDINGS OF THE AUSTRALASIAN COMPUTER SCIENCE WEEK MULTICONFERENCE (ACSW '17), 2017. **Proceedings [...]**. New York: Association for Computing Machinery, 2017. p. 1-4.

SHEN, X.; WANG, L.; LI, Z.; CHEN, Y.; ZHAO, W.; SUN, D. *et al.* PentestAgent: Incorporating LLM Agents to Automated Penetration Testing. In: PROCEEDINGS OF THE 20TH ACM ASIA CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 2025. **Proceedings [...]**. [S. l.]: ACM, 2025. p. 375-391.

STEIN, K.; MAHYARI, A. A.; FRANCIA, G.; EL-SHEIKH, E. Towards Novel Malicious Packet Recognition: A Few-Shot Learning Approach. In: IEEE MILITARY COMMUNICATIONS CONFERENCE (MILCOM), 2024. **Proceedings [...]**. [S. l.]: IEEE, 2024. p. 847-852.

TOUVRON, H.; MARTIN, T.; STONE, P. *et al.* **LLaMA: Open and Efficient Foundation Language Models**. [S. l.]: ArXiv preprint, 2023.

TULCZYZJEW; BIRUK, I.; BILGIC, M.; ABONDO, C.; WEILL, N. PCAPVision: PCAP-Based High-Velocity and Large-Volume Network Failure Detection. In: ACM SIGCOMM WORKSHOP ON NETWORKS FOR AI COMPUTING, 2024. **Proceedings [...]**. [S. l.]: ACM, 2024. p. 26-33.

TULCZYZJEW; JARRAH, K.; ABONDO, C.; BENNETT, D.; WEILL, N. LLMcap: Large Language Model for Unsupervised PCAP Failure Detection. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS (ICC WORKSHOPS), 2024. **Proceedings [...]**. [S. l.]: IEEE, 2024. p. 1559-1565.

WILBUR, L. G. **Implementing Selective Signature Scanning to Optimize Malware Detection**. 2024. Dartmouth College Computer Science Senior Theses.

WILSON, S.; HOMAN, J. **Which YARA Rules Rule? A Guide to Improving Detection and Hunting Capabilities**. Virus Bulletin Conference (VB2021) Presentation Slides, 2021. Disponível em: https://vt-gtm-wp-media.storage.googleapis.com/2.0-Which-YARA-Rules-Rule-Basic-or-Advanced-1.pdf. Acesso em: 08 nov. 2026.

WRIGHT, C. S. **Searching for Exploits**: SCAPY Fuzzing. [S. l.]: SSRN Electronic Journal, 2008.

YANG, Z.; JIN, Y.; LIU, J.; XU, X.; ZHANG, Y.; JI, S. Research on Cloud Platform Network Traffic Monitoring and Anomaly Detection System based on Large Language Models. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, INFORMATION SYSTEM AND COMPUTER ENGINEERING (CISCE), 2025. **Proceedings [...]**. [S. l.]: IEEE, 2025. p. 1029-1032.

ZHANG, J.; BU, H.; WEN, H.; LIU, Y.; FEI, H.; XI, R. *et al.* When LLMs meet cybersecurity: a systematic literature review. **Cybersecurity**, [S. l.], v. 8, p. 55, 2025.

ZHOU, X.; CAO, S.; SUN, X.; LO, D. Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead. **ACM Transactions on Software Engineering and Methodology**, [S. l.], v. 34, n. 5, p. 145, 2025.

ZHURAVCHAK, A.; PISKOZUB, A.; SKORYNOVYCH, B.; LAKH, Y.; ZHURAVCHAK, D.; HLUSHCHENKO, P. *et al.* Design and development of a large language model-based tool for vulnerability detection. **Eastern-European Journal of Enterprise Technologies**, [S. l.], v. 2, n. 2 (134), p. 75-83, 2025.