I.J. Intelligent Systems and Applications, **2026, 2, 125-138**
Published Online on April 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2026.02.09

Framework for Incident Identification Based on LLMs and Cybersecurity Ontologies

Wallace A. Pinheiro*
Center for Systems Development, Brazilian Army, Brazil
Military Institute of Engineering (IME), Brazilian Army, Brazil
E-mail: wallaceapinheiro@gmail.com
ORCID iD: https://orcid.org/0000-0001-7076-8785
*Corresponding author

Ricardo Q. A. Fernandes
National Government CSIRT, Brazilian Presidency, Brazil
E-mail: ricardo.queiroz@presidencia.gov.br
ORCID iD: https://orcid.org/0000-0001-7998-9445

Received: 11 September 2025; Revised: 07 November 2025; Accepted: 06 January 2026; Published: 08 April 2026

Abstract: Accurate and immediate incident identification is essential in the cybersecurity area, as it allows the timely detection of threats, along with countermeasures and mitigation, ensuring security for organizations and individuals. This reduces false positives and enables efforts to be concentrated on real risks. This paper presents a framework that integrates ontologies and Large Language Models (LLMs) to identify incidents from events within the context of security threats. Ontology rules are employed to infer probable incidents, resulting in an initial set of incidents for analysis. Furthermore, ontologies provide contextual information, which is combined with event data to formulate queries for LLMs. These interactions with LLMs produce a second set of probable incidents. The outputs from ontology-based inferences and LLM-driven responses are then compared, and the discrepancies are leveraged to refine ontology rules and adjust LLM responses. Experimental results, focusing on context generation and incident detection, demonstrate that the integration of ontologies and LLMs significantly enhances the accuracy of incident identification when compared to using only LLMs.

Index Terms: LLM, Ontology, Security, Event, Incident.

1. Introduction

LLMs have brought significant advancements in recent years, allowing people to interact with these models as if they were oracles with answers to a wide range of questions, regardless of the domain of knowledge. However, many times, the obtained responses do not align with reality, which has been termed as hallucinations of LLMs. These can be caused by the data, the training, or the inference [1].

On the other hand, ontologies emerged a few decades ago as a means of formal knowledge representation in an explicit and structured manner, aiming to facilitate vocabulary standardization, clear representation of concepts and their interrelationships, interoperability, and semantic representation. Thus, the sharing and searching of knowledge would be conducted more easily and simply. Several initiatives have sought to improve and expand the use of ontologies towards the Semantic Web, including linked data and knowledge graphs [2,3]. However, criticisms also arose due to the following reasons: the lack of necessity for most users to perform more complex searches (where the Semantic Web could make a significant difference), difficulty in integrating knowledge, the existing scale restriction for the technologies used so far, and the need for input data to be accurate for inferences to be made correctly, as even small errors can lead to incorrect inferences [4].

Both critics and proponents of the Semantic Web point out that one of the ways to overcome these difficulties is the integration with machine learning, especially deep learning, so that both approaches can gain mutual benefits. On one hand, large deep learning models could generate links with noisy data (providing, for example, missing relationships), and on the other hand, ontologies and related technologies could assist in the development of explainable AI [5-7].

In the context of cybersecurity, a clear and shared understanding of concepts is fundamental. It is also important to have the ability to make tacit inferences based on previous learnings for new types of threats. This is because, in many

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

cases, new threats bear similarities to existing threats. Thus, both deep learning and ontologies can provide relevant elements in identifying and mitigating threats.

It is crucial to evaluate that the improvement and development of ontologies and LLMs should incorporate ethical considerations, such as fairness, privacy protection, and misuse prevention, to ensure responsible deployment in security contexts.

Ontology restrictions and rules can be used to infer incidents from events by defining structured relationships and logical constraints within a knowledge domain. Restrictions and rules in OWL (Web Ontology Language) ontologies can be defined using OWL constraints [8] or SWRL (Semantic Web Rule Language) rules [9]. OWL constraints, such as those in OWL-DL (based on Description Logic), impose restrictions on OWL elements (classes, properties, and instances) to enable reasoning, with a computational complexity of NEXPTIME. Alternatively, SWRL rules, built on Horn clauses, enhance the expressivity of OWL-DL. However, without DL-Safe restrictions, SWRL rules may become undecidable.

In this context, the knowledge repository offered by ontologies forms a solid foundation for creating, integrating, and maintaining rules across diverse security systems, often in combination with various programming languages. For example, in a cybersecurity ontology, a rule might state: If a network scan is followed by unauthorized access, a potential intrusion incident can be inferred. Such rules rely on predefined classes, properties, and axioms to identify patterns or events, providing a domain-specific framework for incident detection. Once an incident is detected, scripts in various languages can be triggered to isolate the attacker or close security vulnerabilities.

Ontologies also enhance queries processed by large language models (LLMs). By adding contextual information, ontologies provide a formal, human-readable representation of incidents. When an LLM processes a query, the ontology can enrich the input with relevant concepts, relationships, or constraints, ensuring responses are grounded in domain-specific facts. For instance, a query about data breach incidents can be supplemented with ontology-derived context, such as types of breaches or affected systems, improving the relevance and accuracy of the LLM's response.

When supporting LLMs queries, ontology constraints can be converted into text for better interpretability. In OWL, constraints can be mapped to descriptive phrases using templates. For example, consider the text generated for an LLM: If an event has a risk score greater than 70, it is classified as an incident with a high impact level. This text might stem from an OWL constraint with the following characteristics: Class: Event, Property: riskScore, Constraint: xsd:integer >= 70, Result: The event is inferred to belong to the class Incident with an impactLevel set to High.

In another scenario, SWRL rules can be presented directly as text for users familiar with semantic logic, without requiring further conversion. For instance, the previous rule would be: Event(?e) ^ riskScore(?e, ?score) ^ greaterThan(?score, 70) → Incident(?e) ^ impactLevel(?e, "High").

Regarding the results of rule processing, ontologies produce precise, rule-based inferences, as they are constrained by the ontology's scope. In contrast, LLM outputs are expressed in natural language, which is more flexible but less precise, potentially leading to ambiguities or errors (e.g., LLM hallucinations, where the model generates incorrect information). Comparing these approaches offers several advantages, including:

- Refining LLMs through iterative feedback to produce more accurate and contextually relevant responses;
- Enhancing the ontology by identifying and incorporating missing elements, such as new attack vectors not previously accounted for.

Current technologies for incident identification rely on predefined rules, threat signatures, or statistical models trained on historical data. These approaches are often limited by their lack of semantic context and inability to generalize to new (zero-day) threats.

On the other hand, ontologies provide a structured and semantic representation of the cybersecurity domain (e.g., attack types, vulnerabilities, indicators of compromise, etc.). They provide a logical and traceable basis for decision-making. LLMs, in turn, process unstructured data (logs, reports, threat feeds) and map this information to ontologies. Thus, LLMs can interpret natural language in real time, extracting relevant information from heterogeneous sources (e.g., forum posts, articles, logs). Meanwhile, the ontology guides the extraction of entities and relationships, ensuring that the data is organized in a way that is useful for incident identification. Combining these tools can enable incident identification with greater context and accuracy. However, creating and maintaining comprehensive and up-to-date ontologies for cybersecurity is hard work and requires domain expertise.

To address the challenges of current cybersecurity incident identification technologies, this paper proposes the integration of ontologies and Large Scale Language Models (LLMs). Specifically, we suggest the creation of an application ontology, based on standards such as MITRE ATT&CK[1], that integrates with an existing domain ontology to reuse concepts. This ontology uses rules to infer probable incidents from security events. In addition, we propose an interactive framework that asks contextual questions about these events, enriched with ontological data, to identify incidents with greater accuracy and reduce hallucinations (incorrect answers from LLMs). Experiments comparing the ontology inferences with the LLM results demonstrated a significant reduction in false positives, highlighting the synergy between the technologies and ways to optimize their joint results.

This paper is organized as follows: Section 2 presents the related works, Section 3 introduces the proposed framework, Section 4 describes the application ontology used for the experiments, Section 5 presents the experiments, and

1 https://attack.mitre.org/

Section 6 discusses the remarks and conclusions.

2. Related Works

In the context of LLMs, Vaswani et al. [10] propose a simpler neural network architecture for sequence transduction models, i.e., architectures that process and transform input sequences. Instead of using complex models based on recurrent and convolutional networks, they propose an attention model called Transformer, which dispenses with traditional recurrence and convolution strategies. The attention mechanism involves the use of matrix operations, especially during attention calculation, assigning different weights to parts of the input based on their relevance to the task at hand. In this process, three main matrices are used: queries, keys, and values. Additionally, it allows processing input tokens in parallel, resulting in faster and more efficient training. The Transformer architecture achieved success in the task of processing natural language, generating several families of LLMs [11,12], among them: GPT, Llama, and Gemini.

Some works seek to increase the accuracy of LLMs using knowledge graphs and ontologies. Shang et al. [13] propose a framework to detect unknown facts in LLMs, using knowledge graphs and ontologies. The authors attempt to identify triples that LLMs consider non-factual within the knowledge graphs, revealing potential errors (hallucinations) of the LLMs. Thus, the LLM is asked to construct a template based on a triple (extracted from the ontology). Based on this template and information about instances of triples present in the ontology, questions are posed to the LLM. If these questions do not align with the facts described in the ontology, they indicate potential errors in the LLM.

Allemang & Sequeda [14] explore how to improve the accuracy of LLMs in interactions using questions and answers. The goal is to utilize LLMs to generate SPARQL queries from natural language texts. The query generated by the LLM is compared with the semantics defined in the ontology, and the extracted errors are used to repair the SPARQL query. This approach has increased the overall accuracy of the LLM's responses.

Other works propose ontologies in the domain of cybersecurity. Farooq & Waseem [15] try to group and compare ontologies aiming to propose good practices and future works in this area. In this work, a total of 24 articles from the years 2010-2020 were selected through a systematic process.

In this sense, Syed et al. [16] propose a relevant ontology[2]. They advocate for a unified ontology, called the Unified Cybersecurity Ontology (UCO). This ontology aims to unify data from different standards and repositories to facilitate information exchange and enhance situational awareness. It is based on several concepts of MITRE ATT&CK and the Structured Threat Information Expression (STIX)[3].

Currently, tools like Elastic[4] are integrating LLMs into their cybersecurity incident detection methods. However, no methods have been identified that combine LLMs with ontologies for this purpose. This work addresses this gap by comparing the use of LLMs alone, including Elastic's method, with a combined approach integrating LLMs and ontologies, to evaluate improvements in accuracy and reduction of errors in incident detection.

3. Proposed Framework

Figure 1 shows how information related to event (for instance, from a Splunk Enterprise Security[5], functioning as a Security Information and Event Management [17]) is processed to identify events with potential to become incidents [18, 19]. The green components represent newly developed elements within the framework, while the others correspond to integrations with external systems.

Initially, a data set is collected. This data refers to cybersecurity events, which are mapped and inserted into an application ontology called Situational Ontology for Cyber Security (SOCS). This ontology is integrated with a domain ontology (for instance, the UCO ontology [16]) through different classes.

The Context Generator is an element that analyzes unprocessed events within the ontology, aiming to enrich their context using existing data from the ontology and additional sources through Retrieval-Augmented Generation (RAG) to formulate questions for an LLM [20]. We ask LLMs the following question: “Is the following event in terms of cyber security an incident?”. Then, we use class instances data, restrictions and rules to insert additional context. The Table 1 summarizes the conversions, where:

- INSTANCE – instance identifier ((ex.: event_10)
- CLASS, ANOTHER CLASS – class name (ex.: event);
- ATTRIBUTE, ANOTHER_ATTRIBUTE – class attribute (ex.: event_risk_score);
- COMPARISON – comparison label (ex.: minInclusive, maxInclusive, equals to, etc);
- VALUE, ANOTHER_VALUE – value to be get/set;
- [ ] - text that can be repeated in a list;
- { } - optional text.

2 https://github.com/Ebiquity/Unified-Cybersecurity-Ontology
3 https://stixproject.github.io/about/#top
4 https://www.elastic.co/security-labs/elastic-advances-llm-security
5 https://www.splunk.com/en_us/products/enterprise-security.html

Table 1. Context conversion

Object	Template
Class Instance	The following data are related to the class CLASS: [The instance INSTANCE has ATTRIBUTE COMPARISON VALUE]
OWL constraints	The following rules, related to the CLASS, indicates that an event is considered an incident: [The attribute ATTRIBUTE for the restriction COMPARISON is VALUE]
	If the CLASS instance has the ATTRIBUTE COMPARISON VALUE, then it is the ANOTHER_CLASS instance {with the ANOTHER_ATTRIBUTE equal to ANOTHER_VALUE}.
SWRL rules (direclty to text)	Ex.: Event(?e) ^ riskScore(?e, ?score) ^ greaterThan(?score, 70) → Incident(?e) ^ impactLevel(?e, "High").

The questions aim to identify events with moderate to high risk of generating incidents. Simultaneously, the incident identification restrictions or rules are used in the ontology to infer which events are incident generators. When the LLMs return the answers, we compare them to the results of the inferences processed by the reasoners based on the ontologies.

The Process Differences component is responsible for comparing the results of the ontology and the LLM. When the ontology's predefined restrictions or rules identify incidents that the LLM does not recognize, the LLM's negative response is classified as a hallucination. In such cases, a new interaction is conducted with additional contextual information to refine the LLM's future responses [21]. We assume that the ontology's rules and restrictions have been validated by human specialists and are correct.

Conversely, if the LLM identifies probable incidents that are not covered by the ontology's rules, a discrepancy list is generated, prioritizing high-risk incidents. The Cyber Intelligence Center (CIC) then evaluates these discrepancies, determining whether the ontology's rules require updates or whether the identified cases are simply hallucinations by the LLM. If an incident is deemed a hallucination, new contextual information is provided to the LLM to enhance its responses in future queries.

By continuously refining both the ontology and the LLM in cases of divergent incident identification, this strategy fosters ongoing learning and improvement for both technologies. A critical consideration is that human intervention remains essential in high-stakes applications to ensure reliability and ethical oversight.

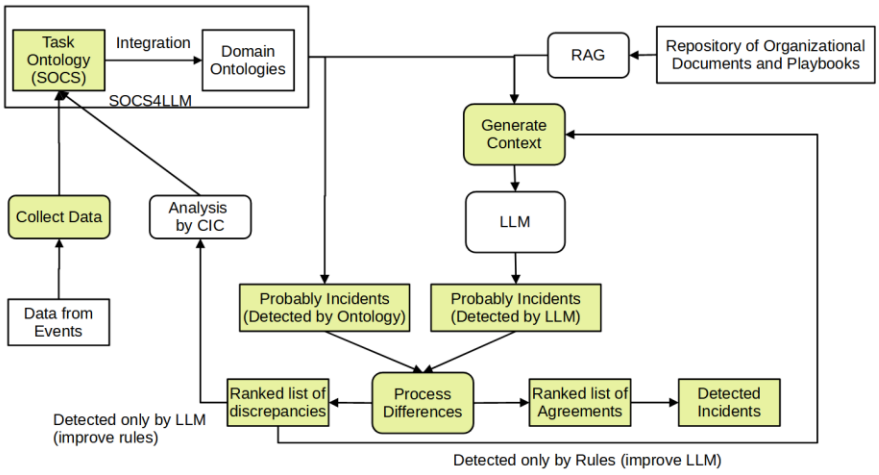


Fig.1. Proposed framework

In terms of architecture scalability, there are some considerations to be made about its two possible bottlenecks: the growth in the number of instances in the ontology and the processing of queries by LLMs in terms of context size and query throughput.

Regarding ontologies, there are well-known solutions that involve dividing the ontology into smaller, interconnected modules, allowing the use of only the modules with the necessary elements (instances). It is also possible to use graph databases to store the elements of an ontology, which improves performance. Regarding rule processing, it is possible to apply inferences only to the parts of the ontology that change, instead of processing everything again (incremental inference).

Considering the processing performed by LLMs in terms of context, it is important to pass only the relevant information to the LLMs, such as that discussed in template generation. It is also possible to use LLMs optimized for long contexts, such as the LLMs discussed in this work. These LLMs store the contexts of previous questions, so that subsequent questions do not need to repeat common elements that have already been passed on previously. It is also possible to maintain a context history, reusing certain excerpts when necessary.

With respect to the query throughput of LLMs, solutions involve managing questions and answers through a cache, which can be incorporated into the Context Generator. This cache allows for filtering repeated questions (events with similar characteristics) and prioritizing high-risk events during peak times. Additionally, queries can be parallelized across different sections of the LLM dedicated to processing requests separately.

4. Application Ontology

The playbook suggestions are also analyzed by the CIC, which will effectively analyze, select, and execute the most appropriate actions to resolve the problem. These data and possible extracted rules are added to the ontology to improve the context in future interactions. In this work, our focus is the 'Generate Context' and 'Detect Incident' components.

The application ontology provides the means to manage events and incidents, offering a common understanding and standardized vocabulary to differentiate events and incidents in the context of a cybersecurity application.

Initially, we take into consideration some competency questions to guide the ontology creation. These questions are:

- What factors allow an event to be considered a generator of an incident?
- What steps should be taken to mitigate or minimize the effects of an incident?
- What resources are affected by an incident?
- What is the origin of an incident and what are its targets?

From these questions, we define concepts, structure relationships, and create a data representation in order to propose the application ontology shown in Figure 2. It is possible to see that this ontology is integrated with the concepts of tactics, techniques, and procedures (mitigation) from MITRE ATT&CK.

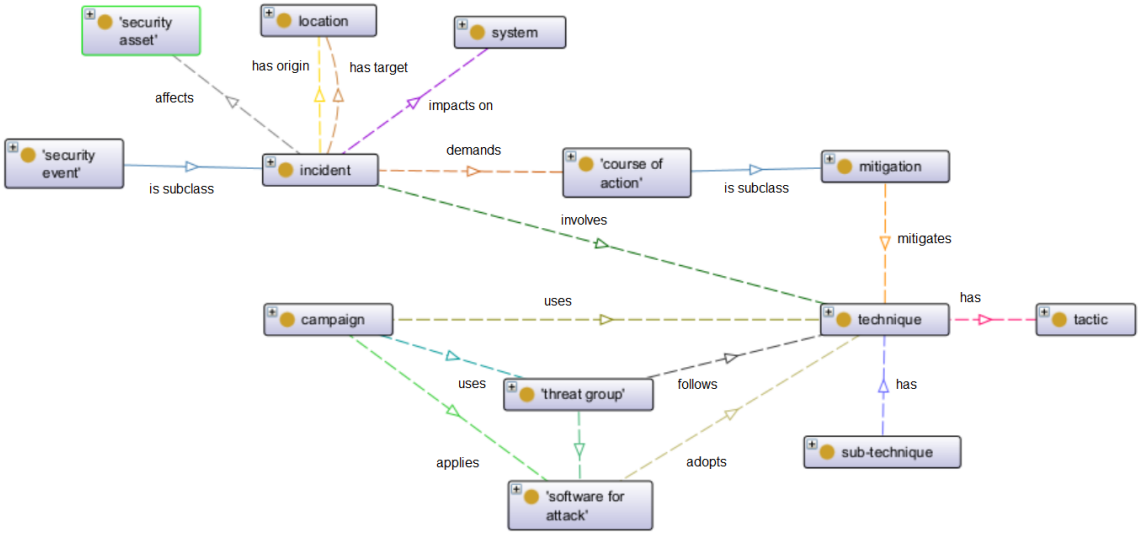


Fig.2. Application ontology

The playbook suggestions are also analyzed by the CIC, which will effectively analyze, select, and execute the most appropriate actions to resolve the problem. These data and possible extracted rules are added to the ontology to improve the context in future interactions. In this work, our focus is the 'Generate Context' and 'Detect Incident' components.

Table 2 presents the metadata defined for SOCS ontology classes, with the socsSecurityEvent class being particularly significant due to its attributes that characterize incident occurrences. These metadata collectively provide contextual information to support LLM queries within the proposed framework.

Table 3 details the attributes of SOCS classes, including their ranges, emphasizing the attributes of the socsSecurityEvent class critical for identifying potential incidents. Additionally, it highlights the ontology's integration with MITRE ATT&CK Tactics, Techniques, and Procedures (TTPs) concepts.

5. Experiments

The experiments were divided into two parts. The first tests the use of OWL constraints and LLMs for incident detection. The second explores more complex rules, incorporating SWRL and LLMs to detect zero-day threats.

5.1. Based on OWL Constraints

The initial experiments focus on evaluating context generation and incident detection using OWL constraints within an ontology framework. Security events, treated as independent occurrences, serve as the basis for this analysis. The objective is to assess the coherence of LLM outputs against simple OWL constraints, which define potential incidents derived from security events. These potential incidents are represented in the ontology as instances of a subclass of socsSecurityEvent, named socsSecurityEvent_restricted.

Additionally, we investigate whether this coherence varies according to context size and rule complexity. To further assess the applicability of our approach, we conduct an exploratory analysis of the incidents generated from security events. As a preliminary step, we determine an appropriate sample size for this evaluation.

Table 2. SOCS context

Class	Label	Description	Comment
socsSecurityEvent	security event	A security event is any identifiable occurrence that may have implications for the security of information, infrastructure, or operations within an organization.	These events are monitored and managed by security teams to prevent, detect, and respond to potential threats, ensuring the continuous protection of the organization's assets.
socsIncident	incident	An incident is defined as an event that, originating from a medium or high risk, has the potential to compromise the confidentiality, integrity, security, or availability of information, systems, or IT networks, along with all their stakeholders.	Such events can significantly impact the proper functioning and reliability of organizational processes, necessitating prompt and effective response measures to mitigate potential harm and restore normal operations.
socsCourseOfAction	course of action	It is a set of steps or strategies that are planned and implemented to address a particular incident. These actions are designed to mitigate risks, remediate impacts, and enhance the security posture of an organization.	By following a structured Course of Action, organizations can manage and respond to cybersecurity incidents, minimizing the impact.
socsSecurityAsset	security asset	It refers to any resource, component, or data that is considered valuable to an organization's information security posture.	These assets need to be protected against threats to ensure the integrity, confidentiality, and availability of information.
socsLocation	location	It refers to the actual geographic location of data centers, servers, network hardware, or other physical infrastructure.	It allows for better management and protection of their physical environments.
socsSystem	system	It represents a computing system or infrastructure component (e.g., server, workstation, or network device) that may be involved in or affected by a cybersecurity incident. It serves as a target or context for incident response actions.	Beyond serving as a target for attacks, it provides the essential context for incident response actions, helping to identify vulnerabilities, track threat vectors, and implement effective mitigation strategies.
socsMitigation	mitigation	It refers to a security measure or strategy designed to prevent or reduce the success of adversarial techniques or sub-techniques. Mitigation represents concepts, configurations, or technologies that organizations can implement to protect their systems and data from specific threats.	For example, a mitigation might involve enforcing strong password policies, using multi-factor authentication, or implementing network segmentation.
socsTactic	tactic	It represents an adversary's strategic objective at a given moment.	For example, a tactic might be “Access Credentials” or “Persistence.” These answer the “why” of an adversary’s action.
socsTechnique	technique	It describes how an adversary achieves a tactical objective.	For example, the tactic "Access Credentials", a technique might be "Credential Dumping". They detail the method used to achieve the objective.
socsSubTechnique	sub-technique	It is a more specific method or technique variation, providing finer granularity in describing adversary behaviors.	For example, within the "Credential Dumping" technique, a sub-technique might be "Access Local Security Authority (LSA) Secrets". These provide a more granular level of detail.
socsCampaign	campaign	It refers to a grouping of intrusion activities conducted over a specific period of time, often with common targets and objectives. Campaigns are typically associated with threat actors or groups and may involve the use of specific techniques, software, and tactics to achieve their goals.	For example, the 2015 Ukraine Electric Power Attack was a campaign by the Sandworm Team targeting the Ukrainian power grid.
socsThreatGroup	threat group	Threat group refers to a cluster of adversary activities tracked under a common name by the cybersecurity community. These groups are often associated with specific threat actors or organizations and are characterized by their use of particular techniques, software, and campaigns. Analysts use groups to understand patterns of adversarial behavior and attribute attacks to known entities.	For example, the FIN7, a financially motivated group that targets businesses for monetary gain.
socsSoftwareForAttack	software for attack	It refers to tools, malware, or legitimate applications that adversaries use to execute their tactics and techniques. This includes a wide range of programs, such as remote access tools (RATs), command-and-control frameworks, credential dumpers, and more. These tools can be custom-built by attackers or repurposed from legitimate software.	For example, the tool Cobalt Strike, a legitimate penetration testing tool often misused by attackers for post-exploitation activities.

Table 3. Class attributes of SOCS

Class	Property	Range
socsSecurityEvent	event_id	Integer
	_time	dateTime
	description, event_type, risk_score, action, status, protocol, geo_location, user_agent, source_address, source_hostname, source_port, dest_address, dest_hostname, dest_port, system_identifier_type, system_identifier_address, system_identifier_hostname, asset_sensitivity	String
socsIncident	incident_id	Integer
	category, impact_level	String
socsCourseOfAction	course_of_action_id	Integer
	goal, description	String
socsSecurityAsset	asset_id	Integer
	address, asset_type, hostname, description, sensitivity	String
socsLocation	location_id	Integer
	country, region, city,	String
socsSystem	system_id	Integer
	system_name, system_version	String
socsMitigation	mitigation_id	Integer
	mitigation_name, mitigation_description	String
socsTactic	tactic_id, tactic_name, tactic_description	String
socsTechnique	technique_id, technique_name, technique_description	String
socsSubTechnique	subTechnique_id, subTechnique_name, subTechnique_description	String
socsCampaign	campaign_id, campaign_name, campaign_description	String
socsThreatGroup	threatGroup_id, threatGroup_name, threatGroup_description	String
socsSoftwareForAttack	softwareForAttack_id, softwareForAttack_name, softwareForAttack_description	String

The following formula is used to estimate the sample size of the events [22,23]:

$$sampleSize = Z^2 * p * (1 - p)^2 \quad (1)$$

Where:

- Z - critical value of the normal distribution for the desired confidence level. We use 1.44 (90%);
- p - the estimated proportion. We apply 20% (2 out of every 10 events are probable incidents);
- e - the margin of error. We accepted 10%.

The sample size formula resulted in 44 samples. Thus, we simulated, using Microsoft Copilot free version[6], 50 security events [24] that could be obtained by Splunk ES. Then, we converted them into JSON format to populate the attributes of the *socsSecurityEvent* class. After that, we tested these samples in three different LLMs in our experiments: Awan, Gemini, and Grok.

The Awan LLM[7] is a platform that provides access to LLM without a priori limiting the number of tokens used in the context of input sequences in the free access. However, it limits the number of interactions. In the experiments, we used Meta-Llama-3-8B model.

The Gemini model[8], developed by Google DeepMind, released in 2023, is a LLM from Google that allows a certain number of interactions freely, but limits the number of tokens for each interaction. We used gemini-1.5-flash model.

The Grok model[9], developed by Xai, released in 2024, is still a beta version and, in this version, a certain number of interactions is free. We used grok-beta model.

In our experiments, we consider two different sets of rules registered in the application ontology. Table 4 shows these sets (conditions on different lines indicate an "OR" connection). All rules restrict the risk score to a maximum value of 100. These rules in context use terms such as *minInclusive*, *maxInclusive*, and *hasValue*, which are terms commonly used in ontologies, particularly in the context of the Web Ontology Language (OWL). For example, for R1, we have:

- The attribute *event_risk_score* for the restriction *minInclusive* is 50.
- The attribute *event_risk_score* for the restriction *maxInclusive* is 100.

6 https://copilot.microsoft.com/
7 https://www.awanllm.com/
8 https://deepmind.google/
9 https://x.ai/grok

Table 4. Rule used in SOCS in the experiments for identification of probable incidents

Rules Sets	Attribute	Condition
Set 1 (R1)	Risk Score	risk score >= 50
Set 2 (R2)	Risk Score	risk score >= 70
	Risk Score and Sensitivity	risk score >= 50 and Sensitivity >= medium
	Risk Score and Sensitivity	risk score >= 20 and Sensitivity >= high
	Risk Score and User Role	risk score >= 40 and User Role is administrator
	Risk Score and User Role	risk score >= 30 and User Role is finance
	Risk Score and Type of Event	data exfiltration with risk score >= 30
	Risk Score and Type of Event	unauthorized access with risk score >= 35
	Risk Score and Source IP	risk score >= 55 from an unknown IP address

In the experiments, we also consider three different contexts that indicate the data included in the questions:

- NSC: Besides the rules set, it includes the attributes of the *socsSecurityEvent* class, except for risk score, action, and status. The idea is to force the LLMs to produce a risk score by itself, without using the support of risk score, action taken, or event status.
- SNC: Besides the rules set, int includes all attributes of the *socsSecurityEvent* class.
- SC: Besides the rules set, all contextual data are provided, including all attributes of the *socsSecurityEvent* class, and the metadata of its neighbor classes (label, description, and comment).

Then, the experiments consider the combination of three LLM models, two rule sets, and three additional contexts for the questions (RSC, RSNC, and RNSC). In order to evaluate the results, we compared the LLM responses to the ontology inferences and assembled a set of confusion matrices. Additionally, we applied the chi-square test to each confusion matrix, considering:

- Null Hypothesis (H0): Assumes that there is no association between the classes inferred by the ontology and classes predicted by the LLM. Class distributions are independent.
- Alternative Hypothesis (H1): Assumes that there is an association between the classes inferred by the ontology and classes predicted by the LLM. Class distributions are not independent.

Thus, if the p-value obtained from the chi-square test [25] is less than 0.05, we reject the null hypothesis and conclude that there is a significant association between the variables. Table 5 shows these results. We observe that more context (SC or SNC) generally leads to the rejection of the null hypothesis, except for the XAI LLM. In this case, there is a better opportunity for improving the results.

Table 5. Confusion matrices, p-value and rejection of null hypothesis

	TP	FN	FP	TP	p-value	reject
AW_R1NSC	1	37	0	12	1.00E+00	no
AW_R1SC	38	0	1	11	3.32E-10	yes
AW_R1SNC	38	0	1	11	3.32E-10	yes
GE_R1NSC	37	1	8	4	1.11E-02	yes
GE_R1SC	38	0	3	9	4.64E-08	yes
GE_R1SNC	38	0	4	8	4.65E-07	yes
XAI_R1NSC	1	37	0	12	1.00E+00	no
XAI_R1SC	38	0	1	11	3.32E-10	yes
XAI_R1SNC	38	0	1	11	3.32E-10	yes
AW_R2NSC	26	16	2	6	1.24E-01	no
AW_R2SC	39	3	4	4	8.15E-03	yes
AW_R2SNC	39	3	4	4	8.15E-03	yes
GE_R2NSC	32	10	3	5	7.71E-02	no
GE_R2SC	39	3	4	4	8.15E-03	yes
GE_R2SNC	41	1	3	5	2.64E-05	yes
XAI_R2NSC	6	36	0	8	5.85E-01	no
XAI_R2SC	25	17	1	7	4.00E-02	yes
XAI_R2SNC	27	15	2	6	9.44E-02	no

To better analyze the experiments, we also use the following metrics [26-28]:

- TPR (True Positive Rate) / Recall: the proportion of actual positives that are correctly identified;
- TNR (True Negative Rate) / Specificity: the proportion of actual negatives that are correctly identified;
- FPR (False Positive Rate): the proportion of actual negatives that are incorrectly identified as positives;
- FNR (False Negative Rate): the proportion of actual positives that are incorrectly identified as negatives;
- PPV (Positive Predictive Value) / Precision: the proportion of positive results that are true positives;
- NPV (Negative Predictive Value): the proportion of negative results that are true negatives;
- ACC (Accuracy): the overall proportion of correctly identified instances;
- F1 Score: the harmonic mean of precision and recall.

Table 6 shows the results using the proposed framework. The prefixes identify the LLM, being: AW for Awan, GE for Gemini and XAI for Grok.

Table 6. Metrics of performance

	TPR	TNR	FPR	FNR	PPV	NPV	ACC	F1
AW_R1NSC	0.026	1	0	0.974	1	0.245	0.26	0.051
AW_R1SC	1	0.917	0.083	0	0.974	1	0.98	0.987
AW_R1SNC	1	0.917	0.083	0	0.974	1	0.98	0.987
GE_R1NSC	0.974	0.333	0.667	0.026	0.822	0.8	0.82	0.892
GE_R1SC	1	0.750	0.250	0	0.927	1	0.94	0.962
GE_R1SNC	1	0.667	0.333	0	0.905	1	0.92	0.95
XAI_R1NSC	0.026	1	0	0.974	1	0.245	0.26	0.051
XAI_R1SC	1	0.917	0.083	0	0.974	1	0.98	0.987
XAI_R1SNC	1	0.917	0.083	0	0.974	1	0.98	0.987
AW_R2NSC	0.62	0.75	0.25	0.38	0.93	0.27	0.64	0.74
AW_R2SC	0.93	0.5	0.5	0.07	0.91	0.57	0.86	0.92
AW_R2SNC	0.93	0.5	0.5	0.07	0.91	0.57	0.86	0.92
GE_R2NSC	0.76	0.63	0.38	0.24	0.91	0.33	0.74	0.83
GE_R2SC	0.93	0.5	0.5	0.07	0.91	0.57	0.86	0.92
GE_R2SNC	0.98	0.63	0.38	0.02	0.93	0.83	0.92	0.95
XAI_R2NSC	0.14	1	0	0.86	1	0.18	0.28	0.25
XAI_R2SC	0.6	0.88	0.13	0.41	0.96	0.29	0.64	0.74
XAI_R2SNC	0.64	0.75	0.25	0.36	0.93	0.29	0.66	0.76

Table 7. Metrics of correlation, similarity and distance

	Pearson	Cosine	Euclidean
AW_R1NSC	-0.21	-0.33	468.11
AW_R1SC	0.92	0.98	90.83
AW_R1SNC	0.93	0.99	72.8
GE_R1NSC	0.46	0.87	230.22
GE_R1SC	1	1	10
GE_R1SNC	1	1	10
XAI_R1NSC	0.07	0.43	428.98
XAI_R1SC	1	1	10
XAI_R1SNC	1	1	10
AW_R2NSC	0.06	0.57	386.72
AW_R2SC	0.93	0.99	67.27
AW_R2SNC	0.73	0.95	147.39
GE_R2NSC	0.36	0.86	242.74
GE_R2SC	1	1	10
GE_R2SNC	1	1	10
XAI_R2NSC	0.01	0.79	322.65
XAI_R2SC	0.96	1	42.43
XAI_R2SNC	0.93	0.99	55.9

The results show that a simpler set of rules (Set 1, indicated by R1) in most cases enables better results compared to a more complex set of rules (Set 2, indicated by R2). Additionally, having more context (SC or SNC) generally leads to superior outcomes. This indicates that restriction rules present a more complex challenge to overcome compared to a simpler, informative context. On the other hand, the lack of context (risk score, action, and status in NSC) makes the task of classifying the events into incidents even more arduous. All these results consider the inferred rules by the ontology as a reference. These results reflect how the LLMs' results match the ontology inference rules. Thus, assuming the rules are correct, these differences demand new questions, informing the LLMs that some incidents were not detected because the rules were not correctly interpreted.

However, the FPR may indicate opportunities to improve the rules in the ontology, since these elements may actually be true positives that are not correctly identified by the ontology rules, as they may be incomplete. Therefore, a more thorough analysis is required in this case, as predicted in the framework. A Cyber Intelligence Center (CIC) should evaluate these cases, looking for incidents not predicted by the rules, if applicable. Overall, we can consider that LLM Gemini achieved slightly superior results.

To complement our analysis, we include three additional metrics: Pearson correlation, Cosine similarity, and Euclidean distance [29,30]. Table 7 shows these metrics.

Generally, these metrics corroborate the previous analysis, showing that, typically, more context with simple rules works better for obtaining a match with the ontology inferences. However, it is possible to notice that, in some cases, SNC obtains competitive results when compared to the SC context.

5.2. Based on SWRL Rules

The second set of experiments evaluates incident detection during the analysis phase, leveraging SWRL rules within an ontology framework. This section aims to examine case studies involving simulated zero-day attack detections. SWRL rules, aligned with MITRE ATT&CK Tactics, Techniques, and Procedures (TTPs), are designed to identify these attacks. The proposed detection approach relies on SWRL rules to enable behavioral analysis, and anomaly detection, utilizing the ontology's classes and properties. Zero-day detection presents inherent challenges due to the identification of previously unknown vulnerabilities or attack patterns, necessitating a focus on these advanced analytical techniques.

In this experiment, the rules will target TTPs commonly associated with zero-day exploitation, such as (MITRE code in parenthesis):

- Command and Scripting Interpreter (T1059, e.g., unusual script execution);
- Exploit Public-Facing Application (T1190);
- Phishing (T1566, as an initial vector for zero-day delivery).

The ontology classes (e.g., socsSecurityEvent, socsIncident, socsCourseOfAction, socsSecurityAsset, socsLocation) and their properties will be used to define conditions and actions in SWRL rules. Zero-day exploits often bypass traditional signature-based detection, so the rules will:

- Monitor Anomalies: use ontology properties like `risk_score`, `source_ip_address`, `dest_port`, and `user_agent` to identify unusual patterns;
- Trigger Actions: Use `socsCourseOfAction` to define responses (e.g., isolate assets, block Ips).

For the following experiments, we created four additional classes that store allowed values, which are:

- 'untrustedAgent' with the property 'untrusted_agent';
- 'trustedIp' with the property 'trusted_ip';
- 'untrustedLocation' with the property 'untrusted_location'; and
- 'trustedServer' with the property 'trusted_server'.

A. Detecting Zero-Day Exploitation via Unusual Command Execution

In this event, attackers may exploit a zero-day vulnerability to execute malicious scripts or commands via interpreters like PowerShell, Python, or Bash. Unusual command execution patterns (e.g., untrusted 'user_agent', unexpected 'source_address', or high 'risk_score') can indicate a zero-day exploit.

SWRL Rule:

socsSecurityEvent(?e) ^ event_id(?e, ?id) ^ event_type(?e,"command_execution") ^ event_risk_score(?e, ?rs) ^ swrlb:greaterThan(?rs,70) ^ event_user_agent(?e, ?ua) ^ untrustedAgent(?ut) ^ untrusted_agent(?ut, ?ut_name) ^ swrlb:equal(?ua,?ut_name) ^ event_source_address(?e, ?sip) ^ trustedIp(?tip) ^ trusted_ip(?tip, ?tip_value) ^ swrlb:notEqual(?sip, ?tip_value) ^ socsSecurityAsset(?a) ^ event_dest_address(?e, ?saddr) ^ asset_address(?a, ?saddr) ^ asset_sensitivity(?a, "high") -> socsIncident(?e) ^ event_description(?e, "Potential Zero-Day Command Execution")

This rule checks if the event is a command execution (‘event_type = "command_execution"’) and if the ‘risk_score’ is high (>70), indicating suspicious activity. It checks also if the ‘user_agent’ is in an untrusted list, suggesting an unusual client and if the ‘source_address’ is trusted, and if the affected asset (‘socsSecurityAsset’) has high sensitivity.

Based on this check, the rule infers that it is an ‘socsIncident’ with description "Potential Zero-Day Command Execution". This rule detects anomalies in command execution contexts, which are common in zero-day attacks.

B. Detecting Zero-Day Exploitation of Public-Facing Applications

In this event, zero-day vulnerabilities in web applications or APIs are often exploited to gain initial access. Indicators include unusual ‘dest_port’, high ‘risk_score’, and untrusted ‘geo_location’ for requests to public-facing assets.

SWRL Rule:

_swrlRule ruleB socsSecurityEvent(?e) ^ event_id(?e, ?id) ^ event_type(?e, "web_request") ^ event_risk_score(?e, ?rs) ^ swrlb:greaterThan(?rs, 75) ^ event_dest_port(?e, ?dp) ^ swrlb:notEqual(?dp, 80) ^ swrlb:notEqual(?dp, 443) ^ socsLocation(?loc) ^ event_geo_location(?e, ?country) ^ location_country(?loc, ?country) ^ untrustedLocation(?ul) ^ untrusted_location(?ul, ?ul_name) ^ swrlb:equal(?country, ?ul_name) ^ socsSecurityAsset(?a) ^ event_dest_address(?e, ?saddr) ^ asset_address(?a, ?saddr) ^ asset_type(?a, "public_facing") -> socsIncident(?e) ^ event_description(?e, "Potential Zero-Day Web Exploit")

This rule verifies if the event is a web request (`event_type = "web_request"`) and if ‘risk_score’ is high (>75). It also verifies if the ‘dest_port’ is non-standard (not 80 or 443), suggesting a targeted exploit, and if the ‘geo_location’ originates from an untrusted country, and if the asset is public-facing (‘asset_type = "public_facing"’).

This rule can infer that it is ‘socsIncident’ with description "Potential Zero-Day Web Exploit". In this scenario, we consider that zero-day exploits often target unpatched vulnerabilities in public-facing applications. This rule focuses on anomalous web requests that deviate from normal traffic patterns.

C. Detecting Zero-Day Phishing for Initial Access

For this event, we consider that phishing emails or links may deliver zero-day exploits to execute malicious payloads. Indicators include checking ‘source_address’ and ‘user_agent’, and a high ‘risk_score’ for email-related security events.

SWRL Rule:

_swrlRule ruleC socsSecurityEvent(?e) ^ event_id(?e, ?id) ^ event_type(?e, "email_access") ^ event_risk_score(?e, ?rs) ^ swrlb:greaterThan(?rs, 80) ^ event_source_address(?e, ?sip) ^ trustedIp(?tip) ^ trusted_ip(?tip, ?tip_value) ^ swrlb:notEqual(?sip, ?tip_value) ^ event_user_agent(?e, ?ua) ^ trustedServer(?ts) ^ trusted_server(?ts, ?ts_name) ^ swrlb:notEqual(?ua, ?ts_name) ^ socsSecurityAsset(?a) ^ event_dest_hostname(?e, ?shost) ^ asset_hostname(?a, ?shost) ^ asset_type(?a, "user_endpoint") -> socsIncident(?e) ^ event_description(?e, "Potential Zero-Day Phishing")

This rule checks if the event is an email access (‘event_type = "email_access"’) and if the ‘risk_score’ is very high (>80). It also checks if the ‘source_address’ is a trusted email server and if the ‘user_agent’ is a trusted email client and if the asset is a user endpoint (‘asset_type = "user_endpoint"’).

This rule may infer that it is a ‘socsIncident’ with description "Potential Zero-Day Phishing". In this rule, we consider thar phishing is a common vector for delivering zero-day exploits. This rule detects suspicious email interactions that may indicate a novel attack.

D. Applicaton of SWRL based on ontologies and LLM

In the conducted simulations, we utilized a modified set of security event instances [24] compared to those described previously, while maintaining the same number of elements. This set included simulations of zero-day attacks and routine events representative of operational scenarios. The SWRL rules defined in the ontology were employed to filter and classify these attacks based on their specific characteristics. For detection, we also applied the Gemini model, which demonstrated satisfactory performance in the experiments described earlier. Finally, we compared the obtained results, analyzing the effectiveness of the ontological rules and the Gemini model in identifying these attacks. The attacks are represented in the ontology as instances of the class socsIncident.

Table 8 presents a comparison of two LLM configurations against a baseline established by SWRL rules and ontology alone for identifying security incidents. The first configuration, labeled GE_NSWRL, employs the Gemini LLM model supported by ontology concepts but excludes SWRL rules. The second configuration, labeled GE_SWRL, integrates the Gemini LLM model with both ontology concepts and SWRL rules.

Table 8. Metrics of performance comparing SWRL results and Gemini LLM

	TPR	TNR	FPR	FNR	PPV	NPV	ACC	F1
GE_NSWRL	0.5	1.0	0.0	0.5	1.0	0.864	0.88	0.667
GE_SWRL	0.917	0.974	0.026	0.083	0.917	0.974	0.96	0.917

The evaluation revealed that the Gemini LLM model, when provided only contextual ontology data without SWRL rules, failed to detect approximately half of the zero-day threats, posing a significant limitation. In contrast, evaluations incorporating the SWRL rules enabled Gemini to detect 92% of these attacks, based on correct classification of socs Secrity Event instances, with an overall accuracy of 96%, achieved by accurately interpreting the ontology's rules.

To ensure the reproducibility of the experiments, all security event data, ontologies, and LLM responses used in the study have been made publicly available at the repository referenced in [24].

6. Conclusions

This paper proposes a framework that integrates ontologies in the process of identifying incidents from events simulated as detections by Splunk ES. Ontologies are used both in processing the context of questions directed to the LLMs and in comparing the results obtained through inference rules that indicate which incidents should have been returned. Thus, a continuous improvement process for the LLMs can be implemented through additional questions that highlight which incidents were not identified. Simultaneously, it is proposed that incidents detected by the LLMs but not identified by the rules be analyzed to determine if additional rules should be created to improve the identification of incidents through the ontology's inferences. In this analysis, different contexts were examined, as well as different sets of rules. Therefore, this work presents the following contributions:

- Development of a semantic enrichment mechanism for events;
- Creation of a continuous learning system for ontologies and LLMs, through the proposed framework;
- Proposal of a hybrid explainability model by integrating the formal logic of ontologies with the natural language processing capabilities of LLMs;
- Integration with existing cybersecurity standards, such as MITRE ATT&CK;
- Incident triage automation;

For future work, we intend to study how different playbooks proposed by the LLMs can be integrated and evaluated through existing rules in the ontology, aiming to enhance the automation process of incident handling. Besides, we have considered the events as independent entities. In some cases, they can act as a collaborative or coordinated attack. Then, the rules should consider a set of events as a probable incident, and the context of the LLMs also needs to be extended to incorporate multiple events. By leveraging continuous learning through the interplay of ontology-based inferences and feedback from LLMs, we aim to propose a set of actionable rules derived from this process in future research.

Author Contributions Statement

Wallace A. Pinheiro – Conceptualization, Methodology, Data Curation, Software Implementation, Statistical Analysis, and Writing: Proposed research ideas, constructed the overall framework, handled data acquisition and preprocessing, implemented the research method, ensured the statistical robustness of the evaluation, and drafted the initial manuscript. Ricardo Queiroz de Araujo Fernandes – Formal Analysis, Validation, and Writing: Performed the analysis of the propositions, validated the results using standard metrics, and reviewed and edited the manuscript.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration

This research was supported by FINEP, Brazil (Projeto Ciber Semântica, Convênio para Pesquisa, Desenvolvimento e Inovação nº 0.1.22.0335.00, Ref.: 0172/22, November 29, 2022).

Data Availability Statement

This study analyzed publicly available datasets. The results obtained and datasets can be found at: https://zenodo.org/records/14557808?preview=1&token=eyJhbGciOiJIUzUxMiJ9.eyJpZCI6IjQ5ZTkxYmY0LTFmZmMtNDdhZi1hZTY0LTk1Yzg5MmIzMWFlOCIsImRhdGEiOnt9LCJyYW5kb20iOiI1NzBjNjMxZmU4ZDM3ZTQ3ZmI4ODUzNDdhMGIwYzYwNyJ9.8ufEtO0rI6UK_cLNsIjzBm6SpR6OQSF3l5TOFwOPEs8zcdRXWwmIW1zU-Kz87mkgL5VS7QzCgCbPMfykpCVarQ, accessed on March 21, 2026.

Ethical Declarations

This study does not involve human participants or animals. Therefore, ethical approval was not required.

Acknowledgments

The authors gratefully acknowledge the financial support from FINEP, Brazil (Projeto Ciber Semântica, Convênio para Pesquisa, Desenvolvimento e Inovação nº 0.1.22.0335.00, Ref.: 0172/22, of 29 November 2022).

Declaration of Generative AI in Scholarly Writing

The authors used Gemini (Google) for spelling checks, paraphrasing, and rewording. After using these tools, the authors carefully reviewed and edited the content and take full responsibility for the publication's content.

Abbreviations

The following abbreviations are used in this manuscript:

AI – Artificial Intelligence
DL – Description Logic
LLM – Large Language Model
OWL-DL – Web Ontology Language
SWRL – Semantic Web Rule Language
NEXPTIME - Nondeterministic Exponential Time
RAG – Retrieval-Augmented Generation
JSON – JavaScript Object Notation
UCO – Unified Cybersecurity Ontology
SOCS – Situational Ontology for Cyber Security
STIX – Structured Threat Information Expression
CSIRT – Computer Security Incident Response Team
CIC – Cyber Intelligence Center
TPR – True Positive Rate
TNR – True Negative Rate
FPR – False Positive Rate
FNR – False Negative Rate
PPV – Positive Predictive Value
NPV – Negative Predictive Value
ACC – Accuracy
H0 – Null Hypothesis
H1 – Alternative Hypothesis

References

[1] L. Huang et al., "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," ACM Transactions on Information Systems, 2023.
[2] S. Grimm, A. Abecker, J. Völker, and R. Studer, "Ontologies and the Semantic Web," in Handbook of Semantic Web Technologies, J. Domingue, D. Fensel, and J. A. Hendler, Eds., Berlin: Springer, 2011, doi: 10.1007/978-3-540-92913-0_13.
[3] P. Hitzler, "A Review of the Semantic Web Field," Communications of the ACM, vol. 64, no. 2, pp. 76–83, 2021.
[4] A. Hogan, "The Semantic Web: Two Decades On," Semantic Web, vol. 11, no. 1, pp. 169–185, 2020.
[5] F. Lecue, "On the Role of Knowledge Graphs in Explainable AI," Semantic Web, vol. 11, no. 1, pp. 41–51, 2020.
[6] C. d'Amato, "Machine Learning for the Semantic Web: Lessons Learnt and Next Research Directions," Semantic Web, vol. 11, no. 1, pp. 195–203, 2020.
[7] P. Hitzler, F. Bianchi, M. Ebrahimi, and M. Sarker, "Neural-Symbolic Integration and the Semantic Web," Semantic Web, vol. 11, no. 1, pp. 3–11, 2020.
[8] P. F. Patel-Schneider, P. Hayes, I. Horrocks, F. van Harmelen. OWL Web Ontology Language: Semantics and Abstract Syntax. W3C Candidate Recommendation. 2002. Disponível em http://www.w3.org/TR/owl-semantics.
[9] I. Horrocks, et al.. SWRL: A Semantic Web Rule Language. *W3C Submission*. 2004.
[10] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.
[12] H. Naveed et al., "A Comprehensive Overview of Large Language Models," arXiv preprint arXiv:2307.06435, 2023, doi: 10.48550/arXiv.2307.06435.
[13] Z. Shang et al., "OntoFact: Unveiling Fantastic Fact-Skeleton of LLMs via Ontology-Driven Reinforcement Learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 17, pp. 29859–29869, 2024, doi: 10.1609/aaai.v38i17.29859.
[14] D. Allemang and J. Sequeda, "Increasing the LLM Accuracy for Question Answering: Ontologies to the Rescue!", arXiv preprint arXiv:2405.11706, 2024, doi: 10.48550/arXiv.2405.11706.

[15] M. S. Farooq and M. T. Waseem, "Developing and Building Ontologies in Cyber Security," arXiv preprint arXiv:2306.00377, 2023, doi: 10.48550/arXiv.2306.00377.
[16] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, "UCO: Unified Cybersecurity Ontology," in AAAI Workshop on Artificial Intelligence for Cyber Security, 2016. [Online]. Available: http://ebiq.org/p/722.
[17] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures," Sensors, vol. 21, no. 14, p. 4759, 2021, doi: 10.3390/s21144759.
[18] A. C. Simonneau, "Security Events, Alerts, Incidents: What Are the Differences?" Oversoc.com, 2024. [Online]. Available: https://www.oversoc.com/en/ressources/article/evenements-de-securite-alertes-incidents-quelles-differences.
[19] C. Pham, "From Events to Incidents," SANS GSEC Practical Assignment, 2001. [Online]. Available: https://www.sans.org/white-papers/646/.
[20] Y. Zhang et al., "A Comprehensive Survey of Scientific Large Language Models and Their Applications in Scientific Discovery," in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 2024, pp. 8783–8817, doi: 10.18653/v1/2024.emnlp-main.670.
[21] T. B. Brown et al., "Language Models Are Few-Shot Learners," arXiv preprint arXiv:2005.14165, accessed Dec. 13, 2024, doi: 10.48550/arXiv.2005.14165.
[22] T. S. Nanjundeswaraswamy and S. Divakar, "Determination of Sample Size and Sampling Methods in Applied Research," Proceedings on Engineering Sciences, vol. 3, no. 1, pp. 25–32, 2021.
[23] W. G. Cochran, "Sampling Techniques," 3rd ed., John Wiley & Sons, 1977.
[24] W. Pinheiro, "Experimental Data [Data Set]," Zenodo, 2025. [Online]. Available: https://zenodo.org/records/14557808?preview=1&token=eyJhbGciOiJIUzUxMiJ9.eyJpZCI6IjQ5ZTkxYmY0LTFmZmMtNDdhZi1hZTY0LTk1Yzg5MmIzMWFlOCIsImRhdGEiOnt9LCJyYW5kb20iOiI1NzBjNjMxZmU4ZDM3ZTQ3ZmI4ODUzNDdhMGIwYzYwNyJ9.8ufEtO0rI6UK_cLNsIjzBm6SpR6OQSF3l5TOFwOPEs8zcdRXWwmIW1zU-Kz87mkgL5VS7QzCgCbPMfykpCVarQ
[25] S. D. Bolboacã, L. Jäntschi, A. F. Sestraş, R. E. Sestraş, and D. C. Pamfil, "Pearson-Fisher Chi-Square Statistic Revisited," Information, vol. 2, no. 3, pp. 528–545, 2011, doi: 10.3390/info2030528.
[26] O. Rainio, J. Teuho, and R. Klén, "Evaluation Metrics and Statistical Tests for Machine Learning," Scientific Reports, vol. 14, no. 1, p. 6086, 2024, doi: 10.1038/s41598-024-56706-x.
[27] P. M. Marcus, "Performance Measures," in Assessment of Cancer Screening, Cham: Springer, 2022, pp. 23–28, doi: 10.1007/978-3-030-94577-0_3.
[28] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009.
[29] I. Batyrshin, "Towards a General Theory of Similarity and Association Measures: Similarity, Dissimilarity and Correlation Functions," Journal of Intelligent & Fuzzy Systems, vol. 36, no. 4, pp. 2977–3004, 2019.
[30] A. Huang, "Similarity Measures for Text Document Clustering," in Proceedings of the 6th New Zealand Computer Science Research Student Conference (NZCSRSC), Christchurch, New Zealand, 2008, vol. 4, pp. 9–56.

Authors' Profiles



Wallace A. Pinheiro graduated in Electronic Engineering in 1998 from the Federal University of Rio de Janeiro (UFRJ). He earned a master's degree in Computer Systems in 2004 from the Military Engineering Institute (IME) and a doctorate in Systems and Computer Engineering in 2010 from UFRJ. From 2010 to 2014, he was a professor at IME, working on topics such as command and control, databases, data quality, and information retrieval. Between 2014 and 2017, he worked at the Systems Development Center (CDS), an organization responsible for developing various systems used by the Brazilian Army. In 2018, he completed a postdoctoral fellowship at UFRJ in Systems and Computer Engineering. From 2019 to 2022, he returned to CDS, where he applied his postdoctoral research. Currently, he is an associate researcher at CDS and an academic collaborator at IME. Currently, he is interested in data mining and artificial intelligence.



Ricardo Queiroz de Araujo Fernandes holds a degree in Systems and Computer Engineering from the Military Institute of Engineering (2002), a master's degree in Pure Mathematics from the Federal University of Rio Grande do Sul (2008), a master's degree in Systems and Computing from the Military Institute of Engineering (2009) and a PhD in Computer Theory from the Pontifical Catholic University of Rio de Janeiro (2012). He completed the Post-Doctorate period in Command and Control at George Mason University (2016). His professional goals involve a diverse range of interests, with a particular focus on data mining, artificial intelligence, and command and control methodologies.

How to cite this paper: Wallace A. Pinheiro, Ricardo Q. A. Fernandes, "Framework for Incident Identification Based on LLMs and Cybersecurity Ontologies", International Journal of Intelligent Systems and Applications(IJISA), Vol.18, No.2, pp.125-138, 2026. DOI:10.5815/ijisa.2026.02.09