Using LLMs as AI Agents to Identify False Positive Alerts in Security Operation Center

Pasha Rafiey

p-rafiee@agri-bank.com

Agricultural Bank

Amin Namadchian

Agricultural Bank

Research Article

Keywords: LLM, llama, AI Agent, SOC, SIEM, False Positive

Posted Date: November 28th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5420741/v1

Additional Declarations: No competing interests reported.

Using LLMs as AI Agents to Identify False Positive Alerts in Security Operation Center

Pasha Rafiey[1*] and Amin Namadchian[2†]

[1*]Cybersecurity Dep, Agricultural Bank, Jalal-Al-ahmad, Tehran, Iran.
[2]Cybersecurity Dep, Agricultural Bank, Jalal-Al-ahmad, Tehran, Iran.

*Corresponding author(s). E-mail(s): p-rafiee@agri-bank.com;
Contributing authors: a-namadchian@agri-bank.com;
†These authors contributed equally to this work.

Abstract

This paper addresses the challenges and solutions related to identifying false positive (FP) alerts in Security Information and Event Management (SIEM) systems, which often overwhelm security operators. To tackle this issue, we propose a novel approach that employs a Large Language Model (LLM), specifically Llama, as an AI agent through a contextual-based approach to identify FPs in security alerts generated by multiple network sensors and collected in Security Operations Centers (SOCs). Our method follows three key steps: data extraction, enrichment, and playbook execution. First, Llama normalizes security alerts using a common schema, extracting key contextual elements such as IP addresses, host names, filenames, services, and vulnerabilities. Second, these extracted elements are enriched by integrating external resources such as threat intelligence databases and Configuration Management Databases (CMDB) to generate dynamic metadata. Finally, this enriched data is analyzed through predefined false positive investigation playbooks, designed by security professionals, to systematically evaluate and identify FPs.By automating the false positive identification process, this approach reduces the operational burden on human security operators, enhancing the overall efficiency and accuracy of SOCs, and improving the organization's security posture.

Keywords: LLM,llama,AI Agent,SOC,SIEM,False Positive

1 Introduction

As cybersecurity threats continue to grow in number and complexity, Security Operation Centers face an overwhelming volume of security alerts generated by Security Information and Event Management (SIEM) systems. False positives alerts that incorrectly signal malicious activity are particularly problematic as they clutter the threat detection environment and divert attention from genuine threats, thereby increasing overall risk. To ensure optimal protection, it is crucial to focus on true attacks among the multitude of detected alerts. This necessitates the identification and reduction of false positive alerts generated by security sensors. False positive identification by machine learning (ML) models is utilized in various domains. In the realm of security, professionals are increasingly leveraging ML models to reduce false positives in security sensors. Techniques such as Adaptive Learning and Clustering (ALAC), Clustering Large Applications (CLARAty), Weighted Support Vector Machine (WSVM), Decision Tree Classification, and Rule-based Classification have shown promise in reducing false positives in Intrusion Detection Systems (IDS). Asieh Mokarian (2013)Ban et al (2023)

However, these models may Not be applicable within the scope of Security Operations Centers. Alerts generated by IDS typically follow consistent patterns and attack vectors, making clustering or classification models particularly effective. While above ML models are often successful in reducing false positives in **Security Sensors** and threat detection like **IDS** Spathoulas and Katsikas (2010)Pietraszek (2004)Al Jallad et al (2020), They may **not be Applicable** when applied comprehensively across an entire network monitored by SOCs. Security Operations Centers deal with a variety of attack vectors, multiple sensors, and numerous MITRE tactics and techniques, such as Advanced Persistent Threats (APTs).

Traditionally, security professionals across various SOC tiers manually identify false positives using a contextual-based approach. This method involves analyzing contextual information such as applications, services, and network location information to identify false positives Chergui and Boustia (2020). Additionally, some methods incorporate indicators of compromise (IOCs) and vulnerabilities to filter out false positives Alvas (March 21, 2024)Kullberg (Apr 11, 2024). However, this human-based approach is time-consuming and resource-intensive, requiring a significant number of security professionals.

Another utilization of machine learning for improving true positives in security monitoring systems is the use of LLMs as AI-assistants to enhance collaboration in Security Operations Center components like Security Information and Event Management (SIEM), Security Orchestration, Automation, and Response (SOAR) Gupta et al (2023) , and sensors. In some approaches, AI-assisted tools **provide recommendations** and insights to human analysts. This human-AI teaming leverages the strengths of both humans and AI. Humans bring intuition, contextual understanding, and evaluation, while AI offers computational power, data processing, and pattern recognition capabilities. By working together, humans and AI can improve true threat detection more effectively. For example, in recent approaches, security professionals use AI-assistants like chat bots as interactive knowledge base management tools for

improving their knowledge to detect true threats. Baruwal Chhetri et al (2024)Oniagbi (June 2024) The mention approach primarily focuses on improving true positive detection in security operation centers. By using large language models (LLMs) as AI-assistants, the goal is to enhance the collaboration between human analysts and AI tools, thereby improving the accuracy of threat detection. This human-AI teaming helps in identifying genuine threats more effectively, which in turn improves the overall security posture. Motlagh et al (2024) However, neither of these approaches couldn't entirely cover identifying false positives in security operation centers.
Our method focuses on addressing the issue of false positive alerts in Security Information and Event Management (SIEM) systems. SIEM systems frequently produce numerous alerts, many of which are false positives, posing a significant challenge for security operators who need to differentiate between actual threats and harmless activities. To tackle this problem, we suggest utilizing a Large Language Model as an AI agent to identify false positives in security alerts from various network sensors in Security Operations Centers. Specifically, we use the Llama Chat-bot to identify false positives through a contextual-based approach. This involves extracting key contextual elements in alerts, including related information and indicators of compromise (IOCs) such as applications, services, protocols, IP addresses, hostnames, filenames, usernames, domain names, processes, vulnerabilities, and more. These elements are then processed based on false positive investigation playbooks designed by security professionals to identify false positive alerts.

To achieve our goals with this method, we provide a detailed explanation in Section 2. Subsection 2.2 covers the use of Llama to extract contextual elements and information from security alerts. Subsection 2.3 describes how to enrich the extracted information with additional resources to generate dynamic metadata using Llama tools. In Subsection 2.4, we illustrate the implementation of algorithms and playbooks using Llama tools to identify false positives. Following this, we analyze the results of our method simulation in Section 3. We then discuss the limitations, challenges, and future work in Section 4. Finally, we present the benefits of our method in Section 5.

2 Method

This paper aims to clarify the challenges and solutions associated with false positive alert identification. SIEM systems often generate a large number of alerts, many of which are false positives. This creates a significant challenge for security operators who must distinguish between true attacks and benign activities. To address this challenge, we propose an approach that uses a Large Language Model (LLM) as an AI agent to identify false positives in security alerts generated by multiple sensors in a network and collected by SIEM in SOCs.

2.1 Study Design

Specifically, we utilize Llama to identify false positives through a contextual-based approach, which is explained in three key steps shown in figure 1. For deploying the LLM as an AI agent, we use Meta-Llama-70b-instruct in 8-bit mode on the NVIDIA A100 GPU with the H2O LLM Studio tool set.Llama Team (July 23, 2024)

Data Extraction: In this step, after a security alert is generated in the SIEM, the Llama agent analyzes the alert context and extracts key contextual elements and normalize based on Common Schema. These elements include related information and indicators of compromise (IOCs) such as victim or attacker IP addresses, involved applications, services, protocols, hostnames, filenames, usernames, domain names, processes, exploited vulnerabilities, and more.

Data Enrichment: In this step, the extracted information is enriched with additional external resources to generate dynamic metadata. The Llama agent enriches and validates the contextual elements against known external and relevant resources such as CMDB, Asset Inventory, Threat Intelligence databases and other Correlated Alerts in SIEM.

False Positive Playbook Execution: In this step, the enriched data is processed using alert false positive investigation playbooks. These playbooks, designed by security professionals, provide structured procedures for analyzing and identifying false positive alerts. Llama applies these algorithms and playbooks to analyze the data extracted in the previous step, identifying false positive alerts.

Also there is a bidirectional relationship between the playbook and enrichment units, allowing for additional information to be requested and incorporated into the playbooks as needed.

Through this comprehensive analysis, Llama identifies alerts that are likely to be false positives. The AI agent's ability to process large volumes of data quickly and accurately reduces the burden on human security operators, allowing them to focus on genuine threats. By leveraging Llama's capabilities, our approach aims to enhance the efficiency and accuracy of false positive identification in SOC environments, ultimately improving overall security posture.

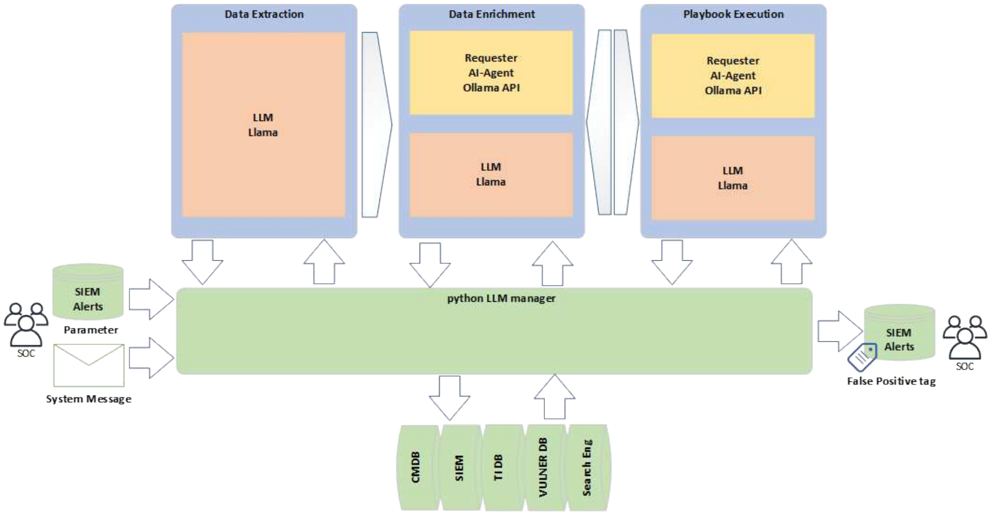
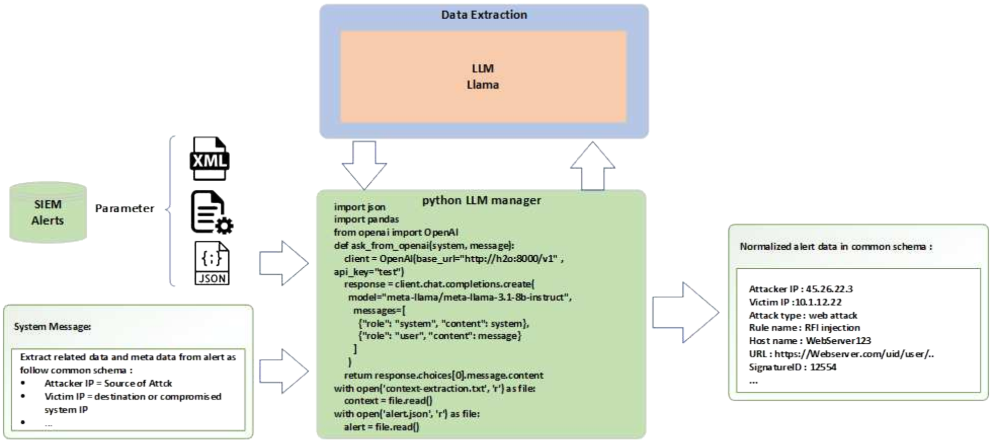


Fig. 1 method step diagram

2.2 Data Extraction

Security alerts are generated in various formats, such as XML, CEF, LEEF, JSON, and Syslog. To better recognize patterns and correlations using alert context information, it is necessary to normalize alerts based on a common schema. For normalizing alerts and extracting their information into a common schema format, we use Large Language Models (LLMs) like Llama, which employ advanced Natural Language Processing (NLP) techniques to manage these complexities. The initial step in this process is to identify the format of the incoming alert. The next step is to normalize the data to a our common schema. To achieve this, we leverage the capabilities of Llama to perform comprehensive data extraction from security alerts. The process begins with understanding the context of each alert, which involves parsing the data and identifying relevant elements that indicate a potential security incident. Once the format is identified, the alert data is parsed using a format specific parser (e.g., a syslog parser for syslog alerts). After parsing, the data is normalized into a standardized structure that the LLM can effectively interpret. This normalization process ensures that alerts originating from different formats are converted into a unified representation for analysis. Following normalization, Llama analyzes the data to extract crucial contextual information in the common schema. This involves understanding the relationships between various components within the alert. For example, Named Entity Recognition (NER) techniques are employed to identify and extract specific entities such as applications, services, protocols, IP addresses, host names, filenames, usernames, domain names, processes, vulnerabilities, and more. The output of this step is the extracted data in the common schema.Yang et al (2023). In this paper, the raised alerts are sent to Llama from SIEM. Since alerts generated by various external sources contain different fields and criteria, Llama functions as a large language model, analyzing the context of each alert to extract relevant information and normalize it according to a common schema closely aligned with designed Common Schema.shown in figure 2



```
import json
import pandas
from openai import OpenAI
def ask_from_openai(system, message):
   client = OpenAI(base_url="http://h2o:8000/v1" ,
api_key="test")
   response = client.chat.completions.create(
    model="meta-llama/meta-llama-3.1-8b-instruct",
     messages=[
      {"role": "system", "content": system},
      {"role": "user", "content": message}
     ]
    )
   return response.choices[0].message.content
with open('context-extraction.txt', 'r') as file:
   context = file.read()
with open('alert.json', 'r') as file:
   alert = file.read()
```

Extract related data and meta data from alert as follow common schema :

- Attacker IP = Source of Attck
- Victim IP = destination or compromised system IP
- ...

Attacker IP : 45.26.22.3
Victim IP :10.1.12.22
Attack type : web attack
Rule name : RFI injection
Host name : WebServer123
URL : https://Webserver.com/uid/user/..
SignatureID : 12554
...

Fig. 2 data extraction diagram

2.3 Data Enrichment

After Llama completes the initial data extraction, the information is further enriched by integrating external resources, creating dynamic metadata. The Llama AI agent, as seen in implementations like Llama tools, interacts with various systems and databases, including Configuration Management Databases (CMDB), Asset Inventories, Threat Intelligence platforms, SIEM systems, Vulnerability Management tools, and Search Engines. For example, it can use internet search engines to find vendor-published false positive hints for signature-based sensors or locate Common Vulnerabilities and Exposures (CVE) information, such as the Common Platform Enumeration (CPE) Dictionary, to identify false positives, as depicted in the figure. 3 This connectivity allows the agent to query, retrieve related metadata, correlate alerts, and deepen its understanding of security incidents.

By generating dynamic queries to access relevant external data, the AI agent enriches extracted information with comprehensive metadata. This ensures Llama delivers enhanced situational awareness by leveraging both up to date external resource and historical data, enabling it to effectively differentiate false positives from genuine alerts. Additionally, the agent maintains a bidirectional relationship with the playbook, responding to and assisting with playbook's requests.In the enrichment step, we use extracted data that is normalized based on a common schema. In this paper, we achieve this by using Llama as an AI agent.

For example, Llama requests additional information about the reputation of the attacker address from various threat intelligence databases via API. This enriched alert data about the attacker's reputation status is then used to evaluate the likelihood of false positives based on Bianco's Pyramid of Pain, which we explain further in the next section. Additionally, we request information about the operating system

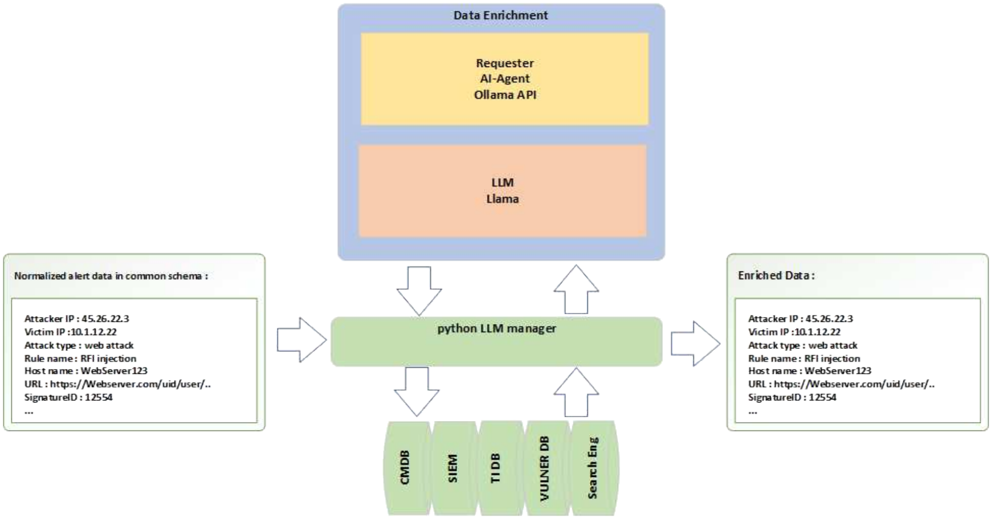


Fig. 3 data enrichment step diagram

and operational services on the victim server from external resources, such as the organization's CMDB, using SQL queries. This helps us enrich the victim server information and identify false positive alerts triggered by irrelevant Snort IDS signatures, such as SID: 1:58723, which is not applicable to a victim Windows server based on the metadata obtained during the enrichment process.

2.4 Playbook Execution

In cybersecurity, a playbook is a collection of predefined workflows and procedures that guide professionals in responding to specific incidents, threats, or tasks, such as identifying false positives. When an alert is triggered, the playbook provides clear, step by step actions, reducing decision making time and improving the accuracy of false positive identification. In the final stage, enriched data is analyzed using false positive investigation playbooks. These playbooks, designed by SOC Tier 1 analysts, provide structured methodologies for evaluating and identifying false positive alerts.

The creation of a false positive identification playbook is based on the expertise and knowledge of security professionals as SOC Tier 1 analysts. These experts draw on their understanding of the threat landscape, including attack vectors, common exploits, and adversary tactics, techniques, and procedures (TTPs), to design effective playbooks. Frameworks like MITRE ATT&CK and David J. Bianco's Pyramid of Pain are often used to classify threats and interpret attacker behaviors. For example, Bianco's Pyramid of Pain helps assess the likelihood of false positives in SIEM alerts. Indicators from higher levels of the pyramid (such as TTPs) are less likely to result in false positives, as they represent more sophisticated attacker behaviors, which are harder to alter and less likely to be mistakenly triggered by benign activities. In contrast, lower-level indicators (like IP addresses or hashes) require careful validation and should be corroborated with enriched metadata for accuracy.Bianco (2014) To refine the detection process, alerts should be categorized and labeled according to their false positive rate. Thresholds can be applied to distinguish normal behavior from malicious actions, while exception management can help account for legitimate activities that might otherwise generate unnecessary alerts. Regular updates to threat intelligence and business asset data are crucial for maintaining detection quality. Some playbook procedures may require additional data to ensure accurate execution, making it important to provide supplementary information during the playbook's operation. A bidirectional relationship between playbooks and enrichment units allows for the

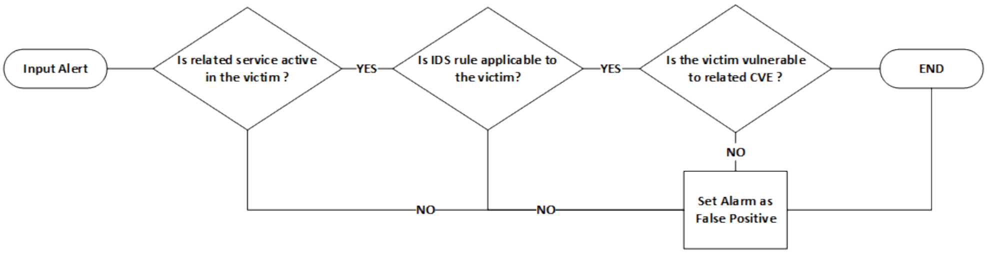

Input Alert
Is related service active in the victim ?
YES
Is IDS rule applicable to the victim?
YES
Is the victim vulnerable to related CVE ?
END
NO
Set Alarm as False Positive
NO
NO

Fig. 4 playbook diagram

request and integration of additional data when necessary. In this method, Llama applies playbooks designed by SOC experts to systematically evaluate enriched data and identify false positive alerts. To ensure successful implementation, the playbook must be precisely executed by AI agents like Llama.For example, to identify false positive alerts triggered by an IDS, Llama runs a playbook and analyzes it to evaluate conditions at each step and determine the next action based on previously provided enriched data. In this case, Llama systematically evaluates the playbook conditions at each step to decide the next step, as shown in Figure 4.This playbook is designed for IDS alerts, the initial step involves Llama searching for enriched data to determine the existence of active services on the victim server. If this information is unavailable, the Llama AI agent queries the Configuration Management Database (CMDB) or uses scanner tools to gather data from the victim server. If the service involved in the IDS rule is active, proceed to the next step. Conversely, if the related service is not active, mark the alert as a false positive and end the process. Next, Llama analyzes the IDS rule description to identify the applicable vendor, operating system (OS), or platform that triggered the IDS rule. If this information is not available in the enriched data, Llama searches the internet to find it. If the IDS rule is applicable to the victim, proceed to the next step. If the IDS rule is not applicable, mark the alert as a false positive and end the process. Finally, Llama looks for vulnerabilities (CVEs) related to the IDS rule on the victim server. If this information is not available in the enriched data, Llama queries the vulnerability scanner database to check for related CVEs on the victim server. If the victim is vulnerable, proceed to the end of the playbook. If the victim is not vulnerable, mark the alert as a false positive and end the process. In summary, this playbook helps determine whether an alert is a true positive or a false positive by checking the status of the related service using CMDB or scanner tools, analyzing the IDS rule to find applicable vendors, OS, or platforms, and querying the vulnerability scanner database to check for related CVEs on the victim server. If any of these checks fail, the alert is marked as a false positive, and the process ends.

2.5 Dataset

In this research, we utilize the Next-Generation Intrusion Detection System Dataset (NGIDS-DS), which was generated at the next-generation cyber range infrastructure of the Australian Centre for Cyber Security (ACCS) at the University of New South Wales (UNSW) at the Australian Defense Force Academy (ADFA) in Canberra. This dataset is part of ongoing cybersecurity projects at ADFA and serves as a realistic Intrusion Detection System (IDS) dataset. The NGIDS-DS dataset comprises both normal and abnormal host and network activities, which were performed during simulations. It was created as a major component of the PhD thesis Haider (2018) and serves as ground truth data with 313,000 records of IDS triggered alerts. This extensive dataset provides a robust foundation for evaluating the effectiveness of our proposed approach in identifying false positive IDS alerts within SIEM systems.

2.6 Data Analysis

In this simulation scenario, we apply the described method to identify false positives in a simulated environment where IDS alerts target a Windows Server 2016 hosting websites through Microsoft's IIS 10.0 service on port 80. The dataset used is NGIDS-DS, which contains a substantial number of IDS alert records simulating threats detected on the server within a Security Operations Center. For false positive identification, we utilize the playbook shown in Figure 4. To evaluate our method, we selected 1,000 records with various attack types across different services and we consider 12% of the records as false positives, equating to 120 records, for comparing its performance against manual analysis.

We assess the accuracy and efficiency of our method compared to manual FP identification by measuring accuracy and time efficiency. In the manual method, information about the target server, as mentioned in the scenario, must be manually requested from the organization's CMDB. Additionally, we manually researched attacks and related vulnerabilities using external sources, all based on the referenced playbook, to identify FPs in the 1,000 simulated alerts.

In contrast, our automated method begins with normalizing the dataset records using our AI agent, Llama. Each alert is analyzed and standardized into a common schema for consistency and further processing. This normalization involves parsing various data formats and extracting key contextual elements such as source and destination IP addresses, port numbers, alert descriptions, and CVEs to identify related services and attack information.

The next step enriches this data. Llama requests additional details on the attacks and associated CVEs by querying external resources like the CVE database or Snort rule documentation. Simultaneously, information about the target server is pulled from the organization's CMDB based on the destination IP and port, providing details on the operating system, active services, and running applications.

This enriched analysis is then processed through a predefined playbook, which runs a series of checks to systematically identify false positives. By comparing the active services, operating system, and applications of the target server with the expected conditions of the triggered IDS rule, the playbook assesses whether the alert represents a true attack or a false positive.

This simulation demonstrates how our method efficiently identifies false positives by leveraging contextual data, external resources, and automated playbook execution. It significantly reduces the manual workload of security operators, providing accurate and timely false positive identification in a complex SOC environment, as shown in Table 2.6. Our method's AI agent identified 112 records as false positives, correctly identifying 108 of them. However, 4 records were incorrectly flagged due to insufficient information in the alert body This process took 40 minutes for 1,000 records. In contrast, the manual method by SOC experts identified 110 FPs, correctly identifying 105 of them, with only 5 records falsely identified, and took 325 minutes. According to the mentioned results, the accuracy is assessed using the F-score. The AI method achieved a higher accuracy 108 with score 0.931, while the manual method accurately classified 105 of them with achieved a score of 0.913. The AI agent method is more

Table 1 compare of models

Model	Accuracy	f-score	Time(min)
LLM	**108**	**93%**	**40**
Manual*	105	91%	325

* one tier-1 soc analyst.

effective at correctly identifying false positives while minimizing incorrect identifications. Additionally, the AI method significantly outperforms the manual method in terms of time efficiency. The AI agent completed the identification process in just 40 minutes for 1,000 records, whereas the manual method took 325 minutes for the same number of records. This demonstrates that the AI method is not only more accurate but also much faster, reducing the time required by security operators to identify false positives by a substantial margin. This efficiency allows security teams to focus on more critical tasks and respond to threats more quickly.

3 Result

The AI-Agent method successfully identified 112 false positives and accurately classified 108 of them, achieving an F-score of 0.931. This higher accuracy demonstrates the effectiveness of the AI method, particularly in executing predefined playbooks for false positive identification. In comparison, the manual approach identified 110 false positives, correctly classifying 105 of them, with an F-score of 0.913. While still accurate, the manual method was slightly less effective in terms of precision and made more errors in classifying false positives.

The AI-Agent method's use of automated data enrichment such as querying external CVE databases and retrieving server information from the CMDB resulted in a more comprehensive, data-driven analysis compared to the manual approach. This automated process allowed the AI method to provide richer and more detailed reasoning for its decisions. In contrast, the manual method relied on human operators to request server details and research vulnerabilities, which led to less detailed and slower analysis due to human limitations in processing large datasets.

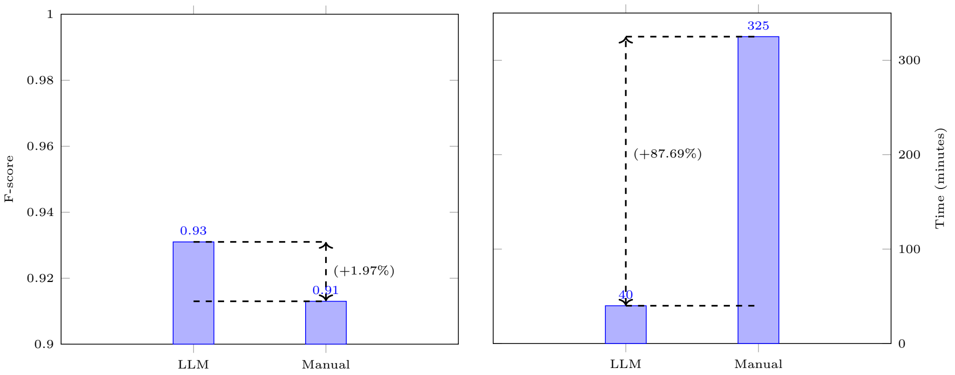


Fig. 5 comparison diagram

As shown in figure 3 in terms of efficiency, the AI-Agent method processed 1,000 records in just 40 minutes, compared to 325 minutes for the manual method. This represents an over 8-fold improvement in speed, enabling security teams to quickly focus on critical alerts and respond to threats by identifying and eliminating more false positives in a timely manner.

4 Discussion

The AI-Agent's superior performance in both accuracy and efficiency highlights the significant benefits of integrating automated intelligence into security workflows. This approach not only reduces the burden on human operators but also achieves an eight-fold improvement in speed. It provides a robust and scalable solution for threat detection and false positive identification, aligning with the increasingly data-intensive demands of modern cybersecurity environments.

4.1 Limitations and Challenges

Integrating an AI-Agent into SOC workflows for false positive identification presents several limitations and challenges. AI systems require continuous oversight to mitigate errors caused by ambiguous contextual information or incorrect assumptions. This is especially critical in complex environments where nuanced interpretation of data is essential, and false positives may not adhere to clear, predictable patterns. Moreover, integrating AI with existing playbooks and SOC processes can be resource-intensive, requiring ongoing system tuning, model retraining, and ensuring compatibility with tools like SIEM and SOAR.

Another key challenge is building and maintaining the infrastructure necessary for seamless communication between the AI-Agent and external resources, such as Configuration Management Databases (CMDB), vulnerability scanners, and organizational threat feeds. Establishing efficient communication channels is both technically demanding and costly, but essential for enabling the AI to access and process real-time information effectively. These challenges highlight the importance of careful planning, infrastructure investment, and resource allocation to maximize the benefits of AI-driven SOC operations.

4.2 Future Work

Integrating an AI-Agent with Security Operations Center (SOC) knowledge management systems (KMS) for interpreting knowledge trees represents a promising future direction. This approach would enable more advanced false positive identification by conducting deeper analysis of historical data, contextual threat patterns, and optimizing or designing new playbooks based on insights from the KMS. Such integration could significantly improve the speed and accuracy of false positive detection, making SOC workflows more dynamic and adaptable to emerging threats. Ultimately, this would strengthen the organization's security posture by continuously refining detection and response strategies.

5 Conclusion

The AI-Agent method clearly outperforms the manual approach in nearly every category. It is not only more accurate, achieving a higher F-score, but also significantly faster completing the task in a fraction of the time. Additionally, the AI-Agent method offers better identification of both false positives and true positives, along with more comprehensive reasoning through automated data enrichment. While the manual method is accurate and reliable, its slower processing time and lower accuracy make it less suitable for large scale operations in Security Operations Centers. In summary, the AI-Agent method is the superior choice for both performance and efficiency.

6 Statements and Declarations

Competing Interests:

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Funding Information:

No funding was received to assist with the preparation of this manuscript.

Author contribution:

All authors whose names appear on the submission made substantial contributions to the conception or design of the work; or the acquisition, analysis, drafted the work or revised it critically for important intellectual content; approved the version to be published; and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Data Availability Statement:

The data that support the findings of this study are available from the corresponding author, [author initials], upon reasonable request.

Informed Consent:

I understand that my participation is voluntary and that I am free to withdraw at any time, without giving a reason and without cost. I understand that I will be given a copy of this consent form.

Research Involving Human and /or Animals:

Not Applicable

References

Al Jallad K, Aljnidi M, Desouki M (2020) Anomaly detection optimization using big data and deep learning to reduce false-positive. Journal of Big Data 7(1). https://doi.org/10.1186/s40537-020-00346-1, URL https://link.springer.com/article/10.1186/S40537-020-00346-1

Alvas I (March 21, 2024) Non-human identities (nhi) - cybersecurity challenges and solutions. URL https://entro.security/blog/use-case-secure-non-human-identities

Asieh Mokarian AGDAhmad Faraahi (2013) False positives reduction techniques in intrusion detectionsystems-a review. IJCSNS International Journal of Computer Science and Network Security 13(10). URL http://paper.ijcsns.org/07_book/201310/20131020.pdf

Ban T, Takahashi T, Ndichu S, et al (2023) Breaking alert fatigue: Ai-assisted siem framework for effective incident response. Applied Sciences 13(11):6610. https://doi.org/10.3390/app13116610, URL https://www.mdpi.com/2076-3417/13/11/6610

Baruwal Chhetri M, Tariq S, Singh R, et al (2024) Towards human-ai teaming to mitigate alert fatigue in security operations centres. ACM Transactions on Internet Technology 24(3):1–22. https://doi.org/10.1145/3670009, URL https://dl.acm.org/doi/pdf/10.1145/3670009

Bianco D (2014) The pyramid of pain. Enterprise Detection and Response blog URL https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html, update

Chergui N, Boustia N (2020) Contextual-based approach to reduce false positives. IET Information Security 14(1):89–98. https://doi.org/10.1049/iet-ifs.2018.5479, URL https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/iet-ifs.2018.5479

Gupta M, Akiri C, Aryal K, et al (2023) From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. arXiv https://doi.org/10.48550/ARXIV.2307.00691, arXiv:2307.00691 [cs.CR]

Haider W (2018) Developing reliable anomaly detection system for critical hosts: a proactive defense paradigm. phdthesis, School of Engineering and Information TechnologyThe University of New South WalesAustralia, URL https://unsworks.unsw.edu.au/bitstreams/3ca5f3db-73e3-4bcb-b7e1-454d9fb2b178/download

Kullberg R (Apr 11, 2024) Identifying and mitigating false positive alerts. URL https://panther.com/blog/identifying-and-mitigating-false-positive-alerts/

Llama Team A (July 23, 2024) The llama 3 herd of models. Tech. rep., meta, URL https://llama.meta.com/

Motlagh FN, Hajizadeh M, Majd M, et al (2024) Large language models in cybersecurity: State-of-the-art. Computers and amp; Security https://doi.org/10.48550/ARXIV.2402.00891, arXiv:2402.00891 [cs.CR]

Oniagbi O (June 2024) Evaluation of llm agents for the soctier 1 analyst triage process. Master's thesis, University of TurkuDepartment of Computing, URL https://www.utupub.fi/bitstream/handle/10024/178601/Oniagbi_Openime_Thesis.pdf?sequence=1%26isAllowed=y

Pietraszek T (2004) Using adaptive alert classification to reduce false positives in intrusion detection. In: Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp 102–124, https://doi.org/10.1007/978-3-540-30143-1_6

Spathoulas G, Katsikas S (2010) Reducing false positives in intrusion detection systems. Computers and amp; Security 29(1):35–44. https://doi.org/10.1016/j.cose.2009.07.008, URL https://doi.org/10.1016/j.cose.2009.07.008

Yang J, Chen YL, Por L, et al (2023) A systematic literature review of information security in chatbots. Applied Sciences 13(11):6355. https://doi.org/10.3390/app13116355, URL https://www.mdpi.com/2076-3417/13/11/6355/pdf?version=1684816473