

# Reducing False Alerts in Cybersecurity Threat Detection Using Generative AI

Xiao Lin  
SMLS Group  
Splunk Inc.  
San Jose, CA, USA  
[xlin@splunk.com](mailto:xlin@splunk.com)

Glory Avina  
SMLS Group  
Splunk Inc.  
San Francisco, CA, USA  
[gloryavina@splunk.com](mailto:gloryavina@splunk.com)

Javier Santoyo  
Security Group  
Splunk Inc.  
Santa Monica, CA, USA  
[jsantoyo@splunk.com](mailto:jsantoyo@splunk.com)

## ABSTRACT

In the dynamic landscape of cybersecurity, the prevalence of false alerts significantly challenges the efficiency and effectiveness of threat detection systems. This paper explores the integration of Generative AI (GenAI) into the threat detection and response workflow to automatically suppress false alerts based on analysts' feedback on historical anomalies. Initially, a traditional approach was employed, utilizing ranked anomaly representation vectors to establish a baseline for false positive suppression. Subsequently, a GenAI agent was integrated into the workflow, enabling the suppression of false positives based on the agent's responses. Furthermore, a generative recommender system was employed to rank false positive anomaly candidates. Experimental results demonstrate a substantial reduction in false alert rates, suggesting that a GenAI-driven approach not only optimizes resource allocation for cybersecurity teams but also enhances the overall resilience of digital infrastructures against evolving threats.

## CCS CONCEPTS

• Security and privacy → Intrusion/anomaly detection and malware mitigation → Intrusion detection system • Computing methodologies → Artificial intelligence → Knowledge representation and reasoning → Probabilistic reasoning • Computing methodologies → Machine learning → Machine learning approaches → Learning latent representations

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
*KDD '24 AI4Cyber, August 2024, Barcelona, Spain*  
© 2024 Association for Computing Machinery. 978-1-4503-0000-0/18/06...\$15.00  
<https://ai4cyber-kdd.com>

## KEYWORDS

Generative AI, Cybersecurity, Threat detection, Recommender system, False alert, Feature representation

### ACM Reference format:

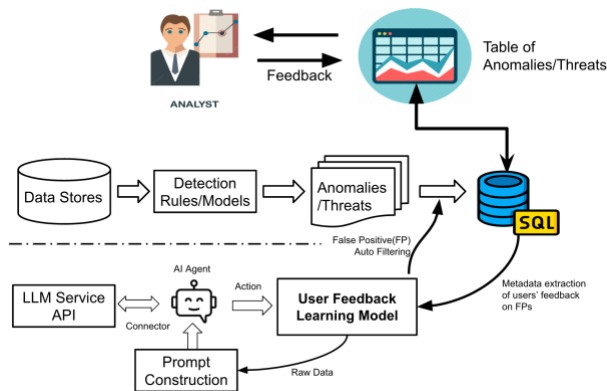
Xiao Lin, Glory Avina and Javier Santoyo. 2024. Reducing False Alerts in Cybersecurity Threat Detection Using Generative AI. In *the 4<sup>th</sup> Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics (KDD'24 AI4Cyber)*, Barcelona, Spain, 6 pages. <https://ai4cyber-kdd.com>.

## 1 Introduction

Anomaly and threat detection tools are critical for tracking anomalies, investigating incidents, and responding to cybersecurity threats (Samtani et al., 2022; Ban et al., 2023). However, the effectiveness of these tools is challenged by the diverse nature of user and entity behaviors across various organizations. Predetermined thresholds often fail to account for unique behavioral baselines, making accurate anomaly detection difficult (Gelman et al., 2023). Temporal variations within the same environment further complicate monitoring efforts, as significant differences in signals can arise between regular workdays and weekends, work hours and off-hours, and during holiday seasons. Additionally, user behavior and device usage across different applications introduce further uncertainties into detection models. Consequently, completely preventing false alerts from these detection tools is exceedingly difficult. The recurrence of false alerts previously flagged by analysts exacerbates the issue, as anomaly detection tools typically base their conclusions solely on input data without incorporating analysts' feedback on reported anomalies.

To address the growing need for tool accuracy and to prevent analysts from falling into false alert fatigue, we propose a user feedback-based false alert automatic suppression system, as shown in Figure 1. The upper part of the figure above the dot-dash line represents the existing tool capability, where analysts interact with tables of anomalies and threats. In this design, in addition to detection models generating anomalies, a user feedback learning

model is added as a post-detection step to rank and filter false alerts. We experiment with three options for this false alert suppression:



**Figure 1: System Diagram of User Feedback based False Alert Automatic Suppression**

1. Traditional approach that vectorizes the anomalies and then ranks them against user feedback.
2. Connecting to a Large Language Model (LLM) API service to leverage its few-shot learning and probability reasoning power.
3. Adopting a generative recommender system as an end-to-end flow to rank false alerts.

The following sections of this paper will include a review of related work, details of these three result implementations, and the corresponding results.

## 2 Related Work

False positive reduction techniques can be categorized into two main approaches (Gupta, Srivastava and Sharma, 2016). The first approach involves enhancing the detection techniques themselves during the detection phase to improve accuracy and reduce false alarms. Enhancing detection methods implicitly reduces false positives but may increase detection complexity. For example, Landress (2016) adopted an unsupervised learning method to profile users' normal behavior, which triggered alerts without generating many false positives. Other machine learning algorithms, including k-means clustering, decision trees, and self-organizing maps, then must be combined to form a hybrid approach that effectively reduces false positives.

The second approach focuses on implementing second stage alert processing techniques after first stage detection. This involves utilizing methods such as data mining, clustering, and machine learning to process and filter alerts generated in the first stage. Pietraszek (2004) introduced the Adaptive Learner for Alert Classification (ALAC), a system designed to classify alerts into true positives and false positives using machine learning techniques. ALAC learns from the feedback provided by analysts

and continuously updates its classification model to improve accuracy. It can autonomously handle alerts classified with high confidence, thereby reducing the workload on human analysts.

Additionally, McRee (2022) employed a quantitative, quasi-experimental approach using the Technology Acceptance Model to compare the usability of visualized alert output versus text-based alert output. The findings indicate that visualized alerts significantly enhance usability and efficiency, leading to better performance and reduced data fatigue among security analysts. Similarly, to address the challenge of dealing with a large number of false alerts in Security Operations Centers (SOCs) due to centralized data and threat detection, Gelman (2023) presented a system designed to reduce response time to actionable incidents, suppress false positives, and optimize the number of alerts that analysts need to investigate.

The second approach fundamentally involves a ranking problem and can be tackled using a wide range of machine learning methodologies, such as convolutional neural networks and recurrent neural networks (Choi et al., 2020), graph neural network techniques (Chen et al., 2023), and reinforcement learning (Chavali, Gupta, and Saxena, 2022). Esposte et al. (2016) developed a recommender system designed to filter and suggest relevant cybersecurity alerts to network administrators based on their ratings and preferences. This system employs a hybrid approach, combining collaborative filtering and content-based filtering. This hybridization effectively addresses the cold-start problem inherent in collaborative filtering and the lack of user data in content-based filtering, thus providing more accurate recommendations. As a result, network administrators can focus on the most pertinent alerts, enhancing their ability to manage network security efficiently.

The development and application of large language models (LLMs) represent a significant breakthrough in artificial intelligence, demonstrating remarkable performance across various tasks (Vaswani et al., 2017; Zhao et al., 2023). The introduction of ChatGPT in 2022 catalyzed the proliferation of Generative AI (GenAI), leading to its integration into numerous applications, including virtual assistants and operational automation, due to its unparalleled capabilities and near-human level intelligence (Brown et al., 2020; Touvron et al., 2023; Zhou et al., 2024). Notably, its capacity for reasoning and assistance in zero/few-shot learning scenarios is particularly compelling (Xie et al., 2024; Schaeffer, Miranda, and Koyejo, 2023). GenAI also revolutionizes recommender systems, elevating them to a new state-of-the-art level (Wu et al., 2023; Zhai et al., 2024). LLMs significantly enhance recommender systems by providing high-quality textual representations and leveraging extensive external knowledge to establish item-user correlations (Ren et al., 2023). These models excel in capturing contextual information, understanding user queries, and item descriptions, thereby improving recommendation accuracy and relevance. They also address the data sparsity issue through zero/few-shot recommendations, making reasonable suggestions without extensive prior data (Ding et al., 2021; Hou et al., 2024).

In the realm of cybersecurity, LLMs and GenAI have found numerous successful applications (Nourmohammadzadeh et al., 2024). These technologies not only enhance threat detection models but also play crucial roles in adversarial simulations and SOC automation. They strengthen risk assessment through comprehensive text data analysis and assist in detecting emerging threats by analyzing security logs and generating sophisticated alerts (Yigit et al., 2024).

We thus conducted a pioneering experiment to integrate GenAI into the SOC workflow to automatically suppress false alerts based on analysts' feedback regarding false positives. In this experiment, we initially employed a traditional approach, utilizing ranked anomaly representation vectors to establish a baseline for false positive suppression. Subsequently, we integrated a GenAI agent endpoint into the workflow, enabling the suppression of false positives based on the agent's responses. Finally, we employed a generative recommender system to rank false positive anomaly candidates. The results of this experiment represent an original and valuable contribution to the community, particularly for those interested in enhancing SOC efficiency.

### 3 Experiments and results

#### 3.0 Common Settings

Given a collection of anomaly items  $A[e_i][c_j][t_k]$ , where  $e_i$  refers to entity (user or device) index  $i$ ,  $c_j$  refers to anomaly category index  $j$ ,  $t_k$  refers to a time moment within the range  $t_k \in [T_c - \Delta T, T_c]$ . Here,  $T_c$  is the current time, and  $\Delta T$  is the time span over which event history is stored. Each element of  $A$  contains at least the following attributes, as listed in the table below:

Attribute	Type	Description
B	boolean	false positive flag
N	string	anomaly name
D	string	anomaly summary
M	string	model/rule name
F	list of string	feature names
V	list of numeric	feature values
S	numeric	anomaly score
T	string	event timestamp

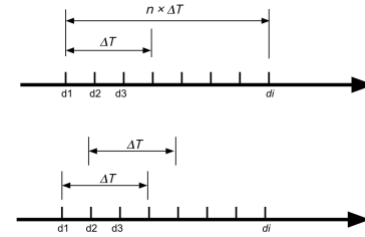
**Table 1: Attributes of an Anomaly Item**

The goal is to determine whether  $B = true$  for  $A[e_i][c_j][T_c]$ , if a subset of  $A[e_i][c_j][t_k]$  has been flagged as false positives, where  $t_k \in [T_c - 1 - \Delta t, T_c - 1]$ , and  $\Delta t < \Delta T$  is the length of time for which analyst flags will be kept.

The experiments are conducted using Azure compute instance with A100 80GB GPU memory and 220GB RAM.

#### 3.1 Anomaly Vectorization and Ranking

To train the vector representation, it is preferred to use the entire span of anomaly history  $n \times \Delta T$  as illustrated in Figure 2. When computational resources are insufficient to handle the anomaly size, it is possible to calculate the vectors incrementally using a fixed sliding window  $\Delta T$ . This approach ensures that the model retains past knowledge while adapting to evolving behaviors and emerging patterns.



**Figure 2: Incremental Anomaly Vector Vectorization**

Let the vectors (total number =  $N$ ) generated from the previous window be,

$$\vec{V}_o = \{\vec{v}_{o1}, \vec{v}_{o2}, \dots, \vec{v}_{oP}, \vec{v}_{o(P+1)}, \dots, \vec{v}_{oN}\}$$

and the new vectors (total number =  $M$ ) for the added anomalies

$$\vec{V}_n = \{\vec{v}_{n1}, \vec{v}_{n2}, \dots, \vec{v}_{nP}, \vec{v}_{n(P+1)}, \dots, \vec{v}_{nM}\}$$

We can solve for transform matrix  $[T]$

$$\vec{v}_{oi} = [T] \cdot \vec{v}_{ni} \quad \text{for } i \in [1, P]$$

$P$  represents the shared anomalies for the two calculations. Then, construct complete vectors as

$$\vec{V}_t = \{\vec{v}_{t1}, \vec{v}_{t2}, \dots, \vec{v}_{tP}, \vec{v}_{t(P+1)}, \dots, \vec{v}_{tN}, \vec{v}_{t(N+1)}, \vec{v}_{t(N+2)}, \dots, \vec{v}_{t(N+M-P)}\}$$

where,

$$\vec{v}_{ti} = \left( \frac{r}{r+1} \vec{v}_{oi} + \frac{1}{r+1} [T] \cdot \vec{v}_{ni} \right) \quad \text{for } i \in [1, P]$$

$$\vec{v}_{ti} = \vec{v}_{ni} \quad \text{for } i \in [P+1, N]$$

$$\vec{v}_{ti} = [T] \cdot \vec{v}_{n(i-N+P)} \quad \text{for } i \in [N+1, N+M-P]$$

In the above equations,  $r$  is defined as the ratio of anomaly size of previous calculation over new calculation and  $r \geq 0$ .  $r$  could be chosen as a large number where common anomalies in previous model are considered to have better accuracy, or as a smaller number if the current model is considered as comparable with the old one. Using this transformation matrix for incremental training not only effectively reduces computational resources but also provides a way to balance the weight of newly encountered anomalies with previously seen ones. We calculate the cross

product of vectors of all common anomalies between two consecutive time windows and compare the results before/after vector alignment through matrix transformation, as listed in the table below. The effectiveness of the transformation is demonstrated by the mean value changing from 0.09 to 0.99, with a near-zero standard deviation (0.01).

	Min.	Max.	Mean	Stdev
Before	-0.13	0.54	0.09	0.12
After	0.51	1.00	0.99	0.01

**Table 2: Vector Transformation Validation**

$A[e_i][c_j][T_c]$  is then ranked against  $A[e'_i][c'_j][t'_k]$  based on the similarity of their vector representations. Although we do not compare the vectors generated from incremental training with those trained over the entire history of anomalies due to their different vector spaces, we observed no significant discrepancy in false positive detection results when applying these two sets of vectors.

### 3.2 LLM API Connector

We use the Mistral 7B Instruct v0.2 model hosted through Azure AI Studio to conduct this experiment. We chose the Mistral LLM because it is claimed to have stronger reasoning capabilities compared to other models of similar size because of its use of grouped-query attention (GQA) and sliding window attention (SWA), which enhance its efficiency and inference speed while handling sequences of arbitrary length effectively (Jiang et al., 2023). Another reason for this choice is its API compatibility with OpenAI's APIs. We construct few-shot prompts to leverage the LLM's contextual understanding and reasoning power. We first ingest cases in  $A[e'_i][c'_j][t'_k]$  to the LLM API endpoint as examples of false positives. We then ask the LLM to check the anomalies in  $A[e_i][c_j][T_c]$  and determine whether they are "similar" to the ones in  $A[e'_i][c'_j][t'_k]$ . We then prompt the LLM should answer YES if these anomalies are considered false positives.

To validate the integration flow, we present example responses from the LLM based on the test dataset prompts, listed in the below table. Notably, the LLM occasionally responds with "unknown", which is valuable as it provides additional confidence in the system's reliability. Furthermore, these examples illustrate another potential application of LLMs within a Security Operations Center (SOC): automated detection report generation.

LLM Response	FP?
Based on the given anomaly detection summaries, I would say that the second anomaly is similar to the first one, but not identical. In both summaries, there is a mention of rare values or uncommon occurrences in the "Destination Zone" category. In the first summary, the specific zones "High_LB_App" and "Old_Prod_Temp" were identified as unusual. In the second summary, the zone "Old_Prod_Temp" is identified again, along with other destination zones. Both summaries also mention	YES

uncommon associations between the destination zones and either a "Source Zone" or "ouGroup". With this information, it's hard to be 100% confident, but I would still lean towards the answer "Yes", given that the second anomaly detection is similar in nature to the first one, and the same complication zone is mentioned in both summaries.

The two summaries are not very similar. The first summary is about the occurrence of specific processes and event descriptions, while the second summary is about the association between event types, ouGroups, and formats. The first summary mentions "wscript.exe" and "A new process has been created," while the second summary mentions "Account Records" and "SymantecDLP." Additionally, the first summary includes information about the number of occurrences of each process and event description, while the second summary includes information about the number of occurrences of each event type, ouGroup, and format when observing a specific ouGroup. Therefore, I would answer NO to indicate that the two summaries are not very similar.

Based on the information provided, the two anomaly detection summaries are similar in that they both report the detection of rare values for the same type of anomaly (process and event description) over a period of 30 days. However, the specific processes and events identified as anomalies, as well as their observed frequencies in the environment, differ between the summaries. Therefore, it is not clear if the second anomaly is a false alert based on the given information alone. To make a definitive determination, additional context or information would be needed, such as a comparison of the behavior of the anomalous process or event to normal behavior, and the significance of the process or event in the specific environment. Therefore, I cannot answer definitively based on the given information alone. Answers could range from "Maybe" to "No," depending on the interpretation of the information.

NO

?

### 3.3 Generative Recommender

A recommender system is used to suppress false positives by leveraging the assumption that a false alert for one user or device is likely to be a false alert for similar users or devices. The recommender system can be effectively adopted for false positive suppression by treating false positive anomalies as "items," a common practice in recommendation systems. The anomalies, whether regular or tagged by analysts, are stored in an SQL database and serve as input data, just as they do for the other two approaches discussed in section 3.1 and 3.2. We selected the state-of-the-art Generative Recommender model (Zhai et al., 2024) for our experiment. In this model, the decoder-style transformer block is replaced by Hierarchical Sequential Transduction Unit (HSTU) as shown in Figure 3. The HSTU splits the block input into the conventional **K** (key), **Q** (query), and **V** (value) matrices, and introduces a new **U** matrix designed to learn the hierarchy of feature interactions at different depths. The model reformulates ranking and retrieval tasks as sequential transduction tasks using the HSTU encoder design in conjunction with the M-FALCON algorithm. Numerical results demonstrate that this approach significantly improves both efficiency and speed.

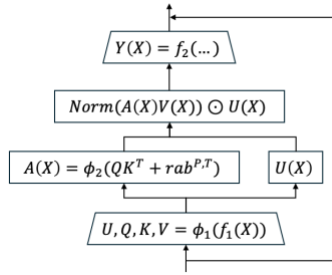


Figure 3: Hierarchical Sequential Transduction Unit (HSTU)

We use a synthesized dataset to evaluate the effectiveness of the GenAI recommender-based false positive detection in comparison to the two other approaches: anomaly similarity threshold and LLM agent reasoning. The dataset comprises 4 million events synthesized primarily from real world enterprise logs, with obfuscation applied for data privacy protection. These logs include:

- Cisco ASA
- Palo Alto Network Firewall
- Symantec DLP endpoint
- Web gateway
- Windows event log

The experiment simulated log data streaming into the detection environment, with anomalies identified by pre-defined rules and models. Analyst feedback on false alert was also simulated to trigger the false positive suppression modules. The results are presented in Figure 4. The experiments indicate that, for a limited number of anomalies, the generative recommender approach does not substantially enhance the hit rate. This might be due to the modeling of only the anomaly ID, without incorporating additional context, which limits the ability to fully leverage the multi-modal capabilities of generative AI. More extensive quantitative experimental results are required to draw more definitive conclusions.

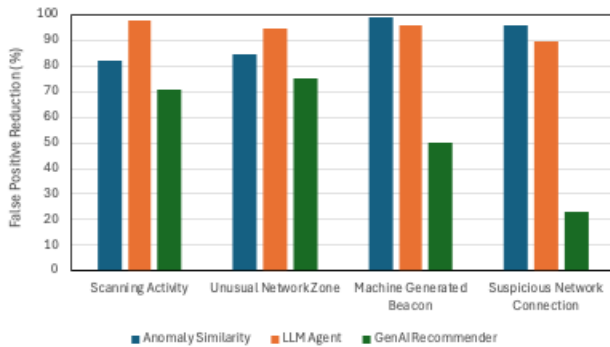


Figure 4: Effectiveness Comparison of False Positive Reduction

## SUMMARY

This paper examines the integration of Generative AI (GenAI) into the threat detection and response workflow, focusing on the automatic suppression of false alerts based on analysts' feedback on historical anomalies. Experimental results show a significant reduction in false alert rates, indicating that a GenAI-driven approach not only optimizes resource allocation for cybersecurity teams but also enhances the overall resilience of digital infrastructures against evolving threats.

Future work will aim to further improve the accuracy of GenAI-driven false alert suppression by implementing Retrieval Augmented Generation (RAG) (Lewis et al., 2020), leveraging the extensive domain knowledge contained within enterprise databases.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments to improve this work.

## REFERENCES

- [1] Tao Ban et al., 2023, Breaking Alert Fatigue: AI-Assisted SIEM Framework for Effective Incident Response, *Applied Sciences*, 13, 6610, <https://doi.org/10.3390/app13116610>.
- [2] Tom Brown et al., 2020, Language Models are Few-Shot Learners, in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada.
- [3] Lalitha Chavali, Tanay Gupta, Paresh Saxena, 2022, SAC-AP: Soft Actor Critic based Deep Reinforcement Learning for Alert Prioritization, in *IEEE World Congress on Computational Intelligence (WCCI2022)*, Padova, Italy <https://doi.org/10.48550/arXiv.2207.13666>.
- [4] Yiru Chen et al., 2023, Dynamic Graph Neural Networks-Based Alert Link Prediction for Online Service Systems, in *38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 79-90, <https://doi.ieeecomputersociety.org/10.1109/ASE56229.2023.00177>.
- [5] Yoon-Ho Choi et al, 2020, Using Deep Learning to Solve Computer Security Challenges: A Survey, *Cybersecurity*, vol. 3, no. 15, <https://doi.org/10.1186/s42400-020-00055-5>.
- [6] Hao Ding et al., 2021, Zero-shot Recommender Systems, arXiv:2105.08318 [cs.LG], <https://doi.org/10.48550/arXiv.2105.08318>.
- [7] Arthur de Moura Esposte et al., 2016, A Collaboration Model to Recommend Network Security Alerts Based on the Mixed Hybrid Approach, in *Proceedings of the Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*.
- [8] Ben Gelman et al., 2023, That Escalated Quickly: An ML Framework for Alert Prioritization, arXiv:2302.06648, <https://doi.org/10.48550/arXiv.2302.06648>.
- [9] Neha Gupta, Komal Srivastava and Ashish Sharma, 2016, Reducing False Positive in Intrusion Detection System: A Survey, *International Journal of Computer Science and Information Technologies*, Vol. 7, no. 3, pp. 1600-1603.
- [10] Yupeng Hou et al., 2024, Large Language Models are Zero-Shot Rankers for Recommender Systems, in *Proceedings of the 46th European Conference on Information Retrieval (ECIR 2024)*, Glasgow, UK, pp. 364-381, [https://doi.org/10.1007/978-3-031-56060-6\\_24](https://doi.org/10.1007/978-3-031-56060-6_24).
- [11] Albert Q. Jiang et al., 2023, Mistral 7B, arXiv:2310.06825 [cs.CL], <https://doi.org/10.48550/arXiv.2310.06825>.
- [12] Angela Denise Landress, 2016, A Hybrid Approach to Reducing the False Positive Rate in Unsupervised Machine Learning Intrusion Detection, *IEEE SoutheastCon 2016*, USA, <https://doi.org/10.1109/SECON.2016.7506773>.
- [13] Patrick Lewis et al., 2020, Retrieval-augmented generation for knowledge-intensive NLP tasks, in *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*, pp. 9459-9474, <https://doi.org/10.48550/arXiv.2005.11401>.



- [14] Griffith Russell McRee, 2022, Improved Detection and Response via Optimized Alerts: Usability Study, *Journal of Cybersecurity and Privacy*, vol. 2, pp. 379-401, <https://doi.org/10.3390/jcp2020020>.
- [15] Farzad Nourmohammadzadeh et al., 2024, Large Language Models in Cybersecurity: State-of-the-Art, arXiv:2402.00891 [cs.CR], <https://doi.org/10.48550/arXiv.2402.00891>.
- [16] Tadeusz Pietraszek, 2004, Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection, *Recent Advances in Intrusion Detection (RAID2004)*, LNCS 3224, pp 102-124.
- [17] Xubin Ren et al., 2023, Representation Learning with Large Language Models for Recommendation, in *Proceedings of the ACM on Web Conference (WWW'24)*, pp. 3464-3475, <https://doi.org/10.1145/3589334.3645458>.
- [18] Rylan Schaeffer, Brando Miranda, Sanmi Koyejo, 2023, Are Emergent Abilities of Large Language Models a Mirage? in *Proceedings of 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, <https://doi.org/10.48550/arXiv.2304.15004>.
- [19] Sagar Samtani, Gang Wang, Ali Ahmadzadeh, Arridhana Ciptadi, Shanchieh Yang, Hsinchun Chen, 2022, ACM KDD AI4Cyber/MLHat: Workshop on AI-enabled Cybersecurity Analytics and Deployable Defense, *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, August 2022, pp. 4900-4901, <https://doi.org/10.1145/3534678.3542894>.
- [20] Hugo Touvron et al., 2023, LLaMA: Open and Efficient Foundation Language Models, arXiv:2302.13971 [cs.CL], <https://doi.org/10.48550/arXiv.2302.13971>.
- [21] Ashish Vaswani et al., 2017, Attention Is All You Need, in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, <https://doi.org/10.48550/arXiv.1706.03762>.
- [22] Likang Wu et al., 2023, A Survey on Large Language Models for Recommendation, arXiv:2305.19860 [cs.IR], <https://doi.org/10.48550/arXiv.2305.19860>.
- [23] Qiming Xie et al., 2024, Ask Again, Then Fail: Large Language Models' Vacillations in Judgement, arXiv:2310.02174 [cs.CL], <https://doi.org/10.48550/arXiv.2310.02174>.
- [24] Yagmur Yigit et al., 2024, Review of Generative AI Methods in Cybersecurity, arXiv:2403.08701, <https://doi.org/10.48550/arXiv.2403.08701>.
- [25] Jiaqi Zhai et al., 2024, Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations, in *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, PMLR 235, Vienna, Austria, <https://doi.org/10.48550/arXiv.2402.17152>.
- [26] Wayne Xin Zhao et al., 2023, A Survey of Large Language Models, arXiv:2303.18223 [cs.CL], <https://doi.org/10.48550/arXiv.2303.18223>.
- [27] Qihang Zhou et al., 2024, AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection, in *Proceedings of the 20th International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, <https://doi.org/10.48550/arXiv.2310.18961>.