

DOI: https://doi.org/10.48009/3_iis_2026_106

System of cybersecurity analysis using generative AI as an assistive tool

Jacob Dilley, *Carson-Newman University*, jadilley@cn.edu
Seongyong Hong, *Carson-Newman University*, shong@cn.edu

Abstract

Intrusion detection systems are imperative for identifying emerging cyber threats; however, traditional anomaly-based approaches often generate high false-positive rates, which reduce analyst efficiency and delay response times. The impetus for this study was the necessity to develop an intelligent, multilayered detection framework that integrates machine learning and large language models to enhance alert accuracy, improve threat interpretation, and generate actionable security insights. Conventional anomaly-based intrusion detection systems have proven efficacious in identifying novel threats; however, they frequently exhibit elevated false-positive rates, which can inundate security analysts and diminish response efficiency. To address this issue, this research proposes a three-layer system that integrates machine learning-based anomaly detection with large language model analysis and automated incident report generation. The system utilizes the NSL-KDD dataset to simulate network traffic and employs the Isolation Forest algorithm to identify anomalous behavior. These anomalies are then analyzed using a large language model to classify threats, assign severity levels, and correlate the related alerts. The system generates structured incident reports that translate technical findings into actionable insights, aligned with the NIST Cybersecurity Framework. The findings indicate that while the system enhances the interpretability and correlation of alerts, challenges persist in reducing false positives and enhancing classification precision.

Keywords: Intrusion detection, Cybersecurity, Generative AI, Anomaly detection

Introduction

The increasing sophistication of cyber threats ranging from automated polymorphic malware to complex social engineering has outpaced traditional signature-based defense mechanisms. In the contemporary context, Security Operations Centers (SOCs) are grappling with a significant challenge known as "alert fatigue." This phenomenon arises due to the overwhelming amount of security telemetry, which makes manual triage an arduous task. The impetus for this research stems from the pressing necessity to address the disparity between the rapid evolution of threats and the limited analytical capacity of human beings. The integration of Generative AI (GenAI) has the potential to transform cybersecurity from a reactive, manual process into a proactive, augmented discipline. This transformation will enable analysts to interpret complex logs and neutralize threats with unprecedented speed and accuracy. As the digital landscape expands, the attack surface for modern enterprises has become increasingly fragmented and difficult to secure. Conventional cybersecurity systems frequently encounter issues such as elevated false-positive rates and a paucity of contextual intelligence during incident response. This paper puts forth a novel system for cybersecurity analysis that utilizes Generative AI as a primary assistive tool. In contrast to conventional automation, GenAI possesses the distinctive capacity to synthesize heterogeneous data points into coherent threat narratives, to elucidate complex code vulnerabilities in natural language, and to propose real-time remediation strategies. The system functions as a "force multiplier" for security professionals, with the

objective of reducing the Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR). Modern cybersecurity systems generate many network alerts, with many of them ending up as false positives. Due to security analysts manually reviewing these alerts, it can increase workload and slow response time, it also raises the risk of an analyst ignoring threats due to the assumption that it is a false positive. As threats grow in complexity and number, traditional systems can struggle to properly identify threats. Artificial intelligence has emerged as a promising solution to improve threat detection and alert analysis. Large language models (LLMs) offer the ability to analyze structured alert data and provide contextual reasoning that goes beyond traditional signature-based methods. The effectiveness of AI driven approaches in reducing false positives and improving accuracy remains an important problem. This research develops a Python based alert analysis system that uses an AI agent to evaluate network alerts. The system simulates real world alert generation and sends alerts to an AI model for classification and then organizes groups threats to more easily understand what to do next. The design has the AI model align recommended actions to the National Institute of Standards and Technology (NIST) Cybersecurity Framework to ensure it aligns with established cybersecurity practices. The objective of this study is to assess the effectiveness of AI assisted alert analysis in identifying malicious activity while minimizing false positives. By evaluating the system's performance through controlled experiments, this research aims to determine whether combining anomaly detection with large language model-based analysis can improve the interpretability and usefulness of intrusion detection systems, particularly in reducing analyst workload and improving understanding of complex attack patterns.

Background and Literature Review

Modern cybersecurity systems rely on intrusion detection systems (IDS) to monitor network activity and identify potential threats. These systems play a critical role in protecting digital infrastructure by analyzing network traffic and detecting malicious behavior (Buczak & Guven, 2016; Kaur et al., 2023). As cyber threats continue to grow in both complexity and scale, traditional detection methods face increasing challenges in maintaining both accuracy and efficiency. Intrusion detection systems are typically categorized into two primary approaches: signature-based detection and anomaly-based detection. Signature-based systems rely on predefined patterns of known attacks, often referred to as signatures. These systems are highly effective at detecting previously identified threats with low false positive rates. They are limited in their ability to detect new or unknown attacks as they rely on prior knowledge of attack behavior. Anomaly based systems establish baseline or normal network behavior and identify deviations from this baseline as potential threats. This approach enables the detection of new attacks, making it useful in dynamic and evolving environments. Anomaly based systems often suffer from higher false positive rates as legitimate behavior could be classified as malicious if it has not been seen before. This trade-off between detection capability and accuracy represents a central challenge in intrusion detection research (Sommer & Paxson, 2010).

The Network Security Laboratory Knowledge Discovery in Databases (NSL-KDD) dataset is a benchmark dataset for evaluating intrusion detection systems (Tavallaei et al., 2009). It is derived from the original KDD Cup 1999 dataset but is refined from its predecessor. The dataset contains simulated network traffic labeled as either normal or as belonging to various attack categories. These attack categories include denial of service (DoS) attacks, which aim to disrupt system availability, probe attacks, which involve scanning and reconnaissance activities, remote to local (R2L) attacks, where an attacker gains unauthorized access from a remote system, and user to root (U2R) attacks, which involve privilege escalation after initial access. Each record in the dataset contains a set of features describing network connections, including protocol type, service, duration, byte count, and error rates. This dataset is suitable for research because it provides

a controlled environment for evaluating detection methods while still capturing a range of realistic attack behaviors. It is important to note that the dataset does not fully reflect modern network conditions.

Isolation Forest is an unsupervised anomaly detection algorithm that identifies anomalous observations through recursive random partitioning rather than probabilistic profiling or distance-based clustering. Unlike traditional machine learning classifiers, which attempt to model both normal and malicious behavior, Isolation Forest operates on the assumption that anomalous data points are both rare and significantly different from the majority of observations. Because of these properties, anomalies are more susceptible to isolation and therefore require fewer partitions to separate from the remaining data (Liu et al. 2008). The algorithm constructs an ensemble of isolation trees. Each tree is generated by recursively partitioning the dataset using randomly selected features and randomly selected split values within the range of the selected feature. At each node of the tree, a feature is chosen at random, and the data are divided into two subsets according to the generated split point. This recursive process continues until either a single observation is isolated, all remaining observations possess identical values, or a predefined tree depth limit is reached. The fundamental measurement used by Isolation Forest is the path length, defined as the number of edges traversed from the root node of the isolation tree to the terminating node containing the observation. Normal observations generally require a larger number of recursive partitions because they exist within dense regions of the feature space and are surrounded by many similar observations. In contrast, anomalous observations tend to lie in sparse regions and are therefore isolated closer to the root of the tree, resulting in shorter path lengths. To improve robustness and reduce the influence of random partitioning, the algorithm constructs multiple isolation trees and computes the average path length for each observation across the ensemble. The average path length is then normalized into an anomaly score ranging from 0 to 1. Observations with shorter average path lengths receive higher anomaly scores and are considered more likely to represent malicious or abnormal activity. To visualize this effort, Figure 1 shows the algorithm in action by isolating two different data points (Liu et al., 2008). Point X_i represents what a normal data point looks like when being isolated. The point requires multiple partitions to isolate the point from the surrounding data, showcasing the characteristic that normal data is difficult to isolate. Point X_o represents what anomalous data can look like when being isolated. The point is quickly isolated in a minimal number of partitions showcasing the characteristics of anomalous data in which they are quickly and easily isolated due to their tendency to disperse from the normal data.

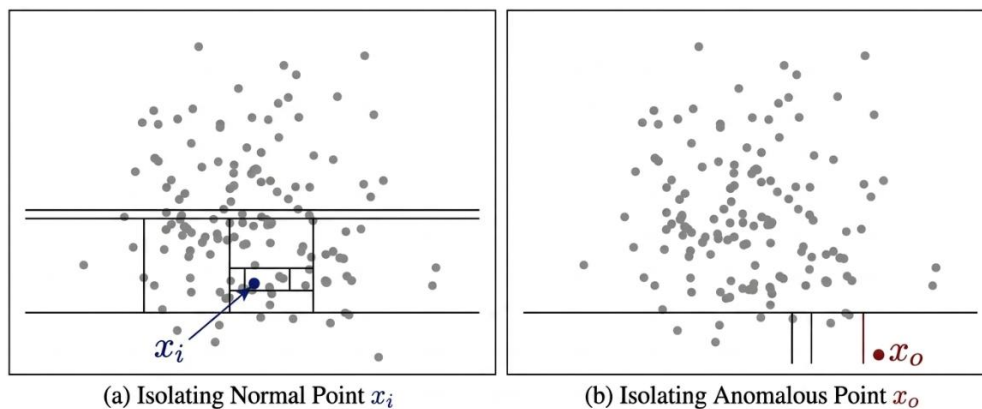


Figure 1. Visualization of isolation of anomalies, a normal point (X_i) vs an anomaly (X_o)

To improve reliability the algorithm builds many trees and computes the average path length for each data point across all trees. This average is then converted into an anomaly score. Data points with shorter average path lengths receive higher anomaly scores and are more likely to be flagged as suspicious. The anomaly score represents how different a given data point is from normal network behavior. In this system, the model is trained using only normal network traffic to establish a baseline of expected behavior. During testing, any data point that deviates significantly from this baseline receives a high anomaly score and is flagged

for further analysis. This approach allows the system to detect previously unseen attacks, since it does not rely on predefined attack signatures. It also introduces a limitation in that legitimate but uncommon behavior may be flagged as anomalous, contributing to a high false positive rate.

Large language models have recently emerged as powerful tools for analyzing and interpreting complex data. In cybersecurity these models can assist in tasks such as log analysis, threat classification, vulnerability identification, and report generation. Unlike traditional rule-based systems, LLMs are capable of processing contextual patterns in text and generating human readable explanations based on learned statistical relationships in data, making them useful for supporting security analysts (Balogh et al., 2024; Gupta et al., 2023). One of the key advantages of LLMs is their ability to integrate multiple sources of information and provide structured reasoning. For example, an LLM can analyze network alerts, identify patterns across multiple events, and generate explanations that describe the nature of potential attacks. This capability addresses a major limitation of traditional intrusion detection systems, which often produce large volumes of alerts without sufficient context. Artificial intelligence has become a critical component of modern cybersecurity systems due to the increasing scale, complexity, and frequency of cyber threats. Traditional security approaches, such as rule-based and signature-based systems are often limited in their ability to detect novel or evolving attacks. As a result, researchers have increasingly explored AI driven methods to enhance threat detection, automate analysis, and improve overall security posture. A comprehensive review by Kaur et al. highlights that AI can significantly improve cybersecurity operations by enabling automated detection, faster response, and improved accuracy in identifying threats (Kaur et al., 2023). Their work categorizes AI applications using the NIST Cybersecurity Framework, which organizes cybersecurity activities into five core functions: Identify, Protect, Detect, Respond, and Recover (National Institute of Standards and Technology, 2018). This structured approach provides a standardized way to evaluate how AI technologies contribute across the full lifecycle of cybersecurity operations. The NIST Cybersecurity Framework itself serves as a foundational model for managing cybersecurity risk. It emphasizes a flexible, risk-based approach that allows organizations to prioritize and align security activities with business objectives. In this research, the framework is used to guide system design and evaluation, ensuring that AI assisted alert analysis aligns with established cybersecurity practices, particularly in detection and response functions. Recent advancements in generative AI and large language models have introduced new capabilities in cybersecurity. These systems can analyze large volumes of structured and unstructured data, generate human-like reasoning, and assist in decision making processes. Generative AI can improve threat detection, automate incident response, and enhance threat analysis. However, these technologies also introduce new risks as they can be exploited by adversaries for tasks such as phishing, malware generation, and automated attacks. The dual-use nature of generative AI is emphasized in the “From ChatGPT to ThreatGPT” study, which demonstrates how the same technologies used for defense can also be leveraged for offensive cybersecurity operations (Gupta et al., 2023). This creates a significant challenge in cybersecurity, as improvements in defensive capabilities are often accompanied by increased adversarial sophistication. Similarly, other research highlights that generative AI can both strengthen security systems and expand the attack surface through automation and adaptive malware generation. Beyond generative models, the evolution towards agentic AI represents a major shift in cybersecurity systems. Agentic AI systems are capable of autonomous reasoning, planning, and continuous operation, enabling tasks such as incident response and threat monitoring (Lazer et al., 2026). These capabilities align closely with real world cybersecurity workflows, where continuous analysis and rapid decision making are essential. However, the increased autonomy of these systems introduces new risks, including coordination of attacks, evasion of detection, and reduced human oversight. Despite the advantages of AI, significant limitations remain, particularly in intrusion detection systems. Research by Sommer and Paxson highlights that machine learning based detection often struggles in real world environments due to high false positive rates, lack of reliable training data, and difficulties in interpreting results (Sommer & Paxson, 2010). These challenges are especially relevant to anomaly detection systems,

which frequently misclassify normal behavior as malicious activity. Similarly, surveys of machine learning methods for intrusion detection emphasize the tradeoff between detecting unknown attacks and minimizing false alarms (Zhan et al., 2025). Recent work on LLM assisted cybersecurity analysis suggests that AI systems can support human analysts by processing logs, identifying vulnerabilities, and generating summaries of security events (Balogh et al., 2024). However, these systems are not fully reliable and may produce incorrect or inconsistent results, reinforcing the need for careful evaluation and human oversight.

Methodology

The present study implements a three-layer IDS integrating anomaly detection, LLM analysis, and automated reporting. The system utilizes the NSL-KDD dataset, employing training on normal traffic and testing on mixed data. The system is organized into three layers: the first layer is the anomaly detection layer, the second layer is the LLM alert analysis layer, and the third layer is the incident reporting layer. Each subsequent layer is designed to enhance the output of the preceding layer, thereby transforming network alerts into a comprehensive and actionable report. The dataset under consideration encompasses both numerical and categorical features. Categorical features, including protocol type, service, and flag, are converted into numerical values through a process known as label encoding. Encoding mappings are reused across datasets to maintain consistency. Unseen values are mapped to default categories to prevent errors. In order to facilitate a more profound comprehension of the process, please direct your attention to Figure 2, which is presented below for your perusal.

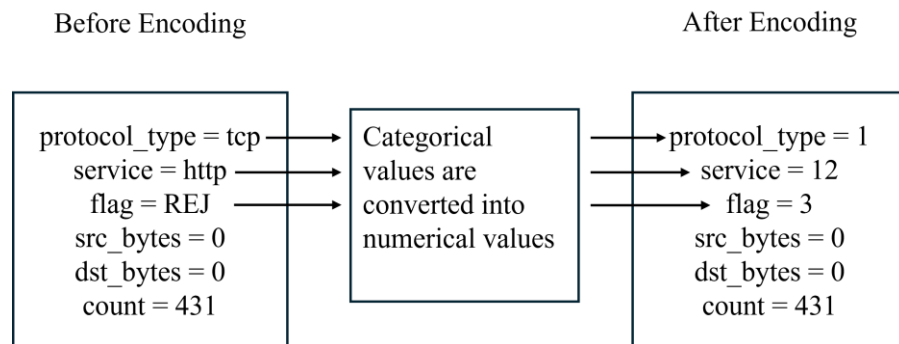


Figure 2. Data encoding example

In layer one, the Isolation Forest is applied to detect anomalous network behavior. The input to this layer consists of preprocessed network traffic data derived from the NSL-KDD dataset. Each record is represented as a numerical feature vector after categorical values have been encoded. These features include protocol type, service, connection count, byte counts, and error rates, all of which provide context about network behavior. During processing, each record is passed through the trained isolation forest model. The model evaluates how easily each data point can be isolated within its tree structure and assigns an anomaly score accordingly. Records that deviate significantly from the baseline learned during training receive higher anomaly scores. A threshold is applied to determine which records should be flagged as anomalous. The severity thresholds are as follows: low is 0, medium is 0.35, high is 0.45, and critical is 0.55. Any record that has a severity score above a certain threshold is assigned to that severity score when structuring alerts. The output of this layer is a set of structured alerts. Each alert includes the anomaly score and relevant network attributes such as source IP, destination IP, protocol, and traffic characteristics. An example of one of these alerts can be seen in Figure 3 below. The information put into each alert is used for further analysis. Each alert is ranked in descending order based on its anomaly scores.

```
ALERT-001 | CRITICAL | Score: 0.7041
Time: 2026-03-10 11:28:16
Source IP: 17.47.111.60 -> Dest IP: 97.217.28.16
Category: Probe (saint)
Protocol: TCP / OTHER [REJ]
Traffic: src_bytes=0 dst_bytes=0 count=431
Error Rate: 0.05 | Logged In: False
NIST: Detect / Identify -> Actions: ['DE.CM-1', 'ID.RA-3']
What it means: Reconnaissance: attacker is scanning and mapping the network before a follow-up attack.
```

Figure 3. Layer 1 alert example

The second layer uses the Claude Haiku model accessed through an Application Programming Interface (API) (Anthropic, 2024). The model was selected due to its ability to process structured data efficiently while providing detailed outputs. The input to this layer consists of the alerts generated in layer 1. Due to the large number of anomalies detected, only the top 50 alerts are selected based on their anomaly scores. These alerts represent the most significant deviations from normal behavior and are used as the basis for further analysis. During processing, each alert is formatted into a structured input and passed to the large language model along with a predefined prompt. The prompt provides context about the anomaly detection system and includes explicit rules for prioritization, threat scoring, and correlation. The model evaluates each alert based on factors such as anomaly score, frequency of activity, traffic volume, and error rate.

The prompt is then sent to the LLM using the Claude Haiku API, where the model processes the input and generates a structured response. During this process, the model evaluates each alert using the provided rules. The model assigns a priority level to each alert, calculates a threat score, and determines whether the alert is likely to be a false positive. It also identifies relationships between alerts by analyzing shared characteristics, such as common source IP addresses or similar activity patterns. This allows the system to group individual alerts into broader attack campaigns. Figure 4 provides a flowchart of prompt processing to allow for a better visualization of these events. To ensure that the model evaluates alerts based on specific factors, these criteria are defined within the prompt. The prompt includes prioritization rules, scoring guidelines, and correlating instructions, which guide the model's reasoning process. By embedding these constraints directly into the input, the system reduces variability in the model output and ensures that responses are aligned with the intended analytical framework. The LLM receives structured alert data and contextual information along with predefined analytical rules. These inputs guide the model in performing classification, correlation, and reasoning tasks. Upon receiving the alert data, the model evaluates each alert based on multiple factors, including anomaly score, frequency of activity, traffic volume, and error rates. It assigns a priority level to each alert and calculates a corresponding threat score. In addition, the model identifies relationships between alerts by analyzing shared characteristics such as source IP addresses and temporal proximity. This enables the grouping of individual alerts into broader attack campaigns.

The third layer uses multiple prompts to generate a comprehensive incident report from the analyzed alerts. While layer 2 focuses on structured classification, layer 3 focuses on generating a narrative report that can easily convey what threat types may be faced. The narrative report includes an executive summary, which describes the events in plain language without using technical jargon, and attack timeline, which shows the timeline of events breaking down when the attacks were carried out, and incident response plan, which is a response plan structured around the NIST Cybersecurity Framework organized into short term and long term follow up actions, and risk assessment, a breakdown of what could happen to your system if the threats are left unmanaged. During processing, multiple prompts are used to generate different sections of the report. The executive summary prompt instructs the model to translate technical findings into plain language suitable for non-technical personnel. It explicitly prohibits the use of technical jargon and requires the output to be written in paragraph form, ensuring clarity and accessibility for anyone who requires it.

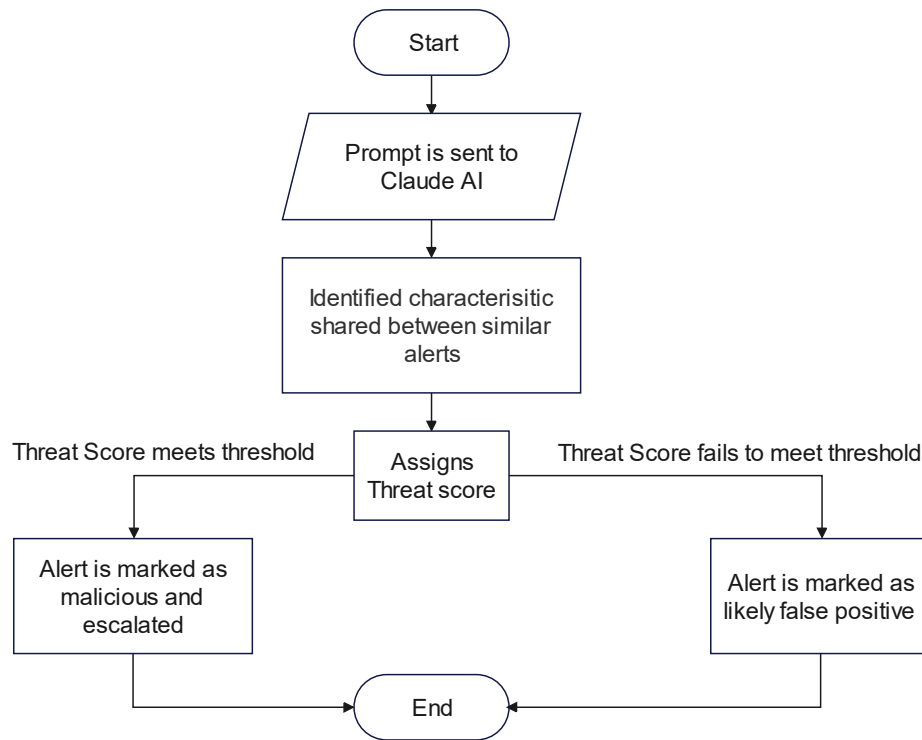


Figure 4. Flowchart of layer two prompt processing

The attack timeline prompt focuses on reconstructing the sequence of events. It guides the model to organize incidents chronologically and describes how individual alerts connect into larger attack campaigns. Instructions encourage the use of narrative transitions to help improve readability and narrative cohesiveness. The incident response plan prompt is structured around the NIST Cybersecurity Framework. It requires the model to generate actionable recommendations mapped to specific NIST control categories and organized into phases, immediate, short-term, and follow-up actions. This is to ensure that the output is not only descriptive but also helpful. The risk assessment prompt shifts focus on business impact. It directs the model to evaluate the consequences of unmitigated threats, including potential system compromise, data exposure, and operational disruption. The model is also required to assign an overall risk rating and justify why it has assigned it. Once all the individual components are generated, the system combines them all into a single report. This modularity allows each section to be customized to a person's individual needs.

Results

The performance of the three-layer intrusion detection system was evaluated using both quantitative metrics and qualitative analysis. Each layer of the system contributes differently to the overall effectiveness, with layer 1 focusing on detection, layer 2 on interpretation and classification, and layer 3 on communication and reporting. The first layer of the system demonstrated a strong capability in identifying anomalous behavior within the dataset. As shown in Table 1, the model processed a total of 22,544 records, including 12,833 labeled attacks and 9,711 normal instances. The model was able to correctly identify 12,262 of the attacks and 3,460 of the normal records. The model incorrectly labelled 6,251 normal records as attacks and 571 attacks as normal. This performance gives the system a 95.60% accuracy rating when identifying true alerts with 35.60% of alerts identified being false positives. The ability to achieve such a high detection

rate reinforces the strengths of anomaly-based intrusion detection systems while also showing the tradeoff of such systems with a high false positive rate. To better visualize these results a chart has been created in Figure 5 below.

Table 1. Layer one alert results

Layer 1 Results	Predicted Attack	Predicted Normal
Actual Attack	12,262 (True Positive)	571 (False Negative)
Actual Normal	3,460 (False Positive)	6,251 (True Negative)

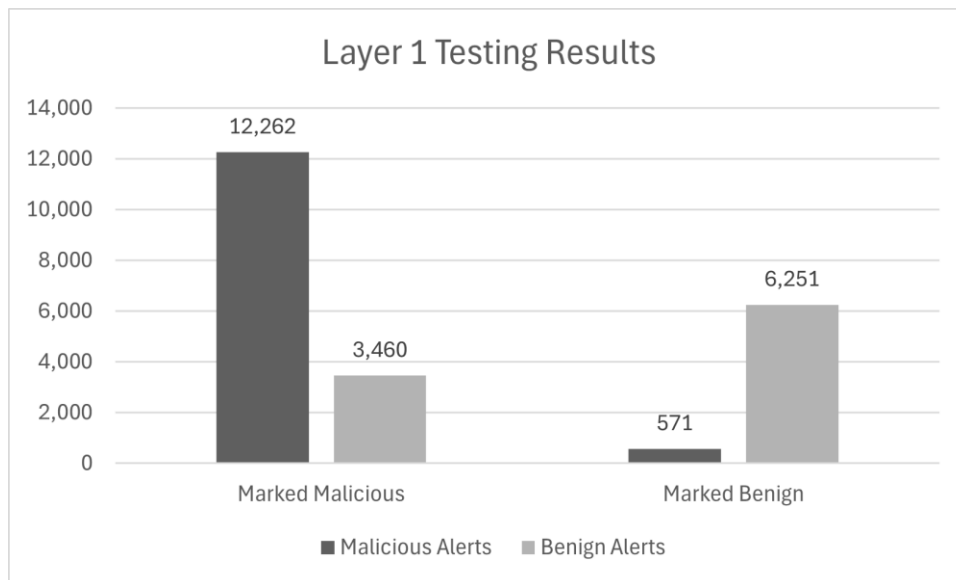


Figure 5. Layer one testing results

From table 1 additional performance metrics can be calculated. The positive predictive value (PPV), defined as $TP / (TP + FP)$, represents the probability that a detected attack is malicious. In this case, the PPV is approximately 78%, indicating that a large portion of flagged alerts are false positives. The negative predictive value (NPV), defined as $TN / (TN + FN)$, represents the probability that a non-flagged record is truly normal. The NPV in this study is approximately 92%, indicating that the model is highly reliable when identifying normal traffic. These metrics provide a more complete understanding of system performance, highlighting the tradeoff between detection sensitivity and false positive rates. A visualization of each factor can be seen through Figure 6.

To better evaluate the performance of the AI based alert analysis layer, three controlled test cases were introduced. These cases were designed to isolate different aspects of the system behavior by controlling the composition of the alerts passed from layer one into layer two. The first case consisted entirely of malicious alerts, allowing evaluation of the model's ability to correctly identify threats. The second case consisted entirely of benign alerts that were incorrectly flagged as malicious by the detection layer, allowing evaluation of the system's ability to detect false positives. The third case consisted of a balanced mixture of malicious and benign alerts, simulating a more realistic operational environment where both types of alerts are present. Each case included 50 alerts and was evaluated independently to assess classification accuracy, false positive detection, and reasoning consistency.

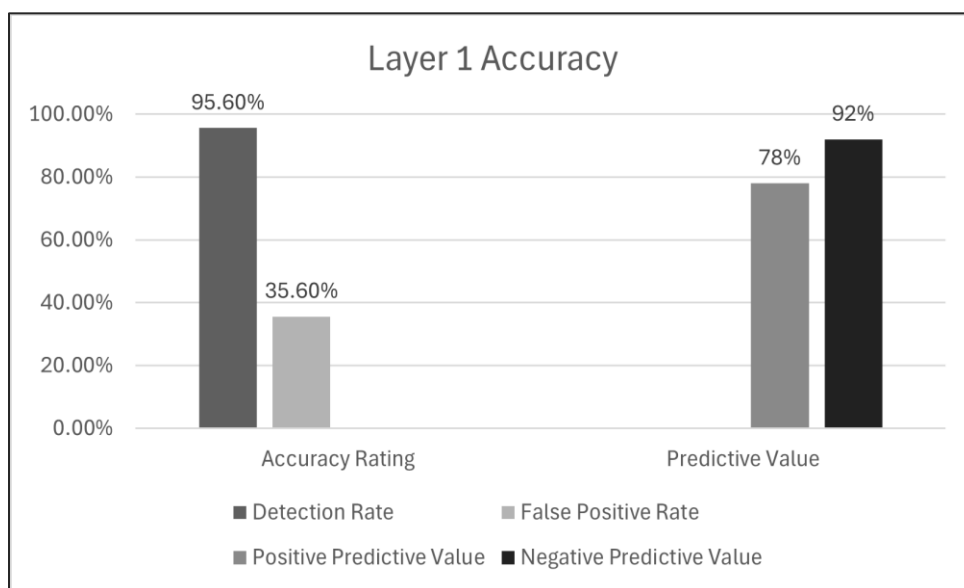


Figure 6. Layer one accuracy metrics graph

Case one results showed a 100% accuracy rating. The model was able to correctly identify that every alert presented was malicious without labeling any as false positives and forwarded the information to layer 3. Case two results had a 56% accuracy rating. The model was only able to identify 28 of the false positives correctly while incorrectly escalating 22 of the false positives as malicious. Case three demonstrated behavior consistent both prior cases. In line with case one, the model was able to accurately assess that all 25 of the malicious alerts were malicious and escalated them properly. The benign alerts were similar to case two, but the model had increased accuracy. The model correctly identified 17 of the alerts as benign and only escalated 8 alerts as malicious. This increased accuracy rate of 68% suggests that the presence of malicious alerts may provide contextual clues that assist the model in distinguishing benign anomalies more effectively. The overall accuracy of case 3 was 84%. A summary of the results for all 3 cases can be found in table 2 with a visualization of the accuracy rating to be found in Figure 7. These results show that while the model is highly effective at dealing with malicious alerts, it falls behind when dealing with false positives. The model correctly identifies over half of the false positives, reaching 56% detection for case two and 68% detection for case three.

The third layer was evaluated based on its ability to generate structured reports across the three test cases defined in layer two. In case one where all alerts were malicious, the system produced highly accurate and coherent reports. The generated reports successfully identified coordinated attack campaigns, including repeated connection attempts from multiple source IP addresses. The system grouped related alerts into unified attack narratives, clearly describing large scale reconnaissance activity. For example, the report identified sustained probing activity with high connection counts and elevated error rates, correctly interpreting these as indicators of systematic network reconnaissance. These results demonstrate that the report generation layer performs effectively when the input data is correctly classified. In case two where all alerts were benign, the performance of the system was dependent on the accuracy of layer two. Because the model incorrectly classified a portion of benign alerts as malicious, the generated reports included descriptions of attack activity that did not actually exist while still including genuine attack patterns. This resulted in reports that were misleading. For example, the normal connection rejection behavior was interpreted as reconnaissance activity, demonstrating how classification errors directly affected the accuracy of the generated reports. In case three the system produced mixed results. The reports correctly identified real attack patterns from malicious alerts while also incorporating incorrectly classified benign

alerts into the analysis. As a result, reports were generated that were partially misleading, where real threats were correctly described but were accompanied by misleading interpretations of normal activity.

Table 2. Results and accuracy metrics for layer 2 with 3 different cases

Layer 2 Results	Total Alerts	Malicious Correct	Benign Correct	False Positives	Malicious Accuracy	Benign Accuracy	Overall Accuracy
Case 1 (Malicious Only)	50	50	—	—	100%	—	100%
Layer 2 Results	Total Alerts	Malicious Correct	Benign Correct	False Positives	Malicious Accuracy	Benign Accuracy	Overall Accuracy
Case 2 (Benign Only)	50	—	28	22	—	56%	56%
Case 3 (Mixed)	50	25	17	8	100%	68%	84%

The presence of both correct and incorrect inputs highlights the sensitivity of the reporting layer to upstream classification performance. While the system maintained strong narrative structure and clarity, the presence of false positives reduced the overall reliability of the reports. These results show that the report generation layer demonstrates a strong capability in organizing and communicating cybersecurity events when properly informed. When given only correctly identified malicious alerts the reports were accurate and easier to comprehend, however once false positives started to get mixed into the system the accuracy of the reports went down causing a flow of misinformation to occur. The effectiveness of the reports generated in layer 3 is directly dependent on the accuracy of layer 2. When layer falsely escalates a benign alert, it can cause the accuracy of the entire layer 3 report to be called into question. Improving false positive detection in layer 2 is the best improvement that can be made for layer 3 to ensure that the reliability of final reports is not soiled.

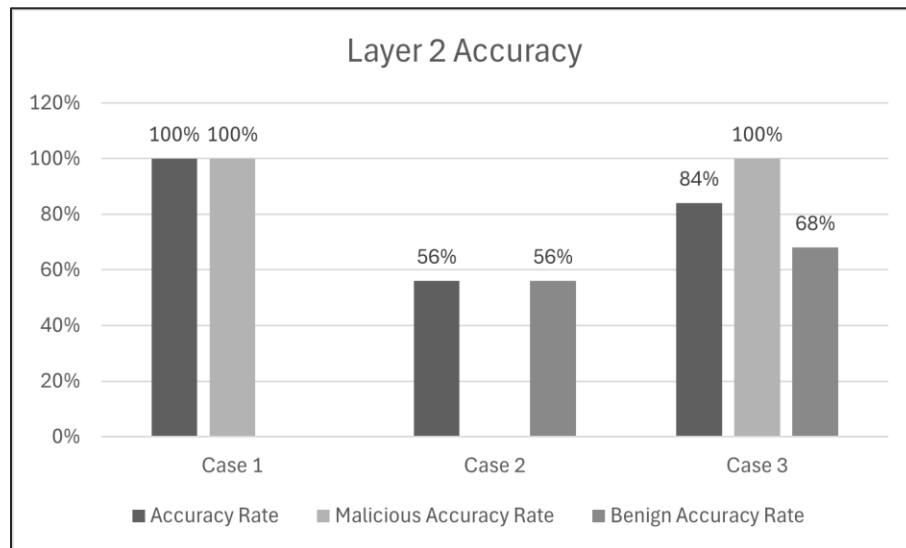


Figure 7. Visualization of accuracy rates for all three cases

Discussion

The results of this study demonstrate both the strengths and limitations of combining anomaly detection with large language model-based analysis in cybersecurity systems. While the system shows strong performance in identifying malicious behavior and generating structured intelligence, several observations emerge while examining the results in detail. One of the most significant details is the high detection capabilities of the anomaly detection layer. The Isolation Forest model managed to achieve a detection rate of 95.6%, successfully identifying the majority of attack instances within the dataset. This supports the existing research that suggests anomaly-based intrusion detection systems are highly effective at identifying previously unseen or unknown threats. This performance comes at a cost of a high false positive rate of 35.6%, which also reflects the limitations of unsupervised detection methods. Because the model is only trained on normal behavior, any deviation from this baseline can be classified as anomalous, whether malicious or not. The introduction of the second layer significantly improves the interpretability of these results. The LLM demonstrates strong capability in analyzing alerts, assigning severity levels, and correlating related events into broader attack patterns. As observed in the evaluation, the model achieved high accuracy in identifying threats and was able to group alerts into coordinated campaigns involving multiple source IP addresses and targets. This represents a major improvement over traditional intrusion detection systems, which typically produce isolated alerts without contextual explanation. Another important observation is the effectiveness of the third layer in transforming analytical results into actionable intelligence. The generated reports successfully integrate multiple aspects of the detected anomalous activity, including a timeline, threat correlation, and response strategies. This demonstrates the potential of LLMs to bridge the gap between technical detection systems and decision-making processes within organizations. By showcasing the ability to translate raw data into a structured narrative the system can enhance both usability and communication. When compared to existing research, the findings of this study align with prior research highlighting both the potential and the limitations of AI in cybersecurity. As discussed in the literature review, anomaly detection systems are known to produce high false rates, and LLM based approaches introduce challenges related to reliability and consistency. This study reinforces these observations while also demonstrating that combining these approaches can provide meaningful benefits. The LLM was able to detect over half of false positives and dismiss them correctly, showing an ability to help with the largest issue of anomaly detection systems, alert fatigue from a high false positive rate. The results also indicate that further refinement is necessary to ensure that such systems can be used reliably. Improvements in false positive detection, model calibration, and integration with real time data sources would be essential for deployment in a production environment. Additionally, the choice of language model would affect performance, as more advanced models are shown to increase accuracy rates as seen in Balogh et al. discussion on the uses of generative AI in cybersecurity, but the more advanced models also come at a rapidly increasing price compared to the minor improvements in accuracy they can provide (Balogh et al., 2024).

Conclusion

This study presented a three-layer AI assisted intrusion detection system that integrates anomaly detection, large language model reasoning, and automated incident reporting. By combining traditional machine learning with generative AI, the system demonstrates how cybersecurity detection can move beyond raw alerts toward actionable intelligence. The results show that layer 1 achieved a high detection rate of 95.6%, successfully identifying the majority of the malicious activity, but this performance came at the cost of a high false positive rate of 35.6%, highlighting a common limitation of anomaly-based intrusion detection systems. Layer 2 addressed this challenge by applying AI driven reasoning to analyze alerts, managing to correctly identify 84% of threats when presented a mix of malicious and benign alerts and successfully

correlate related alerts into coordinated attack patterns. Additionally, layer 3 demonstrated the ability to transform technical alert data into structured reports aligned with the NIST Cybersecurity Framework, providing clear guidance for incident response and decision making. Despite these strengths, the system exhibits important limitations. Most notably, the AI analysis layer showed difficulty in correctly identifying false positives, often classifying less than half as malicious. This indicates that while large language models are highly effective at interpreting and explaining threats, they may lack the precision required for filtering false positives without additional constraints or tuning. A factor that could have influenced AI's ability to detect false positives would be the model used. More expensive models have improved accuracy when used in cybersecurity tasks, but the costs associated can be multitudes higher than the cheaper models for only a slight increase in accuracy. Future work should focus on improving false positive reduction through hybrid approaches, such as incorporating supervised classification models, refining prompt engineering, or introducing feedback mechanisms to guide AI decision making. Expanding the system to operate real-time network traffic and integrating additional datasets would also enhance its practical applicability. Overall, this research demonstrates that combining machine learning and generative AI can significantly enhance intrusion detection systems by improving interpretability, correlation, and response planning. While challenges remain, particularly in reducing false positives, the proposed approach represents a meaningful step toward more intelligent and automated cybersecurity defense systems.

References

- Anthropic. (2024). Claude Haiku (Mar 7 version) [Large language model]. anthropic.com
- Balogh, Š., Mlynček, M., Vraňák, O., & Zajac, P. (2024). Using generative AI models to support cybersecurity analysts. *Electronics*, 13(23), 4718. <https://doi.org/10.3390/electronics13234718>
- Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, <https://doi.org/10.1109/COMST.2015.2494502>
- Gupta, M., Akiri, C., Aryal, K., Parker E., & Praharaj, L. (2023). From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access*, vol. 11, pp. 80218-80245
- Kaur, R., Gabrijelčič, D., & Klobučar, T. (2023). Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion*, 97, 101804. <https://doi.org/10.1016/j.inffus.2023.101804>
- Khan, M. I., Arif, A., & Khan, A. R. A. (2024). The most recent advances and uses of AI in cybersecurity. *BULLET: Jurnal Multidisiplin Ilmu*, 3(4), 566–578. https://www.researchgate.net/publication/390740851_The_Most_Recent_Advances_and_Uses_of_AI_in_Cybersecurity
- Lazer, S. J., Aryal, K., Gupta, M., & Bertino, E. (2026). A survey of agentic AI and cybersecurity: Challenges, opportunities and use-case prototypes. *arXiv*. <https://arxiv.org/pdf/2601.05293>
- Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. In 2008 eighth IEEE international conference on data mining (pp. 413-422).
- National Institute of Standards and Technology. (2018). Framework for improving critical infrastructure cybersecurity (Version 1.1). U.S. Department of Commerce.

- Tavallae, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD Cup 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1–6). <https://www.ee.torontomu.ca/~bagheri/papers/cisda.pdf>
- Sai, S., Yashvardhan, U., Chamola, V., & Sikdar, B. (2024). Generative AI for Cyber Security: Analyzing the Potential of ChatGPT, DALL-E, and Other Models for Enhancing the Security Space. IEEE Access, vol. 12, pp. 53497-53516.
- Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. 2010 IEEE Symposium on Security and Privacy, 305–316. <https://doi.org/10.1109/sp.2010.25>
- Zhang, Z., Li, J., Manikopoulos, C. N., Jorgenson, J., & Ucles, J. (2001, June 5–6). HIDE: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proceedings of the 2001 IEEE Workshop on Information Assurance and Security (pp. 85–90), West Point, NY, United States. IEEE.