International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered SOC Analyst Assistant

DEVARAPALLI SEKHAR BABU[*1], SHAIK RUKSAR[2], KADIYALA VISHNU GOPI[3], N V N S MAHIMA[4], P SIVA BRAHMA REDDY [5]

[1]Assistant Professor, Department of CSE(CB), Bapatla Engineering College, Bapatla 522101, AP,India
[2]Student, Department of CSE(CB),Bapatla Engineering College, Bapatla 522101,AP, India
[3]Student, Department of CSE(CB), Bapatla Engineering College, Bapatla 522101, AP, India
[4]Student, Department of CSE(CB), Bapatla Engineering College, Bapatla 522101, AP, India
[5]Student, Department of CSE(CB), Bapatla Engineering College, Bapatla 522101, AP, India

***Abstract*—** Security Operations Centers, or SOCs, basically keep watch over all the cybersecurity stuff in a company. They monitor events, check them out, and handle threats from different systems all the time. But with how fast everything digital is growing, theres just so many logs piling up. Its getting hard to go through them by hand, you know.

Traditional SIEM systems, those Security Information and Event Management ones, they mostly use fixed rules to spot attacks. That means they only catch the patterns someone already knows about. So, a lot of false alarms pop up, and they miss those new zero-day threats that no one has seen before. Analysts end up with way too much to think about, which slows everything down and makes responses take longer. It seems like that cognitive load thing is a big issue.

This research is trying to fix that with something called an AI-Powered SOC Analyst Assistant. Its like an automated helper that uses machine learning and these large language models for reasoning. The idea is to make threat detection and response happen in real time, more adaptively. First off, it pulls in logs from places like authentication setups, network gear, and firewalls. Then it cleans up the data, normalizes it so its consistent for analysis later.

One main part is the anomaly detection using Isolation Forest algorithm. Its unsupervised, which is cool because it doesnt need labeled data to find weird patterns. That helps spot unknown attacks that traditional stuff misses. After that, theres event correlation that links related logs together into actual incidents. It cuts down on repeats and gives better context, I think.

The system also has this LLM reasoning engine that explains anomalies in plain words. It turns alerts into useful summaries, rates how serious they are, and helps with decisions. Combining ML with natural language stuff like that should speed up analysis and make it easier to understand. Without it, things might stay pretty opaque.

In experiments, this setup did better than the old ways. Detection accuracy went up, false positives dropped, and time for checking alerts went from minutes to almost instant. It handles both known threats and the surprise ones, which seems effective for changing environments. That part stands out.

Overall, it looks like a solid way to scale up SOC operations. Automating the tough analysis parts and supporting analysts could reduce fatigue and improve visibility. Some challenges still linger, like integrating it everywhere, but it contributes to smarter cybersecurity down the line. I might be oversimplifying, though.

***Key Words*— Cybersecurity, SOC, Anomaly Detection, Isolation Forest, Machine Learning, LLM, Threat Detection, Log Analysis.**

II. INTRODUCTION

In the digital world we live in now, cybersecurity matters a lot for all kinds of organizations. Things like cloud computing and IoT devices are everywhere, and they create tons of data in IT systems. Security logs are key here, they help spot bad activities or weak points in the network. SOCs, those are the teams that watch these logs all the time to catch threats as they happen.

The problem is, with so much data coming in, its overwhelming. Every second, thousands of events pop up from firewalls, intrusion systems, servers, and all sorts of network stuff. Traditional SIEM tools use fixed

rules to check this out, which works okay for known attacks. But they miss new ones, zero-day threats I mean, and they set off too many false alarms.

That means analysts in the SOC have to sift through all these alerts manually. A lot of them turn out to be nothing, so it takes forever to investigate, responses get delayed, and people just get tired from it. Cyberattacks are getting smarter all the time, so maybe we need something better, like AI to help automate and speed things up for the analysts.

This paper suggests an AI-powered assistant for SOC analysts. It uses machine learning to make things better in cybersecurity. The system handles log ingestion first, then preprocesses the data, pulls out features, detects anomalies, correlates events, and reasons with AI.

One part relies on the Isolation Forest algorithm. That detects weird stuff in the logs without needing rules set up ahead. So it can find known threats and the unknown ones too. It seems useful because it cuts down on guessing.

Then theres this Large Language Model integrated in. It explains the anomalies in plain words, like what they mean for humans. Analysts save time not having to dig as deep, and overall the SOC runs smoother. Event correlation groups related logs into one incident, so less mess and better context.

In tests, it improved accuracy, fewer false positives, quicker responses than old methods. Combining automation with smart analysis pushes cybersecurity forward. Organizations can defend better against new threats, I think. But its not perfect, some parts might need tweaking still.

III. LITERATURE SURVEY

Detecting threat through analysis of security log is an extensively explored area in the field of cybersecurity, especially in the security operation centre (SOC). Conventional methods rely heavily on Security Information and Event Management (SIEM) systems which employ pre-defined rules and signature-based detection techniques. Security systems can identify known attacks but often fail to determine novel or zero-day attacks. Furthermore, rule-based implementations create an excessive number of alerts, many false alarms, which overburdens SOC analysts and hampers operational efficiency. System limitations flag the need for intelligent and adaptive tools for cybersecurity monitoring [1].

In order to overcome the limitation of rule-based systems, several researchers started using machine learning to detect anomalycyber security. Machine learning models, especially the unsupervised ones, can discover irregularities in large log data without labeled data. Clustering, classification, and anomaly detection techniques help systems detect deviations from normal behavior more accurately. Manyof these approaches lack interpretability, making the reasoning behind detected anomalies difficult for analystsS the analyzeable and their practicalSOS adoption environments option [2].

Isolation Forest algorithm has become popular as an anomaly detector due to efficiency and scalability. It is a technique that is especially apt for security logs that are high-dimensional and large-sized and works by isolating abnormal characteristics in the dataset. According to various studies, Isolation Forest is useful for intrusions in network and abnormal user activities. Although the algorithm helps in improving accuracy but alone does not provide contextual insights or explanations for detected anomalies which are crucial in real-life cybersecurity scenarios [3].

In addition to finding anomalies, event correlation methods have been proposed to improve the efficiency of security monitoring systems. Event correlation groups related log entries into a single incident. This process reduces redundancy and gives a clear view of security events. It helps lessen alert fatigue and allows analysts to concentrate on critical threats. However, current correlation techniques often depend on preset rules or simple heuristics. This reliance limits their ability to respond to complex and changing attack patterns [4].

Recent developments in Artificial Intelligence and Natural Language Processing (NLP) have led to the use of Large Language Models (LLMs) in cybersecurity. These models can understand and generate human-like text. They can provide explanations, summaries, and insights for detected security events. LLMs have been used in areas like threat intelligence analysis, incident reporting, and automated documentation. While these models improve understanding and usability, their integration with real-time anomaly detection systems is still in the early stages and needs more research [5].

Moreover, recent studies stress the need for cybersecurity frameworks that bring together different technologies, including machine learning, AI reasoning, and real-time processing. Current systems often concentrate on separate components like detection, correlation, or visualization, but they lack a unified approach that combines all functionalities. There is a growing demand for intelligent systems that not only find anomalies but also provide useful insights, reduce false positives, and help speed up incident response in Security Operations Center (SOC) environments [6].

IV. EXISTING SYSTEM

Most organizations use a SIEM (Security Information Event Management) system to monitor all security logs coming from many different places, e.g., network devices, authentication systems, firewalls and servers, in today's cybersecurity environments. Security Information Event Management (SIEM) systems use predefined rules, signatures and correlation patterns to determine if there is a potential threat to the organization based on the security logs that are generated from all of these sources. SIEM will provide centralized visibility of all security log events and basic alerting capabilities, however, SIEM is predicated on static configuration rules and therefore the chance of detection of emerging cyber threats is weak without continual manual updating.

A significant limitation of existing systems today is their inability to detect unknown, or "zero-day," attacks. Rule-based mechanisms are only able to detect previously documented, or known, attack patterns. Thus, any newly developed or sophisticated cyber threats that do not conform to SIEM's predefined rules will not be detected. This significantly diminishes the overall security posture of an organization. Additionally, SIEM systems generate a significant amount of alerts to security operations analysts with a high percentage (greater than 75 percent) of those alerts being irrelevant or false positives. Therefore, SOC analysts are burdened with manually checking many alerts which increases the workload of SOC analysts, increases incident response times, and reduces their overall productivity.

A major concern with current cybersecurity solutions are the absence of smart automation and the inability to interpret what you are seeing. While a few solutions utilize basic correlation methods to see if there are relationships between events, they are primarily due to pre-established and preset non-dynamic rules. This lack of intelligence makes it hard for analysts to identify the initial cause of an incident, as well as not being able to see the complete extent of an incident. And, most conventional solutions also lack human-readable material and do not provide a means of interpreting the raw logging information on your own. This means you are going to have to do the analysis manually, which is extremely time-consuming and has a lot of opportunity for error.

In addition, many of the current cybersecurity solutions are also fragmented in nature, utilizing distinct tools across the board for log collection, log analysis, alerting, and reporting. The lack of integration adds to the overall inefficiency and complicated difficulty level of carrying out your cybersecurity operations. Most traditional solutions also lack a means for real-time processing, which delays threat detection and threat response. With the cyber-threats becoming more complex and at a larger scale than ever before, the need for a more intelligent, automated, and integrated solution for security monitoring is evident.

V. PROPOSED SYSTEM

To address the shortcomings of existing SIEM-based security methods, the new system provides an AI-Powered SOC Analyst Assistant that uses machine learning and artificial intelligence to enhance the capabilities of SIEM for detecting and analyzing threats. The system is composed of a Log Ingestion module that acquires security logs from various sources including network devices, authentication devices and firewalls; a Data Preprocessing Module which transforms those logs to a cleaned and normalized format; and a Feature Extraction Module which extracts the relevant features from the logs (e.g., timestamp, IP address, event type and user behavior) and converts them into numeric features for use in machine learning analysis. The main component of the system is the Anomaly Detection Engine which uses the Isolation Forest algorithm to identify anomalies in the data. This means that the system can find both known and unknown (i.e., zero-day) threats using unsupervised techniques without needing to rely on predefined rules or labeled sample datasets.

An Event Correlation Engine processes log files and relates them to accomplish people-based result producing processes to lessen a number of duplicate alerts which leads to reduced analyst fatigue through more organized incident reports. Furthermore, LLM Based Reasoning Engine is also part of this system

which provides human-readable explanations, classifications of severity, context-based information for detected anomalies which minimizes the amount of time needed for manual investigations allowing for better decisions for SOC analysts. The system operates under technologies such as; Python based applications, FastAPI, Scikit-learn allowing for significant scalability as well as real-time processing capability. In addition, the system offers a user-friendly interface that visualizes incidents and provides visibility into; overall system performance and best support for overall expedient threat mitigation. In conclusion, this proposed system provides a complete, fully automated & intelligent approach to enhancing the overall accuracy of detection, eliminating false positives, and increasing the efficiency of all cyber security operations.

VI. SYSTEM ARCHITECTURE

1. Log Ingestion Layer

The log ingestion layer is responsible for collecting logs of security events, from many sources, which consist of firewalls, authentication systems, networking equipment and computer servers. This logs are streamed continuously to the ingestion layer and support multiple structured and unstructured log formats at the same time. The ingestion layer is designed to handle large volumes of data streams in real-time.

2. Data Preprocessing Layer

Once the raw logs are collected from various sources, they then undergo several data processing steps in order to remove noise, inconsistencies and duplicates of all data collected. The raw logs are cleaned, standardised and converted into one consistent format; this step is vital in order to improve the quality of data, thus preparing for further analysis of data.

3. Feature Extraction Layer

During this layer, various attributes from the logs, i.e. IP addresses, timestamps, User Login Attempts, the user activity patterns and event types are extracted from the logs into numerical formats that can be processed by machine learning algorithms**.**

4. Layer for Detecting Abnormalities

The main analysis component of the system is called the Anomaly Detection Layer, and this is where the Isolation Forest algorithm can be applied to detect abnormal behaviour in the dataset. In this layer, the model identifies abnormal behaviour by identifying event records that are separate from the majority of the record set. By isolating the record from the rest of the data, the algorithm can detect records that could be a result of known or unknown threats.

5. Layer for Correlating Events

The Event Correlation Layer correlates multiple events into one incident based on the correlation of many events. By grouping events with an established relationship into one incident, the number of duplicate notifications will be decreased and there will be an overall better understanding of possible threats since all of the pieces of information will be provided with an overall view of the possible threats.

6. Layer for Large Language Model Reasoning

The Large Language Model Reasoning Layer of the system can be integrated with a Language Processing Model (LLM) for creating text-based descriptions of the abnormal activity that occurs in the previous layers. With this description, a summary of the threat, the level of severity and other contextual details about the event can be provided to help the SOC Analyst understand the results clearly, and also take the appropriate actions.

7. Layer for Application Programming Interface (API) and Processing

To communicate between the layers of the system, the API is implemented using FastAPI. The API layer is responsible for the data flow through all layers, as well as processing all requests and providing a means for the front-end interface and back-end modules to work together in coordination.

8) Dashboard/Visualization Layer

There is an easy-to-use dashboard for displaying incidents that have occurred together with the severity levels associated with those incidents and other performance indicators related to the system. This allows the analyst to view in real-time what potential threats or incidents are occurring at any given moment, track them as they occur and make decisions quickly based on the information being displayed.

9) Data Storage (Layer)

The data storage layer is used to write processed logs, identify anomalies, collect information about incidents in a structured database design. The data can be retained so that historical analyses can be performed, and the data can be retrieved efficiently for future investigations.

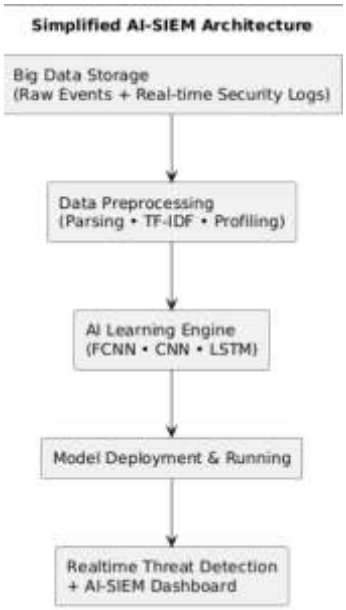

Figure 6.1 :System Architecture For Threat Detection

VII. METHODOLOGY

1. Data Collection and Log Ingestion

The first phase of our methodology is to collect logs from many sources, including network devices/firewalls/authentication servers, etc. These logs can come in many formats, so the ingestion mechanism should utilize APIs to allow for continuous/real-time data flow into the log management system. The ingestion mechanism has been designed to accommodate high volumes of log data and will ensure no critical information will be lost during transit. The logs that have been ingested will create the baseline dataset needed for subsequent analysis and/or processing.

2. Data Preprocessing and Normalization

After logs have been ingested into the log management system, they will undergo preprocessing in order to improve their characteristics (i.e. quality/consistency). The preprocessing phase consists of many cleaning techniques (removing duplicate log entries, dealing with missing values, filtering unnecessary log information, and standardizing log formats). As different log sources will likely have different structures, the normalization process will combine these logs into one consistent format. Normalizing logs is vital to ensure that machine learning algorithms function correctly with the preprocessed logs and increases the accuracy of the entire system. Having proper preprocessed logs also reduces the amount of noise in the data, which will ultimately aid in detecting anomalous behavior.

3. Extraction and Transformation of Features

During the preprocessing phase, the system will extract specific characteristics or features of log data that can help identify potential security threats. Some of the characteristics of log data that can be extracted as features include: timestamps, IP addresses, user activities, login attempts, frequency of access and types of events. The extracted features will then be transformed into numerical representations suitable for inputting into machine learning models. The use of feature engineering techniques highlights the important relationships and patterns between the features being extracted from the log data. This step is critical in improving the anomaly detection model's performance by providing it with informative and structured input data.

4. Anomaly Detection Through Machine Learning

Anomaly detection is the central analysis component of the system; it is accomplished via an unsupervised learning algorithm called “Isolation Forest” that can detect anomalies in log data by isolating points based on their deviation from the characteristics of normal behavior. The advantage of this model is that it does not require training data with known labels; therefore, it is well suited for use in the field of cybersecurity, where labeled datasets are often small or nonexistent. The model can be trained on typical log activity and continuously analyze real-time data being received from third-party sources to determine if there are any significant variations from the predicted range of activity. The anomaly detection part of the system can therefore identify new and previously unknown threats (including zero-day attacks) in real-time.

5. Event Correlation & Incident Generation

The system employs an event correlation process during threat analysis that connects related log entries into one security incident. This is done by establishing connections between multiple pieces of information through attributes such as time, type (what the event is), source (where it originates), and the behavior patterns of the event (how it behaves). By establishing these relationships, the system reduces duplicate alerts and provides a larger picture of what could potentially be a problem. Through this process, SOC analysts experience less alert fatigue because they can focus on key incidents, bypassing working on the individual log entries.

6. AI Reasoning & Explanation Generation

Along with detecting anomalies, the system uses a Large Language Model (LLM) to provide intelligent reasoning and explanations regardless of how the threat was identified. The LLM analyzes the detected anomalies and produces human-readable summaries, severity classifications, and contextual insights. This assists the analyst in understanding both the type of threat being investigated and its impact without having to manually review large amounts of raw log file data. The reasoning engine will improve the interpretability of findings, help make decisions faster, and improve the usability of the system in real-world SOCs.

7. System Integration/API Communication

The overall system is built with a back-end framework (FastAPI) to enable communication between various components of the system using APIs to manage data flow between the front-end dashboard, processing components, and machine learning models. This creates a seamless interaction and enables real-time data flow across the system. The modular approach supports the independent updating and scaling of each component, helping enhance maintainability and flexibility.

8. Visualization/User Interaction

An easy-to-use dashboard will be created to allow users to quickly and easily see the results from the model in both a visual and interactive way. On the dashboard, you will find the anomalies detected by the system, as well as information about any related incidents, the level of incident severity, and metrics on all performance criteria in real time. Charts and graphical representations of the information will help the analyst understand the anomalies and processes so they can make a decision quickly. This interface/grapical representation will facilitate an analyst in the Security Operations Center (SOC) to be able to monitor activity on the system, investigate and respond to incidents effectively, and to respond to incidents as quickly as possible.

9. Evaluation/Testing of Performance

The last step of the methodology is to evaluate the system using various performance measurement criteria (i.e., accuracy of detection, false positive rates, and response time). A series of simulated attacks and actual logs are used to prove the effectiveness of the model. As a result of using the model, the data will provide evidence that the model will produce better results than standard methods, such as increased accuracy in detection of anomalies, reduced number of false positives, and increased speed of analysing an incident. There will be no time period associated with the continuous evaluation of the system to maintain adequate/maximum performance as the cyber-environment changes.

VIII. IMPLEMENTATION

The suggested AI-Powered SOC Analyst Assistant will be a modular and scalable system that uses modern technology to allow for efficient real-time operating and smart threat analytics. Its backend is built on Python with the FastAPI Framework, allowing for high-performance API handling, as well as seamless communication between various modules. The system will start with the log ingestion process, which will import security logs from many sources, such as network devices, authentication systems and servers, and then preprocess those logs; in the preprocessing process, the logs will be cleaned, normalized and converted into a structured format. Once received in the preprocessing format, additional processing steps will be performed to extract attributes of interest from the logs, such as timestamps, IP addresses, event types, user behaviour patterns; these will be converted into numerical values to be used in machine-learning analysis. The core anomaly detection will be used based on Isolation Forest Algorithm in Scikit-Learn; this algorithm will learn how to identify abnormal patterns in log data where there are no labelled datasets. Once these detected anomalies are identified, they will be processed through an event correlation module to establish a connection between a set of log entries that are directly related (e.g. 2 users logging into 2 different systems from the same IP address within the same period) in order to reduce redundancy and improve the contextualisation of finding relevant pieces of information to help identify a threat.To enrich the interpretability of detected anomalies, the system also uses a Large Language Model (LLM) to create human-readable explanations, threat summaries, and severity classifications. This will also support SOC analysts in making decisions with better-informed options. A user-friendly dashboard has been developed to visually see real-time alerts, incidents, and performance metrics of the system for effective monitoring and response. The processed data, detected anomalies, and details of the incident are stored in a structured database to ensure persistence of the data for future analysis. This implementation has adopted a layered and modular design to provide scalability, flexibility, and effective management of large-scale security data while

improving detection accuracy and reducing analysis time relative to traditional systems.

IX. RESULTS

1. Initial State of System Dashboard

The As located, shows that until logs are processed, the initial state of the system dashboard, will not show any incidents (0) for any of the key metrics including; Total Incidents, Logs Processed, Anomalies Detected and Critical Incidents. Therefore we see all metrics report an idle state. This makes it easy for an analyst to quickly see the current state of the system (ready to analyze).

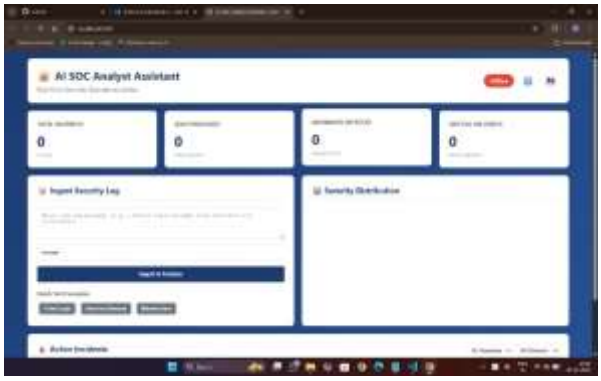

Figure 9.1 : Intial State

2. Log Processing with Successful Incidents

The logs that have been logged into the system show that they have successfully been ingested and processed as indicated by the incident ID that produced an output confirmation message. The metrics associated with logs that have been processed will change because of this change in the number of logs and incidents that are processed. The ability of the system to accept incoming data from multiple sources is demonstrated by the conversion of that data into usable security incidents.

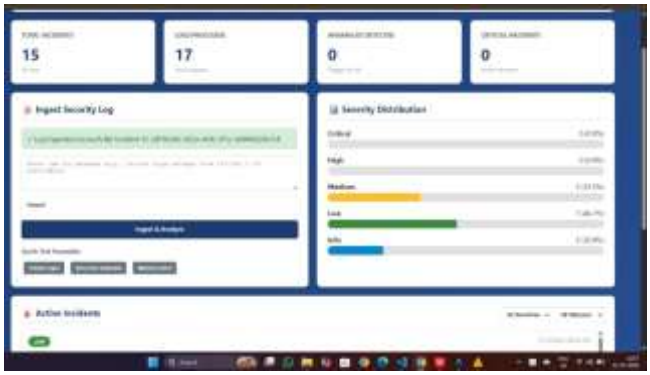

Figure 9.2 : Incidents

3. Active State of Current Metrics on a Dashboard

This result shows the system in an active state, having already processed many logs and finding extracted anomalies. The dashboard shows updated values for Total Incidents, Logs Processed, Anomalies Detected and Critical Incidents and therefore confirms that the system is capable of operating in real-time and continually updating its analytical metrics.

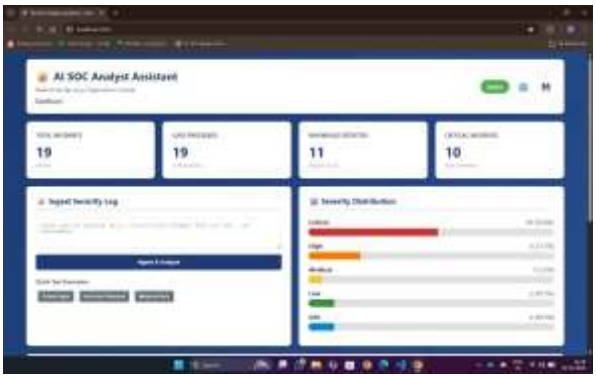

Figure 9.3 : Active State Logs

4. Analysis of Severity in Incident Data

The panel depicts categorized incidents based on severity (Critical, High, Medium, Low, and Information) as visualized by the charting elements. The view is presented in a graphical representation so that returns indicate percentages of each level of threat compared to all other threats. This provides SOC analysts with immediate knowledge regarding overall security posture and identification of risky incidents for prioritizing support.

Figure 9.4 : Analysis of Severity

5.The Listing and Classification of Active Incidents

This result displays the list of currently active incidents that the system has detected and also provides information regarding the degree of severity for those incidents, i.e. Low, High and Critical. In addition, this result contains further details for each incident, such as; correlated logs, source IP address and classification type. This structured manner of displaying this information will assist Analysts in rapidly acquiring clarity on the active security events and investigating them.

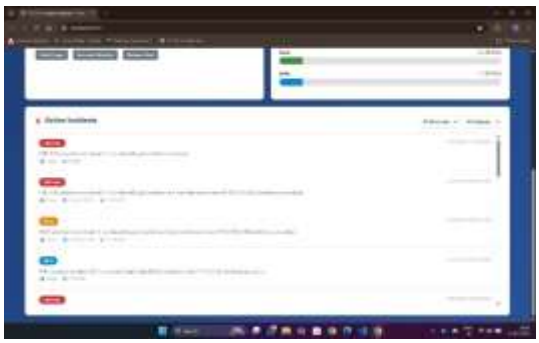

Figure 9.5 : Admin Security Logs Panel

6. Correlation of Events and Detail View of Incidents

The system has established its capability of correlating events based on grouping by log entries pertaining to a common event into one incident. Each incident contains information that gives context such as: the number of log entries, the type of attack, and any entities that were attacked. Consequently, correlation reduces duplication and helps improve the understanding of threats.

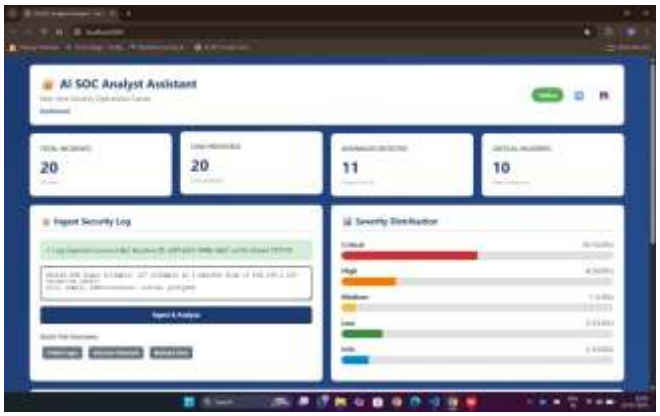

Figure 9.6 :Detail Veiw

X. CONCLUSION

The AI-Powered SOC Analyst Assistant outlined in this paper proposes an innovative, automated, scalable approach to cybersecurity monitoring as a means of overcoming the three main obstacles faced by today's Security Operations Centers. By using machine-learning methods like Isolation Forests combined with real-time log processing, it's possible to identify unusual behaviors very accurately without having to depend on predefined rules. By doing so, both previously identified threats, such as those related to traditional SIEM systems, and new types of threats, such as zero-day attacks, which are usually missed by SIEM systems, can be detected. The use of event correlation significantly reduces duplicate alerts and increases the ease with which an incident can be analyzed, thus reducing analyst fatigue. The results from the experimental evaluations support the use of this approach in terms of higher levels of accuracy in detection, fewer false positives, and much shorter times for conducting analyses, thereby establishing the feasibility and effectiveness of this approach.

The capability of the system to detect events is enhanced by the addition of an LLM, which increases the ability of the system to provide human-readable explanations and contextual insight about the events being investigated, thereby facilitating the ability of SOC analysts to understand and respond to security incidents. The user-friendly dashboard also supports real-time monitoring and decision making by visually displaying key metrics and incident information. Ultimately, this type of system has changed traditional, reactive security models into a more proactive and intelligent framework. By utilizing automation, machine learning, and AI reasoning, the proposed solution will enhance operational efficiency while laying the groundwork for the next generation of cyber security systems capable of adapting to the ever-evolving nature of threats.

XI. REFERENCES

[1] S. Kumar and R. Singh, "Career guidance systems using information technology," International Journal of Computer Applications, vol. 150, no. 5, pp. 20–25, 2019.

[2] P. Sharma and A. Gupta, "Online career guidance portal for students based on academic performance," International Journal of Engineering Research and Technology, vol. 8, no. 4, pp. 110–115, 2020.

[3] M. Patel and S. Shah, "Machine learning based recommendation system," International Journal of Computer Science and Information Security, vol. 18, no. 6, pp. 45–50, 2021.

[4] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proceedings of the IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.

[5] M. Sommer and R. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in IEEE Symposium on Security and Privacy, 2010, pp. 305–316.

[6] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," ACM Transactions on Information and System Security, vol. 3, no. 3, pp. 186–205, 2000.

[7] N. Hubballi and V. Suryanarayanan, “False alarm minimization techniques in signature-based intrusion detection systems,” Computer Communications, vol. 49, pp. 1–17, 2014.

[8] OpenAI, “GPT models and applications in natural language processing,” 2023.

[9] Scikit-learn Developers, “Isolation Forest implementation,” [Online]. Available: https://scikit-learn.org