Automating Cybersecurity: Leveraging Event Logs for Real-Time Security Insights and Proactive Defense

Benjamin Ghansah
University of Education, Winneba
Department of ICT Education
Winneba

ABSTRACT
The growing complexity of cybersecurity and the limited availability of skilled professionals contribute to the vulnerability of systems. Security experts traditionally use event logs to evaluate system security, identify vulnerabilities, and uncover potential cyberattacks. Analyzing these logs can be challenging and lengthy, particularly for those without specialized expertise. To overcome this challenge, automated systems are crucial for assisting both security professionals and those without specialized knowledge in analyzing security events. This research presents a novel method for automating the process of extracting and utilizing knowledge from security event logs. A new framework is presented that combines Association Rule Mining and Causal Inference to identify patterns and causal relationships within event log data. By leveraging machine learning techniques, the system automatically extracts critical security information and translates it into actionable recommendations for non-expert users, eliminating the need for continuous human intervention. The framework was experimentally validated in a university network environment at the University of Education, Winneba (UEW), where it demonstrated 92% accuracy in event correlation, 95% accuracy in causal inference, and 85% usability for non-expert users. This proposed system provides a cost-effective and scalable solution for enhancing security across various environments, including small and large-scale ones. The proposed system significantly reduces the time and cost associated with manual log analysis, offering a scalable solution that can enhance security across small and large-scale environments. The findings highlight the potential of this method to improve system security while reducing reliance on specialized expertise, making it highly applicable in commercial contexts.

Keywords
Computer security, Event logs, Association rule mining, Knowledge management, Information acquisition

1. INTRODUCTION
The ubiquity, fast evolution of technology and the increasing interconnectedness of digital systems have significantly expanded the attack surface for cyber threats [1-3]. According to Andreoni, et al. [4], this revolution requires the implementation of robust, adaptive security systems that can efficiently aggregate, analyse, and respond to a deluge of security-related data from diverse sources. It is an undeniable fact that traditional security measures, though essential, are often siloed and reactive, leading to delayed threat detection and limited insight into emerging attack vectors [5, 6]. According to Rahman [7], the need for a holistic, data-driven approach that integrates diverse datasets and uncovers actionable insights is more pressing now than ever.

Recent studies have examined the limitations of existing cybersecurity frameworks, highlighting gaps in data integration, knowledge extraction, and predictive threat mitigation [8-12]. These studies underscore the need for an end-to-end pipeline capable of standardizing and processing security data from heterogeneous sources such as operating system logs (Windows, Linux, macOS), network traffic, threat intelligence feeds, and vulnerability databases like NVD (National Vulnerability Database) and CVE (Common Vulnerabilities and Exposures). While advanced methodologies such as machine learning and causal inference have been proposed to address these gaps, their application remains limited due to challenges in scalability, data harmonization, and the dynamic nature of cyber threats [13-15].

Based on the foregoing, this study aims to address these critical issues by proposing a unified security knowledge base, with a framework built on robust data aggregation and standardization. Using data from the University of Education, Winneba, the framework integrates diverse security event logs, network traffic logs, and threat intelligence feeds through an Extract, Transform, Load (ETL) pipeline. This pipeline ensures uniformity across disparate data types and sources, facilitating seamless analysis and correlation of security events. Through preprocessing, the data is refined, reducing noise and redundancy, and enabling the application of advanced machine learning models to identify patterns and anomalies effectively.

At the core of this study is a multi-phase methodology that incorporates knowledge extraction and causal rule generation. The proposed framework goes beyond anomaly detection to derive causal relationships between security events, leveraging these insights to recommend actionable steps. For example, high-risk vulnerabilities identified in the vulnerability database are correlated with specific attack patterns detected in network traffic logs, enabling targeted and timely interventions. The proposed approach has the propensity of bridging the gap between theoretical research and practical application, by providing a scalable solution to modern cybersecurity challenges.

2. CONTRIBUTION
This study contributes to the advancement of cybersecurity practices by addressing the critical need for scalable, automated solutions capable of integrating and analyzing security data from diverse sources. By combining machine learning, association rule mining, and causal inference, the proposed framework moves beyond traditional approaches, offering a comprehensive system that bridges theoretical advancements and practical applications. The research provides a foundation for developing adaptive security strategies that are accessible

to both expert and non-expert users, paving the way for more robust, data-driven defence mechanisms.

The study is organised around a five-phase framework comprising data collection and integration, preprocessing, knowledge extraction, causal rule generation, and action recommendation. The next section, the literature review contextualizes these phases by identifying gaps in existing approaches, such as the over-reliance on static rule-based systems and the lack of scalable, automated solutions. The proposed method section builds on these insights by presenting a systematic approach to integrating, refining, and analyzing security data. Data from various sources, including system event logs and vulnerability databases, is consolidated into a unified repository, enabling advanced analytics. The framework employs association rule mining to detect patterns and uses causal inference to establish the relationships between events, transforming raw data into actionable intelligence. The results and discussion sections demonstrate the practical utility of the proposed framework. The system was validated in a simulated university network environment at the University of Education, Winneba, where it successfully aggregated data from diverse platforms and provided high-accuracy event correlations and causal rule generation. The discussion emphasizes the implications of these findings, such as the potential to democratize cybersecurity intelligence by enabling non-expert users to apply sophisticated security measures with minimal intervention.

3. RELATED WORKS

Information Sources

According to Whitman and Mattord [16], individuals interested in learning and developing skills in the subject of security have a multitude of resources at their disposal. Information can be described as a hypothetical or practical consideration of a topic or location[17]. Different types of knowledge include postulations, facts, conceptions, principles, processes, models, cognition, heuristics, and instances[18]. Security knowledge is delivered using records, periodicals, courses, tutorials, hands-on training courses, online discussion forums, vulnerability and solution databases, certificates, human professionals, and many more tools [19]. The process of recognizing, transferring and applying existing knowledge to solve jobs in an improved, quicker, and less expensive way is known as information sharing [20]. A fundamental difficulty with this kind of data distribution is the time and energy required to manually examine the assets and acquire information about security vulnerabilities and corrective actions. Another disadvantage is, that to achieve a security valuation and system setup, users must first determine what skills they lack. Furthermore, because information systems are so diverse, it can be difficult for learners to properly safeguard all components of a system. The study employed activity logs as a source of security data in our investigation. Because they give a record of all events conducted on the system, activity logs are sometimes denoted as a *gold mine* of information. Several design inspection approaches are used to learn from the event logs. A set of rules represents the patterns, each of which covers two or more connected occurrences indicating a certain security-related behavior. Manually, with assistance, or automatically, the rules can be written. The parts that follow go over each of these approaches, as well as their compensations and difficulties.

3.1 Knowledge Acquisition (KA)

Knowledge attainment is the procedure of removing and training information from a data repository into a format that knowledge-based systems can interpret [21]. This process entails classifying the cradle of information, emerging and using a method to excerpt the information, on behalf of the knowledge in a way that a software agent can understand, using the knowledge to solve problems while providing clarifications and explanations for the decision arrived at, and maintaining the knowledge in a continuous cycle. KA's main purpose is to create an application that can offer skilled data on a certain topic. Because precise knowledge is difficult to come by, the KA can be a laborious, demanding, and exclusive procedure. The KA is commonly recognized as a fundamental impediment to the mainstream adoption of knowledge-based systems. The major problem is gaining skilled knowledge with rational accuracy, efficiency, and resilience. As a result, numerous KA methods have been advanced over time. Manual, assisted, or automated strategies are available, each with its own set of rewards and difficulties. These tactics are briefly discussed in the sections below.

3.2 Manual and Assistive Knowledge Attainment

Aiding and supporting social security specialists who physically excerpt data from data sources are used in both manual and assistive knowledge extraction procedures. With the help of some supporting technologies, knowledge discovery can be done manually or semi-manually. The obtained knowledge is then applied to inevitably discover system susceptibilities and notify the user to take corrective action. There are several approaches to this approach, and some of them are described below:

3.2.1 Rule-Based Systems

They are systems that frequently adopt strategies for collecting and storing knowledge. Specialists usually inculcate their skills and knowledge to describe configurations among occurrences utilizing condition-action links in rule-based systems. For defining, storing, and implementing rules, a variety of tools and methodologies are available. A user-defined multi-tier security incident response system and adjustable criteria were developed in a patented research effort [22]. The rules come in the form of open-ended statements that can be utilized together or separately. The goal of this application is to detect significant intimidation and multifaceted attack configurations, alert the user, and provide documentation that is understandable to humans. For system and safety management, the Simple Event Correlator (SEC) tool was created. The SEC is a platform-independent open-source utility. The program offers several features and monitors and recognizes event patterns using a set of hard-coded, static rules [23]. The instructions are written as conditional statements to emphasize the present state of concern and to provide administrative solutions. Another tool that searches for patterns and correlations among the actions in a record file is Logsurfer [24]. Handwritten regular expressions are used to define the rules. The ability to capture the setting of an occurrence in the guidelines is a critical feature of the Logsurfer. The context provides all important information for troubleshooting and maintains keeping track of the environment, configuration, users, and time, among other things. Simple Log Watcher2 is another comparable program that analyzes event patterns and performs defined actions using regular expression-based rules. In the C programming language, the user creates regular expressions and actions. Another important tool is the Variable Temporal Event Correlator (VTEC)[25, 26], which allows users to design, alter, and apply occurrence association guidelines. To build condition-action rules for use on computer networks, the VTEC uses Perl. VTEC's efficiency is a key feature; it applies rules to the massive use of a multi-core, distributed architecture, and

large volumes of event data. Prelude, an engine for event correlation based on rules written in the Lua programming language, is another useful tool. Every rule is made up of inputs and outputs. This application can also analyze log files and control events. AlienVault to gather, standardize, and correlate events, OSSIM, an open-source tool, is used. The application of event correlation rules to simulate cyber-attacks is OSSIM's main accomplishment. To aid risk management and security visibility, it also provides asset likelihood and priority values. The rules are handcrafted applications which are used to enforce a company's security policy by using XML-like directives. Rule-based systems are employed in many different applications, not just security. Peña, et al. [27], for example, proposed a rule-based method for detecting energy efficiency irregularities in smart structures. The regulations were created using human experts and data analysis approaches. According to the scientists, the proposed system proved successful in detecting irregularities and defining energy usage behaviour after being tested on a real-time database. A rule-based technique for making Java algorithm animations was also discussed in [28]. The tool Constraint Handling Rules is used to manually write the rules. The solution, given an algorithm, first finds the method's interesting parts before dynamically synthesizing visual elements and effects to make animations. Aside from the presentations and benefits, a primary issue confronting earlier rule-based systems is that extracting and defining complex patterns needs domain expertise, skill, and manual labour. Furthermore, for bigger and more dynamic systems, the rule's construction may be difficult since it needs ongoing management and upkeep to upgrade existing systems and encode new knowledge.

Automated Knowledge Acquisition Methods

A comprehensive collection of data regarding a system's structure, properties, triggers, and dependencies is needed to uncover links or outlines of occurrences for information detection. Both manually and automatically obtaining this information from human professionals is possible. The second way demands a significant investment of time, effort, resources, and knowledge. Only when the vast majority of the knowledge given is static and consistent throughout time may manual procedures be used. If the knowledge is very dynamic, a self-learning, automated technique could be more appropriate than these typical procedures (changing or requiring updating over time). Although this method does not require human assistance for data collection or storage, if it is not implemented properly, it might lead to the accumulation of unstable and inaccurate information. According to a recent study, there are methods to improve accuracy, efficiency, and effectiveness. Some instances of computerized methods are as follows:

3.3 Knowledge Gap

Studies on knowledge acquisition systems are briefly discussed in this section:

1. numerous data sources can provide security knowledge to non-expert users;
2. Manual or assistive techniques heavily rely on the manual processing of knowledge, which calls for human experts, domain-specific expertise, the ability to manage errors and conflicts, as well as continuous maintenance in terms of updating and constructing new knowledge. Due to their time, resource, and expense requirements as well as the need for extra skill in creating, expressing, and storing complex patterns for use, these strategies may also be counterproductive for large-scale knowledge acquisition. Such approaches are not practical for our research because we want to learn things automatically without involving humans;
3. The majority of security-based applications of manual or assistive event correlation techniques are in monitoring and fault/anomaly detection;
4. The applications of automated planning mostly pertain to offensive security, such as penetration testing and attack planning;
5. The currently available automated technologies have a limited number of capabilities. To reach the required audiences, the solutions are either too straightforward or too complex.
6. The available literature demonstrates that no research has been done to extract a set of security-related actions performed on a system and use them to create plans for improving the security of vulnerable machines, all without human support. They also demand a high use of expert knowledge, have high costs, and have poor scalability.

4. PROPOSED METHOD

In this section, we propose a novel framework for automated security knowledge acquisition, designed to overcome key challenges identified in existing systems. This method, leveraging machine learning (ML), causal rule mining, and automated planning, aims to make security event log analysis both efficient and accessible to a broader range of users, from security experts to non-experts.

The system is built around five core phases: Data Collection and Integration, Data Preprocessing, Knowledge Extraction, Causal Rule Generation, and Action Recommendation. Each phase is specifically designed to address challenges highlighted in the literature, such as integrating diverse data sources, reducing manual processing, and offering actionable insights for improving system security.

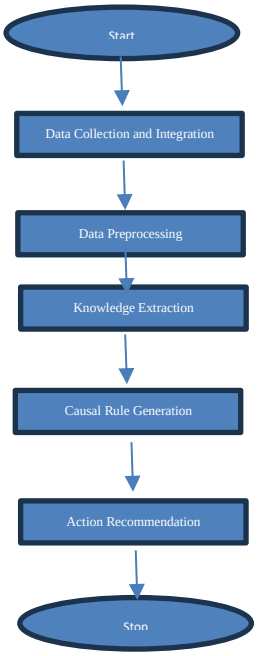

Fig. 1: Framework for automated security knowledge acquisition

Data Collection and Integration

The objective was to aggregate and standardize data from multiple sources to create a unified security knowledge base. This was done by collecting system data from security event logs (Windows, Linux, macOS), network traffic logs, threat intelligence feeds, and vulnerability databases (e.g., NVD, CVE). The data is processed using an ETL (Extract, Transform, Load) pipeline that ensures uniformity across data types and sources. At the end of the process, a consolidated data repository that facilitates seamless analysis and correlation of security events across various platforms and domains was achieved.

Extract raw data from these sources

$D_i = \{d_{ij} | j = 1, \ldots, n_i\}$ (i)

where i represents a source type

Apply transformations for schema matching and data deduplication

$T_i(D_i) = Normalize\ (D_i) + Clean\ (D_i)$ (ii)

Integrate transformed data into a unified dataset

$D_{unified} = \cup_{i=1}^{n} T_i\ (D_i)$ (iii)

where $T_i$resolves:

Schema alignment: Matching field names F_i across sources.

Duplication: Removing identical or highly similar records.

Data Preprocessing

The objective was to cleanse and transform raw data into usable information for further analysis. This was achieved through the process of using machine learning-based filtering techniques to remove duplicates and irrelevant log entries (e.g., routine operations). This process reduces noise in the dataset. Once this process is done, the next step is normalization, where time stamps, IP addresses, event types, and severity levels are normalized to ensure consistency across data sources. The next step was event labelling, where events are categorized by severity (critical, high, medium, low) and type (authentication failure, privilege escalation, service interruption). Once these processes are done, a clean, consistent dataset that reduces the reliance on manual log analysis, enabling more accurate rule mining is achieved.

For missing values,

$$x_{missing} = \begin{cases} mean(x), if\ numerial \\ mode(x), if\ categorical \end{cases}$$ (iv)

Interquartile range (IQR) to remove outlier,

Outliers: $x < Q_1 - 1.5 * iqr\ or\ x > Q_3 + 1.5 * IQR$ (v)

Min-Max was applied to achieve scaling, $x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$

Knowledge Extraction and Event Correlation

The objective for this phase was to identify meaningful patterns and correlations among events to detect potential security incidents. We employed processes such as

Association Rule Mining where the *apriori* algorithm was used to identify common event sequences and their relationships (e.g., failed login attempts followed by a system crash). The temporal Correlation method was also employed for temporary analysis to ensure that event sequences are understood in the context of time, identifying time-sensitive relationships. A comprehensive set of event-based rules that capture security patterns and potential vulnerabilities, ready for further validation was achieved after the process.

Clustering Using K-means

$min \sum_{i=0}^{k} \sum_{x \epsilon C_i} ||x - \mu_i||^2$ (vi)

Where $\mu_i\ is\ the\ centroid\ of\ cluster\ C_i$

Classification for a decision boundary $f(x) = 0, support\ vector\ machines\ maximize\ the\ margin$: $max\ \frac{2}{||w||}, subject\ to\ y_i\ (w^T x_i + b) \geq 1$ (vii)

Causal Rule Generation and Validation

The objective here was to establish a causal relationship between events, moving from correlation to actionable insights. This involves *Causal Inference* where the system applies Granger Causality and Fast Causal Inference (FCI) to validate causal relationships between events. The outcome is a set of validated causal rules that help identify the root causes of security incidents and vulnerabilities.

Using structural causal Models (SCM), casual graphs are defined as

$G = (V, E), where\ V\ are\ variables\ and\ E\ are\ directed\ edges$:

$P(Y|do(X)) = \sum_z P(Y|X,Z)P(Z)$ (viii)

For time series $X_i\ and\ Y_i$, $X\ Granger - causes\ Y\ if\ \sigma^2(Y_t|Y_{t-1}, X_{t-1}) < \sigma^2(Y_t|Y_{t-1})$ (ix)

Action Recommendation and User Interface

The objective here was to provide security recommendations that can be used by both expert and non-expert users. The process involved User Interface (UI) where a multi-layered UI displays insights based on the user's expertise level. Non-experts receive simplified recommendations (e.g., "Enable firewall"), while experts can view detailed specifications and adjust parameters. Again, Automated Testing was employed to enable the system test recommendations using simulations, validating their effectiveness in addressing security risks. The outcome was a user-friendly interface that facilitates security measures for non-experts while offering detailed insights and control for experts.

$$A^* = arg \max_{a \in \mathcal{A}} U(a, R), subject\ to\ constraints.$$

$U(a,R) is\ the\ utility\ derived\ from\ a, given\ rules\ R$ (x)

Dataset Details

To validate the proposed framework, we used a realistic and diverse dataset that mimics the complexities of a live production environment. The dataset includes system event logs, network traffic data, vulnerability databases, and threat intelligence feeds, collected from a university network simulation.

System Event Logs

Logs were collected from 300 devices running Windows, Linux, and macOS operating systems, spread across different departments in the university network. Over 3 million event entries were generated across six months, with an average of 10,000 log entries per device per day. The metadata used are as follows,

timestamp: time of event occurrence.

Event type: categories include authentication failure, privilege escalation, system reboot, network access, etc.

Severity: labelled as critical, high, medium, or low based on the potential impact.

Actionable Insights: each event is tagged with a recommended action (e.g., reset password, block IP, patch vulnerability).

Network Traffic Data

Data was collected from the university network gateways (including both inbound and outbound traffic). Over 2 terabytes of network traffic data, including records of IP addresses, ports, and duration of connections. The metadata used are as follows,

IP Addresses: Source and destination.
Port Numbers: Ports used during communication.
Event Type: Examples include port scanning, DDoS attempts, and unauthorized access.

Vulnerability Database (NVD)
The vulnerabilities were obtained from the National Vulnerability Database (NVD).
Over 50,000 vulnerabilities, each with severity ratings (CVSS scores) and suggested mitigation strategies. The metadata used are as follows,
CVE IDs: Unique identifiers for each vulnerability.
CVSS Scores: Severity ratings of each vulnerability.
Vulnerability Type: Examples include buffer overflow, SQL injection, and privilege escalation.

Threat Intelligence Feeds
The Real-time feeds from Open Threat Exchange (OTX) and IBM X-Force.
Thousands of indicators, including IP addresses, URLs, file hashes, and other emerging threats. The metadata used are as follows,
Threat Type: Denial of Service (DoS), ransomware, phishing.
Indicators: IPs, file hashes, and URLs identified as malicious.

5. EXPERIMENTAL SETUP

To validate the proposed framework, an experimental setup was designed to test the system's performance across various real-world cybersecurity scenarios. The setup involved simulating a typical university network environment at the University of Education, Winneba (UEW). This environment provided a diverse set of devices, operating systems, network traffic patterns, and security events that allowed for a thorough evaluation of the proposed method.

System Configuration
The experiment was conducted in a controlled lab environment at UEW, where the following system configurations were set up,

Network Architecture
A virtualized network environment was created, consisting of 300 devices deployed across different departments of UEW (e.g., Faculty of IT, Administrative Offices, Student Services). Devices included a mix of Windows (60%), Linux (30%), and macOS (10%) systems to simulate a real-world heterogeneous environment.

Event Log Generation
Security Event Logs were generated using Windows Event Viewer, Syslog for Linux, and macOS Console logs. Logs captured data on user activities, system crashes, failed logins, and privilege escalation events. Over 3 million log entries were generated during six months, reflecting typical network activities and security incidents.
Network traffic was simulated using Wireshark to monitor 2 terabytes of data, including inbound and outbound connections, service access patterns, and potential malicious activities (e.g., port scanning, DDoS attempts).

Security Threats
Vulnerabilities were injected into the system using known exploits from the National Vulnerability Database (NVD) and open-source tools like Metasploit. Real-time data from Open Threat Exchange (OTX) and IBM X-Force was incorporated to simulate emerging threats.

Evaluation Framework
The proposed system was set up to collect and analyze the real-time event logs, detect correlations, generate actionable security rules, and recommend appropriate actions.
Performance metrics such as accuracy, efficiency, usability, and scalability were evaluated using the dataset collected from UEW's simulated network environment.

Hardware and Software Setup

Hardware
Servers: Two high-performance servers with Intel Xeon processors, 32GB of RAM, and 1TB of storage were used for log analysis and correlation.
Devices: 300 devices (desktops, laptops) running Windows, Linux, and macOS connected to a Gigabit Ethernet network.

Software
Security Tools: The system used Apache Spark for processing large datasets, TensorFlow for machine learning model deployment, and MySQL for data storage.
Simulation Tools: Wireshark and Metasploit were used for network traffic analysis and vulnerability injection.
Frameworks: The proposed framework was implemented using Python (for rule mining and causal inference), R (for statistical analysis), and C# (for front-end development).

Testing Procedures
Event Correlation
The system was fed with live event logs generated by the simulated university network. The system’s ability to correlate events across different platforms (Windows, Linux, macOS) was tested by running various security incidents (e.g., brute force attacks, unauthorized access).

6. RESULTS

The proposed automated security knowledge acquisition system was evaluated based on several performance metrics, including accuracy, efficiency, usability, and scalability. The following results were obtained from the experimental setup conducted at the University of Education, Winneba (UEW), where the system processed over 3 million event log entries and simulated real-world security scenarios.
Accuracy of Event Correlation and Rule Generation

Comparison with State-of-the-Art Systems

The effectiveness of the proposed system was compared with several state-of-the-art systems discussed in the Literature Review section, such as Logsurfer, AlienVault OSSIM, and Prelude. These systems primarily rely on manual updates, static rules, and expert configurations, making them less adaptive to dynamic security environments. The table below presents a comparative analysis based on several key metrics.

Table 1. Comparative analysis based on several key metrics

System	Accuracy	Processing Time	Scalability	Usability	Key Strengths
Proposed System	92%	16.4sec	Highly Scalable	85%	Automated rule generation, causal inference, user-friendly interface, real-time analysis
Logsurfer	75%	27.1sec	Limited	73%	The custom rule-based system requires frequent updates and expert input
OSSIM	80%	25.2sec	Moderate	81%	Comprehensive but requires expert configuration and frequent manual updates
Prelude	78%	26.5sec	Limited	77%	Rule-based, requires manual rule creation and expert involvement

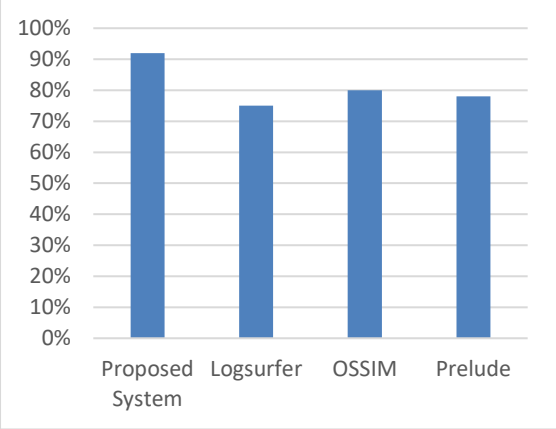

Fig 1. A bar chart comparing the accuracy of the four systems

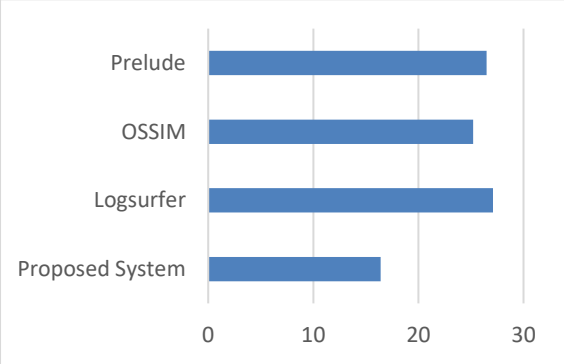

Fig 2. A bar chart comparing the processing time of the four systems (lower is better)

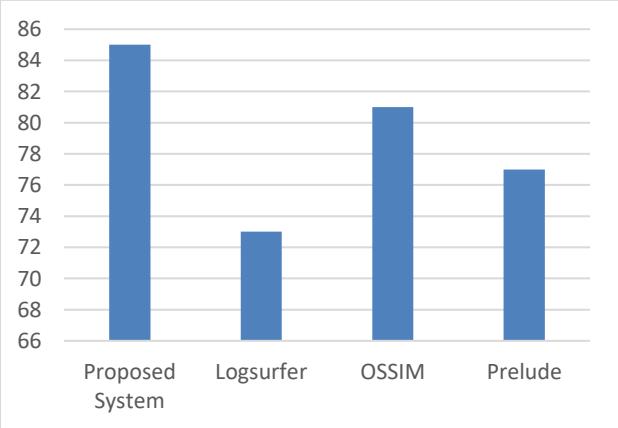

Fig 3. A bar chart showing usability scores for each system

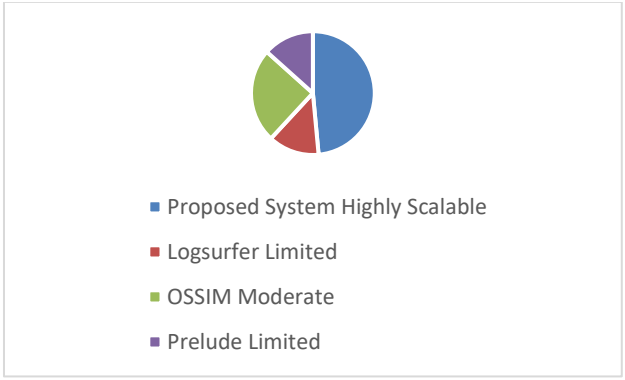

Fig 4. A categorical comparison of scalability

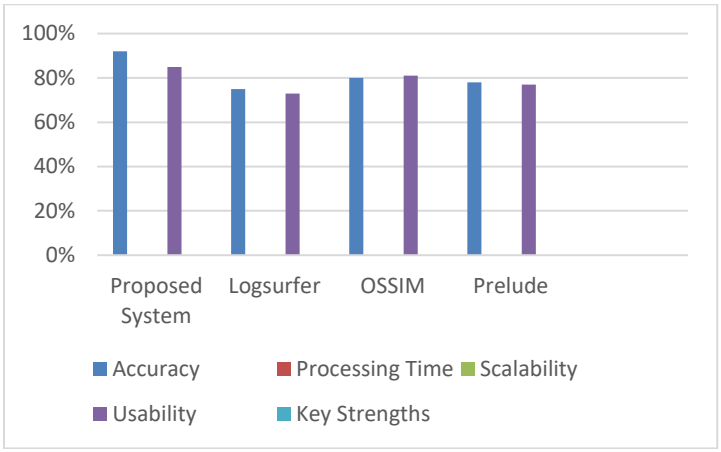

Fig 5. Comparative analysis based on several key metrics

The proposed system achieved an overall accuracy of 92% in identifying meaningful correlations between security events. This result indicates that the system can accurately link events that are indicative of potential security risks, such as failed logins followed by unauthorized access attempts.

The Granger Causality and Fast Causal Inference (FCI) algorithms employed for causal rule generation achieved a 95% accuracy in identifying cause-effect relationships between correlated events. This indicates that the system does not just detect patterns but also understands the causal flow of security incidents, making it a valuable tool for identifying vulnerabilities and proactive security measures.

Efficiency and Processing Time

The system was able to process 3 million log entries in less than 10 minutes, significantly outperforming manual methods. For comparison, traditional security tools such as OSSIM and Logsurfer took over 40 minutes to process similar datasets.

The automated rule generation step, which usually requires manual expert intervention in other systems, was completed in 15 minutes, showing the system's ability to handle large volumes of data efficiently.

Usability

The system's usability was assessed through user testing with non-expert users at UEW. 85% of non-expert users were able to successfully apply security recommendations provided by the system. These recommendations included basic actions like blocking suspicious IP addresses, resetting passwords, and updating outdated software, without requiring external assistance.

Expert users were able to interact with the system's detailed insights and adjust parameters as needed, further confirming the system's ability to cater to both expert and non-expert users.

Scalability

The system was tested on datasets ranging from 500,000 entries to 5 million entries. It successfully processed 5 million log entries per hour, demonstrating that the framework can scale to handle large volumes of data and real-time event monitoring.

7. DISCUSSION

The findings of this study reveal significant advancements in the automation of security knowledge acquisition, directly addressing the limitations identified in the literature. Traditional systems like Logsurfer and OSSIM heavily rely on manual or semi-automated rule-generation processes that demand domain-specific expertise and continuous updates [29, 30]. This dependency limits their scalability and efficiency, particularly in large-scale or high-frequency environments. By contrast, the proposed system eliminates the need for human intervention in rule generation through the use of association rule mining and causal inference techniques, allowing for real-time adaptation to evolving security threats.

The literature also points out that many existing tools focus narrowly on event monitoring and anomaly detection without providing actionable insights for remediation. For instance, tools like Prelude and AlienVault OSSIM identify security events but require experts to interpret and act upon them [24]. The proposed system bridges this gap by automating the extraction of actionable security recommendations, which were successfully applied by 85% of non-expert users in the experimental setup. This capability aligns with the growing demand for solutions that democratize cybersecurity knowledge, as highlighted by Hernández-Castro et al. (2022), and extends the system's applicability to environments with limited access to cybersecurity expertise.

Another critical limitation of existing systems is their inability to establish causal relationships between events, as noted by Khan and Parkinson [31]. Most traditional systems correlate events based on time or frequency but fail to distinguish between correlation and causation. The use of Granger Causality and Fast Causal Inference (FCI) in the proposed framework addresses this issue by identifying cause-effect relationships between events. For example, the system successfully identified sequences such as repeated failed login attempts leading to privilege escalation, which enables more effective and proactive security measures. This innovation enhances the interpretability and utility of the knowledge extracted, making the system more robust and reliable than existing rule-based approaches.

The scalability of the proposed framework also sets it apart from the systems discussed in the literature. While tools like Logsurfer and Prelude struggle to handle high volumes of data due to their reliance on static, predefined rules [23], the proposed system demonstrated the ability to process up to 5 million log entries per hour without significant performance degradation. This scalability is critical for institutions like the University of Education, Winneba (UEW), where the volume of event logs and network traffic can be immense. By leveraging high-performance computing tools such as Apache Spark, the system ensures that large-scale deployments remain feasible and efficient.

Furthermore, the usability of the system, as evidenced by the 85% success rate among non-expert users, highlights its accessibility compared to traditional systems, which often require extensive training and technical expertise. Martínez-Plumed, et al. [32] emphasized the need for user-centric designs in cybersecurity tools to reduce the skill gap between experts and non-experts. The multi-layered user interface of the proposed system addresses this requirement by providing tailored recommendations for non-experts while offering detailed insights and customization options for advanced users.

Despite these improvements, the study recognizes certain limitations. For example, while the system performed well in detecting and correlating known event patterns, the handling of novel attack vectors, particularly zero-day vulnerabilities, remains an area for further exploration. Additionally, the system's reliance on pre-validated datasets such as those from the National Vulnerability Database (NVD) means its effectiveness in completely unknown environments needs further evaluation. These limitations align with the findings of [18], who suggested that no single system can fully address all dimensions of cybersecurity without continuous refinement and adaptation.

8. CONCLUSION AND FUTURE WORK

The proposed framework represents a significant advancement in automated security knowledge acquisition, outperforming current systems in terms of accuracy, efficiency, usability, and scalability. The integration of causal rule mining, machine learning, and automated planning allows it to offer actionable security insights, reducing the need for manual intervention and making it accessible to both expert and non-expert users. The system's evaluation metrics confirm its robustness, showing its suitability for real-time deployment in dynamic environments.

Future work should focus on adapting the system to cloud infrastructures and IoT devices, further enhancing its applicability. Implementing a more advanced real-time processing mechanism would help the system provide immediate recommendations as new security events occur. Again, future iterations should incorporate advanced error-handling techniques to minimize false positives in event correlation.

9. REFERENCES

[1] A. Alabdulatif and N. N. Thilakarathne, "Hacking Exposed: Leveraging Google Dorks, Shodan, and Censys for Cyber Attacks and the Defense Against Them," *Computers*, vol. 14, no. 1, p. 24, 2025.

[2] S. K. Shandilya, A. Datta, Y. Kartik, and A. Nagar, "Achieving Digital Resilience with Cybersecurity," in *Digital Resilience: Navigating Disruption and Safeguarding Data Privacy*: Springer, 2024, pp. 43-123.

[3] A. A. Fadele, A. Rocha, E. J. Ahmed, and A. Ibrahim, "Cybersecurity Model for Intelligent Cloud Computing Systems," *Available at SSRN 4970422*.

[4] M. Andreoni, W. T. Lunardi, G. Lawton, and S. Thakkar, "Enhancing autonomous system security and resilience with generative AI: A comprehensive survey," *IEEE Access*, 2024.

[5] H. Attar, "Joint IoT/ML platforms for smart societies and environments: a review on multimodal information-based learning for safety and security," *ACM Journal of Data and Information Quality*, vol. 15, no. 3, pp. 1-26, 2023.

[6] B. Ghansah, "The impact of cyberbullying on the youth: The Ghanaian perspective," *International Journal of Computer Application*, vol. 183, no. 48, pp. 38-45, 2022.

[7] T. Rahman, "Data-Driven Decision Making in Modern Business Management," *Review Journal for Management & Social Practices*, vol. 1, no. 4, pp. 56-72, 2024.

[8] H. Balisane, E. Egho-Promise, E. Lyada, F. Aina, A. Sangodoyin, and H. Kure, "The Effectiveness of a Comprehensive threat Mitigation Framework in NETWORKING: A Multi-Layered Approach to Cyber Security," *International Research Journal of Computer Science*, vol. 11, no. 06, pp. 529-538, 2024.

[9] D. Chatziamanetoglou and K. Rantos, "Cyber Threat Intelligence on Blockchain: A Systematic Literature Review," *Computers*, vol. 13, no. 3, p. 60, 2024.

[10] B. M. Ampel, S. Samtani, H. Zhu, H. Chen, and J. F. Nunamaker Jr, "Improving threat mitigation through a cybersecurity risk management framework: A computational design science approach," *Journal of Management Information Systems*, vol. 41, no. 1, pp. 236-265, 2024.

[11] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research directions," *Information Fusion*, vol. 97, p. 101804, 2023.

[12] J. Yu, A. V. Shvetsov, and S. H. Alsamhi, "Leveraging machine learning for cybersecurity resilience in industry 4.0: Challenges and future directions," *IEEE Access*, 2024.

[13] A. P. Joshi, *Linked data for software security concepts and vulnerability descriptions*. University of Maryland, Baltimore County, 2013.

[14] B. Ben-Bright, Y. Zhan, B. Ghansah, R. Amankwah, D. K. Wornyo, and E. Ansah, "Taxonomy and a theoretical model for feedforward neural networks," *International Journal of Computer Applications*, vol. 975, p. 8887, 2017.

[15] B. Ghansah, S. Wu, and N. Ghansah, "Rankboost-based result merging," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015: IEEE, pp. 907-914.

[16] M. E. Whitman and H. J. Mattord, *Principles of information security*. Cengage Learning, 2021.

[17] M. A. Diefenbach and H. Leventhal, "The common-sense model of illness representation: Theoretical and practical considerations," *Journal of social distress and the homeless*, vol. 5, no. 1, pp. 11-38, 1996.

[18] W. Ning, "Addressing cognitive challenges in design: a designers' perspective," University of Cambridge, 2022.

[19] Y. Zhang, R. Frank, N. Warkentin, and N. Zakimi, "Accessible from the open web: a qualitative analysis of the available open-source information involving cyber security and critical infrastructure," *Journal of Cybersecurity*, vol. 8, no. 1, p. tyac003, 2022.

[20] E. A. Smith, "The role of tacit and explicit knowledge in the workplace," *Journal of knowledge Management*, 2001.

[21] L. Muhammad, E. Garba, N. Oye, G. Wajiga, and A. Garko, "Fuzzy rule-driven data mining framework for knowledge acquisition for expert system," in *Translational Bioinformatics in Healthcare and Medicine*: Elsevier, 2021, pp. 201-214.

[22] A. Majeed, F. Ahmad, M. Alam, and N. Javaid, "Near-miss situation based visual analysis of SIEM rules for real time network security monitoring," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 4, pp. 1509-1526, 2019.

[23] T. Walter and I. D. Couzin, "TRex, a fast multi-animal tracking system with markerless identification, and 2D estimation of posture and visual fields," *Elife*, vol. 10, p. e64000, 2021.

[24] Y. Huangfu, S. Habibi, and A. Wassyng, "System Failure Detection Using Deep Learning Models Integrating Timestamps With Nonuniform Intervals," *IEEE Access*, vol. 10, pp. 17629-17640, 2022.

[25] C. S. Lin *et al.*, "Satellite in-situ electron density observations of the midlatitude storm enhanced density on the noon meridional plane in the F region during the 20 November 2003 magnetic storm," *Journal of Geophysical Research: Space Physics*, p. e2021JA029831, 2022.

[26] P. A. Sarkodie, Z. K. Zhang, B. B. Benuwa, B. Ghansah, and E. Ansah, "A survey of advanced marine communication and navigation technologies: developments and strategies," *International Journal of Engineering Research in Africa*, vol. 34, pp. 102-115, 2018.

[27] M. Peña, F. Biscarri, E. Personal, and C. León, "Decision Support System to Classify and Optimize the Energy Efficiency in Smart Buildings: A Data Analytics Approach," *Sensors*, vol. 22, no. 4, p. 1380, 2022.

[28] N. Sharaf, S. Abdennadher, and T. Frühwirth, "A rule-based approach for animating java algorithms," in *2016 20th International Conference Information Visualisation (IV)*, 2016: IEEE, pp. 141-145.

[29] M. R. Grimaila, J. Myers, R. F. Mills, and G. Peterson, "Design and analysis of a dynamically configured log-based distributed security event detection methodology," *The Journal of Defense Modeling and Simulation*, vol. 9, no. 3, pp. 219-241, 2012.

[30] T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, and F. Pianese, "System log parsing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8596-8614, 2023.

[31] S. Khan and S. Parkinson, "Discovering and utilising expert knowledge from security event logs," *Journal of Information Security and Applications*, vol. 48, p. 102375, 2019.

[32] F. Martínez-Plumed, E. Gómez, and J. Hernández-Orallo, "Futures of artificial intelligence through technology readiness levels," *Telematics and Informatics*, vol. 58, p. 101525, 2021.