Modeling Time Series from Log Files: An ARIMA Approach for Security-Related Event Detection and Prediction

Marko M. Živanović
Academy of Technical and Art Applied Studies Belgrade,
The School of Electrical and Computer Engineering,
Belgrade, Serbia,
Email: markoz@gs.viser.edu.rs

Marjan Milošević
University of Kragujevac, Faculty of Technical Sciences
Čačak, Serbia
Email: marjan.milosevic@ftn.kg.ac.rs

***Abstract*— The analysis of time series log data provides insights into the dynamics of activities within information systems. This paper explores the application of the ARIMA model for predicting the number of security related events on multiple computer systems based on log data. Data from various environments were tested to assess the reliability and performance of the model in different work scenarios. The results demonstrate that the ARIMA model delivers accurate forecasts. The paper lays the groundwork for further application of the model in anomaly detection and the optimization of security procedures.**

Keywords - Log files; Security analysis; ARIMA; Information systems; Network activities; Anomalies detection; Security threats; Unauthorized access;.

I. INTRODUCTION

Log files represent a crucial element in security analysis and monitoring of information systems, providing a detailed insight into the activities occurring within networks, systems, and applications. These files record events such as login attempts, configuration changes, resource access, and potentially suspicious activities [1]. The analysis of such data enables not only the monitoring of system performance but also the detection of anomalies that may indicate security threats, such as unauthorized access attempts, malicious activities, or data breaches. In modern information systems, security risks are pervasive and diverse. One of the greatest challenges is the detection of sophisticated attacks such as ransomware, DDoS attacks, or phishing, which can severely compromise the integrity, confidentiality, and availability of data [2]. For instance, the failure to promptly identify anomalies in log files may allow attackers to exfiltrate sensitive information or sabotage the system. Additionally, log system overload, the generation of redundant data, and the lack of automation in log processing pose further risks to efficient monitoring and analysis. The assessment of risks associated with log data includes considerations such as the validity and integrity of logs (whether the data is accurate and whether there is a possibility of tampering), timeliness of processing (how quickly anomalies can be identified and reported), the efficiency of analytical methods (whether the models and algorithms used provide sufficiently precise results for decision-making), and resources required for analysis (the demands for storage and processing of large volumes of log data). For the analysis of time series data, algorithms such as ARIMA (AutoRegressive Integrated Moving Average) are commonly employed, as they enable the identification of key patterns and provide accurate predictions for future events [3]. This study focuses on the analysis of time series logs generated on different computers: a personal computer without authentication, a business laptop with working hours from 9 a.m. to 5 p.m., and a computer with flexible working hours. The logs were retrieved from Windows 10 and Windows 11 client operating systems using the Event Viewer program, specifically targeting the security logs of these systems. By employing the ARIMA model for time series analysis, the aim was to identify key patterns and predict future events within the system. Furthermore, this study aims to provide practical recommendations for enhancing security and optimizing monitoring procedures in information systems.

II. RELATED WORK

In the field of time series forecasting and anomaly detection, several studies have explored diverse models and approaches for predicting system behavior, including performance forecasting for operating systems and identifying irregularities within system logs. **Assi et al. (2021)** proposed using the ARIMA and LSTM models for predicting performance issues in Windows OS, such as memory fluctuations [4]. Their study highlighted the strengths of ARIMA in predicting system behavior with the lowest error rates compared to LSTM, suggesting its potential for real-time forecasting of system anomalies. **Lee et al. (2021)** compared the ARIMA model with a Logarithmic Return (LR) model for time series forecasting [6].The LR model was found to be computationally efficient, requiring significantly less CPU time than ARIMA, while still providing comparable accuracy in real-world applications. This work illustrates the ongoing search for more efficient methods in handling autocorrelated time series data, with a focus on reducing computational complexity. In the domain of smart grid energy forecasting, **Alberg and Last (2018)**

developed sliding window-based ARIMA algorithms to forecast hourly electricity load. Their approach, which integrates both non-seasonal and seasonal ARIMA models with online learning methodologies, offers insights into the application of ARIMA for real-time data analysis in energy systems [6]. **Sirisha et al. (2022)** explored the use of ARIMA, SARIMA, and LSTM models for profit prediction in financial time series forecasting. Their findings show that LSTM outperforms both ARIMA and SARIMA in prediction accuracy, underscoring the potential of deep learning models in dynamic financial environments [7]. **Siwach and Mann (2020)** conducted a systematic review on anomaly detection techniques for web log analysis. They evaluated multiple methods and identified the challenges faced by engineers in selecting and implementing effective anomaly detection techniques in large-scale systems. This work highlights the importance of automated log analysis in reducing manual work and improving detection efficiency [8]. In the field of security, **Park and Kim (2021)** investigated the use of Windows Event Logs for corporate security audits and malware detection. Their study emphasized the utility of event log analysis in identifying suspicious behaviors, including external storage connections and process creation, and proposed new tools for more effective forensic analysis of user actions [9]. **Keyogeg et al. (2021)** introduced an automated ransomware detection model that utilizes machine learning and log analysis to identify early-stage ransomware activity in Active Directory environments. Their approach, which integrates Random Forests and focuses on feature engineering, demonstrates significant improvements in detection accuracy and could contribute to the development of real-time monitoring systems for enterprise security [10].

III. Analysis of Log Files from Windows Event Viewer

In this section of the paper, we will focus on analyzing log files obtained from the Windows Event Viewer, specifically from security categories, with each log file having a size of exactly 20 MB for all three categories. The analysis includes log files from three different computers: a work computer with working hours, a home computer with user authentication enabled, and a home computer without user authentication. The work computer uses the Windows 10 operating system, while the other two computers use the Windows 11 operating system. The files are in .evtx format, which is the standard format for Windows Event Viewer, and they cover events related to security authentication and user account administration. To ensure meaningful results, the computers were used in controlled environments simulating real-world scenarios. The work computer was operated during standard business hours, performing typical office-related activities such as accessing shared drives, sending emails, and running productivity software. The home computer with authentication enabled was used for personal tasks, including browsing the internet, accessing cloud storage services, and running multimedia applications. The home computer without user authentication was intentionally left without login credentials and was used to simulate an environment where basic security measures are absent, allowing unrestricted access to the system. This setup aimed to model various levels of security and user interaction for analysis. In the context of security monitoring and system administration, various event categories provide crucial insights into the system's health and the potential risks it may face. User Account Management involves the creation, modification, or deletion of user accounts, which can indicate unauthorized changes or potential breaches if unmonitored, especially when performed outside normal working hours. Logon and Special Logon events track user access to the system, with special logons often indicating privileged access, which, if misused, can pose a high risk. Security Group Management refers to changes in user group memberships, a critical area that, if improperly handled, can allow unauthorized users escalated privileges. Other System Events typically cover less specific system activities that may still require attention if they suggest unusual behavior or failures. Audit Policy Change logs are significant as changes to auditing policies may disable important security monitoring, increasing risk. System Integrity events focus on the health and configuration of the system, with failures potentially signaling vulnerabilities or system tampering. Logoff events are generally low-risk but can help identify abnormal session behaviors, like users not logging off properly after work. Process Creation tracks the initiation of system processes, which is vital for detecting malware or unauthorized applications running on the system. Security State Change logs indicate modifications to the security state of the system, such as firewall or antivirus settings, and these may highlight malicious activities attempting to disable protections. Other Policy Change Events track changes to system policies, which could alter security configurations and create vulnerabilities. Service Shutdown events identify system services being stopped, which can disrupt normal operations or be a precursor to attacks [11, 12].

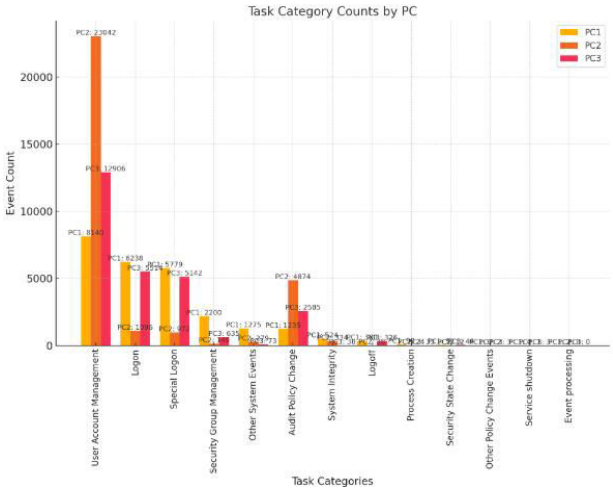

Fig 1. Task Category Counts by PC

The analyzed log files contain numerous Event ID records representing specific events and actions on the computers, such as logons, logoffs, security policy changes, and other activities related to user accounts and system integrity. On the first computer, running Windows 10, the most common events are related to **User Account Management** (12906), **Logon** (5514), and **Special Logon** (5142), with a smaller number of activities related to **Security Group Management** (635) and **System Integrity** (38). A similar pattern is observed on the Windows 11 computer, where events related to **User Account Management** (23042) and **Audit Policy Change** (4874) dominate, while events related to **Security Group Management** (149) and **System Integrity** (334) are also present but in smaller numbers. On the third computer, running Windows 11, the dominant events include **User Account Management** (8140) and **Logon** (6238), with a significant number of events related to **Security Group Management** (2200) and **System Integrity** (524). In addition to the basic analysis of events by Event ID, the differences in the frequency of certain activities between computers with different operating systems were considered. For example, the Windows 10 computer recorded a lower number of events in the **Audit Policy Change** category (2585), while the computers with Windows 11 had a significantly higher number of similar events, indicating differences in system security policy settings and activities. Additionally, it was noted that the number of **Security Group Management** activities was higher on the Windows 11 computers (149 and 2200) than on the Windows 10 computer (635), which may suggest different approaches to user group administration or additional security settings.

TABELA I. SUMMARY OF TASK CATEGORIES BY PC AND THEIR DIFFERENCES

Task Category	PC1 Count	PC2 Count	PC3 Count	Differences
User Account Management	8140	23042	12906	PC2 >>
Logon	6238	1096	5514	PC1 >>
Special Logon	5779	972	5142	PC3 >>
Security Group Management	2200	149	635	PC1 >>
Other System Events	1275	279	73	PC1 >>
Audit Policy Change	1235	4874	2585	PC2 >>
System Integrity	524	334	38	PC1 >>
Logoff	360	38	326	PC1 >>
Process Creation	90	24	33	PC1 >>
Security State Change	52	12	46	PC3 >>
Other Policy Change Events	6	2	3	PC1 >>
Service shutdown	4	1	3	PC1 >>
Event processing	2	0	0	PC1 >>

The graphical representation provides a comparison of event counts across three PCs (PC1, PC2, and PC3) for various event IDs. The bar chart illustrates the frequency of each event for the three PCs, making it easy to identify which PC has the highest count for each specific event. In the plot, the bars are color-coded to represent each PC: blue for PC1, orange for PC2, and green for PC3. This visual comparison highlights the differences in event occurrences, showing patterns such as PC2 having significantly higher counts for certain events, while PC1 and PC3 lead in others. The chart offers a clear overview of the distribution of events across the systems.

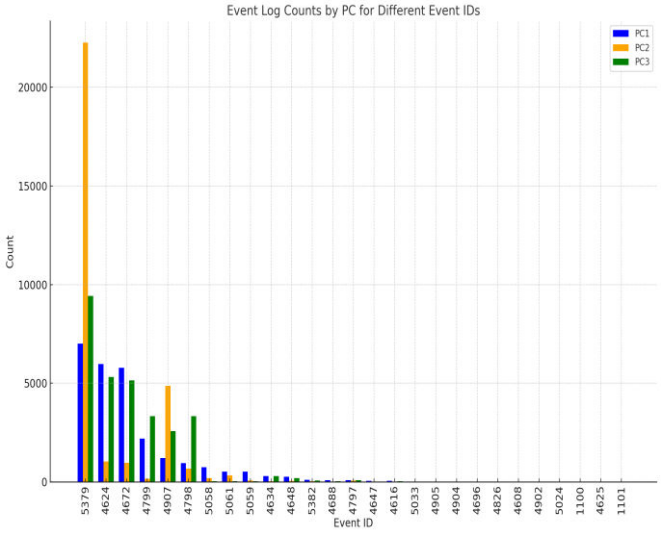

Figure 2. Event Log Counts by PC for Different Event IDs

In the context of monitoring computer system security, various event logs can provide significant insights into user activities and potential security threats. Events such as "Credential Manager credentials were read" (5379) and "Vault credentials were read" (5382) indicate access to sensitive data, which may pose a medium risk if the access is unauthorized. Changes in auditing settings (4907) and assignment of special privileges to users (4672) represent high risk, as they may indicate an attempt to conceal activities or unauthorized privilege assignments. Successful and unsuccessful login attempts, such as "An account was successfully logged on" (4624) and "An account failed to log on" (4625), may signal access attempts to the system, with unsuccessful attempts typically posing a medium risk, while successful ones may indicate unauthorized access if not linked to legitimate users. Operations related to cryptographic key files, such as "Key file operation" (5058) and "Cryptographic operation" (5061), can be associated with high risk if not properly monitored, as the compromise of these data could have severe consequences for system security [13].

Additionally, logs that indicate changes in system configuration, such as "The system time was changed" (4616), are often used to conceal malicious activities and

represent high risk. In the context of system startup and shutdown, logs like "Windows is starting up" (4608) and "Windows is shutting down" (4902) indicate standard operations, while "Audit events have been dropped by the transport" (1101) signals the loss of critical data, which may present a medium risk. Finally, logs related to attempts to unauthorized access or manipulation of settings, such as "An attempt was made to register a security event source" (4904) and "An attempt was made to unregister a security event source" (4905), present high security risks as they may indicate attempts to conceal malicious activities. [13,14]

TABELA II. DISTRIBUTION OF EVENT IDS BY PC WITH DIFFERENCES HIGHLIGHTED"

Event ID	PC1 Count	PC2 Count	PC3 Count	Differences
5379	7016	22261	9423	PC2 has significantly higher count
4624	5975	1039	5315	PC1 and PC3 have higher
4672	5779	972	5142	PC3 has a higher count than PC1 and PC2
4799	2200	149	3330	PC1 and PC3 are higher than PC2
4907	1217	4872	2580	PC2 has the highest count
4798	942	682	3330	PC3 has the highest count
5058	749	199	38	PC1 has the highest count
5061	524	332	38	PC1 has the highest count
5059	514	76	29	PC1 has the highest count
4634	302	28	300	PC1 and PC3 have similar
4648	261	45	197	PC1 has the highest count
5382	102	35	71	PC1 has the highest count
4688	84	22	30	PC1 has the highest count
4797	80	64	80	Equal PC1 and PC3
4647	58	10	26	PC1 has the highest count
4616	46	10	43	PC3 has the highest count
5033	6	2	3	PC1 has the highest count
4905	6	2	1	PC1 has the highest count
4904	6	2	1	PC1 has the highest count
4696	6	2	3	PC1 and PC3 have similar
4826	6	2	3	PC1 and PC3 have similar
4608	6	2	3	PC1 and PC3 have similar
4902	6	2	3	PC1 and PC3 have similars
5024	6	2	3	PC1 and PC3 have similar
1100	4	1	3	PC1 has the highest count
4625	2	12	2	PC2 has the highest count
1101	2	0	3	PC3 has the highest count

IV. ARIMA

ARIMA is a statistical model used for the analysis and forecasting of time series data. This model is particularly useful in analyzing data that is temporally correlated, such as economic indicators, temperatures, energy consumption, and many others. The ARIMA model consists of three key **components: autoregression (AR), moving average (MA), and integration (I),** which enable the model to identify patterns in the data and accurately forecast future values.

AR component uses previous values of the time series to predict the current value. This component is based on the assumption that the current value of the series depends on its past values, employing a linear relationship between the current and previous data points [15]. Mathematically, an AR model of order p can be expressed as:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \in_t , \quad (1)$$

Where Y_t is the current value of the series, ϕ_i are the autoregressive coefficients, c is the constant, and ϵ is the model error (white noise). The integration component transforms non-stationary series into stationary series, making them suitable for analysis. This is achieved by differencing the series, i.e., subtracting the previous value from the current one, and the number of required differences is denoted by the parameter d. Differencing can be represented by the following formula:

$$\Delta Y_t = Y_t - Y_{t-1}, \quad (2)$$

The moving MA component models the current value as a function of the errors from previous predictions. The error is defined as the difference between the actual and predicted values, and the MA model takes into account the average errors from past time points [16]. Mathematically, a MA model of order q is:

$$Y_t = c + \sum_{i=1}^{q} \theta_i \in_{t-i} + \in_t \quad (3)$$

Where θ_i are the moving average coefficients, and $\in$ are the prediction errors. The parameters *p*, *d*, and *q* make up the basic definition of the ARIMA model, written as ARIMA*(p, d, q)*. These parameters are determined through time series analysis, involving differencing the series to achieve stationarity, as well as using autocorrelation and partial autocorrelation techniques to identify the best values for p and q. Mathematically, after the parameters are determined, the Maximum Likelihood Estimation (MLE) method is used to estimate the model coefficients, minimizing the difference between the actual and predicted values [17]. This process allows for the precise determination of parameters that optimize the model for forecasting future data points. The advantages of the ARIMA model are numerous. It is efficient for time series that do not exhibit seasonal effects and provides accurate forecasts for various types of patterns in the data, such as linear trends or cyclical changes. Once the parameters are optimized, the ARIMA model can forecast future values with high precision. However, the ARIMA model also has some limitations. For example, it is not suitable for time series with seasonal fluctuations, as it does not account for seasonal effects. For series exhibiting seasonal patterns, an extended model, SARIMA (Seasonal ARIMA), is used, which adds seasonal components to the basic ARIMA model. Additionally, the selection of parameters p, d, and q can be challenging, especially when the dataset is large [18, 19]. Furthermore, since ARIMA is based on linear assumptions, it is not suitable for series with nonlinear patterns or series with high levels of fluctuation. The graph

displays the event time series with the original data and ARIMA model predictions.

The original time series is clearly shown in blue, while the predictions are represented by a red line with markers indicating each predicted value. The forecast period is highlighted with a red shaded area of low transparency, making it easy to identify where the forecast period begins. Additionally, the last part of the graph is zoomed in to show the final data and predictions in more detail, focusing on the last 50 minutes of the original data and the predicted points (Fig 3).

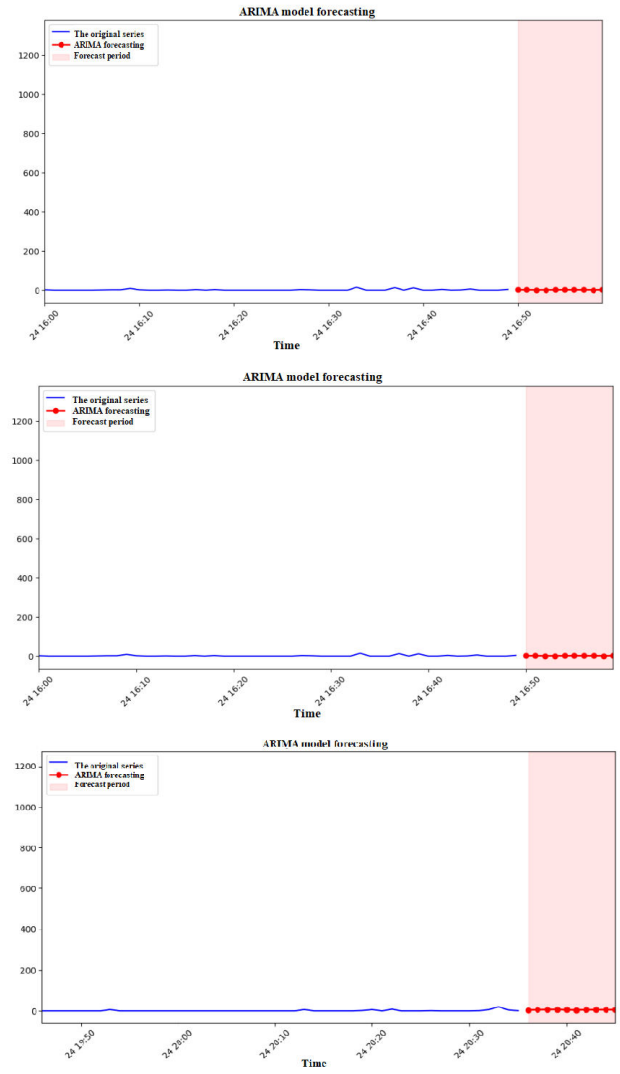

Figure 3. ARIMA model forecasting last 50 miutes

The graph displays a time series of events with the original daily data and ARIMA model predictions at a daily frequency. The original time series is shown in blue, representing the actual event counts on a daily basis. The ARIMA model predictions, marked with a red line and circular markers, show the forecasted event counts for the following 10 days. The forecast period is highlighted with a red shaded area of low transparency, clearly indicating where the predictions start. The x-axis represents the time in days, and the y-axis represents the event count. The last part of the graph includes a rotated x-axis label for better readability (Fig 4).

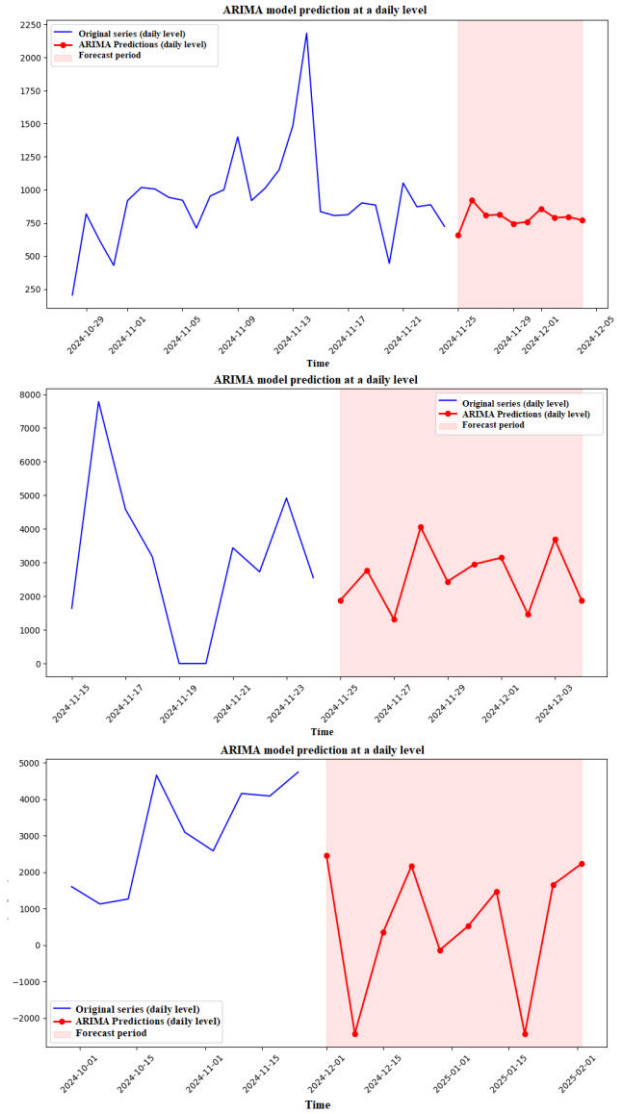

Figure 4. ARIMA model prediction at a daily level

CONCLUSION

Based on the analysis of images and logs, we can conclude that ARIMA models, when applied to log data, offer significant advantages in predicting system events and identifying potential security risks. The displayed images and logs provide better insights into system behavior, vulnerability detection, and anomaly identification that could indicate security threats. Additionally, the event frequency analysis from the logs confirms the importance of regular monitoring and efficient data processing, which is crucial for improving security procedures and preventing security breaches. Furthermore, it was observed that it is better to analyze anomalies at the minute or hourly level rather than at the daily level, as smaller time intervals provide more precise and detailed information about the

logs. This is particularly important when dealing with the type of data used in the analysis, as it offers a deeper insight into the current activities within the system and helps in the early detection of potential issues. These insights highlight the need for implementing advanced data analysis methods to enhance the security and stability of information systems.

ACKNOWLEDGEMENT

This study was partly supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of Grant No. 451-03-66 / 2024-03 / 200132 with the University of Kragujevac - Faculty of Technical Sciences Čačak

REFERENCES

[1] W. Neweva, O. Fitzwilliam, and J. Waterbridge, "Forensic analysis of live ransomware attacks on linux-based laptop systems: Techniques and evaluation," 2024.

[2] O. C. Obi, O. V. Akagha, S. O. Dawodu, A. C. Anyanwu, S. Onwusinkwue, and I. A. I. Ahmad, "Comprehensive review on cybersecurity: modern threats and advanced defense strategies," Computer Science & IT Research Journal, vol. 5, no. 2, pp. 293–310, 2024.

[3] M. Melina, S. Sukono, H. Napitupulu, N. Mohamed, Y. H. Chrisnanto, A. I. Hadiana, et al., "Comparative analysis of time series forecasting models using ARIMA and neural network autoregression methods," BAREKENG: Jurnal Ilmu Matematika dan Terapan, vol. 18, no. 4, pp. 2563–2576, 2024.

[4] M. J. Assi, A. Fahad, and B. Al-Sarray, "Time series forecast modeling for the Windows operating system performance using Box-Jenkins and LSTM models," International Journal of Nonlinear Analysis and Applications, vol. 13, no. 2, pp. 2121–2131, 2022.

[5] S. L. LEE, C. Y. LIEW, C. K. CHEN, and L. L. VOON, "Comparing model building performance of ARIMA model and logarithmic return model," Journal of Information Science & Engineering, vol. 40, no. 5, 2024.

[6] D. Alberg and M. Last, "Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms," Vietnam Journal of Computer Science, vol. 5, pp. 241–249, 2018.

[7] U. M. Sirisha, M. C. Belavagi, and G. Attigeri, "Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison," IEEE Access, vol. 10, pp. 124715–124727, 2022.

[8] D. S. M. Meena Siwach, "Anomaly detection for web log data analysis: A review," Journal of Algebraic Statistics, vol. 13, no. 1, pp. 129–148, 2022.

[9] S. Kang, S. Kim, M. Park, and J. Kim, "Study on windows event log-based corporate security audit and malware detection," *Journal of the Korea Institute of Information Security & Cryptology*, vol. 28, no. 3, pp. 591–603, 2018.

[10] B. Keyogeg, M. Thompson, G. Dawson, D. Wagner, G. Johnson, and B. Elliott, "Automated detection of ransomware in windows active directory domain services using log analysis and machine learning," Authorea Preprints, 2024.

[11] D. A. Bhanage, A. V. Pawar, A. Joshi, and R. G. Pawar, "An efficient failure predictive and remediation system for Windows infrastructure with analysis of log-event records," 2024.

[12] B. Keyogeg, M. Thompson, G. Dawson, D. Wagner, G. Johnson, and B. Elliott, "Automated detection of ransomware in windows active directory domain services using log analysis and machine learning," Authorea Preprints, 2024.

[13] J. Dwyer and T. M. Truta, "Finding anomalies in windows event logs using standard deviation," in 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013, pp. 563–570.

[14] N. R. Koppolu, "Utilizing event logs of windows operating system in digital crime investigations," 2024.

[15] N. M. Ibrahim, A. Al-Nemrat, H. Jahankhani, and R. Bashroush, "Sufficiency of windows event log as evidence in digital forensics," in International Conference on e-Democracy, 2011, pp. 253–262.

[16] C. Y. Tsai, H. I. Cheong, R. Houghton, W. H. Hsu, K. Y. Lee, J. H. Kang, et al., "Predicting fatigue-associated aberrant driving behaviors using a dynamic weighted moving average model with a long short-term memory network based on heart rate variability," Human Factors, vol. 66, no. 6, pp. 1681–1702, 2024.

[17] J. D. Boyko and B. C. O'Meara, "Dentist: Quantifying uncertainty by sampling points around maximum likelihood estimates," Methods in Ecology and Evolution, vol. 15, no. 4, pp. 628–638, 2024.

[18] K. Szostek, D. Mazur, G. Drałus, and J. Kusznier, "Analysis of the effectiveness of ARIMA, SARIMA, and SVR models in time series forecasting: A case study of wind farm energy production," Energies (19961073), vol. 17, no. 19, 2024.

[19] M. W. Nkongolo, "Using ARIMA to predict the expansion of subscriber data consumption," arXiv preprint arXiv:2404.15, 2024.