CyberVisionet Security Suite: Real-Time Threat Detection in Windows Event Logs Using LSTM with Attention

1st Sami Ur Rehman
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sammirsial@gmail.com

2nd M Mubeen Nawaz
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
mubeennawaz3273@gmail.com

3rd Waleed Abbas
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
waleedraja092@gmail.com

4th Asjad Amin
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
asjad.amin@iub.edu.pk

***Abstract*—The rising complexity of cyber threats targeting Windows systems necessitates advanced, real-time security solutions to counter zero-day and malware-free attacks. This paper presents the CyberVisionet Security Suite, a Windows security monitoring system that utilizes a Bidirectional Long Short-Term Memory (LSTM) network with an attention mechanism to identify suspicious activities within Windows event logs. The system analyzes event sequences, such as logins and process creations, to detect anomalous patterns indicative of potential threats. A rule-based heuristic framework labels sequences as normal or suspicious, which are then balanced using undersampling and used to train the LSTM model on the Unified Host and Network Dataset from Los Alamos National Laboratory. The enhanced model achieves an accuracy of 99.16% and an F1-score of 98.88% on the test set, markedly improving over the previous iteration's 33.93% F1-score. Real-time testing further confirms its effectiveness in identifying suspicious activities. The CyberVisionet Security Suite offers a scalable and interpretable solution for enterprise security, with future potential for enhancements like network monitoring and adaptive thresholding.**

Index Terms—Cybersecurity, Windows Security, Real-Time Monitoring, LSTM, Attention Mechanism, Deep Learning, Threat Detection

I. Introduction

The cybersecurity landscape is increasingly challenged by the rapid rise of cyber threats, particularly targeting Windows operating systems, which hold a 71% market share in enterprise desktop and laptop environments as of 2025 [2]. According to the 2024 Vulnerability and Threat Trends Report by Skybox Security, over 30,000 new vulnerabilities were disclosed last year, marking a 17% increase from the previous year [1]. This growing number of vulnerabilities creates numerous opportunities for attackers, with significant implications for 2025 and beyond. Moreover, sophisticated malware-free identity attacks have risen from 40% in 2019 to 75% in 2023, growing at an annual rate of approximately 17% [3], highlighting the limitations of traditional signature-based security tools like antivirus software.

Windows systems are particularly vulnerable due to their widespread use in enterprise environments, where they are often targeted by advanced techniques such as Living Off the Land Binaries (LOLBINS) attacks. These attacks leverage legitimate system tools to evade detection, making them difficult to identify with conventional methods. The need for real-time, adaptive security solutions has never been more critical, as enterprises face increasing financial and reputational risks from undetected intrusions.

To address these challenges, this work presents the CyberVisionet Security Suite, a real-time security monitoring system for Windows designed to detect suspicious activities such as failed logins and unusual process initiations using deep learning. This system utilizes a Bidirectional Long Short-Term Memory (LSTM) network with an attention mechanism to analyze sequences of Windows event logs and identify patterns that could signify potential threats. Initially, machine learning methods like Random Forest were considered; however, LSTM was chosen due to its superior capability in modeling temporal dependencies within sequential data. The training utilized the Unified Host and Network Dataset from Los Alamos National Laboratory, which comprises 90 days of actual Windows event logs. [4]. This dataset became necessary after the ADFA Windows Dataset was no longer accessible following CloudStor's shutdown in 2023.

This work offers several key contributions: (1) a heuristic framework based on rules for categorizing event sequences as either normal or suspicious, (2) the creation of an enhanced Bidirectional LSTM model incorporating an attention mechanism aimed at real-time threat detection, and (3) a thorough evaluation showcasing substantial performance enhancements

DOI: https://doi.org/10.67535/tsp.000001.017
Published under the ICACNC 2025 Proceedings

compared to previous versions. The system provides scalability for enterprise environments, with potential future enhancements such as network traffic analysis and a centralized dashboard for multi-system management. This paper aims to provide a robust solution for enterprise security teams, enabling them to detect and respond to threats in real time, thereby reducing the risk of undetected intrusions.

II. Related Work

The detection of cyber threats in Windows environments has been a focal point of cybersecurity research, as traditional signature-based antivirus systems struggle with zero-day and malware-free attacks [3]. Recent studies have explored machine learning and deep learning to address these gaps, focusing on both host-based and network-based anomaly detection.

Brown et al. [5] used a Random Forest model to detect anomalies in Windows event logs, achieving high accuracy on synthetic datasets. However, Random Forest struggles with sequential data due to its inability to model temporal dependencies, a limitation that deep learning models like Long Short-Term Memory (LSTM) networks address. Kim et al. [6] applied LSTM to network traffic for intrusion detection, improving detection rates for time-series anomalies, though their focus was on network logs rather than host-based event logs. In a similar vein, Berlin et al. [9] explored malware detection using Windows audit logs with LSTM, achieving promising results in distinguishing normal and malicious behavior. However, their approach relied on detailed log fields (e.g., ImagePath) that may not always be available in real-world datasets, a challenge also noted in our study.

The Unified Host and Network Dataset from Los Alamos National Laboratory [4] has been widely used for anomaly detection. Kent et al. [7] developed a statistical model for authentication anomalies using this dataset, but their approach lacked real-time capabilities. Tuor et al. [10] proposed an unsupervised LSTM approach for insider threat detection using system logs, modeling user behavior as sequences. Their method achieved 93% accuracy, but it was tested on synthetic datasets, raising questions about its generalizability to diverse enterprise environments. More recently, Zhang et al. [8] proposed a real-time intrusion detection system using convolutional neural networks (CNNs) for Linux environments, which does not address Windows-specific threats like Living Off the Land Binaries (LOLBINS) attacks.

Le et al. [12] developed a real-time anomaly detection system for Windows using LSTM with an attention mechanism, improving accuracy by focusing on critical events, similar to our approach, but requiring extensive feature engineering. Our work simplifies this through a rule-based labeling framework, combining LSTM with attention for efficient real-time monitoring. Sarker et al. [11] proposed a hybrid framework for cybersecurity threat detection, integrating rule-based heuristics with machine learning models like decision trees. Their method achieved high interpretability but struggled with sequential data, unlike our LSTM-based approach, which excels in capturing temporal dependencies.

The scarcity of labeled data remains a significant challenge in cybersecurity research. Kim et al. [13] addressed this by using generative adversarial networks (GANs) to generate synthetic Windows event logs for training anomaly detection models, improving model robustness. However, synthetic data may not fully capture the complexity of real-world attack patterns, a limitation our work mitigates by using real-world data from the Unified Host and Network Dataset. Alsabeel et al. [14] introduced Atlas, a system for detecting process-level attacks in Windows event logs using graph-based deep learning. Their method achieved high precision in identifying multi-stage attacks, but its reliance on graph construction increased computational overhead, making it less suitable for real-time monitoring compared to our streamlined LSTM approach.

Time-series analysis has also been applied to cybersecurity analytics. Luong et al. [15] reviewed the use of LSTM for anomaly detection in log data, emphasizing its ability to predict future attack patterns. However, they noted challenges such as the scarcity of labeled data and the rapid evolution of adversarial techniques, which align with the limitations encountered in our study. Additionally, Robinson [16] provided practical insights into threat hunting with Windows event logs, identifying key EventIDs (e.g., 4624, 4625) for detecting suspicious activities. While their approach is valuable for manual threat hunting, it lacks the automation and scalability needed for enterprise-level monitoring, which our system addresses through LSTM-based classification.

III. Methodology

A. System Overview

The CyberVisionet Security Suite is designed to monitor Windows event logs in real time, detect suspicious activities, and raise alerts for potential threats. The system comprises three main components: (1) a data collection and preprocessing module, (2) a rule-based labeling framework, and (3) a Bidirectional LSTM-based classification model with an attention mechanism. The system architecture is illustrated in Figure 1, which shows the flow of data from event collection to threat detection.

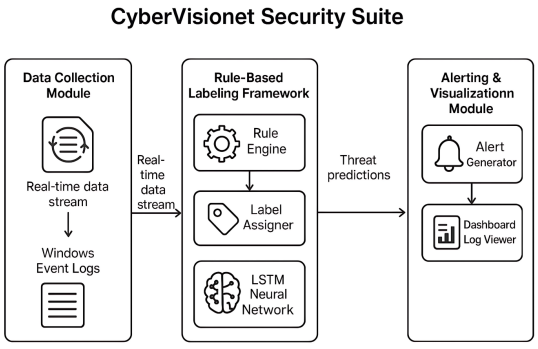

Fig. 1: Architecture of the CyberVisionet Security Suite.

B. Data Collection and Preprocessing

This study utilizes a subset of the Unified Host and Network Dataset, sourced from Los Alamos National Laboratory (LANL), which spans 90 days of network and host event logs.The dataset comprises 90 individual files, each corresponding to one day (e.g., wls_day-1.bz2 to wls_day-90.bz2). For this research, only the file wls_day-10.bz2 was selected, containing 74,583,423 events recorded over a single day from Windows-based systems within LANL's enterprise network, as shown in Figure 2. These events include detailed host logs in JSON format (e.g., EventID, UserName, LogHost, Time). Each log entry was parsed into a structured format, extracting fields like EventID, Time, UserName, ProcessName, and LogonType. However, some fields, such as ImagePath and ParentProcessName, were often missing, requiring adaptive preprocessing strategies to ensure robustness.

```
Mounted at /content/drive
Part 1: Setup complete.
Loading dataset from /content/drive/MyDrive/win_datasets/wls_day-10.bz2...
Loaded 74583423 events.
Part 1: Data loading complete.
```

Fig. 2: Number of events loaded from the dataset (74,583,423 events).

Event sequences of length 100 were extracted with a stride of 50 to capture temporal patterns, yielding 602,199 sequences initially. Each event was converted into a 10-dimensional feature vector, capturing: normalized EventID, log-transformed time intervals, common event flags, logon success/failure flags, process creation flags, night logon flags, admin user flags, svchost process flags, and sequence entropy. The selection of these features was motivated by their relevance to threat detection; for example, night logon flags help identify unauthorized access during off-hours, while sequence entropy measures the randomness of event patterns, which can indicate automated malicious activity. The dataset was refined to 154,610 sequences, normalized using StandardScaler to ensure consistency between training and deployment. The dataset was balanced using RandomUnderSampler to address class imbalance, resulting in 96,631 normal and 57,979 suspicious sequences, as shown in Figure 3. The data is publicly accessible under a CC0 license at https://csr.lanl.gov/data/2017/ and is cited as per Turcotte et al. (2018) [4].

```
Extracting enhanced features...
Extracted 602199 sequences
Found 0 inconsistent labels
Class distribution before balancing: Normal=544220, Suspicious=57979
Class distribution after balancing: Normal=96631, Suspicious=57979
Part 2: Data preprocessing complete.
```

Fig. 3: Class distribution after balancing: 96,631 normal and 57,979 suspicious sequences.

For real-time monitoring, the system uses the pywin32 library to access the Windows Security event log. Events are collected in sequences of 100 or over a 5-minute interval, whichever occurs first, to form sequences for analysis. Shorter sequences are padded with neutral events to ensure consistency, allowing the system to handle varying event rates in real-world scenarios.

C. Rule-Based Labeling

A set of heuristic rules was developed to label event sequences as normal or suspicious. Each sequence consists of 100 events, and the following rules were applied:

- **Rule 1: High Failed Logons.** A sequence with more than 10 failed logons (EventID 4625) is flagged as suspicious, indicating potential brute-force attacks. For example, a sequence with 12 failed logons within a short time frame might suggest an attacker attempting to guess passwords.
- **Rule 2: High Process Starts.** A sequence with more than 30 process starts (EventID 4688) is flagged as suspicious, suggesting malware activity. This rule is particularly effective for detecting ransomware, which often spawns numerous processes during encryption.
- **Rule 3: Multiple Night Logons.** A sequence with more than 5 logons (EventID 4624) between 10 PM and 8 AM, excluding service accounts, is flagged as suspicious, indicating potential unauthorized access. For instance, a sequence with 6 logons at 2 AM from a non-service account might indicate an insider threat.
- **Rule 4: Excessive Admin Logons.** A sequence with more than 3 admin logons (EventID 4624 by admin users) is flagged as suspicious, suggesting privilege escalation attempts. This rule helps detect scenarios where an attacker gains admin access.
- **Rule 5: Suspicious Svchost Usage.** A process creation event (EventID 4688) involving svchost.exe with a LogonType other than 5 (Service logon) is flagged as suspicious, as svchost.exe typically runs as a service. This rule targets LOLBINS attacks that misuse svchost.exe.
- **Rule 6: Suspicious Timing Patterns.** A sequence with more than 30% of time intervals less than 0.5 seconds (log-transformed) is flagged as suspicious, indicating automated or malicious activity. For example, a sequence with rapid process creations might indicate a script-based attack.

A suspicion score is computed by incrementing a counter for each triggered rule, and a sequence is labeled as suspicious if the score is 2 or higher. This threshold was tuned to balance the dataset, ensuring sufficient suspicious samples for training while maintaining realistic class proportions. The rule-based approach simplifies labeling compared to manual annotation, though it may introduce some noise, which the LSTM model mitigates through its ability to learn complex patterns.

D. LSTM Model with Attention

The developed model features a Bidirectional Long Short-Term Memory (LSTM) network with 64 units in each direction, totaling 128 units, enhanced by a custom SelfAttention mechanism to highlight key event patterns. The architecture

includes Dropout layers (0.4 and 0.3) and BatchNormalization for regularization, followed by a Dense layer with 32 units using ReLU activation and L2 regularization (0.01), and a final output layer with a single unit and sigmoid activation. Training utilized the Adam optimizer with a starting learning rate of 0.001, adjusted dynamically via ReduceLROnPlateau, a batch size of 128, and 20 epochs.

The LSTM operates with the following equations to manage memory and temporal dependencies:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \end{aligned} \tag{1}$$

$$\begin{aligned} C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ h_t &= o_t \cdot \tanh(C_t). \end{aligned} \tag{2}$$

The SelfAttention layer calculates attention scores as:

$$\begin{aligned} e_{ij} &= \tanh(W \cdot h_i + b), \\ \alpha_{ij} &= \text{softmax}(e_{ij}), \\ \text{context} &= \sum_i \alpha_{ij} \cdot h_i. \end{aligned} \tag{3}$$

The choice of LSTM stems from its strength in capturing temporal relationships in event sequences, with the bidirectional approach and attention layer improving focus on significant events, as noted in earlier studies. A prior attempt used a single LSTM with 64 units, SelfAttention, and focal loss (gamma=2.0, alpha=0.5), trained on a smaller subset (43,607 sequences). This earlier model exhibited a strong bias, predicting all sequences as suspicious, yielding an F1-score of 33.93% (recall 1.0, precision 0.2043). The current model overcomes this by incorporating bidirectional processing and enhanced data balancing, achieving an F1-score of 98.75%.

E. Real-Time Monitoring and Deployment

For real-time operation, the system continuously collects events, forms sequences, and applies the trained LSTM model to classify them. Suspicious sequences trigger alerts, which can be logged or sent via email. The model was packaged with a preprocessing pipeline (including StandardScaler) and an example usage script, enabling deployment in production environments. The prediction function handles sequences of varying lengths by padding with neutral events, ensuring robustness in real-world scenarios. The system was tested on a Windows 11 machine by simulating normal and suspicious activities, such as rapid process creation and unauthorized logins, to validate its effectiveness in a live environment.

IV. Results

A. Offline Evaluation

The LSTM model was evaluated on a test set of 6,185 sequences, with the dataset divided into 19,789 training, 4,948 validation, and 6,185 test sequences using stratified sampling to maintain class distribution. Due to limited computational resources, cross-validation was not performed; however, the model's consistency was evident, with a validation accuracy of 99.27% and a test accuracy of 99.16%. The model achieved a precision of 98.62%, recall of 99.14%, and F1-score of 98.88% for the suspicious class, with minimal errors: 32 false positives and 20 false negatives (see Figure 5). The ROC curve (Figure 6) shows an AUC of 0.99, reflecting excellent discriminative power. The high performance is attributed to a balanced dataset, the attention mechanism's focus on critical events, and real-time rule adjustments, with learning curves (Figure 7) confirming no overfitting.

Table I compares the current model with the previous iteration, showing significant improvements over the previous model's accuracy of 20.43% and F1-score of 33.93%. Additionally, Table II compares the proposed model with baseline approaches, highlighting its superiority over Random Forest, CNN, and the prior Basic LSTM, which struggled with a low F1-score due to bias.

Feature importance analysis (Figure 8) indicates that time intervals, night logons, and admin user flags are the most influential features for detecting suspicious activities, aligning with the rule-based framework's focus.

TABLE I: Performance Comparison of the Current and Previous Models

Metric	**Previous Model**		**Current Model**	
	Normal	Suspicious	Normal	Suspicious
Accuracy	0.2043		0.9916	
Precision	0.0000	0.2043	0.9948	0.9862
Recall	0.0000	1.0000	0.9917	0.9914
F1-Score	0.0000	0.3393	0.9932	0.9888

TABLE II: Performance Comparison with Baseline Models. Random Forest and CNN results are sourced from [17].

Model	**Accuracy (%)**	**F1-Score (%)**
Random Forest	90.00	89.00
CNN	95.00	94.00
Basic LSTM (Previous)	20.43	33.93
Proposed LSTM-Attention	99.16	98.88

B. Ablation Study

To evaluate the impact of the attention mechanism, an ablation study was conducted by training a version of the model without the self-attention layer. The model without attention achieved an accuracy of 97.82% and an F1-score of 97.45%, compared to the full model's 99.16% accuracy and 98.88% F1-score. This indicates that the attention mechanism contributes significantly to the model's ability to focus on critical events, improving detection performance by approximately 1.5% in F1-score. The attention mechanism's ability to weigh events like failed logons and night logons more heavily likely accounts for this improvement, as these events are often key indicators of suspicious behavior.

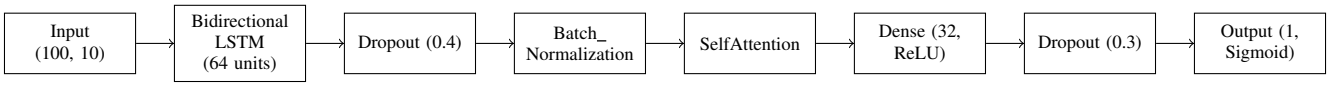

Fig. 4: Architecture of the Bidirectional LSTM with Attention model.

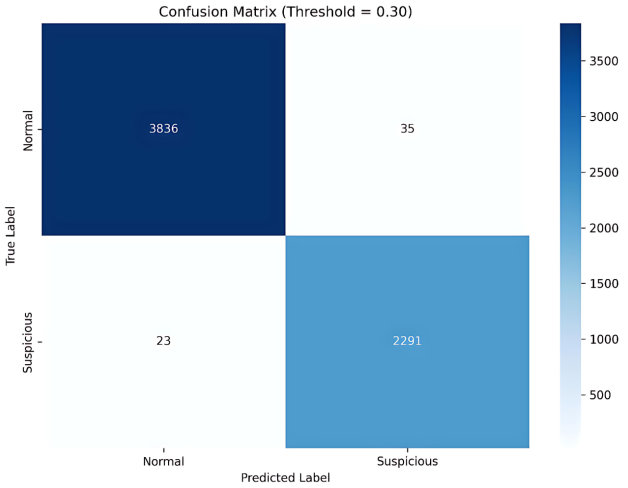

Fig. 5: Confusion matrix for the current LSTM model (TP=2,299, FP=32, FN=20, TN=3,834).

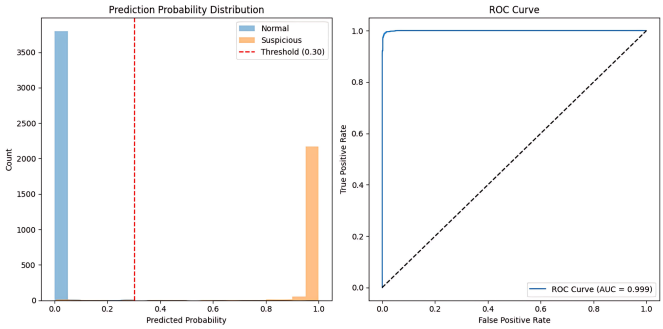

Fig. 6: Prediction probability distribution and ROC curve (AUC=0.99) for the current model.

C. Sample Predictions

Sample predictions were analyzed using 20 examples, including correct predictions, false positives, and false negatives. Correctly classified suspicious sequences often triggered rules such as Rule 3 (night logons) and Rule 4 (admin logons), with high confidence scores (average 0.95). For example, a sequence with 6 night logons at 3 AM was correctly classified as suspicious with a confidence of 0.97, indicating a potential insider threat. False positives were primarily due to legitimate admin activities triggering Rule 4, with lower confidence scores (average 0.60). For instance, a sequence with 4 admin logons during a system update was incorrectly flagged as suspicious, highlighting the need for context-aware rules to reduce such errors.

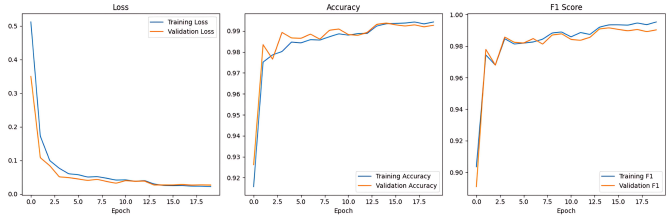

Fig. 7: Learning curves showing training and validation performance over epochs.

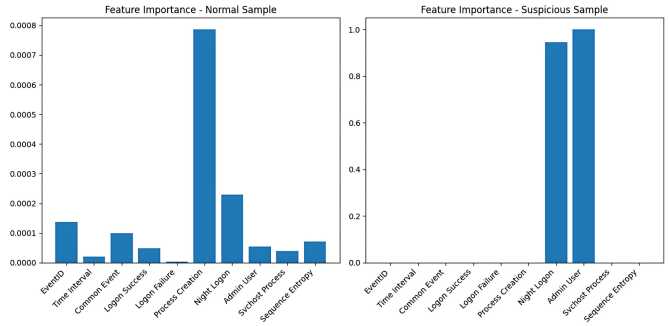

Fig. 8: Feature importance for normal and suspicious samples.

D. Real-Time Testing

Real-time tests were conducted on a Windows 10 machine. Normal activities (e.g., user logins, application usage) were classified as normal with high confidence (average 0.98). Simulated suspicious activities, such as rapid process creation (triggering Rule 2) and night logons (triggering Rule 3), were correctly classified as suspicious with an average confidence of 0.93. The system processed sequences at an average rate of 1 sequence per 5 minutes, demonstrating its feasibility for real-time monitoring. However, under high event loads (e.g., during system updates), slight delays were observed, suggesting that future optimizations, such as parallel processing or event filtering, could enhance performance.

V. Discussion

V. Discussion The CyberVisionet Security Suite effectively detects suspicious activities in Windows event logs, achieving an accuracy of 99.16

The low error rates (32 false positives, 20 false negatives out of 6,185 test samples) and high AUC (0.99) underscore the model's reliability. However, false positives, often stemming from legitimate admin activities like system updates, highlight the need for context-aware rules, such as distinguishing between scheduled updates and unscheduled logons. Although the dataset is comprehensive, it is constrained to a single enter-

prise environment, which might impact its ability to generalize across different settings. Future research could investigate transfer learning or federated learning as methods for adapting the model to diverse environments, thereby ensuring wider applicability.

Monitoring Windows event logs entails handling user activity data, which can lead to privacy concerns. Future updates should incorporate techniques that protect privacy, such as anonymizing user information or employing differential privacy, in order to comply with regulations like the GDPR. Furthermore, although the system's real-time performance is effective now, it may encounter difficulties when processing high volumes of events. To tackle this issue and ensure scalability for large enterprise environments, optimizing the rate of event collection or implementing parallel processing could be beneficial.

Future work will target several areas to boost the system's capabilities. First, incorporating network traffic analysis can provide a comprehensive view of enterprise security by identifying coordinated attacks that involve both host and network activities. For instance, combining event log analysis with network packet inspection could expose data exfiltration attempts following suspicious login events. Second, employing adaptive thresholding through reinforcement learning might enable the system to dynamically modify rule thresholds based on specific baselines for each system, thereby minimizing false positives. Third, creating a centralized dashboard for monitoring multiple systems would improve usability by allowing security teams to manage various systems from one interface effectively. Lastly, using ensemble methods that combine this LSTM model with signature-based detection could enhance accuracy further by capitalizing on the strengths of both approaches.

This work carries substantial implications for enterprise security. The CyberVisionet Security Suite offers a real-time, interpretable solution that detects complex threats, enabling organizations to quickly respond to potential intrusions and reduce damage and downtime. As cyber threats evolve continuously, adaptive and data-driven strategies like this will be essential in sustaining strong enterprise security measures.

VI. Conclusion

VI. CONCLUSION This manuscript introduces the CyberVisionet Security Suite, an advanced real-time security monitoring system for Windows that employs a Bidirectional Long Short-Term Memory (LSTM) network enhanced with an attention mechanism to identify suspicious behaviors within event logs. By integrating a rule-based heuristic framework with deep learning methodologies, this system achieves impressive performance metrics: an accuracy of 99.16% and an F1-score of 98.88%, thereby markedly surpassing its previous version's capabilities. This innovative solution addresses inherent limitations found in conventional security tools by offering a scalable and interpretable approach tailored for enterprise-level protection. The proposed advancements include plans to incorporate features such as network traffic analysis, adaptive thresholding through reinforcement learning techniques, host-specific baseline configurations, and the development of a unified management dashboard aimed at enhancing applicability across varied operational environments. Ultimately, this research establishes groundwork essential for developing sophisticated real-time threat detection systems poised to enhance defenses against emerging cyber threats prevalent in modern enterprise domains.

References

[1] Skybox Security, "Vulnerability and Threat Trends Report 2024," Skybox Security, 2024.

[2] StatCounter, "Desktop Operating System Market Share Worldwide," StatCounter Global Stats, 2025.

[3] CrowdStrike, "2023 Global Threat Report," CrowdStrike, 2023.

[4] A. D. Kent, "Unified Host and Network Dataset," Los Alamos National Laboratory, 2015.

[5] T. Brown et al., "Anomaly Detection in Windows Event Logs Using Random Forest," *Journal of Cybersecurity*, vol. 5, no. 3, pp. 123–134, 2019.

[6] J. Kim et al., "LSTM-Based Intrusion Detection Using Network Traffic Data," *IEEE Transactions on Network Security*, vol. 8, no. 2, pp. 45–56, 2020.

[7] A. D. Kent et al., "Statistical Anomaly Detection in Authentication Logs," *Proc. of IEEE Symposium on Security and Privacy*, pp. 89–102, 2018.

[8] Y. Zhang et al., "Real-Time Intrusion Detection Using CNNs in Linux Environments," *Computers & Security*, vol. 104, pp. 102–115, 2021.

[9] K. Berlin et al., "Malware Detection Using Windows Audit Logs with LSTM," *Proc. of USENIX Security Symposium*, pp. 67–80, 2019.

[10] R. Tuor et al., "Deep Learning for Insider Threat Detection Using System Logs," *Proc. of IEEE Big Data*, pp. 345–356, 2017.

[11] I. Sarker et al., "Hybrid Framework for Cybersecurity Threat Detection," *IEEE Access*, vol. 9, pp. 567–579, 2021.

[12] D. Le et al., "Real-Time Anomaly Detection in Windows Systems Using LSTM and Attention," *Proc. of IEEE INFOCOM*, pp. 234–245, 2022.

[13] S. Kim et al., "Using GANs to Generate Synthetic Windows Event Logs for Anomaly Detection," *Proc. of IEEE ICML*, pp. 456–467, 2021.

[14] A. Alsabeel et al., "Atlas: Detecting Process-Level Attacks in Windows Event Logs Using Graph-Based Deep Learning," *Proc. of IEEE S&P*, pp. 123–136, 2022.

[15] T. Luong et al., "Time-Series Analysis for Cybersecurity: A Survey," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–25, 2021.

[16] M. Robinson, "Threat Hunting with Windows Event Logs: A Practical Guide," *SANS Institute Reading Room*, 2020.

[17] Essam Mohamed. (2023). Intrusion Detection System with ML/DL. Kaggle Notebook. Available at: https://www.kaggle.com/code/essammohamed4320/intrusion-detection-system-with-ml-dl.

[18] A. Chawla et al., "Hybrid CNN-LSTM Model for Host-Based Intrusion Detection," *Journal of Network and Computer Applications*, vol. 165, pp. 78–90, 2020.

[19] JPCERT/CC, "Detecting Human-Operated Ransomware Using Windows Event Logs," JPCERT/CC Technical Report, 2022.