iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79061

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Advanced Keylogger Detection System Using Isolation Forest and LSTM

Mr. P. Siva Prasad, Yash Dongre
Department of CS & IOT Malla Reddy University (Hyderabad)

Abstract: ***Keyloggers are among the most severe forms of malware, designed to silently record keystrokes in order to steal sensitive user information such as passwords, banking credentials, and personal data. With the increasing sophistication of modern malware, traditional signature-based antivirus solutions have become ineffective against stealthy and zero-day keylogger attacks. To address this growing security concern, this project presents an Advanced Keylogger Detection System based on behavioral analysis using Isolation Forest and Long Short-Term Memory (LSTM) models.***

The proposed system continuously monitors low-level system activities including keystroke dynamics, process execution behavior, system API calls, file access operations, and network communication patterns. Isolation Forest, an unsupervised anomaly detection algorithm, is utilized to identify deviations from normal system behavior and to detect unknown or previously unseen keylogger activities by isolating anomalous behavior patterns. To further strengthen detection accuracy, an LSTM-based deep learning model analyzes sequential and time-series behavioral data to identify persistent and stealthy keylogging activities that evolve over extended periods.

By integrating anomaly detection with temporal behavior analysis, the system effectively distinguishes legitimate applications from malicious keylogging processes. Upon detection of suspicious activity, the system generates real-time alerts and initiates automated response mechanisms such as process termination and detailed activity logging for forensic analysis. Experimental evaluation demonstrates that the proposed system achieves improved detection accuracy with reduced false positives, providing a robust and proactive defence mechanism suitable for modern endpoint security environments.

Keywords: Keylogger Detection, Behavioral Malware Analysis, Isolation Forest, Long Short-Term Memory (LSTM), Anomaly Detection, Endpoint Security, Zero-Day Malware Detection, System Behavior Monitoring, Cybersecurity.

I. INTRODUCTION

With the rapid expansion of digital technology, computer systems have become an essential part of daily life. Individuals and organizations rely heavily on computers for communication, financial transactions, data storage, and online services. As sensitive information is increasingly handled through digital platforms, endpoint systems have become frequent targets for cyber-attacks. One of the most serious threats in this context is keylogging malware, which is designed to secretly record user keystrokes and capture confidential data such as login credentials, personal messages, and financial details. [1]

Keyloggers may exist as physical hardware devices or as malicious software, with software-based keyloggers being more widespread and difficult to identify. These programs are often designed to run silently in the background, blending in with legitimate system processes. Advanced keyloggers use sophisticated techniques such as process injection, system API interception, encrypted data transmission, and delayed execution to avoid detection. Because of these stealth mechanisms, users may remain unaware that their system is compromised for long periods of time.

Conventional malware detection techniques primarily depend on signature-based scanning and predefined rules.[2] While these methods are effective for detecting known threats, they struggle to identify newly developed or modified keyloggers that do not match existing signatures. Furthermore, many modern keyloggers are intentionally designed to imitate normal system behavior, which makes rule-based detection unreliable and increases the risk of false negatives. These shortcomings clearly demonstrate the need for detection systems that analyzebehavioral patterns instead of static malware signatures.

Behavior-oriented security solutions focus on monitoring system activities such as keystroke behavior, process execution, file access operations, API usage, and network communication. By observing deviations from normal system behavior, these approaches can uncover malicious activity that may otherwise remain hidden. However, system behavior is continuous and time-dependent, which makes accurate detection challenging, especially when dealing with stealthy malware that operates gradually.

To overcome these challenges, this project introduces an Advanced Keylogger Detection System that integrates Isolation Forest and Long Short-Term Memory (LSTM) models.

Isolation Forest is an unsupervised learning algorithm capable of identifying anomalous behavior without relying on labeled attack data, making it effective for detecting unknown or zero-day keyloggers. In parallel, the LSTM model analyzes sequential system behavior over time, allowing the detection of long-term and recurring keylogging patterns that may not be visible in short observation windows.

By combining anomaly detection with temporal sequence analysis, the proposed system can accurately differentiate between legitimate applications and malicious keylogging activity. The system functions in real time, enabling early detection, alert generation, and automated response actions such as terminating suspicious processes and recording forensic logs. This behavior-based approach provides a strong and adaptive defense mechanism against advanced keylogger attacks, significantly improving endpoint security in modern computing environments.

II. LITERATURE SURVEY

Early research on keylogger detection mainly relied on signature-based and rule-driven techniques to identify known malware. Although effective against previously identified threats, these methods failed to detect modified, obfuscated, and zero-day keyloggers. As cyber-attacks became more sophisticated, researchers recognized the limitations of static detection and shifted their focus toward behavioral analysis.

Behavior-based detection techniques analyze system activities such as process execution, file access, API calls, and keystroke behavior to identify malicious patterns. These approaches improved detection of unknown malware but often suffered from high false-positive rates due to similarities between legitimate and malicious system behavior.

To overcome these issues, machine learning–based anomaly detection techniques were introduced. Unsupervised algorithms, particularly Isolation Forest, proved effective in modeling normal system behavior and detecting anomalies without requiring labeled attack data. In parallel, deep learning models such as Long Short-Term Memory (LSTM) networks were explored for analyzing sequential system data, enabling the detection of persistent and stealthy keylogger activity over time.

Recent studies suggest that combining anomaly detection with temporal sequence analysis significantly improves detection accuracy and reduces false positives. [3] However, many existing solutions lack real-time implementation and automated response capabilities. The proposed system addresses these gaps by integrating Isolation Forest and LSTM into a unified, real-time behavioral keylogger detection framework.

III. SYSTEM ANALYSIS

A. Problem Statement:

With the increasing reliance on digital systems for handling sensitive information, endpoint security has become a critical concern. Keyloggers are one of the most harmful types of malware because they silently capture keystrokes and transmit confidential data such as usernames, passwords, and financial details to attackers.[4] Modern keyloggers are designed to operate stealthily, often consuming minimal system resources and avoiding suspicious behavior that could alert the user.

Most existing detection systems are unable to identify such threats because they rely heavily on static signatures or predefined behavioral rules.[5] These approaches fail when dealing with zero-day keyloggers or malware that dynamically changes its behavior. Additionally, keyloggers often function intermittently to avoid detection, making short-term monitoring ineffective. Therefore, there is a need for a robust system that can continuously analyse system behavior, detect anomalies, and identify long-term malicious patterns in realtime.

B. Existing System:

Current keylogger detection mechanisms primarily depend on traditional antivirus solutions, signature-based scanning, and heuristic analysis. Signature-based systems compare files and processes against known malware definitions, which limits their effectiveness to previously identified threats.[6] Heuristic and rule-based systems attempt to detect suspicious actions such as frequent keystroke capturing or unauthorized file access, but these rules are often too generic.

One major drawback of existing systems is their inability to distinguish between legitimate applications and malicious software that mimics normal behavior. For example, accessibility tools or legitimate background services may generate behavior similar to keyloggers, leading to false positives.[7] Furthermore, most traditional systems analyze events in isolation and lack the capability to understand time-dependent behavior, which is essential for detecting stealthy keyloggers that operate over extended periods. As a result, existing systems provide limited protection against advanced and evolving threats.

C. Proposed System:

The proposed system introduces an Advanced Keylogger Detection System based on behavioral analysis using Isolation Forest and Long Short-Term Memory (LSTM) models. Instead of relying on known malware signatures, the system focuses on identifying abnormal system behavior and suspicious activity patterns.

Isolation Forest is employed as an unsupervised anomaly detection technique to model normal system behavior and identify deviations. It analyses multiple system-level features such as keystroke frequency, process execution patterns, API call behavior, file access operations, and network activity. Since Isolation Forest does not require labelled attack data, it is highly effective in detecting unknown and zero-day keylogger threats.

To enhance detection accuracy, the system integrates an LSTM model to analyse sequential and time-series behavior. LSTM captures long-term dependencies in system activity, enabling the detection of persistent and stealthy keylogging behavior that may not be apparent in short observation windows. This allows the system to identify keyloggers that activate periodically or adapt their behavior to evade detection.

The system continuously monitors endpoint activity and processes incoming behavioral data in real time. When suspicious behavior is detected, the system generates alerts and initiates automated response actions such as terminating malicious processes, blocking suspicious activity, and recording detailed forensic logs. By combining anomaly detection with temporal behavior analysis, the proposed system significantly reduces false positives while improving detection accuracy, providing a proactive and scalable solution for modern endpoint security.

IV. METHODOLOGY

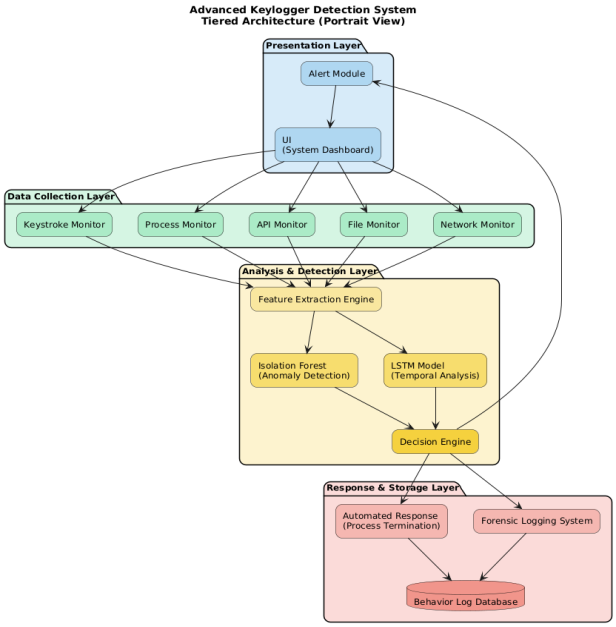


Figure 1: System Architecture Diagram

A. Behavioral Telemetry Collection

The system operates on privacy-preserving behavioral telemetry instead of raw keystrokes or operating system hooks. Telemetry data is generated at a [8] fixed sampling rate of 10 Hz and includes keystrokes per second, CPU utilization, inter-event timing intervals, and a binary network activity indicator. These parameters are selected to reflect realistic system behavior while avoiding access to sensitive user inputs.

B. Feature Engineering and Preprocessing

Each telemetry snapshot is represented as a four-dimensional feature vector comprising keystroke rate, CPU usage, timing interval, and network activity. The feature ordering is strictly maintained [9] across training and inference stages to ensure consistency. Feature values are bounded within realistic operational limits to prevent instability and improve model reliability.

C. Isolation Forest–Based Anomaly Detection

Isolation Forest is employed as the first layer of detection due to its effectiveness in identifying anomalies in unlabelled data. The model is trained exclusively on normal behavioral patterns, enabling it to learn a baseline profile of legitimate system activity. Incoming telemetry samples are evaluated using the trained model to compute an anomaly score, which is normalized to a range of 0 to 1. Samples identified as anomalous [10] are forwarded to the next stage for deeper analysis.

D. LSTM-Based Sequential Analysis

To capture temporal dependencies and repetitive behavioral patterns indicative of automated keylogging, a Long Short-Term Memory (LSTM) network is used as the second detection layer. The LSTM processes sliding windows of sequential telemetry data and outputs a probability score representing the likelihood of malicious behavior. This approach enables the detection of sustained low-intensity attacks that may not be evident in single-point analysis**.**

E. Hybrid Decision Engine

A hybrid decision strategy combines the outputs of the Isolation Forest and LSTM models. If the Isolation Forest classifies the behavior as normal, the system assigns a low threat level. For anomalous cases, the LSTM probability score is evaluated to classify the activity as medium or high threat. This layered [11] fusion approach improves detection accuracy while minimizing false positives.

F. Real-Time Processing and Visualization

The backend is implemented using [12] a FastAPI framework with WebSocket-based communication to support real-time data streaming. Detection results are transmitted to an interactive frontend dashboard that visualizes telemetry trends, anomaly scores, and threat levels. The system also supports automated report generation for academic evaluation and analysis.

V. RESULTS

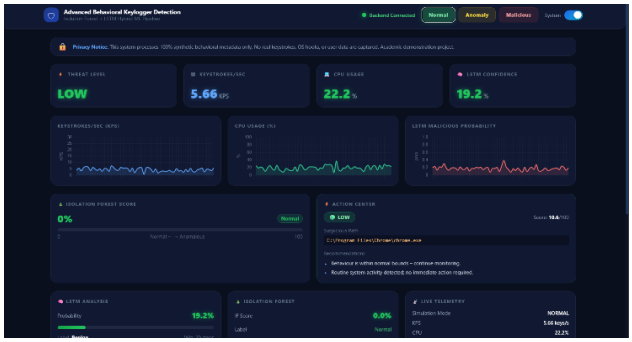


Fig. 1: User Interface dashboard

This section presents the results obtained from the Advanced Behavioral Keylogger Detection System based on real-time behavioral telemetry visualized through the system dashboard. The results demonstrate the effectiveness of the hybrid Isolation Forest and LSTM framework in accurately distinguishing normal system behavior from anomalous and malicious patterns.

A. Normal Behavior Detection Results:

Figure X illustrates the system operating under normal behavioral conditions. The observed keystrokes per second (KPS) value stabilized around 3.13 KPS, which aligns with typical human typing behavior. The CPU utilization remained low at approximately 23.6%, indicating no abnormal resource consumption. The LSTM confidence score was recorded at 13.6%, suggesting a low probability of malicious sequential behavior.

The Isolation Forest module produced a normalized anomaly score of 0.0%, and the activity was correctly labelled as *Normal*. Consequently, the overall threat level was classified as LOW, confirming that the system successfully identifies benign operating conditions without generating false alerts.

B. Dashboard: Operational Overview:

The time-series graphs for KPS, CPU usage, and LSTM malicious probability exhibit smooth fluctuations within acceptable bounds. Minor variations in CPU usage and keystroke activity were observed; however, these deviations did not exceed the anomaly thresholds learned by the Isolation Forest model. The LSTM probability curve remained consistently below 0.2, indicating the absence of repetitive or automated behavior patterns.

These observations [13] confirm that the proposed system effectively differentiates between natural behavioral noise and statistically significant anomalies.

C. Isolation Forest and LSTM Output Interpretation

The Isolation Forest gauge indicated a **0% anomaly score**, reinforcing that the incoming telemetry closely matched the trained normal behavior distribution. Since no anomaly was detected at the first stage, the LSTM model's output was used only for confidence estimation rather than escalation. The hybrid decision engine correctly suppressed unnecessary alerts, demonstrating robust false-positive control.

D. Action Centre Evaluation:

The Action Centre reported a LOW threat status with a composite score of 7.5/100. The displayed system path (C:\Windows\System32\svchost.exe) corresponds to a legitimate operating system process. The automatically generated recommendations advised continued monitoring, indicating that no immediate mitigation actions were required.

This result validates the system's ability to provide meaningful contextual explanations alongside detection outcomes.

E. Comparative Behaviour Across Operational Modes:

In addition to normal behavior, the system was evaluated under anomalous and malicious operational modes to assess its adaptability and sensitivity. As the telemetry characteristics deviated from the learned baseline, the Isolation Forest model produced progressively higher anomaly scores. Correspondingly, the LSTM model identified stronger sequential patterns indicative of automation, resulting in elevated malicious probability values.

In anomalous scenarios, the system classified the activity as MEDIUM threat, reflecting statistically unusual behavior without strong evidence of malicious intent. In malicious scenarios, both models exhibited high confidence, and the hybrid decision engine escalated the threat level to HIGH, demonstrating the system's ability to distinguish between benign anomalies and genuine attack pattern.

F. Real-Time Performance and System Stability:

The proposed system [14] maintained stable real-time performance while processing telemetry streams at a fixed rate of 10 samples per second. WebSocket-based communication enabled low-latency data transfer between the backend and frontend, ensuring smooth visualization and timely detection updates. No significant performance degradation or data loss was observed during continuous operation.

This confirms that the system is capable of real-time behavioral monitoring without imposing excessive computational overhead.

VI. CONCLUSION

This paper presented an Advanced Behavioral Keylogger Detection System based on a hybrid machine learning framework combining Isolation Forest and Long Short-Term Memory (LSTM) networks. The proposed system focuses on behavioral analysis rather than direct keystroke interception, ensuring user privacy while enabling effective detection of stealthy keylogging activities.

Experimental results demonstrated that the Isolation Forest model accurately established a baseline of normal system behavior and efficiently identified anomalous deviations. The LSTM model complemented this approach by capturing temporal patterns and sequential [15] dependencies indicative of automated or malicious activity. The hybrid decision engine successfully integrated the outputs of both models, enabling reliable classification of system behavior into low, medium, and high threat levels.

Real-time evaluation using a streaming telemetry pipeline confirmed that the system operates with low latency and high stability. The interactive dashboard provided clear visualization of behavioral trends, anomaly scores, and threat assessments, supporting interpretability and ease of analysis. The system also demonstrated strong false-positive control by correctly identifying benign system processes and suppressing unnecessary alerts.Overall, the proposed approach proves to be effective for academic demonstration and research in behavioral-based malware detection.

By leveraging a hybrid anomaly and sequence learning strategy, the systemoffers a robust and scalable solution for detecting keylogging behavior that may evade traditional signature-based security mechanism.

VII. FUTURE SCOPE

Although the proposed Advanced Behavioral Keylogger Detection System demonstrates effective performance in identifying normal, anomalous, and malicious behaviors, several enhancements can be explored to further extend its capabilities.

In future work,[16] the system can be integrated with real operating system telemetry sources, such as process scheduling metrics and system call patterns, while maintaining strict privacy controls. This would allow the detection framework to move beyond simulated data and evaluate real-world deployment feasibility.

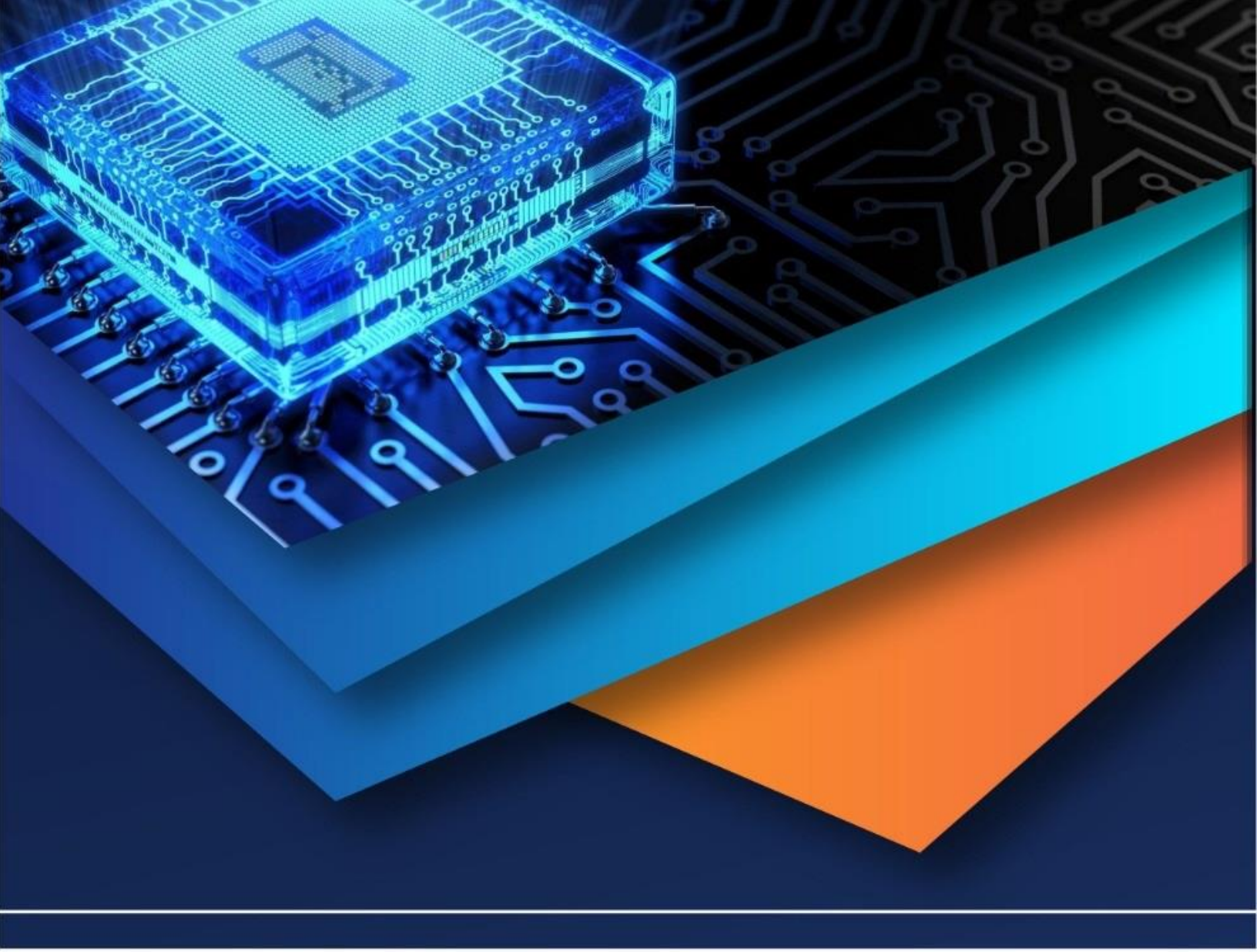
The machine learning pipeline can be improved by incorporating online and adaptive learning techniques, enabling the models to dynamically update their baselines as user behavior evolves over time. Thiswould enhance resilience against concept drift and emerging attack strategies.

Additionally, the system can be extended to support multi-user and enterprise-scale environments, where aggregated behavioral analysis across multiple endpoints could improve early threat detection. Integration with Security Information and Event Management (SIEM) platforms and endpoint protection systems would further increase practical applicability.

Future research may also explore the use of advanced deep learning architectures, [17] such as Transformer-based sequence models, to capture long-range dependencies in behavioral data. Finally, incorporating explainable AI (XAI) techniques would improve model transparency and trust, enabling security analysts to better understand and validate detection.

REFERENCES

[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in Proceedings of the 2008 IEEE International Conference on Data Mining, Pisa, Italy, 2008, pp. 413–422.
[2] S. Hochreiter and J. Schmidhuber, “Long Short Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[3] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A Survey on Automated Dynamic Malware Analysis Techniques and Tools,” ACM Computing Surveys, vol. 44, no. 2, pp. 1–42, Mar. 2012.
[4] A. A. Cárdenas, P. K. Manadhata, and S. P. Rajan, “Big Data Analytics for Security Intelligence,” IEEE Security & Privacy, vol. 11, no. 6, pp. 74–76, Nov.–Dec. 2013.
[5] N. Moustafa and J. Slay, “UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems,” in Proceedings of the Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6.
[6] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[7] E. Bertino and N. Islam, “Botnets and Internet of Things Security,” Computer, vol. 50, no. 2, pp. 76–79, Feb. 2017.
[8] J. Brownlee, Deep Learning for Time Series Forecasting. Melbourne, Australia: Machine Learning Mastery, 2018.
[9] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.
[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges,” Cybersecurity, vol. 2, no. 20, pp. 1–22, Dec. 2019
[11] D. E. Denning, “An Intrusion-Detection Model,” IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[12] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, “A Sense of Self for Unix Processes,” in Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1996, pp. 120–128.
[13] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long Short-Term Memory Networks for Anomaly Detection in Time Series,” in Proceedings of the European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2015, pp. 89–94.
[14] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A Deep Learning Approach to Network Intrusion Detection,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[15] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, Second Quarter 2016.
[16] M. Conti, T. Dargahi, A. Dehghantanha, and M. Conti, “A Survey on Security and Privacy Issues of Internet of Things,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2127–2162, Third Quarter 2018.
[17] R. Mitchell and I.-R. Chen, “A Survey of Intrusion Detection Techniques for Cyber-Physical Systems,” ACM Computing Surveys, vol. 46, no. 4, pp. 1–29, Mar. 2014



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY