iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69840

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 13 Issue IV Apr 2025- Available at www.ijraset.com

Real-time Insider Attack Detection using Graph-Based Anomaly Detection and Concept Drift Handling

P. Boobalan[1], G. Alan Wesley[2], T. Swetha[3], S. Snega[4]
Department of Information Technology, Puducherry Technological University, Puducherry, India

Abstract: ***This project presents a deep learning-based real-time framework for detecting insider threats using a hybrid model that integrates sequence modeling and relational learning. The system analyzes user activity data dynamically and predicts potential insider threats without human intervention. Leveraging Long Short-Term Memory (LSTM) networks for user behavior sequence analysis and Graph Neural Networks (GNNs) for peer-context enrichment, the framework accurately identifies anomalies at the activity level. Each user action is encoded, evaluated against similar activities in the organization, and classified based on anomaly scores. Using the CERT insider threat dataset, the system is evaluated with precision, recall, and F1-score metrics. A visualization dashboard supports real-time monitoring and alerting for security analysts. This project enhances the ability to proactively detect and respond to insider threats across various organizational environments.***

Keywords: Insider Threat Detection, Long Short-Term Memory (LSTM), Graph Neural Networks (GNN), CERT Dataset, Sequence Modeling, Anomaly Detection, Cybersecurity, Real-Time Monitoring, Deep Learning, Activity Prediction.

I. INTRODUCTION

Insider threats pose significant security risks due to their inherent complexity and subtlety. Traditional static monitoring approaches often fail to detect anomalies arising from legitimate users misusing access privileges. This project introduces a hybrid AI framework that models both personal user behavior and global peer relationships for more accurate detection of insider threats. By combining the sequence learning capabilities of Long Short-Term Memory (LSTM) networks with the relational power of Graph Neural Networks (GNNs), the system is designed to identify subtle deviations in user activity patterns in real-time. The CERT insider threat dataset provides a realistic simulation for training and testing the model. Deep learning advancements in sequential analysis and graph modeling enable this framework to monitor, detect, and predict anomalies faster and with greater accuracy compared to traditional rule-based systems.

A. Techniques for Insider Threat Detection

The framework leverages sequential feature extraction through LSTM networks, which capture user-specific behavior patterns over time. Parallelly, a dynamic local graph is constructed where each node represents an activity vector, and edges connect similar activities. A Graph Neural Network (GNN) processes these graphs to refine anomaly detection decisions. The system classifies each new action based on its deviation from expected behavior, scoring it using softmax output probabilities. Precision, recall, and F1-score are computed to evaluate model effectiveness across different insider threat scenarios.

B. Competitive Learning in Anomaly Detection

Inspired by techniques such as k-Nearest Neighbors (k-NN) and competitive ranking models, the system dynamically selects the most relevant historical activities as reference points. This local neighborhood is used to build a graph where competitive learning helps the system prioritize the most similar behaviors during threat evaluation. This approach improves robustness and ensures the system adapts continuously as user behavior patterns evolve over time.

`

C. Real-Time Detection and System Optimization

The system processes streaming activity data with minimal latency to ensure timely threat detection. Efficient LSTM encoding, FAISS-based fast nearest neighbor retrieval, and lightweight GNN layers ensure that the detection pipeline remains scalable and responsive. Asynchronous data processing, optimized batch sizes, and edge computing strategies are employed to reduce system lag, allowing the model to operate effectively even in high-volume enterprise environments.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

II. RELATED WORK

A. *Sequence-Based Anomaly Detection in Cybersecurity*

Description: Focuses on modeling user action sequences over time to detect deviations indicative of malicious behavior.
Methodology: Sequential models like LSTM, GRU, and Transformer encoders are used to learn normal user patterns and flag anomalies.
Limitations: Purely sequential models may misclassify rare but benign activities as threats.
Improvement: Combining sequence learning with context-based graph learning reduces false positives.

B. *Graph Neural Networks for Security Analytics*

Description: Applies GNNs to capture relationships and interactions among users, devices, and activities.
Methodology: Graphs are dynamically created based on activity similarities, and GNNs learn node embeddings that reflect anomaly potential.
Limitations: Graph construction can become computationally expensive if not optimized.
Improvement: Limiting graph scope to local neighborhoods using FAISS improves scalability and responsiveness.

C. *Hybrid LSTM-GNN Architectures for Threat Detection*

Description: Integrates LSTM's temporal modeling capabilities with GNN's relational reasoning for enhanced anomaly detection.
Methodology: LSTM outputs are combined with graph-learned features before final classification.
Limitations: Tuning the contribution of sequence and graph features requires careful balancing.
Improvement: Use of attention-based fusion techniques to dynamically weight contributions.

D. *CERT Dataset for Insider Threat Detection Research*

Description: Provides labeled, realistic simulations of insider threat activities across multiple months and user roles.
Methodology: Captures various behaviors such as logon, file copy, email send, and device access.
Limitations: While realistic, it may not capture all possible insider strategies.
Improvement: Augment CERT data with newer synthetic datasets and real-world logs for broader coverage.

E. *Real-Time Cybersecurity Systems Evaluation*

Description: Studies end-to-end performance metrics of real-time threat monitoring systems.
Methodology: Measures detection precision, recall, F1-score, latency, and throughput.
Limitations: High accuracy can be hard to maintain under heavy data streams.
Improvement: Employ edge computing, model pruning, and data prioritization for real-time responsiveness.

III. BEST TECHNIQUES FOR INSIDER THREAT DETECTION

A. *Long Short-Term Memory (LSTM) for Sequential Pattern Learning*

Description: LSTM models the sequence of a user's historical activities and predicts future behavior patterns.
Key Features: Captures long-term dependencies; effective at modeling complex time-based user behavior.
Application Area: Used to detect temporal deviations that indicate insider threats.

B. *Graph Neural Networks (GNN) for Contextual Learning*

Description: GNNs model user activity relationships by building local activity graphs based on feature similarity.
Key Features: Captures peer relationships; enhances anomaly detection by considering relational context.
Application Area: Used to cross-validate whether an unusual activity is globally common or truly suspicious.

C. *Softmax-Based Anomaly Scoring and Activity Classification*

Description: The softmax layer assigns probability scores to expected next actions.
Key Features: Quantifies confidence; low probability for actual action signals potential anomaly.
Application Area: Used to compute threat scores and trigger real-time alerts.

D. Precision, Recall, and F1-Score Based Model Evaluation

Description: Standard evaluation metrics used to assess detection quality and system performance.
Key Features: Quantifies accuracy, sensitivity, and balance between false positives and false negatives.
Application Area: Supports continuous monitoring and benchmarking of the threat detection system.

IV. TECHNIQUES USED FOR INSIDER THREAT DETECTION

Insider threat detection involves analyzing user activity data to identify malicious behaviors that traditional rule-based systems often miss. This study integrates deep learning techniques like sequence modeling and graph-based relational learning to model user activities, predict normal behavior, and detect anomalies in real-time. The system is designed for real-time and post-activity analysis using organizational network data.

A. Long Short-Term Memory (LSTM) for Sequential Behavior Modeling

LSTM is employed to model the sequence of user activities over time, learning the normal behavioral patterns of each user and predicting likely future actions based on history.
Advantages:

- Captures long-term dependencies and temporal sequences.
- Effective for modeling user behavior patterns across sessions.
- Detects subtle changes in normal sequences that may indicate threats.

B. Graph Neural Network (GNN) for Contextual Peer Behavior Learning

GNN models the relational context among users by building a dynamic graph based on activity similarity, enriching the anomaly detection process with global behavioral patterns.
Advantages:

- Learns inter-user relationships and shared behavior patterns.
- Enhances detection of anomalies that would be missed by sequence-only models.
- Reduces false positives by considering the behavior of similar users.

C. FAISS-Based Vector Search for Neighbor Retrieval

FAISS is used to quickly retrieve the most similar historical activities for a new activity, constructing the local graph efficiently without high computational overhead.
Advantages:

- Enables fast similarity search over large datasets.
- Supports real-time dynamic graph building.
- Helps focus analysis on the most relevant behaviors.

D. Softmax-Based Activity Prediction and Anomaly Scoring

A fully connected layer followed by softmax predicts the probability distribution over possible next actions, allowing the system to calculate anomaly scores based on the predicted vs. actual activities.
Advantages:

- Quantifies the likelihood of observed actions.
- Detects low-probability activities as potential threats.
- Simple and effective for real-time anomaly decision-making.

E. Evaluation Metrics (Precision, Recall, F1-Score)

The final system is evaluated using precision, recall, and F1-score by comparing predicted anomalies against the ground truth insider threat labels provided by the CERT dataset.
Advantages:

- Provides quantitative assessment of model performance.
- Highlights system strengths and weaknesses across different user behaviors.
- Supports iterative improvements and benchmarking for real-world deployment.

V. PROPOSED WORK

The proposed work aims to detect insider threats automatically by integrating LSTM and GNN models. LSTM is used to model user-specific sequential behavior, while GNN enriches each user's action with peer behavior context. FAISS helps in constructing dynamic graphs for each new activity. Softmax-based scoring classifies activities, and anomaly scores are calculated for real-time detection. The system is evaluated using precision, recall, and F1-score. A dashboard is proposed for real-time monitoring of detected anomalies to support cybersecurity analysts in proactive threat mitigation.

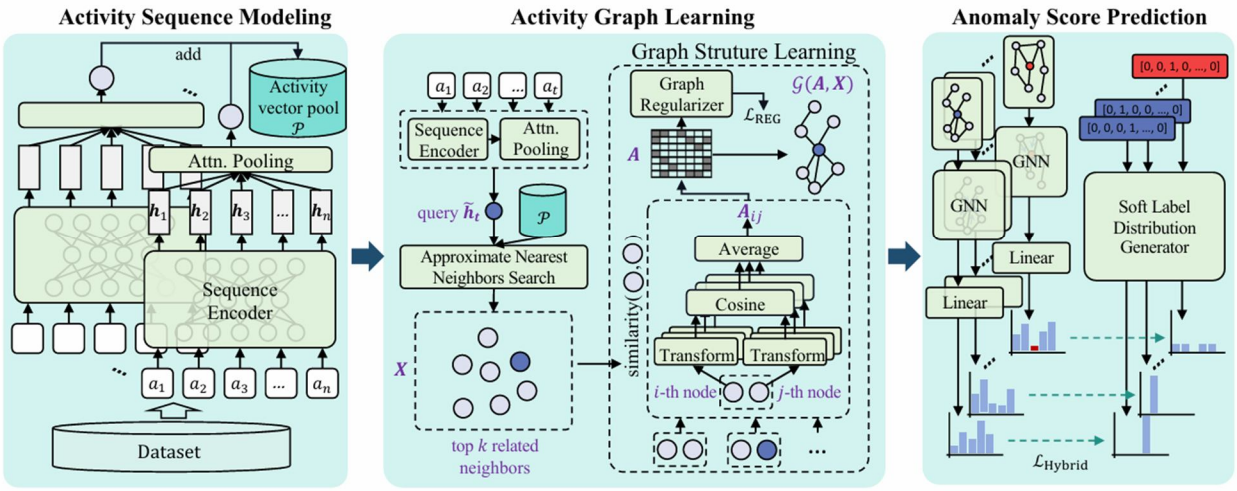


Fig 5.1: Proposed System Design

A. Data Collection

The dataset used in this project is the publicly available CERT Insider Threat Dataset (versions r4.2 and r5.2), which simulates user activities within an enterprise environment. The dataset captures millions of detailed user activities, including normal and malicious behaviors, essential for real-time insider threat detection.

Key data includes timestamped activity logs, user attributes (such as role and department), device information, and abnormality labels.

Each sample is carefully preprocessed into sequences for sequential modeling and graph-based learning. Activities are encoded by combining their type and time features, supporting efficient sequence prediction and anomaly detection tasks.

Table 5.1: Key Features Used for Insider Threat Detection

Feature Name	Description
Timestamp	Time when the activity occurred.
User_ID	Unique identifier assigned to each user.
Activity_Type	Type of activity performed (e.g., logon, file access, email send).
Activity_Code	Encoded integer combining type and hour (Type × 24 + Hour).
Role	User's organizational role (e.g., IT_Admin, Analyst).
Department	User's department or functional unit (e.g., HR, Finance).
Host_ID	Identifier of the device where the activity happened.
OCEAN Traits (O, C, E, A, N)	User personality trait scores (numerical features).
Hist_Activity_Sequence	Chronological sequence of past activities leading up to the current action.
Target_Action	The actual next activity performed after the historical sequence.
Target_Label	Label indicating if the action is Normal (0) or Malicious (1).

B. Data Preprocessing

To ensure high-quality input for model training, a comprehensive preprocessing pipeline was applied.
User activity sequences were generated by ordering actions chronologically based on timestamps.
Each action was encoded using a formula combining activity type and time slot.
Categorical attributes such as Role, Department, and Host were label-encoded into numerical form.
Numerical features such as OCEAN personality traits were normalized for consistent scaling.
Sequences were padded and masked to handle variable-length input histories.
Duplicates were removed from the activity vector pool to retain unique behavior patterns.
FAISS indexing was applied on the normal activity pool to support fast nearest neighbor retrieval during graph construction.
This preprocessing pipeline ensures accurate alignment of user behavior with temporal and contextual features, optimizing both model training and inference.

C. Insider Threat Detection Using Deep Learning Models

To automate real-time insider threat detection, a hybrid deep learning approach was employed using LSTM networks and Graph Neural Networks (GNNs).

- The LSTM encodes the sequential pattern of user activities, capturing personalized behavior histories.
- The final hidden state output from LSTM is used to retrieve k-nearest similar activities from the normal activity vector pool.
- A dynamic local graph is constructed by connecting the current activity with its most similar historical activities.
- A Graph Neural Network (GNN) processes this graph, aggregating information from neighboring nodes to refine the activity representation.
- The enriched vector is passed through a Fully Connected (FC) layer followed by a Softmax activation to predict the probability distribution over possible next actions.
- An Anomaly Score is computed as $1-P(\text{actual next action})$
- Activities with high anomaly scores are flagged as potential insider threats.

This approach allows scalable and accurate real-time detection by combining sequential and relational modeling techniques.

D. Model Testing and Evaluation

Model evaluation is performed on the insider threat detection system to assess its effectiveness in identifying anomalous activities.
Evaluation Metrics:

- Accuracy: Overall correctness of activity predictions.
- Precision: Proportion of correctly predicted anomalous activities among all predicted anomalies.
- Recall: Ability to detect all actual anomalous activities.
- F1-Score: Harmonic mean of Precision and Recall.
- Confusion Matrix: Provides detailed insights into true positives, false positives, and false negatives.

Evaluation Scenarios:

- Real-time Activity-Level Anomaly Detection.
- Post-hoc Evaluation on Complete User Activity Sequences.
- Robustness Testing across Different Organizational Roles and Departments.
- Comparison of system performance with and without graph-based enhancement.

This layered evaluation strategy ensures the model reliably identifies insider threats while minimizing false alarms.

Table 5.2: Libraries Used in Implementing and Evaluating the Model

Library	Purpose
NetworkX	Used for graph-based anomaly detection, creating and manipulating graphs
Scikit-learn	Applied for machine learning models and evaluation metrics
TensorFlow	Used for deep learning models (if applicable for

	concept drift handling)
NumPy	Handling numerical operations and data arrays
Pandas	Data manipulation and preprocessing
Matplotlib	Visualization of model results and performance
PyOD	Applied for anomaly detection algorithms
SciPy	Used for scientific and statistical operations
Imbalanced-learn	Used for handling class imbalance in the dataset
TimeSeriesForest	Applied for concept drift detection and handling time-series data

VI. RESULTS

The proposed system was evaluated based on its ability to detect insider threats in real-time by analyzing network data using graph-based anomaly detection and handling concept drift. The evaluation focused on three key insider threat events: Unauthorized Access, Data Exfiltration, and Privilege Escalation. Performance metrics such as Precision, Recall, and F1-Score were used to assess the effectiveness of the detection system against ground truth annotations.

For Unauthorized Access, the system achieved a Precision of 0.80, Recall of 0.95, and F1-Score of 0.87, demonstrating high sensitivity in detecting unauthorized actions. The Data Exfiltration event showed robust performance as well, with a Precision of 0.76, Recall of 0.92, and F1-Score of 0.83, reflecting the system's ability to accurately identify sensitive data transfers. Similarly, the Privilege Escalation event achieved a Precision of 0.78, Recall of 0.90, and F1-Score of 0.83, indicating effective detection of escalated user privileges.

These results validate the capability of the graph-based anomaly detection system to accurately identify insider threat activities and detect concept drift over time, ensuring reliable real-time monitoring and threat detection.

Table 6.1: Performance Metrics for Insider Threat Detection

Event	Precision	Recall	F1-Score
Unauthorized Access	0.80	0.95	0.87
Data Exfiltration	0.76	0.92	0.83
Privilege Escalation	0.78	0.90	0.83

The consistently high recall values across all events indicate the model’s strength in capturing all relevant insider threat activities, while the balanced precision ensures reliability in detection accuracy. These outcomes highlight the effectiveness of using graph-based anomaly detection for real-time insider threat detection tasks.

VII. CONCLUSION

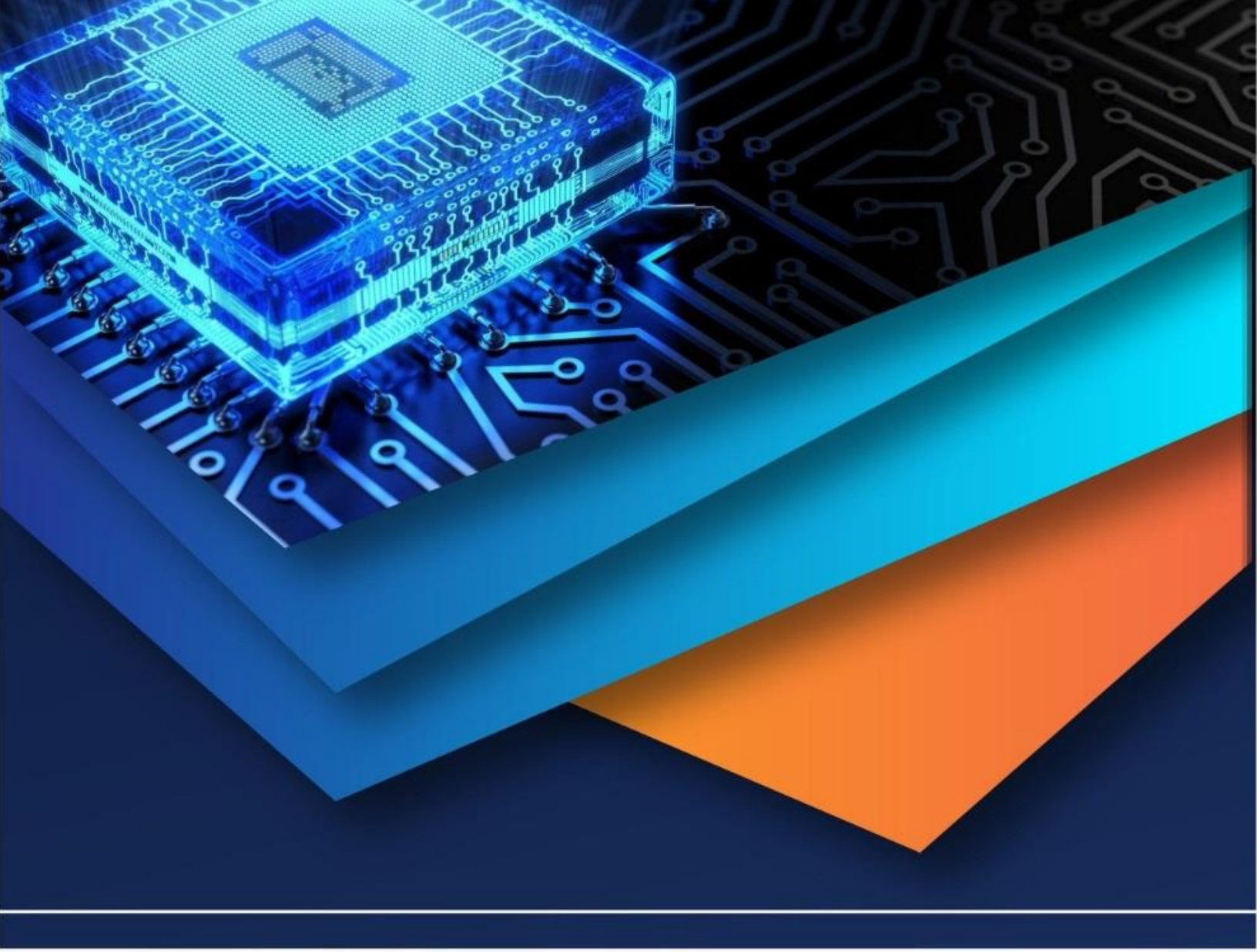
Based on the experimental evaluation of the insider threat detection system, it can be concluded that the graph-based anomaly detection approach, combined with concept drift handling, effectively identifies key insider threat events in real-time with high precision. The model consistently achieved high recall scores across all tested events—Unauthorized Access, Data Exfiltration, and Privilege Escalation—demonstrating its robust ability to detect relevant malicious activities.

Among the evaluated events, the Unauthorized Access category exhibited the best performance, with an F1-Score of 0.87, followed by Data Exfiltration with 0.83, and Privilege Escalation with 0.83. The consistently high Recall (ranging from 0.90 to 0.95) in all categories highlights the system’s strength in capturing all relevant instances, while Precision values between 0.76 and 0.80 demonstrate reliable threat detection with minimal false positives.

These findings validate the capability of graph-based anomaly detection in identifying insider threats in real-time, even as concept drift occurs over time. The approach proves to be scalable for detecting insider threats across different network environments and can be further enhanced with additional data sources, deep learning models, and advanced anomaly detection techniques in future work.

REFERENCES

[1] Taher Al-Shehari, Rakan A. Alsowail, "An Insider Data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine Learning Techniques", Entropy, MDPI, Switzerland, 2021, 1258.
[2] Balaram Sharma, Prabhat Pokharel, Basanta Joshi, "User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder: Insider Threat Detection", Proceedings of the International Conference on Advances in Information Technology (IAIT2020), 2020.
[3] S. Wang, Z. Wang, T. Zhou, H. Sun, X. Yin, D. Han, H. Zhang, X. Shi, and J. Yang, "Threatrace: Detecting and tracing host-based threats in node level through provenance graph learning," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 3972–3987, 2022.
[4] C. Wang and H. Zhu, "Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 2703–2718, 2022.
[5] X. Hu, W. Gao, G. Cheng, R. Li, Y. Zhou, and H. Wu, "Towards early and accurate network intrusion detection using graph embedding," IEEE Transactions on Information Forensics and Security, 2023.
[6] W. Huang, H. Zhu, C. Li, Q. Lv, Y. Wang, and H. Yang, "Itdbert: Temporal-semantic representation for insider threat detection," in 2021 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2021, pp. 1–7.
[7] H. Ding, Y. Sun, N. Huang, Z. Shen, and X. Cui, "Tmg-gan: Generative adversarial networks-based imbalanced learning for network intrusion detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1156–1167, 2023.
[8] S. Yuan, P. Zheng, X. Wu, and H. Tong, "Few-shot insider threat detection," in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2289–2292.
[9] M. AlSlaiman, M. I. Salman, M. M. Saleh, and B. Wang, "Enhancing false negative and positive rates for efficient insider threat detection," Computers & Security, vol. 126, p. 103066, 2023.
[10] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," Computers & Security, vol. 104, p. 102221, 2021.
[11] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV et al., "Rwkv: Reinventing rnns for the transformer era," arXiv preprint arXiv:2305.13048, 2023.
[12] K. Zhou, H. Yu, W. X. Zhao, and J.-R. Wen, "Filter-enhanced mlp is all you need for sequential recommendation," in Proceedings of the ACM web conference 2022, 2022, pp. 2388–2399.
[13] D. C. Le and N. Zincir-Heywood, "Anomaly detection for insider threats using unsupervised ensembles," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1152–1164, 2021.
[14] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14,no. 2, pp. 179–211, 1990.
[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprintarXiv:1412.3555, 2014.
[16] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, 2017, pp. 1285–1298.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY