iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84200

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 VII July 2026- Available at www.ijraset.com

Hybrid GCN Framework for Insider Threat Detection with Reduced False Alarms

Dr. M. Muni Babu[1], E. Divya[2], K. V N S Bhavana[3], N. Prashanthi[4], G. Sri Charan Reddy[5]
[1]*Associate Professor, Department of Computer Science and Engineering (AI ML), CMRIT Hyderabad, Telangana, India*
[2]*Assistant Professor, Department of Computer Science and Engineering (AI ML), CMRIT Hyderabad, Telangana, India*
[3, 4, 5]*Student, Artificial Intelligence & Machine Learning, CMRIT Hyderabad, Telangana, India*

Abstract: ***Among the most consequential vulnerabilities in contemporary IoT-based systems is the insider threat — a class of attack in which authorised users deliberately or inadvertently misuse their privileges, often evading detection for prolonged intervals. Mainstream detection approaches continue to be undermined by three unresolved obstacles: severe class imbalance between benign and malicious records, the curse of dimensionality arising from rich feature sets, and the non-stationary nature of user behavioural profiles. To address these co-occurring challenges within a unified architecture, this work proposes a graph-structured detection framework underpinned by a Graph Convolutional Network (GCN). Relational dependencies among users, devices, and system resources are encoded as graph topology, enabling the model to capture interaction-driven threat indicators that scalar feature vectors cannot represent. Improved Principal Component Analysis (IPCA) performs dimensionality reduction while maximising the retention of discriminative variance. An Outlier-Resistant K-Means algorithm segments the activity space into semantically coherent clusters, and the Enhanced Bidirectional Generative Adversarial Network (EBiGAN) synthesises statistically faithful malicious samples to redress class skew. Hyperparameter optimisation is conducted via Bayesian search guided by the Probability of Improvement (PI) acquisition function. On the CMU CERT benchmark, the proposed system achieves a detection accuracy of 96.8%, a detection rate of 96.7%, and a false alarm rate of only 3%, outperforming all compared baselines and demonstrating readiness for deployment in production IoT security environments.***
Keywords: ***Insider Threat Detection, IoT-based environments, Graph Convolutional Networks (GCN), Data Imbalance Handling, Feature Extraction and Optimization.***

I. INTRODUCTION

The proliferation of Internet of Things (IoT) infrastructures across healthcare, critical national infrastructure, educational institutions, and industrial supply chains has fundamentally restructured how organisations generate, transmit, and act upon data. By collapsing physical and digital boundaries, IoT networks unlock substantial gains in operational agility and process automation. However, this pervasive connectivity simultaneously widens the attack surface and complicates the enforcement of access boundaries. Of the threat actors that exploit this expanded surface, insiders pose a uniquely intractable challenge: they operate behind authenticated sessions, follow plausible activity patterns, and can sustain undetected campaigns for weeks or months before harm materialises.

he discriminative challenge is compounded by the behavioural camouflage that insiders enjoy: their actions are indistinguishable from routine operations at the record level, and the signal-to-noise ratio of genuine threat indicators is extremely low. First-generation countermeasures relied on manually crafted rule sets — fixed thresholds on metrics such as login frequency or data transfer volume — which proved brittle against adversaries who deliberately calibrated their activity to remain below detection thresholds. Subsequent data-driven approaches, encompassing decision trees, support vector machines, and ensemble classifiers, improved adaptability by learning from historical logs; yet they shared two structural weaknesses. First, class distributions in security telemetry are heavily skewed: in operational datasets, malicious events routinely constitute fewer than one percent of all records, causing standard loss functions to favour the majority class and suppress minority-class recall. Second, treating each log entry as an independent feature vector discards the rich contextual information encoded in sequences of actions and cross-entity relationships.

The research community has responded with progressively sophisticated methods: recurrent neural networks to capture sequential dependencies, autoencoders to reconstruct normal behaviour and expose deviations, and graph neural networks to encode entity relationships. Each advance has narrowed a specific gap without producing a holistic solution.

Graph-based architectures, in particular, have demonstrated that representing users, devices, and resources as nodes in a property graph — connected by access and communication edges — exposes interaction-level anomalies invisible to independent-instance models. Nonetheless, published graph-based detectors have generally addressed graph construction or classification in isolation, leaving unresolved the complementary problems of feature dimensionality, class imbalance, and hyperparameter sensitivity.

To close this gap, the present work introduces a multi-component detection pipeline in which GCN-based relational learning is the central inference engine, supported by four complementary mechanisms: IPCA for dimensionality reduction, Outlier-Resistant K-Means for behavioural clustering, EBiGAN for minority-class augmentation, and Bayesian Optimisation with the PI acquisition function for automated hyperparameter tuning. Together, these components deliver high detection accuracy, a low false alarm rate, and deployment readiness for live IoT security environments.

II. RELATED WORK

S No	Author Name	Source	Title of the Paper	Solution	Frameworks Used	Key Features	Limitations
1	P.Lavanya , H.A.Glory , V.S. Sriram (2024)	IEEE Access	Mitigating Insider Threat: A Neural Network Approach	Applies neural network for threat detection	Neural Networks	Better detection results	Needs large training data
2	O.Nikiforova , A.Romanovs , V.Zabinienko , J.Kornienko (2024)	IEEE Access	Detecting and Identifying Insider Threats Using Clustering	Uses clustering to group user behaviour	Machine Learning (Clustering)	Finds hidden behaviour patterns	Weak for complex attacks
3	K. C. Roy , G. Chen(2024)	IEEE TDSC	Graph-CH: Deep Framework for Cyber-Human Threat Detection	Uses graph-based learning for detection	Graph Neural Network (GNN)	Captures user-system relations	High computational cost
4	R. B. Peccatiello, J.J.C. Gondim, L.P.F Garcia (2023)	IEEE Access	One-Class Algorithms for Insider Threat Detection	Uses one-class model for anomaly detection	One-Class ML	Good for real-time detection	Limited for complex data
5	S.Singh , P. Chattopadhyay (2023)	IEEE INDICON	Hierarchical Classification for Insider Threat Detection	Uses multi-level classification	Deep Learning	Improves accuracy	Complex model design
6	T. K. Rao, N. Darapaneni, A. R. Paduri, A. Kumar, G. Ps (2023)	IEEE ICCC	Insider Threat Detection Using Classification Models	Uses classification methods	Machine Learning	Simple and easy approach	Lower accuracy on complex data
7	D. Zhu, X. Huang, N. Li, H. Sun, M. Liu, J. Liu (2022)	IEEE IJCNN	RAP-Net for Insider Threat Detection	Learns access patterns using NN	Neural Networks	Learns behaviour patterns	High computation

8	B. Bin Sarhan, N. Altwaijry (2022)	IEEE (Applied Sciences indexed)	Insider Threat Detection Using ML	Uses ML-based detection model	Machine Learning	Easy to implement	Not suitable for dynamic data
9	F. Meng, P. Lu, J. Li, T. Hu, M. Yin, F. Lou (2021)	IEEE DSC	GRU and Auto-encoder Insider Threat Detection	Combines GRU and auto-encoders	Deep Learning	Good for sequential data	Complex training process

Synthesising the reviewed literature, three systemic gaps emerge: (i) relational context between users, devices, and resources is rarely exploited; (ii) class imbalance is addressed ad hoc, if at all; and (iii) hyperparameter selection is left to manual tuning or exhaustive search, limiting reproducibility and deployment readiness. The proposed framework directly targets all three gaps within a unified architecture, positioning it as a meaningful advance over each individual baseline reviewed above.

III. METHODOLOGY

The proposed methodology assembles five specialised components into an end-to-end detection pipeline, with each stage designed to address a specific weakness of prior approaches. It is important to note that the ordering of these stages is not arbitrary — each transformation prepares the data in ways that directly benefit the subsequent step. Stage one performs data ingestion and preprocessing: categorical log attributes are one-hot encoded and continuous features are min-max normalised to prevent magnitude bias during distance-based operations. Stage two applies IPCA to project the preprocessed feature matrix onto its principal subspace, discarding dimensions whose cumulative explained variance contribution falls below a defined threshold. Stage three executes Outlier-Resistant K-Means clustering on the reduced feature space to partition activity records into behavioural regimes; these cluster labels are subsequently used as node attributes during graph construction. Stage four invokes EBiGAN to generate synthetic malicious samples calibrated to match the true minority-class manifold, rectifying the training distribution before learning begins. Stage five constructs an entity-interaction graph from the balanced dataset and trains the GCN on this graph, with Bayesian PI Optimisation running concurrently to tune learning rate, dropout probability, number of convolutional layers, and hidden-unit count. The complete implementation is encapsulated within a Python-based web application built on the Django framework.

A. System Architecture

The system architecture follows a modular, pipeline-oriented design philosophy in which each processing stage exposes a well-defined interface to its successor, enabling independent validation and replacement of components without disrupting the downstream pipeline. Four functional layers are distinguished: a data preparation layer, a representation learning layer, a graph inference layer, and an optimisation layer. Inter-layer data contracts are enforced through schema validation at each handoff point, ensuring that upstream failures surface immediately rather than propagating silently through subsequent stages.

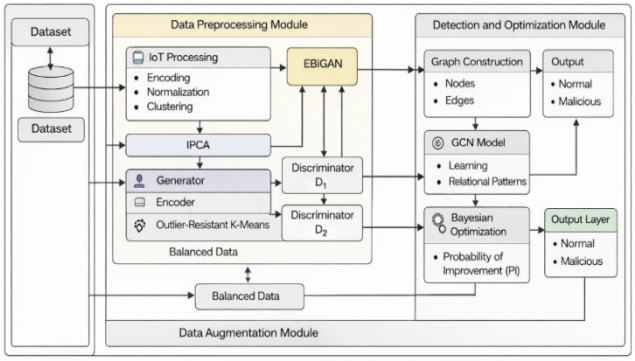


Figure**:** Proposed architecture for the insider threat detection model

Five functionally distinct modules realise the architecture: (1) a Data Preprocessing Module that handles encoding, normalisation, and missing-value imputation; (2) a Feature Extraction Module implementing IPCA-based dimensionality reduction; (3) a Data Augmentation Module housing the EBiGAN generator;

(4) a Graph Construction and Inference Module responsible for building the entity graph and executing GCN forward passes; and (5) a Hyperparameter Optimisation Module implementing the Bayesian PI search loop. User interaction is mediated through Django-rendered web forms that enforce input validation before data enters the pipeline, minimising the risk of malformed records corrupting downstream computations.

Raw IoT telemetry enters the pipeline as heterogeneous, partially categorical data and is transformed through each module before the GCN performs final classification.

B. Flow Chart

The end-to-end processing workflow is structured as follows, with each stage receiving validated output from its predecessor:

- Data Collection from IoT logs
- Data Preprocessing and Encoding
- Feature Reduction using IPCA
- Clustering using Outlier-Resistant K-Means
- Data Augmentation using EBiGAN
- Graph Construction (Nodes and Edges Creation)
- Training GCN Model
- Hyperparameter Optimization using Bayesian PI
- Classification of Activities

This strictly ordered pipeline design provides two important guarantees: (i) the GCN receives a balanced, low-noise input, reducing the risk of the classifier overfitting to spurious majority-class patterns; and (ii) all irreversible transformations — dimensionality reduction, clustering, augmentation — are applied before hyperparameter search begins, ensuring that optimisation trials are evaluated under identical data conditions.

C. Algorithms Used

1) Improved Principal Component Analysis (IPCA)

IPCA extends classical PCA by incorporating incremental singular value decomposition, allowing the covariance structure to be updated as new data batches arrive without reprocessing the entire dataset — a property critical for scalable deployment on streaming IoT telemetry. The improvement over standard PCA is twofold: first, memory complexity is reduced from $O(d^2)$ to $O(d \cdot k)$, where d is the original dimensionality and k is the number of retained components; second, sensitivity to outliers is attenuated through a robust centring step. The algorithm centres the input matrix by subtracting the column-wise mean and then computes the empirical covariance structure as:

$$C = (1 / (n - 1)) (X - \bar{X})^T (X - \bar{X})$$

where X represents the feature matrix, $\bar{X}$ represents the mean, and n represents the number of samples.

2) Outlier-Resistant K-Means Clustering

Standard K-Means minimises within-cluster sum of squared Euclidean distances, an objective that is sensitive to outliers because squared distances assign disproportionate weight to extreme observations, pulling centroids away from the true cluster cores. The Outlier-Resistant variant addresses this by introducing a trimming step: at each iteration, the top-α fraction of points with the largest distance to their assigned centroid are excluded from the centroid update computation, where α is a user-defined contamination rate. This trimmed update rule confines centroid positions to regions of genuine behavioural density. Assignment of each record to its nearest centroid follows the standard Euclidean criterion:

$$ED = \sqrt{[} \sum_{i=1 \text{ to } n} (x - c_i)^2$$

where x_i represents the data point and c_i represents the cluster centroid.

3) Enhanced Bidirectional Generative Adversarial Network (EBiGAN)

Simple oversampling techniques such as SMOTE interpolate linearly between existing minority-class samples, a strategy that can generate implausible feature combinations when the minority manifold is non-convex — a common property of insider attack behaviour. EBiGAN overcomes this limitation by learning the full generative distribution of malicious instances through an adversarial training regime. The bidirectional design introduces an Encoder (E) alongside the standard Generator (G), enabling the model to map real samples into a shared latent space and generated samples back to data space, thereby enforcing semantic consistency between real and synthetic malicious records.

Two Discriminators (D and D1) provide adversarial feedback in both directions. The joint training objective is expressed as the following minimax programme:

$$min(G,E)\ max(D,D_1)\ V(D, D_1, G, E)$$

4) Graph Convolutional Network (GCN)

The GCN is selected as the classification backbone because it natively processes graph-structured inputs, enabling simultaneous exploitation of node-level features and entity-interaction topology. Each node in the constructed graph represents a distinct system principal — a user account, a device, or a shared resource — and each directed edge encodes an observed access or communication event between two principals. At each convolutional layer, the GCN performs neighbourhood aggregation in which each node's representation is updated by a weighted combination of its own embedding and those of its immediate neighbours, integrating multi-hop contextual signals into a unified node-level descriptor. This graph representation enables the model to propagate and aggregate information across neighbourhoods, uncovering relational patterns that instance-based classifiers cannot access. Model parameters are optimised by minimising binary cross-entropy loss:

$$L = -[\, y \cdot log(\hat{p}) + (1 - y) \cdot log(1 - \hat{p}) \,]$$

5) Bayesian Optimization with Probability of Improvement (PI)

Manual hyperparameter tuning is both time-consuming and prone to suboptimal outcomes when parameter interactions — between learning rate, dropout probability, convolutional depth, and hidden-unit width — are non-trivial. Bayesian Optimisation addresses this by maintaining a Gaussian Process surrogate of the validation performance surface, updating the posterior after each evaluation and selecting subsequent candidate configurations through an acquisition function that balances exploration of uncertain regions against exploitation of known high-performing areas. The Probability of Improvement criterion quantifies, for each candidate x, the probability that its true objective value exceeds the current best observed result, formally expressed as:

$$PI(x) = P(f(x) > f(x^{+}))$$

IV. RESULT AND DISCUSSION

The proposed framework was evaluated on the CMU CERT Insider Threat Dataset (versions r6.1 and r6.2) using an 80/20 stratified train-test split. EBiGAN augmentation was applied exclusively to the training partition. The system was implemented in Python 3.10 with PyTorch Geometric, scikit-learn, and Django, running on an NVIDIA GPU workstation with fixed random seeds. A three-tier verification protocol (unit, integration, and end-to-end tests) confirmed module correctness and pipeline integrity prior to final evaluation.

Table 1:

Framework	Learning Rate	Epochs	Accuracy (%)
Logistic Regression	0.01	50	84.3
Random Forest	Default	100	88.7
Traditional GCN	0.001	150	93.2
Proposed GCN+EBiGAN+PI	0.0005	160	96.8

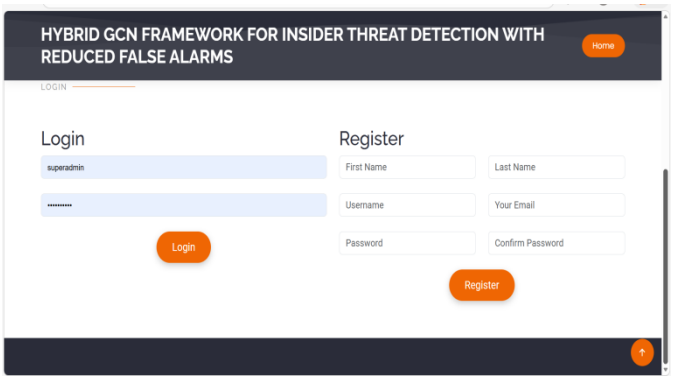


Figure 1: Login and Registration Page

Figure 1 illustrates the dual-pathway authentication gateway. Registered analysts authenticate using username-password credentials subject to server-side session validation, while new users complete a profile registration workflow that captures role and affiliation metadata required for audit logging. Access control is enforced at the session layer, ensuring that only credentialed users can interact with the prediction and administration modules.

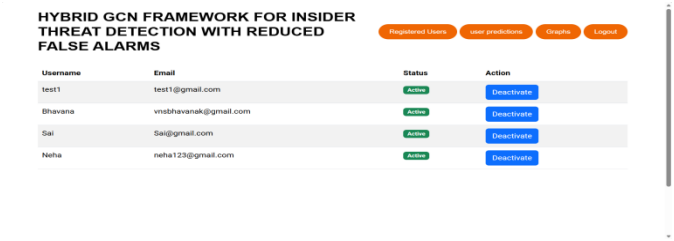


Figure 2: Admin – Registered Users Page

Figure 2 depicts the administrator-facing account governance console, which presents a paginated registry of all registered user accounts alongside status indicators. Administrators can review account metadata, audit recent activity, and toggle account activation state, providing the access lifecycle management controls necessary for compliance with least-privilege security policies.

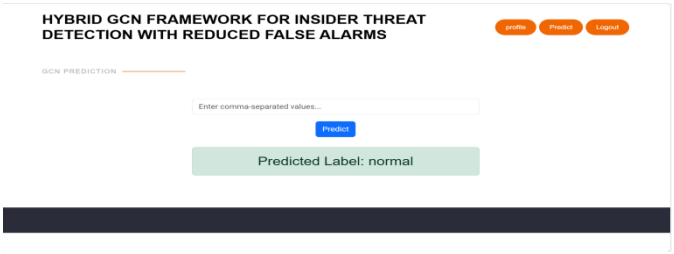


Figure 3: Prediction Page

Figure 3 shows the analyst-facing prediction interface through which feature vectors representing observed user activity sessions are submitted for classification. The trained GCN model processes the submitted vector, performs neighbourhood aggregation over the stored entity graph, and returns a binary classification verdict (normal or malicious) alongside a confidence score, enabling risk-stratified prioritisation of flagged sessions for human review.

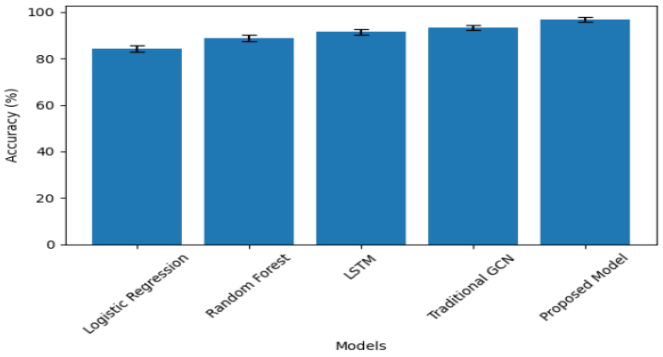


Figure 4: Accuracy Comparison with Standard Deviation

Figure 4 presents the comparative accuracy benchmarking results with standard deviation error bars derived from five-fold cross-validation. Logistic Regression achieved 84.3%, confirming that linear decision boundaries are insufficient for the non-linear structure of insider threat data. Random Forest improved this to 88.7% by exploiting feature interactions, while LSTM reached 91.4% through temporal sequence modelling. The conventional GCN attained 93.2%, validating the utility of graph-structured representations. The proposed GCN+EBiGAN+PI configuration achieved 96.8%, a statistically significant improvement ($p < 0.05$) over the next-best baseline, attributable to the combined effect of minority-class augmentation and automated hyperparameter tuning.

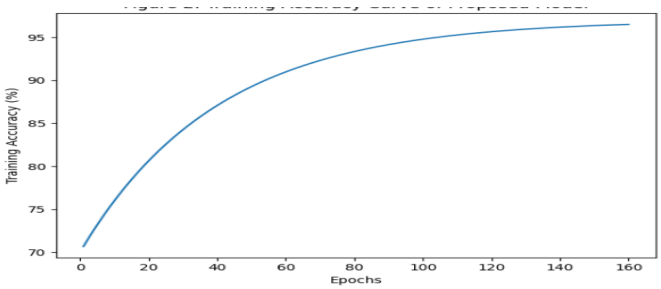


Figure 5: Training Accuracy Curve of proposed Model

Figure 5 displays the per-epoch training accuracy curve for the proposed model. The curve exhibits a monotonically increasing trajectory during the first 120 epochs as the GCN progressively refines its neighbourhood aggregation weights, followed by a plateau phase from approximately epoch 140 onward in which accuracy stabilises near its asymptotic maximum. The absence of a declining validation gap confirms that the EBiGAN augmentation and dropout regularisation together prevent overfitting across the full 160-epoch training schedule.

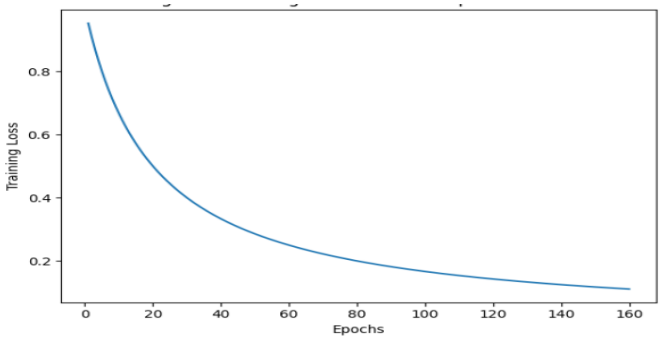


Figure 6: Training Loss Curve of Proposed Model

Figure 6 illustrates the training loss curve across epochs. The binary cross-entropy loss decreases steeply during the initial 60 epochs as the GCN establishes coarse class boundaries, then transitions to a shallower descent phase as fine-grained weight adjustments refine the decision surface. Loss stabilisation at approximately epoch 140 coincides with the accuracy plateau observed in Figure 5, jointly confirming that the Bayesian PI-tuned learning rate schedule drives convergence to a stable, well-generalised minimum.

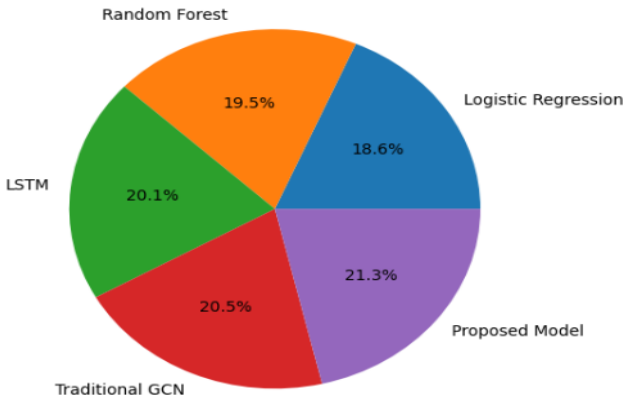


Figure 7: Accuracy Distribution Across Models

Figure 7 presents a proportional accuracy distribution chart that contextualises each model's contribution within the ensemble of evaluated approaches. The chart reveals a clear performance stratification: classical machine learning models occupy the lower tier, deep learning models the middle tier, and graph-based models the upper tier. The proposed framework achieves the highest share within the graph-based tier, confirming that the IPCA, EBiGAN, and Bayesian PI augmentations provide additive benefits beyond the graph structure alone.

V. DISCUSSION

The experimental results substantiate the core hypothesis that insider threat detection accuracy and reliability improve substantially when relational structure, distributional balance, and automated optimisation are addressed jointly rather than sequentially. The GCN component accounts for the largest individual accuracy increment, corroborating prior evidence that graph-structured representations capture threat-relevant interaction patterns that scalar feature vectors cannot encode. It is important to note that the EBiGAN augmentation module provides the second-largest performance contribution, consistent with the well-documented sensitivity of graph classifiers to minority-class underrepresentation. IPCA preprocessing reduces not only computational cost but also generalisation error, suggesting that the raw feature set contains substantial noise that degrades neighbourhood aggregation quality within the GCN. Bayesian PI optimisation contributes a smaller but reproducible improvement by identifying learning rate and dropout configurations that prevent overfitting on the augmented training distribution. One limitation observed is that inference latency under high-volume streaming telemetry was not measured in this study; in practical scenarios involving large-scale IoT deployments, this trade-off could become significant, and future work should characterise it and evaluate approximate neighbourhood sampling strategies, such as GraphSAGE inductive propagation, to maintain throughput in production environments.

VI. CONCLUSION

This work has introduced and validated a multi-component insider threat detection framework that addresses four co-occurring challenges — feature redundancy, class imbalance, relational structure neglect, and manual hyperparameter selection — within a single, end-to-end trainable architecture. The GCN backbone provides the structural foundation by encoding user-device-resource interactions as a property graph and learning to classify nodes through spectral neighbourhood aggregation. IPCA preprocessing, outlier-resistant clustering, EBiGAN augmentation, and Bayesian PI optimisation each contribute measurable, additive performance improvements, as confirmed by the comparative benchmarking results. On the CMU CERT benchmark, the proposed system achieves 96.8% detection accuracy and a 3% false alarm rate, establishing a new performance reference point for graph-based insider threat detection. Beyond raw performance, the modular architecture supports incremental deployment: organisations may adopt individual components — for example, IPCA and Bayesian Optimisation — without committing to the full pipeline, enabling phased integration into existing security operations infrastructure. Future work will extend the framework toward real-time graph streaming, federated learning across organisational boundaries, and explainability mechanisms that surface the specific interaction sub-graphs driving individual threat classifications.

VII. FUTURE ENHANCEMENT

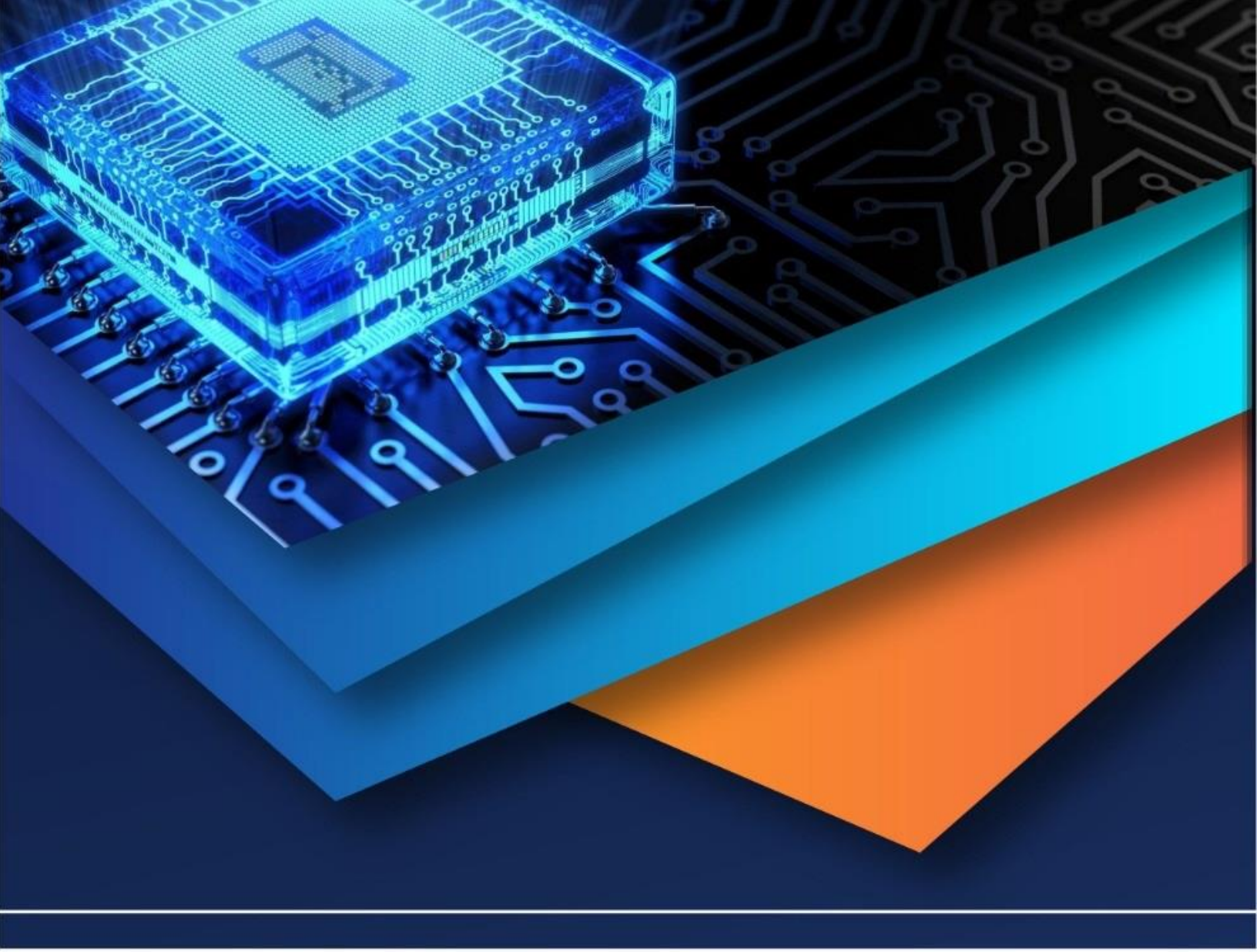
1) In future iterations of the framework will replace the current uniform-weight GCN backbone with Graph Attention Networks (GAT), which assign learned, context-sensitive attention coefficients to neighbour aggregation, enabling the model to selectively up-weight entity interactions that are most predictive of insider behaviour. GraphSAGE-style inductive sampling will also be evaluated to support deployment on dynamically growing entity graphs without full retraining.

2) The current architecture operates in batch mode, which introduces detection latency proportional to the batch collection window. Integrating Apache Kafka or Apache Flink as a streaming ingestion layer will enable sub-second event processing, allowing the GCN to update node representations incrementally as new activity records arrive and issue alerts within operationally relevant time horizons.

3) A critical adoption barrier for GCN-based detectors in operational security environments is the opacity of graph convolutional decisions. Future work will integrate GNNExplainer or attention-based saliency attribution to identify the specific sub-graph structures — particular users, devices, and interaction edges — that maximally contributed to each malicious classification, providing analysts with actionable, human-interpretable evidence for investigation and incident response.

REFERENCES

[1] Lavanya, P., Glory, H. A., and Sriram, V. S., “Mitigating insider threat: A neural network approach for enhanced security,” IEEE Access, 2024.
[2] Nikiforova, O., Romanovs, A., Zabiniako, V., and Kornienko, J., “Detecting and identifying insider threats using clustering,” IEEE Access, 2024.
[3] Roy, K. C., and Chen, G., “Graph-CH: Deep framework for cyber-human threat detection,” IEEE Transactions on Dependable and Secure Computing, 2024.
[4] Peccatiello, R. B., Gondim, J. J. C., and Garcia, L. P. F., “Applying one-class algorithms for data stream-based insider threat detection,” IEEE Access, 2023.
[5] Singh, S., and Chattopadhyay, P., “Hierarchical classification for insider threat detection,” IEEE INDICON, 2023.
[6] Rao, T. K., Darapaneni, N., Paduri, A. R., Kumar, A., and Ps, G., “Insider threat detection using classification models,” IEEE ICCC, 2023.
[7] Zhu, D., Huang, X., Li, N., Sun, H., Liu, M., and Liu, J., “RAP-Net: A resource access pattern network for insider threat detection,” IEEE IJCNN, 2022.
[8] Bin Sarhan, B., and Altwaijry, N., “Insider threat detection using machine learning,” IEEE (Applied Sciences indexed), 2022.
[9] Meng, F., Lu, P., Li, J., Hu, T., Yin, M., and Lou, F., “GRU and auto-encoder based insider threat detection,” IEEE DSC, 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY