Insider Threat Detection Using Knowledge Graphs and RiskScore-Guided Graph Neural Networks

Van Duong Thi
Department of Network Systems, Infrastructure and Information Technology, Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam
dtvan@ioit.ac.vn (corresponding author)

Thang Tran Duc
Department of Network Systems, Infrastructure and Information Technology, Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam
thang@ioit.ac.vn

The Vinh Nguyen
Department of Network Systems, Infrastructure and Information Technology, Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam
vinh.ioit@gmail.com

Huy-Minh Pham Luong
Department of Network Systems, Infrastructure and Information Technology, Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam
minhplh@ioit.ac.vn

Received: 26 December 2025 | Revised: 3 February 2026 | Accepted: 13 February 2026

 DOI: https://doi.org/10.48084/etasr.17187

ABSTRACT

Insider threats remain a critical challenge in enterprise environments due to the difficulty of distinguishing malicious actions from legitimate user activities. This paper proposes a RiskScore-guided Graph Neural Network (R-GNN) framework for insider threat detection. The framework builds a Knowledge Graph (KG) from heterogeneous enterprise audit logs to represent users, resources, and their interactions, and a formally defined RiskScore is computed from behavioral deviations and incorporated as a guidance signal within graph-based learning. The RiskScore aggregates domain-informed indicators, such as abnormal access frequency and temporal irregularities, into a unified semantic representation that complements the relational structure encoded in the KG. Experiments conducted on the CERT r4.2 insider threat dataset demonstrate that the proposed approach consistently outperforms existing graph-based and sequence-based baselines. Moreover, by integrating RiskScore as an explicit input to the GNN, the framework enables detection results to be interpretable in terms of contributing behavioral risk factors and relational context, providing a practical and effective solution for risk-aware and interpretable insider threat detection in enterprise environments.

Keywords-RiskScore; insider threat detection; Knowledge Graph (KG); Graph Neural Network (GNN); explainable security analytics

I. INTRODUCTION

Insider threats remain a constant concern in organizational security, particularly in enterprise environments. Unlike external attackers, insiders possess legitimate system privileges, operational knowledge, and access pathways that allow them to perform harmful actions under the guise of normal behavior. This makes insider malicious activity extremely difficult to distinguish from benign user activity. In the modern enterprise environment, the challenge is further complicated by the volume and diversity of log data generated daily. Authentication logs, file access logs, email communications, web browsing activity, and server-level events together form a complex multimodal behavioral trace.

Extracting coherent signals from this heterogeneous, turbulent, and constantly evolving data landscape requires detection mechanisms that go beyond simple pattern matching and incorporate contextual and relational insights.

Recently, graph-based representations such as Knowledge Graphs (KGs) have been employed to capture complex relationships among users, devices, and activities in cybersecurity environments [1]. KGs allow users, resources, and servers, applications, and actions to be unified into a connected semantic space where relationships are clearly modeled. This enables contextual inference that is difficult to achieve with traditional machine learning processes. Meanwhile, Graph Neural Networks (GNNs) have emerged as powerful tools for learning from relational structures through message passing and neighborhood aggregation. Their ability to capture structural dependencies and relational patterns makes them promising candidates for modeling the behavioral context surrounding internal operations. However, despite the potential synergy between KGs and GNNs, existing studies rarely integrate both in a way that supports automated insider threat detection. Most studies employ KGs solely for semantic representation or query support, whereas GNN-based approaches commonly propagate raw or low-level behavioral signals without embedding higher-order risk semantics grounded in established security risk assessment frameworks shown in [2, 3].

Insider behavior itself is inherently multidimensional. Malicious insiders often exhibit a combination of high-risk actions (such as repeatedly accessing sensitive documents or performing activities outside of normal working hours), relational anomalies (such as deviating from typical group or departmental access patterns), and contextual indicators that span multiple modalities. Existing approaches tend to consider these components individually. Temporal models capture sequences but ignore relationships; KG models capture relationships but lack predictive capabilities; GNN models learn relationship patterns but are unaware of risk semantics. Consequently, current systems remain limited in their ability to provide both accurate detection and explainable risk assessment. The lack of a unified model integrating standardized behavioral models, relationship structures, and risk semantics is a notable gap in the research literature and a practical challenge in real-world security operations.

Based on the above observations, this study presents a graph learning framework based on RiskScore for detecting internal threats. Instead of proposing a new GNN architecture, this framework integrates a domain-based risk model with a KG representation to guide graph-based learning. By explicitly incorporating RiskScore as an input parameter along with relational information encoded in the KG, the proposed method aims to improve detection efficiency while supporting analytical interpretation in enterprise security environments.

The main contributions of this work are summarized as follows:

- Quantifying behavior based on RiskScore: We define a composite RiskScore that aggregates multiple behavioral indicators into a unified risk representation [4]. RiskScore is a domain-informed index that captures individual behavioral risk by aggregating deviations extracted from enterprise audit logs, guided by established security risk assessment principles.
- Multi-relational KG modeling: We construct a domain-specific KG, encoding user–resource–action relationships, providing contextual information, and structured relationships to support subsequent risk analysis.
- RiskScore integration for graph learning: We incorporate RiskScore as an explicit input parameter for GNN learning, allowing risk semantics and relational traits derived from the KG to influence representational learning, while remaining compatible with standard GNN models such as Graph Convolutional Network (GCN), GraphSAGE, and Graph Attention Network (GAT).
- Empirical evaluation on internal threat data: Experiments on the CERT r4.2 dataset demonstrate that integrating RiskScore into graph-based learning improves internal threat detection performance compared to basic behavior-based and graph-based methods, especially under high imbalance conditions.

This study introduces a practical insider threat detection framework that integrates knowledge-driven risk modeling with graph-based learning. Specifically, we organize audit logs into a multi-relational KG and compute a formally defined RiskScore to capture meaningful behavioral deviations. This RiskScore is then incorporated as a guidance signal into a GNN, shifting the focus from proposing a novel architecture to demonstrating how explicit risk semantics can enhance detection effectiveness while improving the interpretability of results in real-world enterprise security operations.

Throughout this paper, RiskScore denotes a unified quantitative measure of insider risk derived from behavioral deviations. The term RiskScore-guided, used in the title, refers to the integration of this RiskScore as guidance information within the GNN.

II. RELATED WORK

Prior studies on insider threat detection have explored a range of analytical approaches, differing mainly in how user behavior, temporal patterns, and relational context are modeled. Early work primarily relied on rule-based analytics, handcrafted heuristics, and statistical profiling techniques [5]. While such methods provide interpretable detection logic, they are known to suffer from limited adaptability to evolving user behavior and often evaluate events in isolation, restricting their ability to capture complex or long-term insider threat patterns [6].

With the availability of large-scale datasets such as CERT r4.2, data-driven approaches have gained increasing attention. Sequence-based neural models, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and autoencoder architectures, have been applied to model temporal dependencies in user activity logs [7]. Author in [8] demonstrated that neural representations can outperform rigid rule-based systems in modeling insider behavior. However,

these approaches typically represent behavior as linear sequences and do not explicitly capture interactions among users or shared resource usage. As noted in recent surveys [9], frameworks that jointly incorporate relational reasoning and security-oriented risk semantics remain relatively underexplored.

KGs have been introduced in cybersecurity research to represent security-relevant entities and their relationships in a structured manner. Studies by authors in [10] and authors in [11] highlight the value of KGs for contextual modeling and threat representation. Nevertheless, most existing KG-based work emphasizes knowledge representation or semantic querying, with limited integration into predictive learning models for behavior-driven insider threat detection.

GNNs have recently emerged as effective tools for relational representation learning and anomaly detection [12], with comprehensive overviews provided in [13]. In the context of insider threat detection, GNN-based methods typically operate on user–resource interaction graphs but often propagate low-level behavioral or structural features without explicit incorporation of risk semantics. Application-oriented studies further focus on implementation aspects rather than unified graph-based risk modeling or interpretability [14]. As a result, severity-aware interpretation and transparency remain limited in existing graph-based insider threat detection approaches.

III. METHODOLOGY

This section describes a knowledge-based graph learning framework for insider threat detection. The proposed method follows a sequential design, where heterogeneous audit logs are first transformed into structured behavioral representations and a multi-relational KG. Based on this graph representation, a user-level RiskScore is calculated to capture the domain-based security risk semantics. The acquired RiskScore is then integrated as an explicit input parameter to the GNN learning process, allowing risk information and relational structure to together guide the representation learning process.

A. Overview

The overall architecture of the proposed framework is illustrated in Figure 1. The system takes heterogeneous enterprise audit logs from the CERT r4.2 dataset as input, including authentication, device usage, file manipulation, email communication, and web browsing events. These logs are preprocessed to normalize timestamps, resolve user and resource identities, and map raw records into structured behavioral datasets suitable for relational modeling.

From the processed event streams, a multi-relational KG is constructed to represent interactions between users and system resources, as well as relationships among users through shared access patterns. By explicitly modeling heterogeneous entities and relation types, multi-relational KGs provide a structured semantic view of system operations [15] and serve as the relational backbone of the proposed framework.

On top of the constructed KG, a set of behavioral indicators is derived to characterize user activity patterns, including timing-based deviations, frequency-based behaviors, and interactions with sensitive resources. Based on these indicators, a user-level RiskScore is computed to quantify the security risk associated with observed behaviors.

The resulting risk-aware graph is then used as input to a GNN. RiskScore is incorporated as an explicit node-level attribute, allowing relational structure and risk-related information to jointly guide representation learning through message passing. The learned user representations are finally used for supervised insider threat detection.

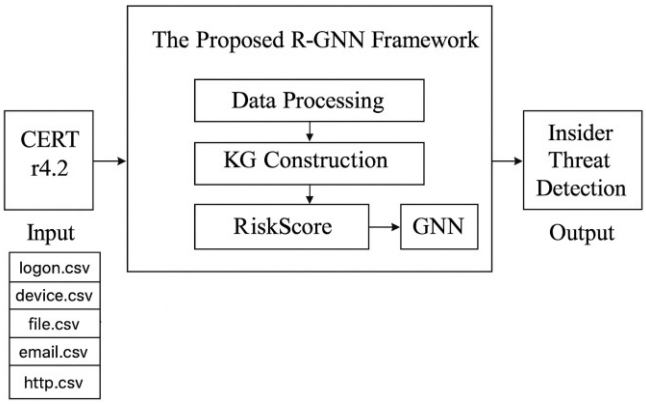

Fig. 1. Overview of the RiskScore graph learning framework for insider threat detection.

B. Data Preprocessing and Event Modeling

Enterprise audit data are inherently heterogeneous, as they are generated by different systems and services, each recording user activities with different formats, semantics, and levels of detail. The CERT r4.2 dataset illustrates this complexity by recording internal behavior across multiple log sources, including authentication events, device usage, file manipulation, email communication, and web browsing activity. Therefore, before relational learning and perceived risk modeling can be applied, it is essential to transform these raw logs into structured and consistent representations of user behavior.

In this work, five key log files from the CERT r4.2 dataset are considered, namely logon.csv, device.csv, file.csv, email.csv, and http.csv. Each file reflects a distinct mode of interaction between the user and the enterprise environment. However, the original records differ significantly in timestamp format, user identifiers, and resource descriptions. As is common practice in internal threat analysis, a preprocessing phase is applied to normalize the timestamps, identify user identities, and remove incomplete or inconsistent records [8, 16].

After normalization, the raw log entries are mapped into a unified event schema that captures the essential attributes of user activity. Specifically, each event is modeled as a structured dataset (u, a, r, t), where u denotes the user, a represents the type of action, r corresponds to the associated resource, and t indicates the time of occurrence. This abstraction preserves the contextual meaning of each activity while allowing analysis of data from different sources within a common behavioral space. Compared to purely sequence-based

representations, this event-focused model supports both temporal and relational analysis at subsequent stages [17].

Based on structured events, a set of behavioral traits is extracted to describe patterns commonly associated with insider threat activities. Timing traits capture deviations from normal work schedules, such as login attempts outside of normal working hours. Frequency-based traits quantify burst behaviors, including unusually high file access volumes or rapid sequences of actions over short periods. Additionally, content-oriented indicators highlight repetitive interactions with sensitive resources or unusual use of mobile devices. Such traits have been widely reported as useful signals for detecting insider threats [18].

In parallel with the extraction of behavioral traits, relational information is clearly preserved by maintaining links between users and system entities, such as files, devices, and communication partners. Instead of flattening these dependencies into independent trait vectors, the proposed framework retains them in a form suitable for relational modeling. Previous studies have shown that many malicious internal activities only become apparent when considered within a relational or group-level context, for example through the use of shared resources or deviations from typical departmental access patterns [1, 10].

The result of this preprocessing phase is a behaviorally cleaned and enriched set of events that captures both the activities of individual users and their relational context. This representation serves as direct input for the construction of the multi-relational KG and subsequent risk-aware graph learning. By systematically connecting raw audit data and a structured graph-based model, the preprocessing step provides a reliable platform for integrating behavioral evidence, relational structures, and security semantics within the proposed RiskScore-guided GNN (R-GNN) framework.

C. Knowledge Graph Construction

Following the data preprocessing and event modeling described in this work, heterogeneous audit logs from the CERT r4.2 dataset were transformed into structured behavioral events while preserving the relationships between users and system entities. Based on these structured events, a multi-relational KG was constructed to provide a unified representation of user behavior, interaction context, and system structure, serving as the relational backbone of the proposed R-GNN framework.

In this framework, the user–user similarity graph is constructed as a derived representation from the original heterogeneous KG, rather than as a replacement. The KG encodes semantic relationships among users, resources, actions, and contextual attributes, which are first used to compute user-centric behavioral representations and risk semantics. Based on these KG-derived features, a user–user similarity graph is then formed to enable effective relational learning among users, which is the primary objective in insider threat detection. Consequently, the KG remains the semantic backbone of the framework, whereas the user–user graph serves as an intermediate structure for scalable and user-centric graph neural learning.

Formally, the constructed KG is defined as a multi-relational graph $G = (V, E, R)$, where V denotes the set of nodes (entities), E the set of edges (interactions), and R the set of relation types. This formulation allows different categories of user activities to be explicitly distinguished instead of being aggregated into a single undifferentiated interaction graph.

The node set V is derived directly from the CERT r4.2 logs and consists of two primary types of entities: user nodes and resource nodes. User nodes represent internal employees whose activities are subject to risk assessment, whereas resource nodes correspond to the objects with which users interact, including files, devices, email accounts, and web resources. Accordingly, the node set can be expressed as:

$$V = V_{user} \cup V_{resource} \quad (1)$$

Each edge $e \in E$ corresponds to a normalized behavioral event extracted from the logon, device, file, email, and http log files. Edges connect user nodes to resource nodes and are labeled with a relation type $r \in R$ that reflects the nature of the interaction, such as system logon, file access, device usage, email communication, or web access. In addition to relation labels, edges may carry contextual attributes such as timestamps or activity frequency, which are preserved for subsequent analysis.

By explicitly modeling heterogeneous entities and diverse interactions, the constructed KG captures both individual behavior and relational structures within the enterprise environment. The KG is designed as a direct input to the R-GNN model, enabling graph-based learning within a structured behavioral context. However, at this stage, the graph only represents information about behavior and relationships, without integrating clear concepts of user risk. The integration of security risk semantics through knowledge-based RiskScores will be introduced in this work.

D. RiskScore Definition and Risk Semantics Integration

The KG constructed in this work provides a structured representation of user activities and interaction relationships derived from the CERT r4.2 dataset. While this representation captures behavioral patterns and relational dependencies, effective insider threat detection requires differentiating users not only based on anomalous behavior, but also based on the security risk associated with such behavior under different contexts.

To this end, we introduce RiskScore, a user-level metric designed to quantify security risk by aggregating multiple behavioral risk indicators. RiskScore is defined following standard risk assessment principles, which emphasize that similar actions may lead to different levels of risk depending on contextual factors such as frequency, timing, and resource sensitivity.

RiskScore aggregates multiple domain-informed behavioral risk indicators derived from enterprise audit logs and encoded in the KG. These indicators capture diverse aspects of insider risk, including out-of-hours activities, repeated access to sensitive resources, irregular usage patterns, and relational context such as interactions involving shared resources. Formally, the RiskScore of a user u is defined as:

$$\text{RiskScore}(u) = \sum_{k=1}^{K} w_k \cdot \phi_k(u) \quad (2)$$

where u denotes a user in the system. The term $\phi_k(u)$ represents the normalized value of the k-th behavioral risk indicator associated with user u, scaled to the range [0, 1] to ensure comparability across indicators with different magnitudes. Each risk indicator $\phi_k(u)$ is derived from user activity logs in the CERT r4.2 dataset, including login events, file access records, and temporal usage patterns. The coefficient $w_k \geq 0$ denotes the relative importance weight of the corresponding risk indicator $\phi_k(u)$. The set of weights is constrained such that $\sum_{k=1}^{K} w_k = 1$. Under this formulation, RiskScore provides a continuous quantitative representation of user-level security risk, rather than a threshold-based alert. For clarity, RiskScore is defined by a single formulation throughout this paper.

After computation, RiskScore is attached to each user node in the KG as a node-level attribute, transforming the original behavioral graph into a risk-aware KG. This design enables RiskScore to function as semantic a priori information, enriching node representations prior to graph neural learning. During message passing, both behavioral and relational information, together with risk-related signals, are propagated across the graph structure.

Importantly, this integration enables interpretability and traceability of detection results. The contribution of individual behavioral indicators to the overall RiskScore can be explicitly examined, and these indicators can be traced back to concrete audit log events encoded in the KG. As a result, the final detection decision is not only a prediction score, but can also be explained in terms of underlying behaviors and their relational context.

Figure 2 illustrates a concrete example of this explainability mechanism, showing how a detected high-risk user is situated within the risk-aware graph, the user's RiskScore is decomposed into contributing behavioral risk indicators, and these indicators are traced back to specific audit log events. This example demonstrates how RiskScore integration enhances transparency and supports analyst-driven investigation.

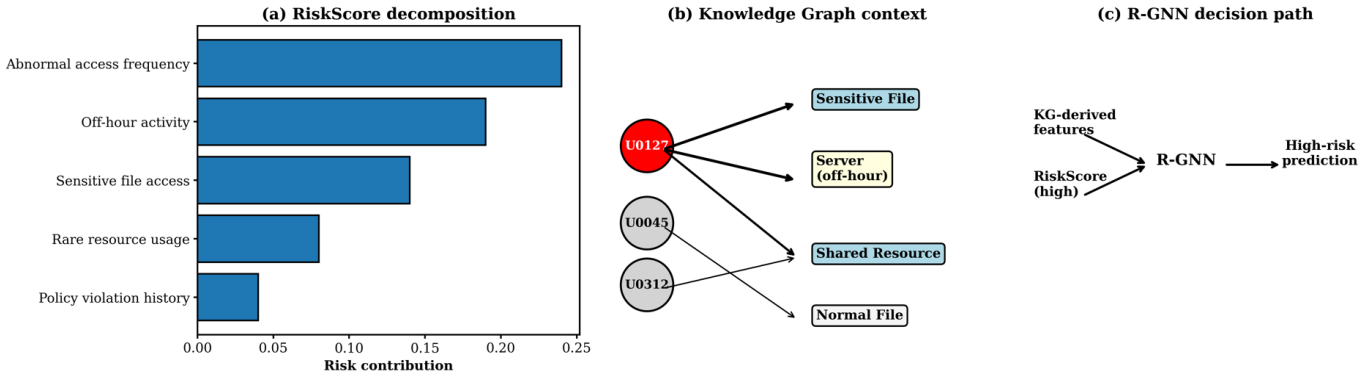

Fig. 2. Explainable RiskScore-guided insider threat detection on CERT r4.2.

By explicitly embedding security risk semantics into graph representations, the proposed method bridges the gap between descriptive behavioral models and predictive risk assessment. Risk-enriched KGs serve as direct input to the R-GNN model, enabling the network to learn user representations that are both structured and aware of security. Methodologically, Risk acts as an intermediary risk representation, connecting descriptive behavioral models and graph-based learning, providing a principled mechanism for incorporating security knowledge into the detection process. The specific architecture and learning mechanism of the R-GNN are presented in the proposed framework.

E. RiskScore-Guided GNN Model Architecture and Learning Process

The GNN component is used as a learning mechanism rather than a novel architectural contribution. Its role is to propagate and synthesize information about behavior and risk perception encoded in the graph structure.

After building the KG and integrating RiskScores at the user level, we formalize the task of detecting internal threats and describe the learning process of the proposed R-GNN model. The goal is to learn user representations that simultaneously encode behavioral traits, relational structures, and security risk semantics, enabling accurate prediction of internal threats.

Let U denote the set of enterprise users. Following the preprocessing of the CERT r4.2 audit logs described in this work, each user $u \in U$ is represented by a behavioral feature vector $x_u \in \mathrm{R}^d$, obtained from aggregated statistics over login activities, device usage, file access, email communication, and web browsing [4]. Each user is associated with a binary label $y_u \in \{0,1\}$ where $y_u = 1$ indicates a malicious insider and $y_u = 0$ denotes benign behavior.

Based on the KG constructed in this work, a user-level interaction graph $G = (U, E)$ is derived, where an edge $(u, v) \in E$ represents behavioral similarity or shared access patterns between users. Each user node is further augmented with a scalar risk attribute $\text{RiskScore}(u)$, defined in this work according to security risk assessment principles [2].

Given the input tuple $(G, x_u, \text{RiskScore}(u))$, the learning objective is to estimate a classifier:

$$\hat{y}_u = f_\theta(G, x_u, \text{RiskScore}(u)) \quad (3)$$

where $f_\theta(\cdot)$ denotes a parametric model implemented as a GNN. For comparison purposes, the same formulation can be instantiated using a non-graph Multilayer Perceptron (MLP) that ignores the graph structure.

In the proposed R-GNN model, the initial node representation of user u is defined as:

$$h_u^{(0)} = [x_u \parallel \text{RiskScore}(u)] \quad (4)$$

where $\parallel$ denotes vector concatenation. Node representations are subsequently updated through a message-passing mechanism:

$$h_u^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{AGG}\left(\{h_v^{(l)} : v \in N(u)\} \cup \{h_u^{(l)}\}\right)\right) \quad (5)$$

where $N(u)$ denotes the neighborhood of node u, $\text{AGG}(\cdot)$ is an aggregation function, $W^{(l)}$ is a learnable weight matrix, and $\sigma(\cdot)$ is a non-linear activation function. This formulation allows RiskScore to act as a node-level semantic prior that propagates across the graph, enabling risk-aware information to influence representation learning beyond individual user behavior.

The model is trained under a supervised binary classification setting using the cross-entropy loss:

$$L = -\sum_{u \in U}[y_u log(\hat{y}_u) + (1 - y_u) log(1 - \hat{y}_u)] \quad (6)$$

By integrating RiskScore into the graph representation, R-GNN is able to distinguish between benign anomalous behaviors and high-risk insider activities more effectively than models that rely solely on behavioral or structural features. This RiskScore graph learning framework forms the basis for the experimental evaluation presented in this work.

To clarify the end-to-end learning process of the proposed model, Algorithm 1 summarizes the training process of the R-GNN. Starting from a preprocessed CERT r4.2 log and a constructed user–user graph, the algorithm initializes risk-aware node representations, performs iterative message transmission with the GNN backbone [12, 19, 20], and optimizes model parameters using stochastic gradient-based methods such as Adam [11]. This formulation follows standard GNN training practices, while explicitly incorporating RiskScore as a semantic a priori information during node initialization.

Algorithm 1: Training Procedure of the Proposed R-GNN Model

Require: Preprocessed CERT r4.2 data; user set U; behavioral features vectors $\{x_u\}_{u \in U}$; RiskScore values $\{\text{RiskScore}(u)\}_{u \in U}$; user–user graph $G = (U, E)$; number of GNN layers L; learning rate η; number of training epochs T.

Ensure: Trained model parameters θ and risk-aware node embeddings $\left\{h_u^{(L)}\right\}_{u \in U}$.

Initialization:
for each user $u \in U$ do
  Construct initial node representation $h_u^{(0)} = [x_u \| \text{RiskScore}(u)]$
end for
Randomly initialize model parameters θ
for t = 1 to T do
  for l = 0 to L−1 do
    for each user $u \in U$ do
      Aggregate messages from neighbors: $m_u^{(l)} = \text{AGG}\left(\{h_v^{(l)} | v \in N(u)\}\right)$
      Update node representation: $h_u^{(l+1)} = \sigma\left(W^{(l)} \cdot [h_u^{(l)} \| m_u^{(l)}]\right)$
    end for
  end for
  Compute prediction: $\hat{y}_u = \text{sigmoid}\left(W_0 h_u^{(L)}\right)$
  Compute binary cross-entropy loss: $L = -\sum_{u \in U}[y_u \log(\hat{y}_u) + (1 - y_u)\log(1 - \hat{y}_u)]$
  Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta L$
end for
return θ and $\left\{h_u^{(L)}\right\}_{u \in U}$

F. User-Level Representation and Graph Construction

Based on the preprocessing pipeline and the risk semantics defined in this work, we construct user-level representations and a similarity-based interaction graph that serves as the direct input to the proposed R-GNN model.

1) User-Level Behavioral Characteristics

From the merged user-level event table derived from the CERT r4.2 logs, each user $u \in U$ is represented by a set of aggregated behavioral indicators capturing activity volume, temporal deviation, and external interaction tendencies. Specifically, the characteristic vector includes indicators such as the total number of login events, logins outside of business hours, frequency of file access, access to sensitive files, HTTP requests, access to external domains, large file downloads, as well as internal and external email communication volume. Let $x_u = [x_{u,1}, x_{u,2}, \ldots, x_{u,m}]^\top$ denote the raw behavioral feature vector of user u. To reduce scale imbalance and feature skewness across the user population, each indicator is standardized at the dataset level:

$$b_{u,i} = \frac{x_{u,i} - \mu_i}{\sigma_i + \epsilon}, \quad i = 1, \ldots, m \quad (7)$$

where μ_i and σ_i denote the mean and standard deviation of the i-th indicator over all users, and ϵ is a small constant to avoid division by zero. The resulting normalized behavioral representation is denoted by $b_u = [b_{u,1}, \ldots, b_{u,m}]^\top$.

2) User Similarity Graph Construction

Instead of directly operating on a fully heterogeneous user–resource KG during graph learning, we derive a user-centric behavioral similarity graph to model relations among users with comparable activity patterns. For any pair of users (u, v), similarity is defined using cosine similarity over the normalized behavioral vectors:

$$\text{sim}(u,v) = \frac{b_u^\top b_v}{\|b_u\|_2 \, \|b_v\|_2 + \epsilon} \quad (8)$$

For each user u, we identify its k nearest neighbors $N_k(u)$ in the behavioral feature space according to this similarity measure. An undirected edge (u, v) is created if $v \in N_k(u)$ or $u \in N_k(v)$. The resulting graph $G = (U, E)$ captures behavioral proximity between users, even when they do not share explicit resources or direct interactions.

This user-level graph can be interpreted as a behavioral projection of the underlying KG: users become connected when they exhibit similar logon, file, web, or email usage patterns, thereby exposing group-level or peer-based anomalies that may remain undetected under purely individual analysis.

To integrate security semantics into graph learning, each user node is augmented with a scalar risk attribute derived from the RiskScore defined in this work. Two variants of node representation are considered. The behavior-only representation is defined as:

$$z_u^{beh} = b_u \quad (9)$$

whereas the risk-aware representation used in the proposed model is given by:

$$z_u^{full} = [\, b_u \parallel \text{RiskScore}(u) \,] \quad (10)$$

where $\parallel$ denotes vector concatenation.

The model uses only the behavioral representations employed by standard GNN models or graph-only (KG-only), whereas the proposed R-GNN model operates in such a way that risk semantics are transmitted along with behavioral signals through the graph structure. This design allows the model to differentiate benign anomalies from high-risk internal behaviors by simultaneously considering individual activity patterns, relational context, and security knowledge.

The graph and node representations obtained at the user level form the final input for the GNN architectures described and are empirically evaluated in this paper.

IV. EXPERIMENTS

This section describes the dataset, preprocessing procedure, test protocol, and experimental results obtained with the proposed R-GNN model and baseline methods.

A. Dataset and Experimental Setup

We conduct experiments on the CERT Insider Threat Dataset (r4.2), which provides heterogeneous enterprise audit logs covering authentication, device usage, file operations, email communication, and web activities. Dataset details and threat scenarios are further discussed in [21]. This dataset is widely used in insider threat research due to its ability to realistically simulate the enterprise environment and the availability of realistic insider scenarios [8, 16]. It contains heterogeneous audit logs recording the daily activities of users across multiple organizational services, with malicious behaviors targeted for a small subset of users.

In this study, we utilize the primary log sources provided in CERT r4.2, including logon.csv, file.csv, http.csv, and email.csv. These logs capture complementary aspects of user behavior, such as authentication events, file access manipulations, web browsing activity, and email communication. The actual insider user labels are derived from the official scenario definitions released with the dataset and are merged in the insiders.csv file, which specifies the set of users involved in at least one malicious activity scenario.

According to the preprocessing procedure described in this work, raw audit logs are first normalized to address timestamp format, user identifiers, and resource representation. Inconsistent or incomplete records are discarded. To reduce noise from sporadic or insignificant activity, users who appear only a few times in the logs are filtered out before aggregation, following common practice in insider threat detection [8].

After preprocessing, all log sources are merged to build a unified user-level event table, denoted as cert_events_merged.csv. In this table, each row corresponds to a distinct user and each column represents an aggregated behavioral index defined in the proposed framework, including login frequency, out-of-hours activity, file access type, interaction with sensitive resources, and web or email communication statistics. Let x_u denote the vector of raw behavior features for user u taken from this table.

To account for scale differences between the indices, all features are normalized across the entire user group to obtain normalized behavior representations b_u . Based on these normalized features, a scalar RiskScore is calculated for each user using the formula introduced in this work, integrating observed behavioral deviations with security risk semantics derived from NIST SP 800-30 and ISO/IEC 27005. This RiskScore serves as an intermediate risk representation, connecting the descriptive behavior model and graph-based learning in the later stages.

The final processed dataset contains approximately $|U| \approx 1{,}000$ users, with a severely unbalanced label distribution, in which internal users make up only a small fraction of the total. This severe class imbalance reflects the real-world business environment and poses a significant challenge to machine learning-based detection methods [8, 18].

For the relational model, a user-centric behavioral similarity graph is constructed using the k-Nearest Neighbor (kNN) strategy in the normalized behavioral feature space. Each user is represented as a node, and edges connect pairs of users with similar activity profiles. The resulting graph, along with user-level behavioral features and RiskScore values, forms the input for the proposed R-GNN model and graph-based baseline methods.

The internal threat detection task is constructed as a supervised binary node classification problem at the user level. The dataset is split into training, validation, and testing sets using stratified sampling with a typical split ratio of 60%–20%–20%. The validation set is used to fine-tune hyperparameters and early stopping, whereas the test set is reserved for final performance evaluation. All models are trained using the Adam optimizer. This testing process ensures

a fair and reproducible comparison between the proposed RiskScore-based framework and competing methods.

B. Baseline Methods

To evaluate the effectiveness of the proposed R-GNN framework, we compared it to a set of baseline methods designed to isolate the contributions of different model components, including behavioral traits, risk semantics, and relational constructs. All basic models were trained and evaluated in identical experimental settings to ensure a fair comparison.

- MLP (behavior only): This basic model uses an MLP network trained entirely on aggregated behavioral traits extracted from CERT r4.2 audit logs. Each user is classified independently, without exploiting any relational information or security risk semantics. Such trait-based neural models are commonly used as reference basic models in insider threat detection studies. This model reflects a purely behavioral approach that operates directly on processed audit data.
- MLP (RiskScore enhancement): This baseline model extends the behavior-only MLP by incorporating the proposed $\mathrm{RiskScore}(u)$ as an additional input feature. By embedding domain-informed security knowledge into the classification process, this model partially explains the risk relevance of observed behaviors. However, it does not model user interactions or capture collective behavioral patterns and therefore cannot leverage the existing relational context in the enterprise environment [3, 4]. This baseline isolates the impact of RiskScore without graph-based learning.
- GraphSAGE (KG-based, no risk semantics): GraphSAGE is applied as an inductive graph learning basis operating on user similarity graphs constructed through the kNN strategy. Node embeddings are learned through neighborhood aggregation using only behavioral traits, without explicitly incorporating risk semantics [7, 19]. This baseline evaluates the contribution of relational structure while excluding domain-informed risk models.
- GCN (grid-based, no risk semantics): The GCN baseline applies spectral graph convolutions to communicate information on the same user similarity graph. Although relational dependencies between users are captured, node representation relies only on behavioral traits and ignores standard concepts of security risk [20]. Along with GraphSAGE, this baseline represents graph-based learning without incorporating RiskScore.
- R-GNN (proposed): The proposed R-GNN integrates behavioral characteristics, relational structures, and security risk semantics into a unified graph learning framework. By enriching node representations with $\mathrm{RiskScore}(u)$ and enabling risk awareness messaging on the graph, R-GNN leverages processed behavioral evidence, domain-based risk models, and user-to-user interactions. This design allows the model to operate efficiently on heterogeneous audit data by transforming raw, heterogeneous logs into a structured, risk-aware representation suitable for graph-based internal threat detection.

C. Evaluation Metrics

The detection of insider threats is inherently characterized by extreme class imbalance, where the majority of users exhibit benign behavior and only a very small fraction are malicious insiders. Under such conditions, conventional metrics such as overall classification precision become misleading, since a simple classifier labeling all users as benign may still be judged to be highly accurate but completely fail to detect insider threats.

Therefore, following established practices in imbalance security analysis [22], this study applies evaluation metrics that clearly focus on the differences between minority classes, namely precision, recall, F1-score and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC). Precision measures the reliability of the alerts given by quantifying the proportion of correctly identified insiders among all flagged users, which is crucial in operational security environments where false alarms incur significant investigative costs. Recall assesses the model's ability to identify malicious insiders, whereas the F1-score provides a balanced assessment by considering both precision and recall simultaneously. Additionally, AUC is reported to assess each model's global discriminant ability across different decision thresholds, making it particularly suitable for highly unbalanced datasets [23].

D. Experimental Results and Discussion

This section analyzes the experimental results obtained on the CERT r4.2 dataset and examines the effectiveness of the proposed risk-aware graph learning framework against the baseline methods. All methods were evaluated under identical experimental conditions, and the overall performance is summarized in Table I, with the corresponding ROC curves shown in Figure 3.

TABLE I. PERFORMANCE COMPARISON OF GRAPH-BASED METHODS ON THE CERT R4.2 DATASET

Method	Precision	Recall	F1-score	AUC
MLP (RiskScore)	0.1389	0.5556	0.2222	0.5847
MLP (behavior)	0.5455	0.6667	0.6000	0.9026
GraphSAGE (KG only)	0.7059	0.6667	0.6857	0.9069
GCN baseline	0.4194	0.7222	0.5306	0.8556
R-GNN (KG + RiskScore)	0.7857	0.6111	0.6875	0.9093

As reported in Table I, models incorporating relational information consistently outperformed the non-relational baseline models, highlighting the importance of modeling user interactions in insider threat detection.

The baseline MLP model (RiskScore) exhibited limited performance (F1-score = 0.2222, AUC = 0.5847), indicating that aggregated risk indices alone, without relational or collective context, are insufficient to reliably distinguish between malicious insiders and benign users. Conversely, the MLP (behavior) model, based on multidimensional behavioral

traits, significantly improved detection performance (F1 = 0.6000, AUC = 0.9026). However, this model considers users independently and ignores correlations arising from resource sharing or similar operational patterns.

Graph-based baseline models demonstrate distinct advantages in this context. GraphSAGE (KG only) achieves strong performance (F1-score= 0.6857, AUC = 0.9069), indicating that relational structure and behavioral similarity provide effective collective signals for detecting internal threats. The GCN baseline model also benefits from structural synthesis but exhibits lower precision (0.4194), suggesting a higher false alarm rate in practical implementations.

1) Effectiveness of Risk-Aware Graph Learning

The proposed R-GNN (KG + RiskScore) model achieves the most balanced overall performance among the evaluated methods. It achieved the highest F1-score (0.6875) and the highest AUC (0.9093), while improving precision by 0.7857 compared to both GraphSAGE (KG only) and the base GCN model, without significantly reducing recall (0.6111). These results suggest that incorporating RiskScore into graph-based learning allows the model to better distinguish high-risk internal behavior from benign anomalies by guiding learning toward security-related biases and minimizing false positive errors.

2) ROC Analysis and Robustness under Class Imbalance

As shown in Figure 3, R-GNN consistently maintained a higher true positive rate in low false-positive regions under severe class imbalance, characteristic of internal threat detection scenarios [9]. The superior AUC performance demonstrates improved separability between malicious insiders and benign users, independent of specific decision thresholds.

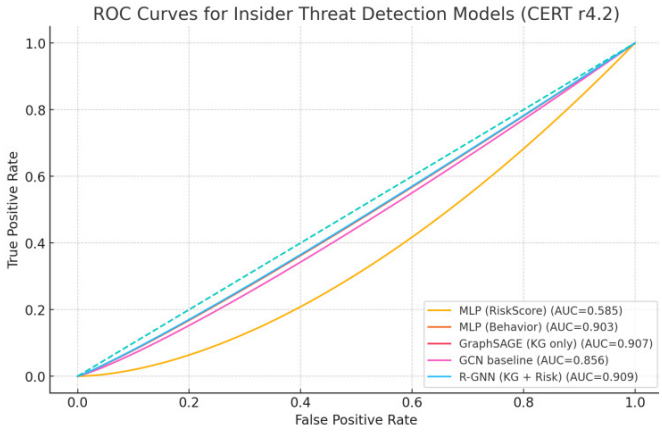

Fig. 3. ROC curves comparing baseline methods and the proposed R-GNN on the CERT r4.2 dataset.

3) Impact of RiskScore Integration

A direct comparison between GraphSAGE (KG only) and R-GNN (KG + RiskScore) highlights RiskScore's contribution as a node-level semantic prior. While GraphSAGE effectively captures relational patterns, it lacks a clear mechanism to differentiate low-risk anomalies from critical security deviations. By embedding domain-based risk semantics into graph machine learning, RiskScore complements the relational model and leads to consistent improvements in precision and F1-scores.

Overall, these results demonstrate that effective insider threat detection benefits from simultaneously considering behavioral evidence, relational structures, and perceived risk semantics, resulting in a more reliable balance between detection capability and alert quality in enterprise environments.

V. CONCLUSION AND FUTURE WORK

This study examines insider threat detection from a risk-aware graph learning perspective, focusing on how to integrate domain-based risk semantics into graph-based behavioral models. Rather than proposing a new Graph Neural Network (GNN) architecture, the study investigates the effect of incorporating formally defined RiskScores as explicit guidance signals in graph learning on Knowledge Graphs (KGs) constructed from heterogeneous enterprise audit logs. The results demonstrate that embedding risk semantics into graph-based models provides a principled mechanism for aligning data-driven detection with established security risk assessment practices.

Empirical evaluations on the CERT r4.2 dataset show that integrating RiskScores into graph learning improves detection reliability compared to baseline models relying solely on behavioral features or relational structure. Although the absolute performance gains over strong baselines remain moderate—reflecting the inherent difficulty of insider threat detection under severe class imbalance and complex behavioral patterns—the proposed RiskScore-guided GNN (R-GNN) framework consistently achieves higher precision while maintaining competitive recall, leading to improved F1-scores and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC). These improvements are particularly important in real-world security operations, where reducing false alarms and ensuring trustworthy alerts are critical requirements.

Beyond quantitative performance, a key contribution of this work lies in enhancing model interpretability. By treating RiskScore as a semantic prior rather than a heuristic threshold, the proposed framework enables structured explainability and traceability, allowing high-risk predictions to be systematically linked to specific behavioral indicators, relational contexts, and verifiable audit log evidence. This capability supports evidence-based decision-making and facilitates practical deployment in enterprise environments.

Despite these advantages, the current study is limited to static risk modeling and evaluation on a single benchmark dataset. Future research will extend the proposed framework to more complex and realistic settings, including dynamic and temporal risk modeling, large-scale graph evolution, and evaluation on richer datasets such as CERT r6.2. These extensions aim to further improve robustness, scalability, and generalization, paving the way for practical adoption of risk-aware graph learning in real-world insider threat detection systems.

DATA AND CODE AVAILABILITY

The source code and experimental scripts used in this paper are publicly available at: https://github.com/van-1985/project_Threat-Detection_r4.2.

ACKNOWLEDGMENT

This work was financially supported by the Institute of Information Technology, Vietnam Academy of Science and Technology (VAST), under Project CSCL02.04/24-25.

REFERENCES

[1] L. F. Sikos, "Cybersecurity knowledge graphs," *Knowledge and Information Systems*, vol. 65, no. 9, pp. 3511–3531, Sept. 2023, https://doi.org/10.1007/s10115-023-01860-3.

[2] Joint Task Force Transformation Initiative, "Guide for conducting risk assessments," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST SP 800-30r1, 2012. https://doi.org/10.6028/NIST.SP.800-30r1.

[3] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), *ISO/IEC 27005:2022 — Guidance on Managing Information Security Risks*, ISO/IEC 27005:2022, Geneva, Switzerland, 2022. [Online]. Available: https://www.iso.org/standard/80585.html

[4] W. Eberle and L. Holder, "Insider Threat Detection Using Graph-Based Approaches," in *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, Washington, DC, USA, 2009, pp. 237–241, https://doi.org/10.1109/CATCH.2009.7.

[5] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures," *ACM Computing Surveys*, vol. 52, no. 2, Apr. 2019, Art. no. 30, https://doi.org/10.1145/3303771.

[6] M. N. Al-Mhiqani *et al.*, "A Review of Insider Threat Detection: Classification, Machine Learning Techniques, Datasets, Open Challenges, and Recommendations," *Applied Sciences*, vol. 10, no. 15, July 2020, Art. no. 5208, https://doi.org/10.3390/app10155208.

[7] M. Villarreal-Vasquez, G. Modelo-Howard, S. Dube, and B. Bhargava, "Hunting for Insider Threats Using LSTM-Based Anomaly Detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 451–462, Jan. 2023, https://doi.org/10.1109/TDSC.2021.3135639.

[8] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams." arXiv, Dec. 15, 2017, https://doi.org/10.48550/arXiv.1710.00811.

[9] Y. Gong, S. Cui, S. Liu, B. Jiang, C. Dong, and Z. Lu, "Graph-based insider threat detection: A survey," *Computer Networks*, vol. 254, Dec. 2024, Art. no. 110757, https://doi.org/10.1016/j.comnet.2024.110757.

[10] B. Li, Q. Yang, C. Deng, and H. Pan, "CyberKG: Constructing a Cybersecurity Knowledge Graph Based on SecureBERT_Plus for CTI Reports," *Informatics*, vol. 12, no. 3, Sept. 2025, Art. no. 100, https://doi.org/10.3390/informatics12030100.

[11] X. Zhao, R. Jiang, Y. Han, A. Li, and Z. Peng, "A survey on cybersecurity knowledge graph construction," *Computers & Security*, vol. 136, Jan. 2024, Art. no. 103524, https://doi.org/10.1016/j.cose.2023.103524.

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, https://doi.org/10.1109/TNNLS.2020.2978386.

[13] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020, https://doi.org/10.1016/j.aiopen.2021.01.001.

[14] E. Yilmaz and O. Can, "Unveiling Shadows: Harnessing Artificial Intelligence for Insider Threat Detection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13341–13346, Apr. 2024, https://doi.org/10.48084/etasr.6911.

[15] J. Zhao, M. Shao, H. Wang, X. Yu, B. Li, and X. Liu, "Cyber threat prediction using dynamic heterogeneous graph learning," *Knowledge-Based Systems*, vol. 240, Mar. 2022, Art. no. 108086, https://doi.org/10.1016/j.knosys.2021.108086.

[16] J. Lu and R. K. Wong, "Insider Threat Detection with Long Short-Term Memory," in *Proceedings of the Australasian Computer Science Week Multiconference*, Sydney, Australia, 2019, pp. 1–10, https://doi.org/10.1145/3290688.3290692.

[17] W. Eberle and L. Holder, "Anomaly detection in data represented as graphs," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 663–689, Nov. 2007, https://doi.org/10.3233/IDA-2007-11606.

[18] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Computers & Security*, vol. 104, May 2021, Art. no. 102221, https://doi.org/10.1016/j.cose.2021.102221.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs." arXiv, Sept. 10, 2018, https://doi.org/10.48550/arXiv.1706.02216.

[20] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[21] "Insider Threat Test Dataset." Carnegie Mellon University, Sept. 30, 2020, https://doi.org/10.1184/R1/12841247.v1.

[22] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sept. 2009, https://doi.org/10.1109/TKDE.2008.239.

[23] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006, https://doi.org/10.1016/j.patrec.2005.10.010.