



## DYNAMIC HETEROGENEOUS GRAPH NEURAL NETWORKS FOR ANOMALY DETECTION IN SUPPLY CHAINS

Nan ZHAO, Chun FENG

Southwest Jiaotong University, School of Transportation and Logistics, Chengdu 611756, China  
Corresponding author: Chun FENG, E-mail: [ifengchun@swjtu.edu.cn](mailto:ifengchun@swjtu.edu.cn)

**Abstract.** Supply chain anomaly detection is a critical task in supply chain management, aiming to enhance system stability and resilience by identifying potential anomalies in the supply chain network in a timely manner. However, the dynamic changes in nodes and relationships, along with the heterogeneous characteristics of the network, pose significant challenges for anomaly detection. Existing approaches struggle to capture the complex multi-relational interactions in dynamic networks. Furthermore, most studies focus on detecting node anomalies while underemphasizing edge anomalies, thus failing to fully reflect the complexity of both node and relational aspects in the supply chain. This paper proposes a dynamic heterogeneous graph neural network-based approach for supply chain anomaly detection, which jointly models the temporal dynamics and multi-relational heterogeneity of the supply chain network to achieve comprehensive anomaly detection for both nodes and edges. The approach extracts multi-relational features from the network via a message-passing module, incorporates temporal evolution information through a dynamic feature update module, and calculates anomaly scores for nodes and edges using a dynamic-heterogeneous joint reconstruction module. To help users interpret the detection results, the paper also introduces feature-level and temporal-level contribution analysis, with visual analytics for anomalous nodes and edges as auxiliary support. Results demonstrate that, on publicly available supply chain network datasets, the proposed dynamic heterogeneous graph neural network approach significantly outperforms baseline models. This study offers a novel and practical solution for supply chain management.

**Keywords:** supply chain anomaly detection, dynamic heterogeneous graph neural networks, temporal dynamics, multi-relational heterogeneity.

### 1. INTRODUCTION

As global supply chains become increasingly complex, supply chain anomaly detection has emerged as a key task for enhancing supply chain resilience and stability [1]. Anomalous events in the supply chain, such as production bottlenecks, transportation delays, and inventory imbalances, not only directly impact operational efficiency but can also lead to significant economic losses. Therefore, the ability to detect and prevent these anomalies in a timely and accurate manner has become a critical issue in modern supply chain management [2].

However, current approaches for supply chain anomaly detection exhibit notable limitations [4]. Most existing approaches rely on static graph models [5], which primarily perform anomaly detection through the aggregation of static node features [6]. These approaches fail to capture the temporal variations of nodes and edges, thus unable to reflect the dynamic nature of the network. Some researchers have proposed time series analysis approaches [7], which offer certain advantages in modeling the temporal dimension, yet they overlook the importance of multi-relational interactions and fail to fully reflect the complex interactions among nodes in the supply chain. Other approaches based on dynamic graph approaches address the dynamic characteristics of nodes and edges over time, but many existing dynamic graph approaches still face limitations when modeling multi-relational heterogeneous graphs [8]. To this end, some studies have introduced heterogeneous graph models [9], which can handle relationships between different types of nodes and edges in the supply chain. However, most existing heterogeneous graph models neglect the impact of temporal changes on network

behavior and fail to capture the temporal evolution within the supply chain. Additionally, most current approaches focus on node anomaly detection, often neglecting edge anomalies, even though changes in edge anomalies are often key factors triggering supply chain risks.

The challenges in supply chain anomaly detection primarily stem from the following aspects: first, the nodes and edges in the supply chain network are heterogeneous, and the relationships between them are complex and diverse, making it difficult for traditional approaches to comprehensively model the interactions among different relationship types [10]. Second, the features of nodes and edges in the supply chain change dynamically over time, requiring models to balance temporal dynamics and heterogeneity to accurately capture the dynamic evolution of the supply chain [11]. Lastly, traditional approaches often focus on node-level anomalies, with insufficient attention given to edge anomalies, resulting in partial and inaccurate anomaly detection [12]. To address these challenges, this paper proposes a supply chain anomaly detection approach based on Dynamic Heterogeneous Graph Neural Networks (DyHetGNN). This approach combines temporal dynamics and multi-relational heterogeneity, enabling accurate detection of both node and edge anomalies. Specifically, the approach utilizes four key modules to achieve this goal: a message-passing module that extracts features from neighboring nodes and associated edges in a multi-relational heterogeneous graph; a dynamic feature update module that captures long-term dependencies through temporal message fusion; a dynamic-heterogeneous joint reconstruction module that reconstructs nodes and edges and computes anomaly scores; and an interpretability analysis module that provides detailed explanations of anomaly detection through feature-level and temporal-level contribution analysis.

The contributions of this paper are as follows:

1. The introduction of dynamic heterogeneous graph neural networks, which jointly model node and edge anomalies, fully considering the heterogeneity and temporal dynamics of supply chain networks.
2. The design of a dynamic-heterogeneous joint reconstruction module, which improves anomaly detection accuracy by jointly considering temporal dynamics and multi-relational features.
3. The incorporation of an interpretability analysis module, which enhances model transparency and usability by helping users understand the sources of anomalies through feature and temporal step contribution analysis.

Results demonstrate that the proposed approach outperforms existing baseline models in both node and edge anomaly detection, validating the effectiveness of the approach in handling multi-relational heterogeneity and temporal dynamics in supply chain anomaly detection. This research provides a practical and valuable solution for supply chain anomaly detection, offering comprehensive and accurate decision support for supply chain managers.

In the following sections, Section 2 introduces the problem definition, the construction of dynamic heterogeneous graphs, and data processing, laying the foundation for the subsequent sections. Section 3 presents the construction of dynamic heterogeneous graphs and the preprocessing steps necessary for our approach. Section 4 details the experimental setup, including the datasets, baseline models, and training details. Finally, Section 5 discusses the results and provides insights into the model's performance.

## 2. PRELIMINARY

Supply chain anomaly detection is a crucial task to ensure the stability and resilience of the supply chain. However, current anomaly detection approaches often struggle with the dynamic and heterogeneous nature of supply chain networks. These approaches are typically limited by static models that fail to capture the temporal evolution and complex relationships between entities in the network.

To address these limitations, the construction and data processing of dynamic heterogeneous graphs form the foundation of the proposed approach, directly affecting model performance and anomaly detection accuracy. Building a dynamic heterogeneous graph for the supply chain network involves extracting key elements from raw data: nodes, edges, attributes, types, and temporal dynamics [13]. Data processing plays a crucial role in generating standardized node and edge feature matrices, providing normalized inputs for subsequent dynamic embedding learning.

## 2.1. Problem definition

In modern supply chain management, anomaly detection is a critical task for ensuring the stability and resilience of supply chains. Existing approaches, however, often rely on static models that fail to capture the dynamic and heterogeneous nature of supply chain networks. These models struggle to address the diversity of nodes (e.g., suppliers, manufacturers, retailers) and edges (e.g., transportation routes, inventory flows) in the network, as well as the temporal evolution of these relationships.

The challenges in supply chain anomaly detection include:

1. Lack of Dynamic Modeling: Existing approaches typically use static graphs that cannot capture temporal changes in the supply chain.
2. Heterogeneity: Supply chains contain diverse nodes and edges, which require models that can account for multi-relational data.
3. Over-focus on Node-Level Anomalies: Traditional approaches often concentrate only on detecting node-level anomalies, overlooking edge-level anomalies that are crucial for identifying disruptions such as transportation delays or inventory imbalances.

To address these challenges, we propose a new approach based on DyHetGNN, which jointly models the temporal dynamics and multi-relational heterogeneity of the supply chain network. The classification problem in anomaly detection involves distinguishing between normal and anomalous nodes and edges. This task is formulated as an optimization problem, where the goal is to minimize the reconstruction error between the actual and reconstructed feature vectors for both nodes and edges. By minimizing this reconstruction error, the model can effectively classify nodes and edges as normal or anomalous.

## 2.2. Construction of dynamic heterogeneous graphs

In the supply chain network, nodes and edges exhibit diversity (heterogeneity) and evolve dynamically over time. The construction of the dynamic heterogeneous graph  $\mathbf{G}$  is fundamental.  $\mathbf{G}$  is defined as a set of graphs over a time series,  $\mathbf{G} = \{\mathbf{G}^t | t = 1, 2, \dots, T\}$ , where each time slice  $\mathbf{G}^t$  represents the state of the supply chain network at time  $t$ , and is formally defined as:

$$\mathbf{G}^t = (\mathcal{V}^t, \mathcal{E}^t, \mathbf{X}_v^t, \mathbf{X}_e^t, \mathcal{R}) \quad (1)$$

In this formulation,  $\mathcal{V}^t$  and  $\mathcal{E}^t$  denote the sets of nodes and edges at time  $t$ , respectively;  $\mathbf{X}_v^t \in \mathbb{R}^{|\mathcal{V}^t| \times d_v}$  and  $\mathbf{X}_e^t \in \mathbb{R}^{|\mathcal{E}^t| \times d_e}$  are the dynamic feature matrices for nodes and edges, with  $d_v$  and  $d_e$  as feature dimensions;  $\mathcal{R}$  denotes the set of relationship types in the supply network.

For each relationship  $r \in \mathcal{R}$ , the adjacency matrix  $\mathbf{A}_r^t$  at time  $t$  is constructed, whose entries are defined as:

$$(\mathbf{A}_r^t)_{ij} = \begin{cases} 1, & \text{if } e_{ij} \in E_r^t \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where  $E_r^t$  denotes the set of edges at time  $t$  associated with relation  $r$ . The adjacency matrices for all relationship types are stacked into a tensor  $\mathbf{A}^t \in \mathbb{R}^{|\mathcal{V}^t| \times |\mathcal{V}^t| \times |\mathcal{R}|}$ , representing the multi-relational structure at time  $t$ . By incorporating multiple time slices, the dynamic heterogeneous graph captures the temporal evolution of the supply chain, accounting for changes in nodes and edges across time steps.

## 2.3. Data processing

Raw data often contains noise, missing values, and uneven distribution, requiring the following processing steps. First, deduplication is performed to eliminate duplicate records in node and edge features, particularly for data from the same time step, which are filtered based on timestamps and unique sources. Next, data types are standardized to ensure consistent dimensions and formats, with features unified to floating-point types. Timestamps are normalized to align different formats to a consistent time granularity. For missing values, interpolation is used for node features, while missing edge features are filled with the mean of adjacent edges. Undefined relationships are supplemented with default types. Finally, feature normalization, time-series modeling, and time-step encoding are applied to capture the temporal evolution of node and edge representations and generate dynamic embeddings.

**1. Feature normalization.** Feature normalization eliminates scale differences across feature dimensions, enhancing comparability and ensuring training stability.

Node features  $\mathbf{x}_{v_i}^t$  and edge features  $\mathbf{x}_{e_{ij}}^t$  at time  $t$  are normalized as follows:

$$\tilde{\mathbf{x}}_{v_i}^t = \frac{\mathbf{x}_{v_i}^t - \boldsymbol{\mu}_v^t}{\boldsymbol{\sigma}_v^t} \quad \tilde{\mathbf{x}}_{e_{ij}}^t = \frac{\mathbf{x}_{e_{ij}}^t - \boldsymbol{\mu}_e^t}{\boldsymbol{\sigma}_e^t} \quad (3)$$

where  $\boldsymbol{\mu}_v^t, \boldsymbol{\sigma}_v^t \in \mathbb{R}^{d_v}$  and  $\boldsymbol{\mu}_e^t, \boldsymbol{\sigma}_e^t \in \mathbb{R}^{d_e}$  are the mean and standard deviation vectors computed across all nodes and edges at time step  $t$ , respectively. The normalization is applied element-wise across each feature dimension, using broadcasting.

**2. Time step encoding.** Time step encoding adds temporal position information to the features to capture the absolute time meaning. For each time step  $t$ , a time embedding  $\mathbf{p}_t$  is added:

$$\hat{\mathbf{x}}_{v_i}^t \leftarrow \tilde{\mathbf{x}}_{v_i}^t + \mathbf{p}_t \quad \hat{\mathbf{x}}_{e_{ij}}^t \leftarrow \tilde{\mathbf{x}}_{e_{ij}}^t + \mathbf{p}'_t \quad (4)$$

where  $\mathbf{p}_t \in \mathbb{R}^{d_v}$  and  $\mathbf{p}'_t \in \mathbb{R}^{d_e}$  are the time embeddings for nodes and edges, respectively. Since the time steps are uniform, they can be generated through learnable parameters or positional encoding.

**3. Time-series modeling.** Time-series modeling captures temporal dependencies in node or edge features, reflecting how past dynamics influence the present. The dynamic feature sequence for each node is constructed as a time window:

$$\hat{\mathbf{X}}_{v_i}^{t-k:k} = [\hat{\mathbf{x}}_{v_i}^{t-k} \hat{\mathbf{x}}_{v_i}^{t-k+1} \dots \hat{\mathbf{x}}_{v_i}^t]^T \in \mathbb{R}^{(k+1) \times d_v} \quad (5)$$

where  $k$  is the window size, and  $\hat{\mathbf{X}}_{v_i}^{t-k:k}$  represents the feature sequence of node  $v_i$  from  $t - k$  to  $t$ .

Similarly, for edges:

$$\hat{\mathbf{X}}_{e_{ij}}^{t-k:k} = [\hat{\mathbf{x}}_{e_{ij}}^{t-k} \hat{\mathbf{x}}_{e_{ij}}^{t-k+1} \dots \hat{\mathbf{x}}_{e_{ij}}^t]^T \in \mathbb{R}^{(k+1) \times d_e} \quad (6)$$

Finally, the node and edge feature matrices,  $\hat{\mathbf{X}}_v^t \in \mathbb{R}^{|\mathcal{V}^t| \times (k+1) \times d_v}$  and  $\hat{\mathbf{X}}_e^t \in \mathbb{R}^{|\mathcal{E}^t| \times (k+1) \times d_e}$  are constructed by stacking the time-embedded feature sequences  $\hat{\mathbf{X}}_{v_i}^{t-k:k}$  and  $\hat{\mathbf{X}}_{e_{ij}}^{t-k:k}$  for all nodes  $v_i \in \mathcal{V}_t$  and edges  $e_{ij} \in \mathcal{E}_t$ , respectively. Together with the multi-relational adjacency tensor  $\mathbf{A}^t \in \mathbb{R}^{|\mathcal{V}^t| \times |\mathcal{V}^t| \times |\mathcal{R}|}$ , the dynamic graph data at time step  $t$  is formatted and ready for input into the DyHetGNN model.

### 3. APPROACH

This study presents a approach for anomaly detection in the nodes and edges of dynamic heterogeneous supply chain graphs, aiming to jointly capture the key characteristics of dynamics and heterogeneity. The approach adopts a hierarchical modeling strategy, leveraging GNN to learn and integrate temporal dynamic features and heterogeneous multi-relational properties, while also designing an efficient anomaly detection mechanism. The primary optimization goal of this study is to minimize the reconstruction error for both nodes and edges, thereby enabling effective anomaly detection. The reconstruction error serves as a critical indicator of abnormal behavior, with anomalies being flagged when deviations from normal patterns exceed a predefined threshold. To achieve this, we formulate the optimization problem as minimizing the reconstruction error between the actual and reconstructed features of nodes and edges. The loss function employed for optimization is the Mean Squared Error (MSE), which quantifies the difference between the original and reconstructed feature vectors.

The approach consists of the following core modules:

1. **Message Passing Module:** This module captures the feature information of neighboring nodes and associated edges in the multi-relational heterogeneous graph, aggregates this information with weighted sums, and then applies an attention mechanism to weight and fuse messages from different relations, generating messages for nodes and edges.

2. **Dynamic Feature Update Module:** Based on multi-time-step message fusion and a temporal attention mechanism, this module combines messages with historical features, and employs time-series modeling to generate dynamic features for nodes and edges in the next time step.

3. **Dynamic-Heterogeneous Joint Reconstruction Module:** By jointly modeling temporal and heterogeneous information, this module reconstructs the features of nodes and edges and calculates the reconstruction error for anomaly detection.

4. **Interpretability Analysis Module:** This module combines reconstruction errors, attention weights, and other factors to analyze the contribution of each feature and different time steps to the anomaly score, providing an explanation for the anomaly detection process.

The following is a detailed introduction to each module.

### 3.1. Message Passing Module

The input is a dynamic heterogeneous graph, including dynamic node features  $\hat{\mathbf{X}}_{\mathbf{v}}^t$ , dynamic edge features  $\hat{\mathbf{X}}_{\mathbf{e}}^t$ , and a multi-relational adjacency tensor  $\mathbf{A}^t$ . For each relation type  $r$  a Graph Attention Network (GAT) based mechanism assigns attention weights to capture relational differences [14], followed by Relational Graph Convolutional Network (R-GCN) to incorporate relational features [15]. The message aggregation for node  $v_i$  at time  $t$  is:

$$\mathbf{m}_{\mathbf{v}_i}^t = \sum_{r \in \mathcal{R}} \beta_r \cdot \sum_{v_j \in \mathcal{N}_i^r} \alpha_{ij}^r \cdot (\mathbf{W}_r^{(1)} \hat{\mathbf{x}}_{v_j}^t + \mathbf{W}_r^{(2)} \hat{\mathbf{x}}_{e_{ij}}^t) \quad (7)$$

For edge  $e_{ij}$  at time  $t$ :

$$\mathbf{m}_{e_{ij}}^t = \sum_{r \in \mathcal{R}} \gamma_r \cdot (\mathbf{W}_r^{(3)} \hat{\mathbf{x}}_{v_i}^t + \mathbf{W}_r^{(4)} \hat{\mathbf{x}}_{v_j}^t + \mathbf{W}_r^{(5)} \hat{\mathbf{x}}_{e_{ij}}^t) \quad (8)$$

where  $\mathcal{N}_i^r$  denotes the set of neighbors of node  $v_i$  under relation  $r$ , and  $\alpha_{ij}^r$  is the attention weight for neighbor  $v_j$  to target node  $v_i$ .  $\beta_r$  and  $\gamma_r$  are the global weights for relation  $r$ ,  $\mathbf{W}_r^{(1)} \mathbf{W}_r^{(2)} \mathbf{W}_r^{(3)} \mathbf{W}_r^{(4)} \mathbf{W}_r^{(5)}$  are the linear transformation matrices for relation  $r$ . Node messages aggregate from neighbors and edges, while edge messages include both node and edge features, as node anomalies often relate to connection patterns, and edge anomalies to dynamic transaction or logistics data.

### 3.2. Dynamic Feature Update Module

The messages generated for nodes and edges,  $\mathbf{m}_{\mathbf{v}_i}^t$  and  $\mathbf{m}_{e_{ij}}^t$ , in the message passing module are input into the dynamic feature update module. This module integrates multi-step message fusion to capture long-term dependencies by combining features and messages from  $K$  past time steps. A GRU-based dynamic autoencoder is used [16], followed by Temporal Graph Attention (TGA) to assign different weights to historical features [17]. The updated node and edge features are:

$$\text{Node update: } \mathbf{h}_{\mathbf{v}_i}^{t+1} = \sum_{k=0}^{K-1} \delta_k \cdot [(1 - z_t) \cdot \tanh(\mathbf{W}_h \cdot [r_t \cdot \mathbf{h}_{\mathbf{v}_i}^{t-k} \mathbf{m}_{\mathbf{v}_i}^{t-k}] + b_h) + z_t \cdot \mathbf{h}_{\mathbf{v}_i}^{t-k}] \quad (9)$$

$$\text{Edge update: } \mathbf{h}_{e_{ij}}^{t+1} = \sum_{k=0}^{K-1} \varepsilon_k \cdot [(1 - z_t) \cdot \tanh(\mathbf{W}_h \cdot [r_t \cdot \mathbf{h}_{e_{ij}}^{t-k} \mathbf{m}_{e_{ij}}^{t-k}] + b_h) + z_t \cdot \mathbf{h}_{e_{ij}}^{t-k}] \quad (10)$$

where  $K$  is the time window size, and  $\delta_k, \varepsilon_k$  are the weights for each historical time step,  $r_t$  and  $z_t$  are the reset gate and update gate of the GRU, while  $\mathbf{W}_h$  and  $b_h$  are the weight matrix and bias term of the GRU, respectively. The "tanh" activation function is defined as  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ , which maps input values to a range between  $-1$  and  $1$ .

In the first time step of the dynamic graph, there is no historical information from the previous time step. Therefore, it is necessary to initialize the historical features of nodes and edges,  $\mathbf{h}_{\mathbf{v}_i}^0$  and  $\mathbf{h}_{e_{ij}}^0$ , to provide a starting point for subsequent dynamic feature updates. The initial historical feature vectors are obtained by applying a linear projection to the processed raw features through an embedding layer, formulated as follows:

$$\mathbf{h}_{\mathbf{v}_i}^0 = \mathbf{W}_{\text{init}}^{\mathbf{v}} \cdot \hat{\mathbf{x}}_{\mathbf{v}_i}^t \quad \mathbf{h}_{e_{ij}}^0 = \mathbf{W}_{\text{init}}^{\mathbf{e}} \cdot \hat{\mathbf{x}}_{e_{ij}}^t \quad (11)$$

where  $\mathbf{W}_{\text{init}}^{\mathbf{v}}, \mathbf{W}_{\text{init}}^{\mathbf{e}}$  are linear transformation matrices for node and edge feature embeddings.

### 3.3. Dynamic-Heterogeneous Joint Reconstruction Module

The input to the dynamic-heterogeneous joint reconstruction module comes from the output of the dynamic feature update module,  $\mathbf{h}_{v_i}^{t+1}$  and  $\mathbf{h}_{e_{ij}}^{t+1}$ . The input features of nodes and edges are reconstructed through dynamic autoencoders and heterogeneous autoencoders, respectively, calculating dynamic anomaly scores and heterogeneous anomaly scores. These scores are then weighted and fused to obtain the total anomaly score. During the training phase, the model is optimized using the MSE loss to fit the normal feature patterns. In the detection phase, the anomaly scores are computed based on the reconstruction error, and anomalies are detected using MES.

**1. Dynamic Autoencoder.** The encoder uses GRU to model temporal dynamics of nodes and edges at time  $t$ , computed as:

$$\begin{aligned} \mathbf{z}_{v_i}^d &= (1 - n_t) \cdot \tanh(\mathbf{W}_z \cdot [l_t \cdot \mathbf{h}_{v_i}^t \mathbf{h}_{v_i}^{t+1}] + q_h) + n_t \cdot \mathbf{h}_{v_i}^t \\ \mathbf{z}_{e_{ij}}^d &= (1 - n_t) \cdot \tanh(\mathbf{W}_z \cdot [l_t \cdot \mathbf{h}_{e_{ij}}^t \mathbf{h}_{e_{ij}}^{t+1}] + q_h) + n_t \cdot \mathbf{h}_{e_{ij}}^t \end{aligned} \quad (12)$$

where  $l_t$  and  $n_t$  are the reset gate and update gate of the GRU, while  $\mathbf{W}_z$  and  $q_h$  are the weight matrix and bias term of the GRU, respectively.

To reconstruct the original features from temporal embeddings, the decoder adopts a two-layer MLP [18], applied to node and edge features as:

$$\begin{aligned} \widehat{\mathbf{x}}_{v_i}^d &= \mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{z}_{v_i}^d + b_1) + b_2 \\ \widehat{\mathbf{x}}_{e_{ij}}^d &= \mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{z}_{e_{ij}}^d + b_1) + b_2 \end{aligned} \quad (13)$$

where  $\mathbf{W}_1$  and  $\mathbf{W}_2$  represent the weight matrices for the first and second layers of MLP, respectively;  $b_1$  and  $b_2$  denote the bias terms for the first and second layers of the MLP; The ReLU activation function is defined as  $\text{ReLU}(x) = \max(0, x)$ , which outputs zero for negative inputs and the input itself for non-negative inputs.

**2. Heterogeneous Autoencoder.** The encoder models heterogeneous features using relational attention [19], and computes low-dimensional representations as:

$$\mathbf{z}_{v_i}^h = \sum_{r \in \mathcal{T}_e} \epsilon_r \cdot f_r(\mathbf{h}_{v_i}^{t+1}) \quad \mathbf{z}_{e_{ij}}^h = \sum_{r \in \mathcal{T}_e} \zeta_r \cdot g_r(\mathbf{h}_{e_{ij}}^{t+1}) \quad (14)$$

where  $\mathcal{T}_e$  is the set of edge types;  $\epsilon_r$  and  $\zeta_r$  are the weights of relation  $r$  learned by the relational attention mechanism;  $f_r$  and  $g_r$  are the relation-specific encoders.

The decoder uses MLP with relation-specific weights to reconstruct heterogeneous features as:

$$\begin{aligned} \widehat{\mathbf{x}}_{v_i}^h &= \epsilon_r \cdot \mathbf{W}'_2 \cdot \text{ReLU}(\mathbf{W}'_1 \mathbf{z}_{v_i}^h + b'_1) + b'_2 \\ \widehat{\mathbf{x}}_{e_{ij}}^h &= \zeta_r \cdot \mathbf{W}'_2 \cdot \text{ReLU}(\mathbf{W}'_1 \mathbf{z}_{e_{ij}}^h + b'_1) + b'_2 \end{aligned} \quad (15)$$

where  $\mathbf{W}'_1$  and  $\mathbf{W}'_2$  represent the weight matrices for the first and second layers of MLP, respectively;  $b'_1$  and  $b'_2$  denote the bias terms for the first and second layers of the MLP; The ReLU activation function is employed to introduce non-linearity into the model.

**3. Loss Function.** The model minimizes both dynamic and heterogeneous reconstruction errors using mean squared error (MSE) loss [20], defined as:

$$\text{Dynamic Loss: } \mathcal{L}^d = \frac{1}{|\mathcal{V}_t|} \sum_{v_i \in \mathcal{V}_t} \|\mathbf{x}_{v_i}^t - \widehat{\mathbf{x}}_{v_i}^d\|_2^2 + \frac{1}{|\mathcal{E}_t|} \sum_{e_{ij} \in \mathcal{E}_t} \|\mathbf{x}_{e_{ij}}^t - \widehat{\mathbf{x}}_{e_{ij}}^d\|_2^2 \quad (16)$$

$$\text{Heterogeneous Loss: } \mathcal{L}^h = \frac{1}{|\mathcal{V}_t|} \sum_{v_i \in \mathcal{V}_t} \|\mathbf{x}_{v_i}^t - \widehat{\mathbf{x}}_{v_i}^h\|_2^2 + \frac{1}{|\mathcal{E}_t|} \sum_{e_{ij} \in \mathcal{E}_t} \|\mathbf{x}_{e_{ij}}^t - \widehat{\mathbf{x}}_{e_{ij}}^h\|_2^2 \quad (17)$$

$$\text{Total loss: } \mathcal{L} = \eta_d \cdot \mathcal{L}^d + \eta_h \cdot \mathcal{L}^h \quad (18)$$

where  $\eta_d$  and  $\eta_h$  are tunable parameters, the loss contribution for dynamics and heterogeneity.

**4. Anomaly Detection.** Anomaly scores are computed based on reconstruction errors, with dynamic and heterogeneous anomaly scores fused to obtain the total anomaly score. These scores are then normalized using significance scores (MES) to mitigate feature scale or global shift effects. The dynamic, heterogeneous, and total anomaly scores are given by the following equations:

$$\text{Dynamic Anomaly Scores: } S_{v_i}^d = \|\mathbf{x}_{v_i}^t - \widehat{\mathbf{x}}_{v_i}^d\|_2^2 \quad S_{e_{ij}}^d = \|\mathbf{x}_{e_{ij}}^t - \widehat{\mathbf{x}}_{e_{ij}}^d\|_2^2 \quad (19)$$

$$\text{Heterogeneous Anomaly Scores: } S_{v_i}^h = \|\mathbf{x}_{v_i}^t - \widehat{\mathbf{x}}_{v_i}^h\|_2^2 \quad S_{e_{ij}}^h = \|\mathbf{x}_{e_{ij}}^t - \widehat{\mathbf{x}}_{e_{ij}}^h\|_2^2 \quad (20)$$

$$\text{Total Anomaly Score: } S_{v_i} = \eta_d \cdot S_{v_i}^d + \eta_h \cdot S_{v_i}^h \quad S_{e_{ij}} = \eta_d \cdot S_{e_{ij}}^d + \eta_h \cdot S_{e_{ij}}^h \quad (21)$$

The anomaly scores are then normalized into significance scores (MES) as follows:

$$\text{MES}_{v_i} = \frac{S_{v_i} - \mu_s}{\sigma_s} \quad \text{MES}_{e_{ij}} = \frac{S_{e_{ij}} - \mu'_s}{\sigma'_s} \quad (22)$$

where  $\mu_s$ ,  $\mu'_s$  and  $\sigma_s$ ,  $\sigma'_s$  are the mean and standard deviation of the anomaly scores for nodes and edges, respectively. The anomaly scores are then compared to predefined thresholds  $\tau_v$  and  $\tau_e$ :

$$\text{MES}_{v_i} > \tau_v \quad \text{MES}_{e_{ij}} > \tau_e \quad (23)$$

### 3.4. Interpretability Analysis Module

The interpretability module aims to explain the sources of anomaly scores through feature-level and time-level contributions, and visualize anomalous nodes and edges.

**1. Feature-Level Contribution.** This analysis quantifies the impact of each feature dimension on the anomaly scores. The contributions to the anomaly scores are calculated as the absolute gradients with respect to the node and edge features:

$$\text{Contribution of the } n\text{-th feature of node } v_i: C_{v_i, n} = |\nabla x_{v_i, n}^t S_{v_i}| \quad (24)$$

$$\text{Contribution of the } n'\text{-th feature of edge } e_{ij}: C_{e_{ij}, n'} = |\nabla x_{e_{ij}, n'}^t S_{e_{ij}}| \quad (25)$$

where  $\nabla x_{v_i, n}^t S_{v_i}$  and  $\nabla x_{e_{ij}, n'}^t S_{e_{ij}}$  are the gradients of the anomaly scores with respect to the node and edge features, respectively.

**2. Time-Level Contribution.** Time-level contribution analysis uses a temporal attention mechanism to quantify the effect of each time step on anomaly scores. The time-level contribution for node and edge is computed as:

$$\text{Time-level contribution for node } v_i: C_{v_i}^t = \sum_{k=0}^{K-1} \theta_k S_{v_i}^{t-k} \quad (26)$$

$$\text{Time-level contribution for edge } e_{ij}: C_{e_{ij}}^t = \sum_{k=0}^{K-1} \vartheta_k S_{e_{ij}}^{t-k} \quad (27)$$

where  $\theta_k$  and  $\vartheta_k$  are the temporal attention weights for node and edge features, respectively.

## 4. TESTING DESIGN

### 4.1. Dataset description

The test uses the publicly available SupplyGraph dataset [21], provided by a leading FMCG company in Bangladesh, which is a key player in the regional supply chain industry. The dataset spans from January 1, 2023, to August 9, 2023, covering 221 days, with each time step representing a day's data.

The dataset is structured as a dynamic heterogeneous graph with three node types and three edge types, each associated with distinct features. The number and distribution of nodes and edges are as follows:

*Table 1*  
Summary of Node and Edge Types in the SupplyGraph Dataset

Category	Type	Quantity	Features
Node	Product Node	41	Production demand, actual shipment, sales order quantity, etc.
	Production Facility Node	25	Factory capacity utilization, failure events, downtime duration, etc.
	Storage Facility Node	13	Storage capacity utilization, inventory volume, turnover rate, etc.
Edge	Product-Production Facility Edge	150	Relationship of products produced at a facility, including daily production volume and order completion rate.
	Production-Storage Facility Edge	300	Relationship of transporting products from a production facility to a storage facility, including transport volume, time, and delays.
	Product-Storage Facility Edge	234	Relationship of products stored in a storage facility, including inventory demand and utilization rates.

## 4.2. Baseline models

To ensure a fair comparison, we carefully selected five baseline models that are widely used in anomaly detection and graph modeling. These models represent various approaches in dynamic graph modeling, heterogeneous graph analysis, and anomaly detection. All baseline models were configured with comparable complexity and trained using similar hyperparameters, including node and edge embedding dimensions, time window size, and the same learning rate and optimizer settings.

1) LSTM: A classic recurrent neural network that excels at capturing long-term dependencies in time-series data [3]; 2) GCN: A standard GNN model that aggregates node features through graph convolutions, suitable for static graph data [22]; 3) HGT: A Transformer-based model for heterogeneous graphs, using multi-head attention to capture complex interactions between different relation types [23]; 4) T-GCN: A GCN extension for dynamic graphs, integrating GRU to model temporal dependencies [11]; 5) DHGNN: A model that jointly handles dynamic and heterogeneous graphs, capturing multi-relational and temporal dynamics [13].

## 4.3. Training details

The tests were conducted on NVIDIA Quadro RTX 4000 hardware, with Python 3.10 as the experimental environment. The training data was divided sequentially, with 70% of the time steps used for training, 15% for validation, and 15% for testing, ensuring temporal consistency to avoid information leakage. The model input includes dynamic feature matrices and multi-relational adjacency tensors, with node and edge embedding dimensions set to 64. A dynamic modeling approach with a time window size of 5 was employed, capturing data from the previous five time steps for modeling the current time step. During training, the Adam optimizer was used with an initial learning rate of 0.001, and a cosine annealing strategy was applied to dynamically adjust the learning rate to enhance convergence. Additionally, weight decay (0.0001) was applied to prevent overfitting. To mitigate gradient explosion, a gradient clipping mechanism with a threshold of 5.0 was used. The training batch size was set to 32, with a maximum of 200 epochs for optimization, and an early stopping strategy was implemented, terminating the training when the validation performance did not improve for 10 consecutive epochs. Model performance was evaluated every 5 epochs on the validation set, and evaluation metrics were recorded.

## 5. RESULTS ANALYSIS

In this section, we present the results obtained by applying the proposed DyHetGNN approach, as described in previous sections, to anomaly detection tasks in supply chains. The theoretical foundations provided in Section 3, including the dynamic heterogeneous graph construction, message passing, and dynamic feature updates, were directly applied to model and evaluate both node and edge anomalies.

We first validated the model using publicly available datasets and compared the results with baseline models. The application of temporal dynamics and multi-relational heterogeneity allowed the model to detect both node and edge anomalies effectively, outperforming traditional approaches. Specifically, the dynamic-

heterogeneous joint reconstruction module helped reduce the reconstruction error, as detailed in the theory section, allowing for better classification of anomalies during the validation phase.

For model validation, we used metrics AUC-ROC and F1-Score to evaluate the model’s performance[24]. The validation process was conducted using the validation set every 5 epochs, as explained in Section 4, with the model’s performance closely monitored for signs of overfitting or underfitting. These metrics were used to adjust model parameters, such as learning rate and weight decay, to ensure optimal performance.

The results show that DyHetGNN effectively captures temporal and relational patterns in the data, addressing the challenges discussed in Section 2. Furthermore, we conducted ablation studies to verify the contributions of each module, as described in Section 3. These studies confirmed the importance of the message passing, dynamic modeling, and dynamic-heterogeneous reconstruction components in enhancing the model’s ability to detect anomalies in both nodes and edges.

### 5.1. Baseline comparison

We compared the proposed DyHetGNN with five baseline models. Results show that DyHetGNN significantly outperforms other models in both node and edge anomaly detection tasks. The detailed comparison is as follows.

Table 2  
Performance Comparison of Models in Node and Edge Anomaly Detection

Model	Node		Edge	
	AUC-ROC (%)	F1-Score (%)	AUC-ROC (%)	F1-Score (%)
LSTM	80.9	68.5	78.3	65.9
GCN	78.2	65.4	76.3	63.5
HGT	82.3	70.1	80.2	67.8
T-GCN	84.5	73.2	83.1	70.5
DHGNN	87.8	75.8	86.4	73.9
<b>DyHetGNN</b>	<b>90.6</b>	<b>77.8</b>	<b>91.8</b>	<b>79.2</b>

DyHetGNN achieves superior performance in both node and edge anomaly detection, with higher AUC-ROC and F1-Score. By jointly modeling temporal dynamics and multi-relational features, DyHetGNN outperforms static and time-series models in identifying complex supply chain anomalies. Its innovative integration of dynamic and heterogeneous features further enhances detection accuracy compared to advanced models like T-GCN and DHGNN.

### 5.2. Ablation Study

To evaluate the contributions of each module in DyHetGNN, ablation experiments were conducted by systematically removing key components: the message-passing mechanism, dynamic modeling, dynamic-heterogeneous reconstruction, and the MES-based normalization.

Table 3  
Results of Ablation Experiments

Experiment	Node		Edge	
	AUC-ROC (%)	F1-Score (%)	AUC-ROC (%)	F1-Score (%)
Without Message Passing	83.1	70.2	84.7	72.5
Without Dynamic Modeling	85.3	72.5	86.4	74.1
Without Dynamic-Heterogeneous Reconstruction	82.1	70.4	84.6	72.8
Without MES Normalization	88.2	75.1	89.3	77.3
<b>Complete Model</b>	<b>90.6</b>	<b>77.8</b>	<b>91.8</b>	<b>79.2</b>

The message-passing mechanism improves edge anomaly detection by capturing multi-relational features, with AUC-ROC dropping significantly (91.8%  $\rightarrow$  84.7%) when removed. The dynamic modeling module enhances temporal dependency learning, especially for nodes (AUC-ROC: 90.6%  $\rightarrow$  85.3%), showing

its critical role. The dynamic-heterogeneous reconstruction module boosts complex anomaly detection, as its removal reduces edge F1-Score notably (79.2%  $\rightarrow$  72.8%). The MES normalization ensures robustness by addressing feature scale differences, with a slight drop in node AUC-ROC (90.6%  $\rightarrow$  88.2%) upon removal.

### 5.3. Interpretability analysis

**1. Feature-Level Contribution Analysis.** Feature-level analysis identifies key factors driving supply chain anomalies, providing insights for risk management and model optimization.

Using a Sunburst chart, we visualize feature contributions across node and edge types, where segment size indicates contribution magnitude. Results highlight capacity utilization, production demand, outbound demand, transport volume, and failure events as primary contributors. Fluctuations in these features often disrupt production and logistics, causing anomalies. In contrast, sales order volume, throughput, and inventory capacity show lower contributions, as they reflect demand patterns rather than direct anomaly triggers.

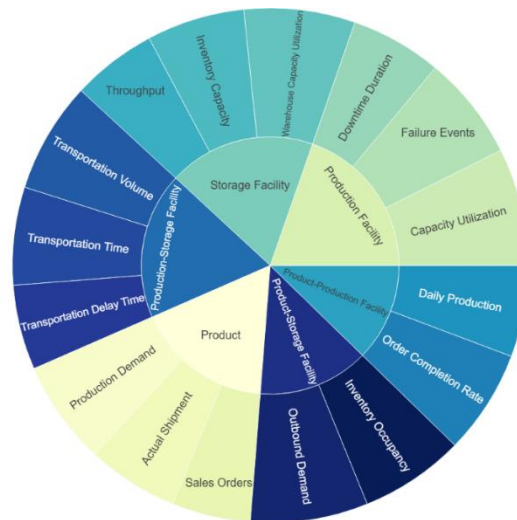


Fig. 1 – Feature-level contribution chart.

These results emphasize that production and transportation variations are critical in anomaly detection. Monitoring high-contribution features enables proactive risk management and system stability.

**2. Time-Level Contribution Analysis.** Time-level analysis quantifies the impact of each time step on anomaly detection, revealing key periods of influence.

Figure 2 shows the time-step contributions over 221 days, with blue representing nodes and orange for edges. Node contributions peak between days 50–100, linked to demand fluctuations and production issues, then decrease during the recovery period (days 100–150) and stabilize afterward. Edge contributions peak between days 120–160 due to transportation and inventory anomalies and remain moderate thereafter.

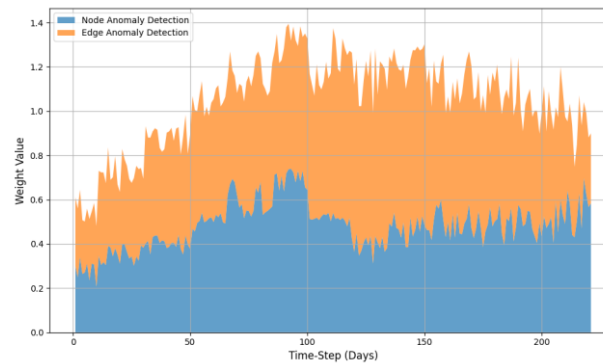


Fig. 2 – Time-level contribution chart.

This analysis identifies critical risk periods, providing insights into temporal anomaly patterns for targeted decision-making.

**3. Visualization of Anomalous Nodes and Edges.** We visualize anomalous nodes and edges to identify key patterns and dynamic behaviors in the supply chain. Nodes and edges are represented with color, shape, and thickness: product nodes (blue circles), production facilities (orange squares), and storage facilities (green triangles). Edges include product-to-production (cyan), production-to-storage (red), and product-to-storage (purple). Nodes and edges exceeding the anomaly threshold turn gray, with edges becoming thicker to highlight anomalies.

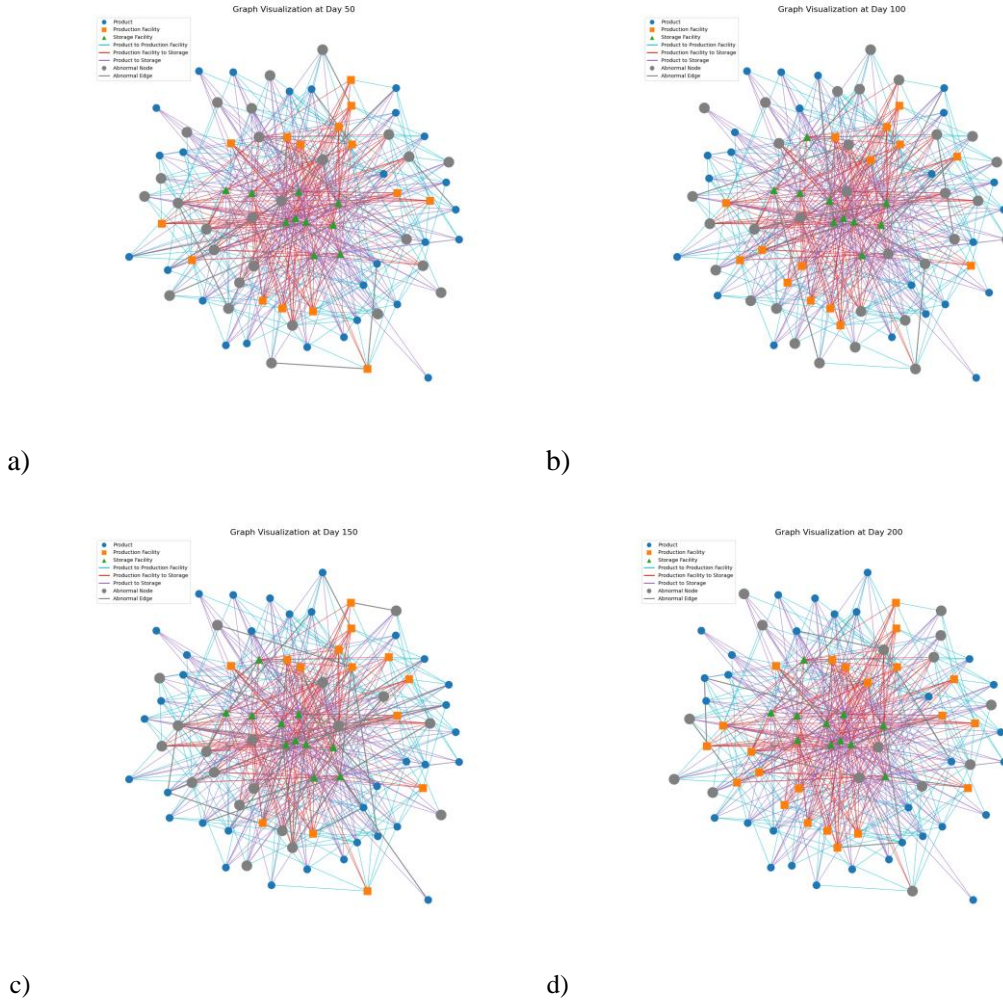


Fig. 3 – Visualization of anomalous nodes and edges at key time points: a) Day 50; b) Day 100; c) Day 150; d) Day 200.

We selected days 50, 100, 150, and 200 as they correspond to critical turning points, such as demand fluctuations, production bottlenecks, and transport delays. These visualizations reveal how anomalies evolve over time, providing insights into their distribution and impact. By highlighting these changes, managers can better locate risks, monitor key periods, and optimize supply chain operations.

## 6. CONCLUSION

This paper proposes a DyHetGNN-based approach for supply chain anomaly detection, addressing challenges of multi-relational heterogeneity and temporal dynamics. By jointly modeling the temporal and relational features of nodes and edges, the approach achieves comprehensive anomaly detection. Results demonstrate that DyHetGNN outperforms baseline models in AUC-ROC and F1-Score, validating its effectiveness in complex supply chain networks. Ablation studies confirm the importance of the message-passing mechanism, dynamic modeling, and dynamic-heterogeneous reconstruction modules. Additionally,

the interpretability module, through feature-level and time-level analysis, along with anomaly visualization, enhances model transparency and practical utility. Future work will focus on improving computational efficiency for large-scale supply chain networks and enhancing the model's robustness to uncertainties and unexpected disruptions.

## REFERENCES

- [1] Khedr AM, Rani S. Enhancing supply chain management with deep learning and machine learning techniques: A review. *J Open Innov Technol Mark Complex*. 2024; 10: 100379. DOI: 10.1016/j.joitmc.2024.100379.
- [2] Glaser AE, Harrison JP, Josephs D. Anomaly detection methods to improve supply chain data quality and operations. *SMU Data Sci Rev*. 2022; 6(1): 3.
- [3] Tang Y, Wang Y, Liu C, Yuan X, Wang K, Yang C. Semi-supervised LSTM with historical feature fusion attention for temporal sequence dynamic modeling in industrial processes. *Eng Appl Artif Intell*. 2023; 117: 105547. DOI: 10.1016/j.engappai.2022.105547.
- [4] Tran KP, Nguyen HD, Thomassey S. Anomaly detection using long short term memory networks and its applications in supply chain management. *IFAC-Pap Online*. 2019; 52(13): 2408–2412. DOI: 10.1016/j.ifacol.2019.11.567.
- [5] Deng C, Liu Y. A deep learning-based inventory management and demand prediction optimization method for anomaly detection. *Wirel Commun Mob Comput*. 2021; 1: 9969357. DOI: 10.1155/2021/9969357.
- [6] Ma X, Wu J, Xue S, Sheng QZ, Xiong H, Yang J, Zhou C, Akoglu L. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans Knowl Data Eng*. 2023; 35(12): 12012–12038. DOI: 10.1109/TKDE.2021.3118815.
- [7] Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A. Deep learning for time series forecasting: a survey. *Big Data* 2021; 9(1): 3–21. DOI: 10.1089/big.2020.0159.
- [8] Peng H, Du B, Liu M, Liu M, Ji S, Wang S, Zhang X, He L. Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. *Inf Sci*. 2021; 578: 401–416. DOI: 10.1016/j.ins.2021.07.007.
- [9] Wang X, Bo D, Shi C, Fan S, Ye Y, Yu PS. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Trans Big Data*. 2022; 9(2): 415–436. DOI: 10.1109/TBDATA.2022.3177455.
- [10] Wiedmer R, Griffis SE. Structural characteristics of complex supply chain networks. *J Bus Logist*. 2021; 42(2): 264–290. DOI: 10.1111/jbl.12283.
- [11] Liu X. Viewpoints on robust supply chain network risk assessment using dynamic Bayesian networks. *Innov Appl Eng Technol*. 2023; 2(1): 1–8. DOI: 10.58195/iaet.v2i1.135.
- [12] Ekle OA, Eberle W. Anomaly detection in dynamic graphs: a comprehensive survey. *ACM Trans Knowl Discov Data*. 2024; 18(8): 1–44. DOI: 10.1145/3669906.
- [13] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans Neural Netw* 2008; 20(1): 61–80. DOI: 10.1109/TNN.2008.2005605.
- [14] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. *arXiv* 2017: 1710.10903. DOI: 10.48550/arXiv.1710.10903.
- [15] Rajmohan R, Kumar TA, Sandhya SG, Rani M, Sivalingam K. R-GCN: a residual-gated recurrent unit convolution network model for anomaly detection in blockchain transactions. *Multimed Tools Appl*. 2024; 83: 87527–87551. DOI: 10.1007/s11042-023-17942-x.
- [16] Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks. In: *Proc IEEE Int Midwest Symp Circuits Syst (MWSCAS)*; 2017, pp. 1597–1600. DOI: 10.1109/MWSCAS.2017.8053243.
- [17] Zhang C, James JQ, Liu Y. Spatial-temporal graph attention networks: a deep learning approach for traffic forecasting. *IEEE Access*. 2019; 7: 166246–166256. DOI: 10.1109/ACCESS.2019.2953888.
- [18] Taud H, Mas JF. Multilayer perceptron (MLP). In: Camacho Olmedo M, Paegelow M, Mas JF, Escobar F, editors. *Geomatic approaches for modeling land change scenarios. Lecture notes in geoinformation and cartography*. Cham, Switzerland: Springer; 2018, pp. 451–455. DOI: 10.1007/978-3-319-60801-3\_27.
- [19] Yu L, Sun L, Du B, Liu C, Lv W, Xiong H. Heterogeneous graph representation learning with relation awareness. *IEEE Trans Knowl Data Eng*. 2023; 35(6): 5935–5947. DOI: 10.1109/TKDE.2022.3160208.
- [20] Meni MJ, White RT, Mayo ML, Pilkiewicz KR. Entropy-based guidance of deep neural networks for accelerated convergence and improved performance. *Inf Sci*. 2024; 681: 121239. DOI: 10.1016/j.ins.2024.121239.
- [21] Wasi AT, Islam MD, Akib AR. SupplyGraph: a benchmark dataset for supply chain planning using graph neural networks. *arXiv* 2024;2401.15299. DOI: 10.48550/arXiv.2401.15299.
- [22] Zhang S, Tong H, Xu J, Maciejewski R. Graph convolutional networks: a comprehensive review. *Comput Soc Netw*. 2019; 6(1): 1–23. DOI: 10.1186/s40649-019-0069-y.
- [23] Mei X, Cai X, Yang L, Wang N. Relation-aware heterogeneous graph transformer based drug repurposing. *Expert Syst Appl*. 2022; 190: 116165. DOI: 10.1016/j.eswa.2021.116165.
- [24] Bobadilla J, Gutiérrez A. Wasserstein GAN-based architecture to generate collaborative filtering synthetic datasets. *Appl Intell*. 2024; 54: 2472–2490. DOI: 10.1007/s10489-024-05313-4.

*Received December 18, 2024*