Contents lists available at www.goldenlightpublish.com/journals/jcemi/2-toc

Journal of Construction Engineering, Management & Innovation

Journal homepage: www.goldenlightpublish.com/journals/2/jcemi

Construction performance classification with controlled information flow in dense engineering activity graphs

Fatemeh Mostofi[1,2,*], Onur Behzat Tokdemir[3,4], Vedat Toğan[2]

[1] Machine Learning Engineer, OPAL AI INC., Los Angeles, California, USA
[2] Karadeniz Technical University, Civil Engineering Department, Trabzon, Türkiye
[3] Istanbul Technical University, Civil Engineering Department, Istanbul, Türkiye
[4] Istanbul Technical University, Vice Chair, Artificial Intelligence and Data Science Research Center, İstanbul, Türkiye

[*] Corresponding author: F. Mostofi (fatemee.mostofi@gmail.com)

https://doi.org/10.31462/jcemi.2026.680
Received 23 January 2026; Revised 12 May 2026; Accepted 15 June 2026; Available online 29 June 2026

Keywords

Graph neural networks
Gated mechanism
High-granularity data
Class imbalance
Construction management
Cost performance index

Abstract

Earned value management (EVM) is critical for monitoring financial and operational performance in large-scale projects. However, its implementation is limited by the complexity of modern construction workflows. Particularly, modeling the cost performance index (CPI) to evaluate the efficiency of activity delivery plans has led to the development of various relational learning models. This study addresses the inability of current relational models to handle high-resolution construction data with dense WBS-based activity networks, where uncontrolled information flow and class imbalance reduce model expressiveness. In this study, we introduce a CPI-specific activity learning mechanism that forecasts CPI using a gated approach to regulate information propagation in dense spatiotemporal data. To contextualize the proposed information modeling framework, this study draws upon a large-scale dataset derived from 77 construction progress reports collected over an 18-month period, which were cleaned and encoded into 52 unique delivery-week categories for model training. We demonstrate that the proposed gated architectures outperform previously proposed attention-based mechanisms. Across eight timeframes of real-world construction datasets, GatedGNN shows superior performance, improving accuracy by 14% and F1 scores by 13% over GCN and GAT models. These improvements enable more reliable CPI classification (efficient vs. inefficient) and support timely corrective decision-making by producing activity-level risk flags that can guide managerial review of resource allocation, schedule coordination, and WBS-specific cost deviations. By improving fidelity in class distribution and managing relational complexity, gated GNNs enable earlier and more accurate identification of inefficiencies. This facilitates proactive decision-making, allowing engineering managers to intervene before performance deviations escalate.

1. Introduction

The construction industry functions within a highly dynamic VUCA environment characterized by volatility, uncertainty, complexity, and ambiguity [1–4]. Variables such as labor availability, logistics coordination, technological adaptation, weather conditions, and regulatory changes collectively introduce substantial uncertainty into project execution. Furthermore, inconsistent information regarding complex operational variables, productivity metrics [5], and logistics performance [6] complicates effective planning and decision-making processes. In real-world construction systems, delays and cost overruns interact recursively, with each reinforcing and amplifying the consequences of the other. Under uncertain conditions [7], these feedback effects accumulate and result in escalating budget inconsistencies and systemic impacts on stakeholders [8, 9]. Construction projects commonly fail to achieve intended outcomes related to cost, schedule, and quality, despite these dimensions collectively defining project success. Such deviations may cascade through supply chains and broader economic systems, negatively influencing national revenue generation and macroeconomic equilibrium [10].

1.1. Progress management in construction projects

In this context, earned value management (EVM) thus plays a strategic evaluative role [11] and remains embedded in monitoring protocols. PMI's methodology [12] defines economic efficiency through the cost performance index (CPI) (Eq. (1)). CPI variations indicate budget variance, though modeling confronts the complexity of work breakdown structure (WBS)-based datasets.

$$CPI = \frac{EV}{AC} \tag{1}$$

Recent work emphasizes the versatility of graph neural networks (GNNs) in handling non-Euclidean engineering data such as BIM and point clouds [13, 14]. As a subclass of neural networks [15] GNNs have proven highly effective for modeling relational structures commonly found in construction engineering data. These models fulfill the growing need for machine learning (ML) approaches that account for both spatial and temporal factors in project data [16, 17].

GNNs integrate the structural capabilities of graph theory with the predictive prowess of neural networks, enabling the extraction of richer and more relevant feature representations from graph-structured data [18]. They excel in representing and reasoning with data in graph form, allowing them to manage complex networks by learning from both the connections and inherent features [19]. By combining the structural capabilities of graph theory with the predictive power of deep neural networks, GNNs create models that can learn richer information from graph representation learning [20].

1.2. Application of GNNs in engineering project management

Ensuring accurate progress tracking is vital in construction [21, 22]. While ML [23], building information modeling (BIM) [24], and video tracking have helped [25], GNNs are gaining attention for their strength in learning from structured graph data [26, 27]. GNNs use message-passing to learn from node dependencies, unlike conventional ML, which makes them ideal for connected data [13, 28, 29].

Recent advances in graph learning have led to the development of architectures such as node2vec [30], graph attention networks (GATs) [31, 32], and spatio-temporal GCNs [31, 32], which have demonstrated strong performance across complex graph-structured datasets [28]. Within this landscape, GCNs and GATs remain among the most commonly implemented GNNs [33]. GCNs generalize convolutional neural networks (CNNs) operations to non-Euclidean domains, effectively learning from node interdependencies [34] and have been applied in various engineering use cases, including rework cost prediction [35, 36]. Meanwhile, GATs enhance this framework by using attention mechanisms to dynamically weight neighbor contributions, allowing the model to generate more nuanced node representations [37, 38]. These capabilities make GATs highly flexible for capturing relational dependencies in construction and engineering tasks [38, 39].

1.3. Research gap and proposed solution

GNNs excel at capturing structured dependencies in construction data [40], and outperform conventional models like random forests, Naïve Bayes, and multilayer perceptrons in relational tasks [36, 41]. However, our findings reveal that common GNN variants, as GCNs and GATs, underperform when managing intricate information flows in high-resolution project datasets. Although attention mechanisms enhance GCNs [33], limited work has investigated gating structures capable of dynamically regulating and prioritizing signal flow. Such methods could inherently mitigate class skew, reducing dependence on augmentation.

Our approach centers on two architectures designed to mitigate class imbalance and feature over smoothing prevalent in dense activity networks. Thus, the proposed model contributes in two primary ways:

- *Contribution 1: Higher-precision prediction in dense activity networks:* To improve CPI classification in highly structured engineering datasets, we develop a gated GNN architecture that selectively modulates message passing between temporally bound and WBS-linked tasks. This architecture outperforms baseline GNNs by reducing oversmoothing and enhancing node-level feature differentiation. Prior research applied similar mechanisms in crack detection [42], sensor fusion [43], and image segmentation [44]. So the proposed architecture is expected to enhance the predictive capabilities of GNNs in the domain of engineering project management through the integration of gating mechanisms.
- *Contribution 2: Reduced bias from imbalanced data distributions:* Class imbalance continues to degrade performance, and targeted message propagation between tasks remains an under-researched approach to improve model reliability in construction forecasting. This limits prediction accuracy, especially in critical domains like fraud or construction analytics [45–47], where accurate predictions are essential for effective decision-making. In response, a growing body of literature has thoroughly surveyed this conundrum and its manifold remedies [48–53]. To address class imbalance, recent studies have explored various hybrid methods to improve the representation of minority classes in GNNs. For instance, [54] proposed, which synthesizes harder samples to enlarge the decision boundaries of minority classes, thereby reducing the compression of minority classes in latent space. [55] proposed Nia-GNNs that oversample minority nodes based on predicted labels and similarities, creating interpolated synthetic nodes to improve representation. In another study, [56] employed a Meta-GCN model that dynamically weights class samples during training, minimizing loss on minority classes without overfitting to outliers. In a recent work, [57] proposed an ensemble module utilizing dynamic weight attention to balance spatial and embedding information, reducing bias

toward majority classes and outperforming benchmark models like GAT.

Our prior research [58] illustrated that integrating a variational autoencoder (VAE) increased GAT's accuracy by about 6%, affirming the utility of generative techniques [59]. Yet, when confronted with more expansive datasets, the VAE faltered, producing NaN values and challenging the model's reliability. Similar difficulties arose with GAN-based approaches used in balancing visual data [53, 60]; their effectiveness hinged upon coupling with undersampling, a process often deemed computationally and practically unwieldy. Therefore, the proposed design stabilizes model performance under data imbalance and feature oversmoothing. Applied to construction graphs, the model enables early identification of inefficient activities by flagging activity records with high predicted risk of CPI inefficiency. These flags can support proactive managerial decisions by directing attention to specific WBS groups, delivery weeks, and activity clusters requiring resource, schedule, or cost-control review.

The primary contribution of this study is the application of recent GNNs advancements to the modeling of interdependent construction tasks, which often require coordination across shared time frames (e.g., weekly schedules) and structured planning frameworks (e.g., WBS alignments). Specifically, the study focuses on improving the classification of activities based on the CPI. Accurate CPI classification is essential for project managers to forecast activity efficiencies and to support proactive decision-making. This enables engineering professionals to assess the planning accuracy based on ongoing project progress. Construction projects are modeled as temporally synchronized, WBS-aligned graphs. Predicting CPI-based efficiency classifications is the core task. This allows managers to recognize underperforming activities early and adjust resource strategies accordingly.

2. Research Methodology

Fig. 1 illustrates the research methodology designed to evaluate the role of gating in enhancing prediction accuracy and information flow control. Data spanning three years from a large-scale construction project were transformed into graph-based networks, with nodes representing activities linked through spatial and temporal relations. Eight datasets reflected real-world variance and imbalance. Baselines included GCN and GAT, while GatedGNN introduced dynamic gating and ResGatedGNN added residual paths for depth. Iterative tuning and evaluation via accuracy, F1, and AUC-ROC confirmed the superiority of gated models, particularly under class-imbalanced conditions. This framework provides a foundation for advancing GNN applications in construction.

As shown in Fig. 1, the workflow consists of five main stages. First, raw progress reports were collected and standardized to extract activity-level cost, progress, labor, WBS, and timing information. Second, CPI values were calculated and converted into binary activity performance labels. Third, the cleaned activity records were encoded into graph structures by representing activities as nodes and linking activities that shared the same delivery week and WBS group. Fourth, baseline GNN models, including GCN and GAT, were trained and compared with the proposed GatedGNN and ResGatedGNN architectures. Finally, model performance was evaluated across eight rolling time-frame datasets using accuracy, precision, recall, F1 score, and AUC-ROC.

3. Configuration of Activity Information Learning Mechanism in CPI Prediction Models

3.1. Activity workflow as Graph Neural Networks (GNNs) models

Engineering research has highlighted the adaptability of GNNs in addressing the multifaceted challenges in project management and safety [35, 61–63]. Modeling the relationships among activities is not possible with most of the existing activity progress prediction approaches. Existing ML approaches in engineering progress prediction typically involve collecting and preprocessing cost records, organizing them into tabular formats where each record represents an activity with various features, as shown in Eq. (2). Here, **X** is an input matrix of size n×d, and **Y** is an output vector of size n. n and d represent the numbers of engineering progress records.

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad (2)$$

$$x_{ij} \in R^{n\times d}, \forall\, i = 1, 2, \ldots, n, \forall\, j = 1, 2, \ldots, d\,, y_i \in R^{n}, \forall\, i = 1, 2, \ldots, n$$

ML model uses a loss function to receive feedback and improve through various optimization and activation functions. However, these ML modeling approaches often fail to capture intricate transition of items, particularly when attempting to assess the sequential performance metrics such as the CPI [64].

3.2. Addressing representational limitations of gated GNNs

GCNs and GATs are now de-facto standard in relational data tasks [14, 54, 58, 65, 66], but suffer under class imbalance [55–57, 67]. Class imbalance, where one class overshadows others not only distorts learned representations but also undermines the veracity of predictive outcomes [51, 68]. In GNN models the skewness in training data affects the latent space, having the major class squeezing the underrepresented one, called the squeezed minority problem [54]. This imbalance represents a significant challenge for predictive models, often compromising their ability to accurately identify minority classes [69].

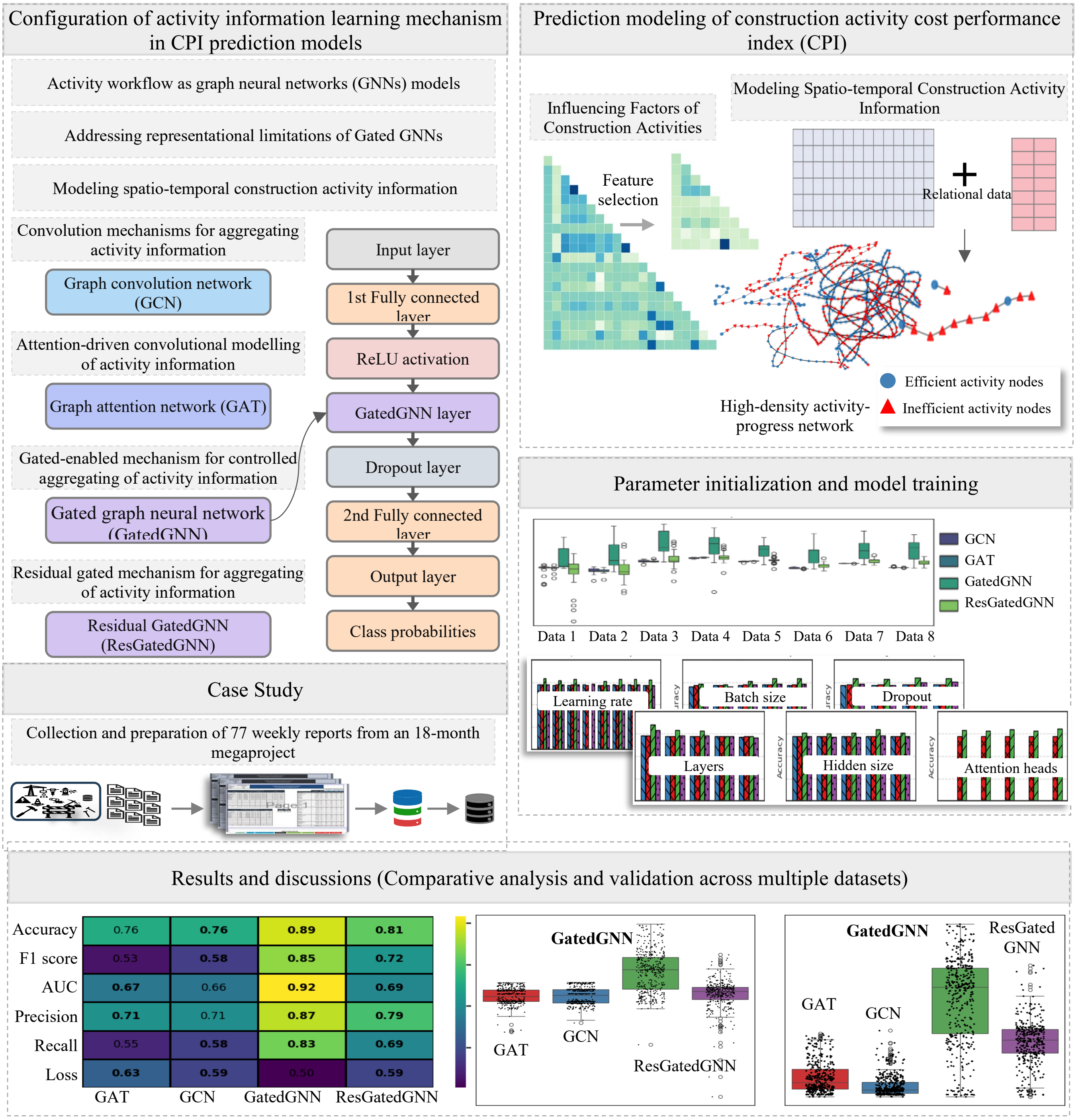


Fig. 1. Research methodology

Earlier results [58] confirmed VAE-augmented GATs improved accuracy by 6% [59], but failed to scale. GAN methods [53, 60] also depended on costly undersampling. To overcome this limitation, the present research proposes gated GNNs that regulate message flow in WBS-aligned construction graphs. These models improve CPI-based prediction and mitigate oversmoothing.

3.3. Modeling spatio-temporal construction activity information

One essential element of methodology implemented in this study was defining a process of transforming construction progress data into graph-based representations, followed by the application of a GatedGNN models to classify construction CPI. In this study, we modeled the CPI classification task with different GNN architecture, through network representation of data (Eq. (3)). The objective was to classify each activity as efficient (1) or inefficient (0). We constructed the network G to capture the spatial and temporal relationships between activities, where its structure can vividly delineate projects and their relationships with strategic goals. As shown in Eq. (4), G as the input network of the GNN models maps nodes, V, into a representation vector [15], where N is the total number of activities. Each node $x_{vi} \in R^F$, where F stores features of each construction activity, including eight feature columns detailed in Table 1 like quantity, person-hours, WBS group, and delivery week. The edges E connected to pairs of nodes if and only if the corresponding activities shared the same delivery week and belonged to the same WBS group (e.g., mechanical, electrical, insulation), being defined as shown Eq. (5).

$$G = f(V, E) \tag{3}$$

$$V = \{v_1, v_2, \dots, v_N\} \tag{4}$$

$$E = \left\{ (v_i, v_j) \mid \begin{array}{c} Activity\ v_i\ and\ v_j\ share\ the\ same \\ delivery\ week\ and\ WBS\ group \end{array} \right\} \tag{5}$$

This modelling of dataset allowed the network to encapsulate both spatial (WBS group) and temporal (delivery week) dependencies between activities. GNNs aggregated neighboring nodes' representation vectors (e.g., v_i) to process and update information for each node v_j, updating the information related to node v_i [15].

3.4. Development of activity progress prediction model

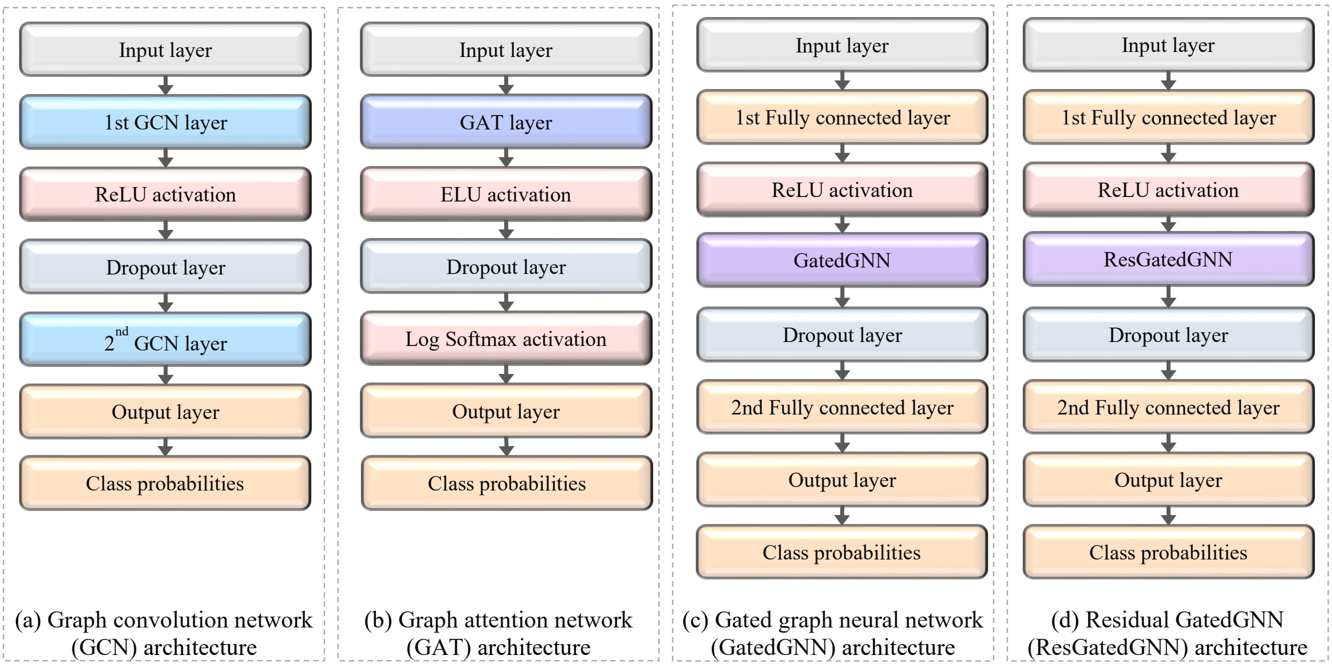
To mitigate vanishing gradients in sequential modeling [15], this study employs ResGatedGNN, combining GRUs [17] residual pathways. Building on early gated approaches in graph learning [70], and foundational GNNs [71], we embed gating mechanisms within GCN and GAT baselines. This allows finer control over information flow, especially in dense, short-span construction networks. Learning behavior across architectures is shown in Fig. 2. Prior successes in text [72] and traffic [73] forecasting further support this approach.

3.5. Convolution mechanisms for aggregating activity information

Fig. 2a illustrates the configured GCN architecture introduced by Kipf and Welling (2016), a seminal model that has gained widespread traction in GNN research [75]. To obtain expressive, high-level representations [76], GCNs apply recursive neighborhood aggregation through graph convolutional layers [77]. By integrating information from both the node itself and its neighbors, GCNs effectively encode local graph topology and feature distributions.

Table 1. Statistical details of the encoded dataset used for training the GNN models

Feature column	Data type	Unique categories	Range	Mode	Mode Freq.
Week	Categorical	52	-	6	10,584
Quantity	Categorical	5,907	-	0	17,883
Weight (%)	Categorical	63	-	0.0	43,684
Planned progress (%)	Numerical	880	[0.0 - 1]	0.0	15,605
Planned unit person-hour	Numerical	167	[0.01 – 3,213.803]	0.26	10,668
Actual unit person-hour	Numerical	1,556	[0 – 14,175.0]	0.0	48,611
WBS group 2	Categorical	8	-	5	100,113
Activity group	Categorical	155	-	36	27,456
CPI class	Categorical	2	-	0	184,392



(a) Graph convolution network (GCN) architecture

(b) Graph attention network (GAT) architecture

(c) Gated graph neural network (GatedGNN) architecture

(d) Residual GatedGNN (ResGatedGNN) architecture

Fig. 2. Architecture of investigated GNN models

Each layer propagates node-held construction progress descriptors and edge-index information, aggregating them to spread context across the graph. In recent years, this framework has been adopted in engineering and construction management applications, addressing problems like quality monitoring [35] safety forecasting [36, 65, 78], building on prior applications in infrastructure sciences [14, 79]. The operation underpinning this aggregation is mathematically formalized in Eq. (6) as in [74].

$$H^{(l+1)} = \sigma\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\,\boldsymbol{W}^{(l)}\right) \tag{6}$$

In Eq. (5), l stands as the layers' output, and is computed by taking the activation matrix from the preceding layer, applying the normalized adjacency matrix $\widetilde{\mathbf{A}}$ of the network, where subsequently multiplied by the current layer's weight matrix, $\mathbf{W}^{(l)}$. In this equation, $\boldsymbol{D}$ represents the degree matrix, while $\boldsymbol{H}^{(l)}$ denotes the activation matrix from the previous layer.

3.6. Attention-driven convolutional modelling of activity information

GAT [38] assigns importance to neighboring nodes using attention mechanisms. Fig. 2b shows the architecture of the utilized GAT model, as described by [38]. The architecture in Fig. 2b begins by processing the input graph: node features are assigned, and connections are defined. The attention layer initializes weights, computes pairwise scores, and normalizes them before feature aggregation. This mechanism is formalized in Eqs. (7, 8).

$$h_i{}' = \sigma\left(\sum_{j\in N_i} \alpha_{ij}\,\boldsymbol{W}\; h_{ij}\right) \tag{7}$$

$$e_{ij} = \sigma\big(a^T\,(\boldsymbol{W}h_i \parallel \boldsymbol{W}h_j)\big) \tag{8}$$

In Eq. (7) σ represents the activation function, while $h_i{}'$ denotes the updated feature representation of node i. The term α_{ij} represents attention coefficients, which are computed for every pair of neighboring nodes i and j. The weight matrix $\mathbf{W}$ is applied to the features h_{ij} of each neighboring node. Additionally, N_i refers to the set of neighbors surrounding node. The attention mechanism, as defined by [17], is implemented as a single-layer neural network that applies a feedforward process. The nonlinearity is introduced by setting σ to the LeakyReLU activation function [80].

The computation of attention coefficients, α_{ij}, involves an intermediate step that is defined in Eq. (8). Here, e_{ij} represents the raw attention score between nodes i and j, updating information in node features h_i and h_j. The operation ‖ denotes the concatenation of the features of nodes i and j, while T represents the transposition operator. The raw attention scores e_{ij} are processed through a learnable attention vector a and further refined using the activation function.

3.7. Gated-enabled mechanism for activity information aggregation

Fig. 2c displays the GatedGNN model configured in this research. The GatedGNN model processes the input graph G to perform the CPI classification task. We first transformed the node features into initial hidden representations using a linear transformation followed by a rectified linear unit (ReLU) [81] activation function. To process the input network information, GatedGNN processed the initial node embedding, $h_v^{(0)}$, computed as shown in Eq. (9). Here, $h_v^{(0)}$is the initial hidden state of node v, $W^{(0)}$ is the weight matrix of the initial linear layer, and $b^{(0)}$ is the corresponding bias vector. The feature vector x_v contains the attributes of node v such as quantity and planned progress.

$$h_v^{(0)} = ReLU\big(W^{(0)}x_v + b^{(0)}\big) \tag{9}$$

The core of the GatedGNN model consists of gated graph convolutional layers, which update node representations by aggregating information from neighboring nodes and applying a gating mechanism via GRU. The gated mechanism in graph neural networks serves as a dynamic control system that modulates the flow of information between nodes in a graph. For each node v and layer l, the update process involves two main steps. This study adjusted the message passing and state updating steps using GRU by [82] by implementing the PyTorch library, as shown in Eqs. (10) and (11).

$$m_v^{(l)} = \sum_{u\,\in\,\mathbb{N}_{(v)}} W^{(l)}\,h_u^{(l-1)} \tag{10}$$

$$h_u^{(l)} = GRU\big(h_u^{(l-1)}, m_v^{(l)}\big) \tag{11}$$

In Eq. (10), $m_v^{(l)}$ represents the aggregated message for node v at layer l, and $\mathbb{N}_{(v)}$ denotes the set of neighboring nodes of v and, $W^{(l)}$ is a learnable weight matrix for layer l. So, $h_u^{(l-1)}$ holds the hidden state of neighbor node u from the previous layer. This aggregation sums the transformed hidden states of neighboring nodes, allowing the model to capture the local structure around each node. At the updating state using the GRU mechanism (Eq. (11)), $h_u^{(l)}$ is the updated hidden state of node v and layer l, and the GRU function manages the flow of information by deciding how much of the previous hidden state $h_u^{(l-1)}$ and the new message $m_v^{(l)}$ to retain. By incorporating gating functions, GNNs can learn the information of the input network using the modern backpropagation [83]. Instead of time steps in a recurrent manner, our study implementation of GatedGNN uses multiple layers to simulate the recurrent updates, thereby controlling the depth of message passing and state updates. This adjustment allowed us to simulate recurrent updates without explicitly unrolling over time steps, as in the original GatedGNN.

In addition, we applied dropout regularization to the hidden states, $h^{(l)}$, after the convolutional layers. The final step involved mapping the hidden representations to output logits using a linear transformation without an activation function, being expressed as Eq. (12).

$$y_u = W^{(l_f)} h_u^{(l)} + b^{(l_f)} \tag{12}$$

Here, y_u denotes the output logits for of u, $W^{(l_f)}$is the weight matrix of the final linear layer, and $b^{(l_f)}$ is the bias vector. Through this layer, the hidden state $h_u^{(l)}$ is transformed into a two-dimensional output corresponding to the efficient and inefficient classes. Considering that Cross-Entropy Loss function in PyTorch expects raw logits and internally applies the softmax function during loss computation, we did not apply an activation function after this layer.

3.8. Residual gated mechanism for aggregating of activity information

The ResGatedGNN model incorporated residual connections into the GatedGNN architecture to address challenges associated with training deeper network architectures, such as the vanishing gradient problem. These residual connections enabled the model to learn deeper representations without losing gradient information across layers, enhancing the stability and depth of learning in graph neural networks. The architecture of utilized ResGatedGNN is given in Fig. 2d. The learning process began with an input layer that encapsulated the essential structural information of the construction activity progress network configured for CPI prediction tasks. The core of this model was the ResGatedGNN layer, where gated graph convolutions were applied along with residual connections. This performed feature propagation by leveraging the gating mechanism to control the flow of information through the network, while the residual connections helped maintain the learning signal across multiple layers. To prevent overfitting, a dropout layer was applied to the output of the ResGatedGNN layer [84]. The transformed features were then fed into a fully connected layer, which performed a linear transformation, mapping the features from the hidden dimension to the output dimension. Finally, the sigmoid activation function was applied to the output of the fully connected layer, scaling the outputs between 0 and 1.

4. Case Study

Fig. 3 displays the structured workflow used to process 77 construction progress reports collected over an 18-month project period. These reports, sourced from Excel and PDF files, varied in format and content. Data extraction focused on summary tables, labor metrics, and schedule information, often found in multi-tab Excel sheets. A standardization step aligned inconsistent headers and merged equivalent fields. Reports with severe data gaps were excluded. Cleaning procedures removed irrelevant rows, and no imputation was applied to incomplete columns to preserve reliability.

Post-cleaning, the dataset emphasized essential predictors like weight (%) and planned progress (%). These, along with labor-hour and classification data, were curated to support reliable CPI forecasting. Table 1 summarizes 267,763 processed project records from 77 progress reports, standardized into 52 delivery-week categories containing task timing, labor utilization, and WBS data. The collected dataset exhibited a significant imbalance in CPI labels, with Efficient (CPI ≥ 1) records vastly outnumbering Inefficient (CPI < 1) records. Of the 267,763 entries, only 83,371 were labeled as Inefficient, compared to 184,392 Efficient records. This imbalance posed a challenge for model training, as it risked biasing predictions toward the majority class.

4.1. Parameter initialization and model training

The training dataset was processed using rolling windows (time frames), segmenting data into training, validation, and test sets, each representing a specific temporal frame of construction activities. These windows, each spanning approximately 21 weeks, account for phase-specific variations in labor allocation, scheduling, and cost performance within a project timeframe. Fig. 4a shows the segmented datasets selected from each timeframe during the projects, while Fig. 4b shows the CPI class distribution of the dataset across project timeline.

To compare GatedGNN, ResGatedGNN, GCN, and GAT fairly, a rigorous hyperparameter tuning was performed. Dropout rates (0.2–0.9), learning rates (0.1, 0.01, 0.001), batch sizes (8–128), and hidden dimensions (8–128) were varied. Model depth was tested at 1–4 layers, and GAT was evaluated using 2–4 attention heads. Gated variants, especially ResGatedGNN, showed sensitivity to tuning—more so than GCN and GAT, which remained stable.

5. Results and Discussion

Fig. 5 shows results across datasets, emphasizing the need for tailored tuning in gated models. Specifically, to identify the optimal configurations for each GNN model across eight dataset segments (Fig. 5a), an initial training phase was conducted with 10 and 100 epochs. Promising setups were then refined with 500 epochs, as extended training consistently improved prediction accuracy. Optimal configurations included a learning rate of 0.1, 0.01, or 0.001 (dataset-dependent), a dropout rate of 0.2, hidden dimensions of 64, and a batch size of 16.

The results, as depicted in Fig. 5b, demonstrate that incorporating gated mechanisms significantly enhances the prediction accuracy and F1 score of GNN models across various datasets. The GatedGNN model consistently outperformed its residual counterpart (ResGatedGNN), GCN, and GAT in both metrics, showcasing the superiority of its architecture in handling the inherent complexities of construction datasets.

To further evaluate this, Fig. 6 illustrates the accuracy and F1 score performance of the proposed GatedGNN, ResGatedGNN, and benchmark models (GCN and GAT) across different datasets.

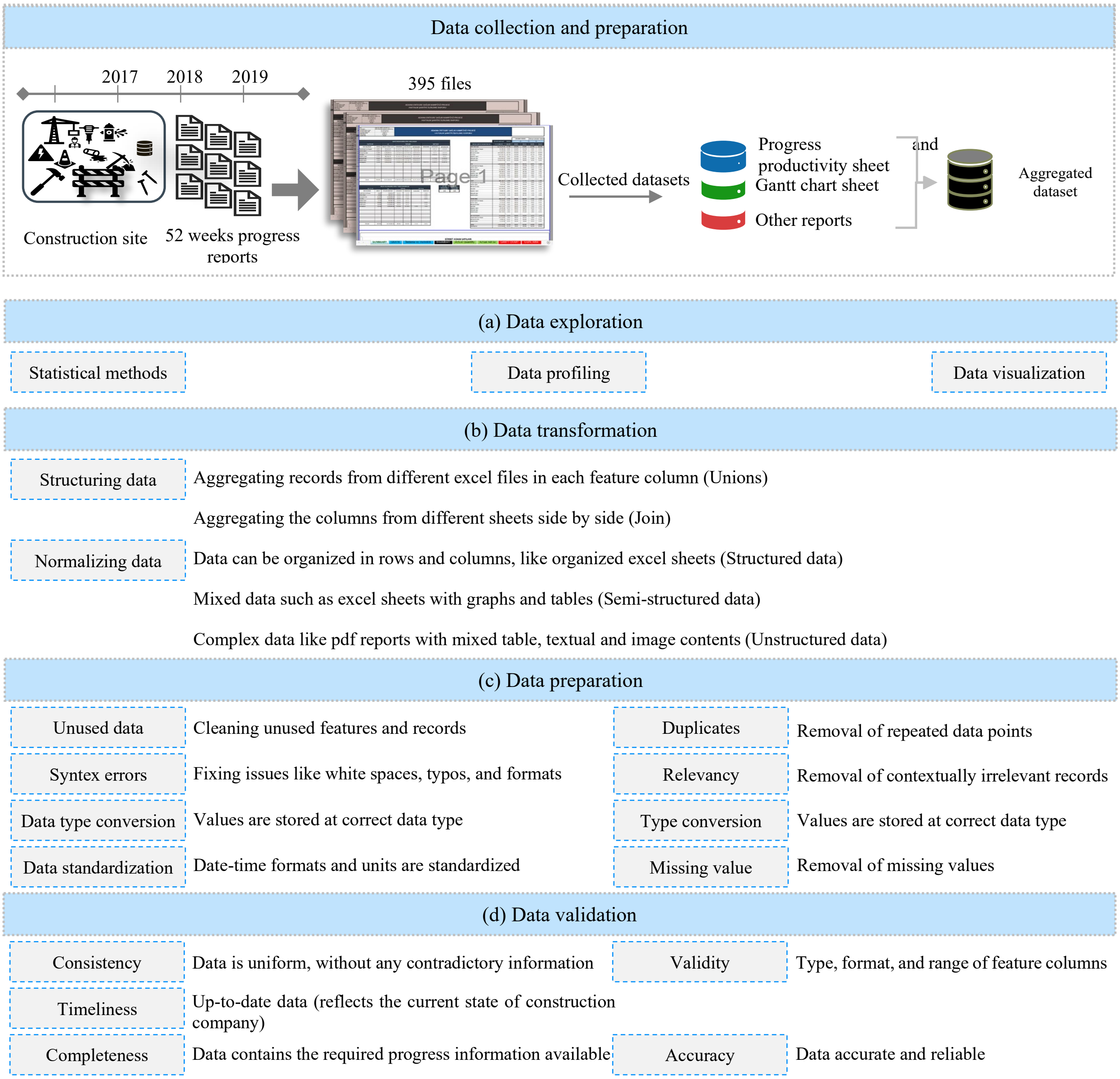


Fig. 3. Outline of undertaken data preparation steps

The GatedGNN achieved accuracy margins of 6% to 14% higher than GCN and GAT models. Notably, while GAT utilizes an attention mechanism, it still performed 10% to 13% lower in accuracy compared to GatedGNN. Similarly, GatedGNN demonstrated 3% to 13% higher accuracy compared to ResGatedGNN across all datasets. As evident from F1 scores, a key advantage of the GatedGNN is its ability to mitigate class imbalance issues, which commonly impact the performance of GNN models. While GCN and GAT experienced a sharp decline in F1 scores relative to accuracy, ranging from a 15% to 30% drop for GCN and 13% to 29% for GAT, where the GatedGNN demonstrated a much smaller deviation. For instance, the ResGatedGNN showed a 7% to 18% drop, whereas the GatedGNN achieved near-consistent F1 scores, indicating its superior ability to handle imbalanced datasets. These results indicate that the proposed GatedGNN mitigates the class imbalance problem commonly observed in GNN models [57]. To further analyze this, Fig. 7 visualizes the class prediction performance of the GatedGNN model compared to GCN and GAT.

The confusion matrices reveal that the GatedGNN achieves more balanced predictions across both majority and minority classes. For instance, in Dataset 8, the GatedGNN correctly classified 95.7% of inefficient activities while achieving a 57.3% prediction rate for efficient activities. In Dataset 3, the model demonstrated its best performance, accurately predicting 95.6% of inefficient activities and 69.9% of efficient activities.

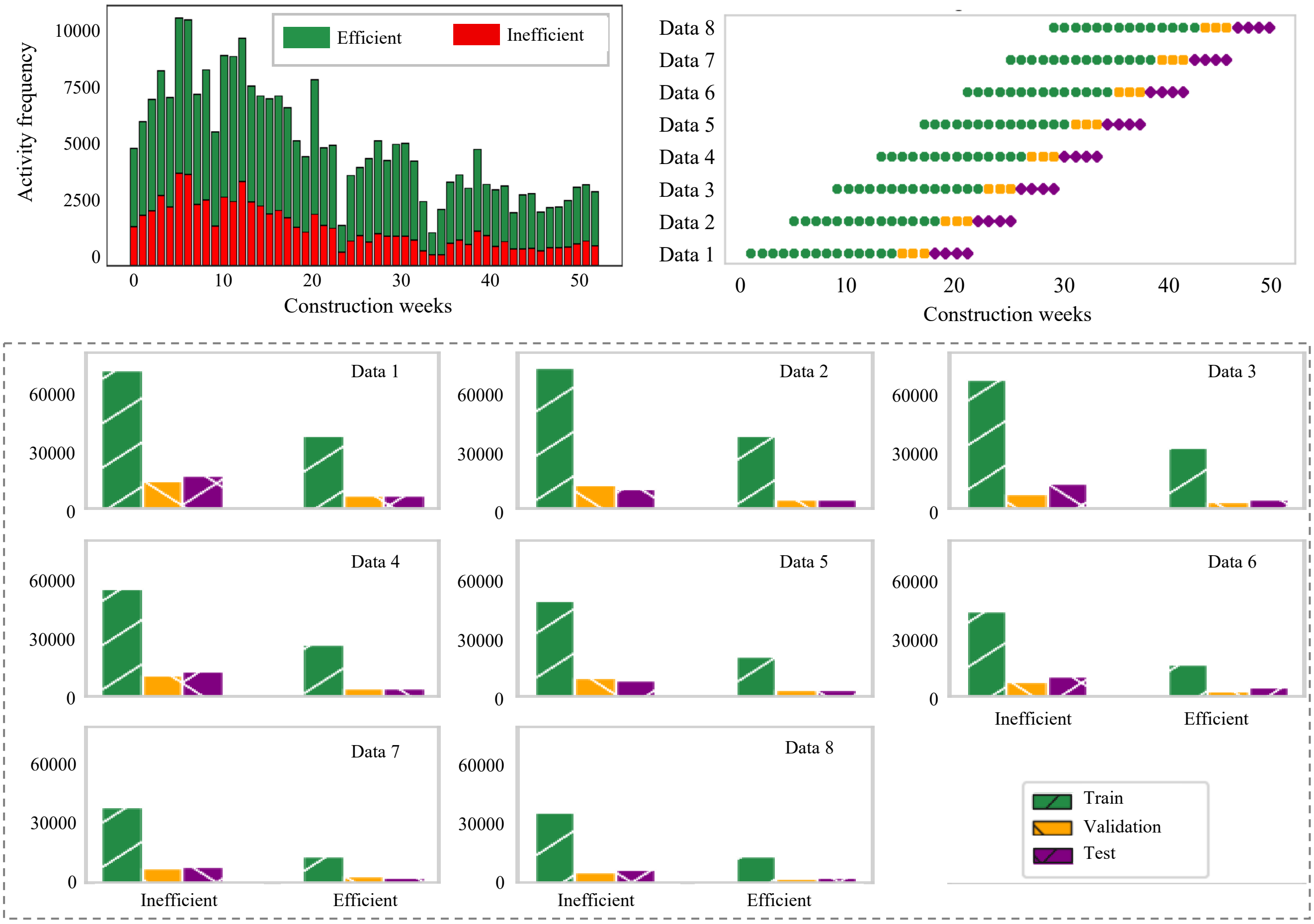
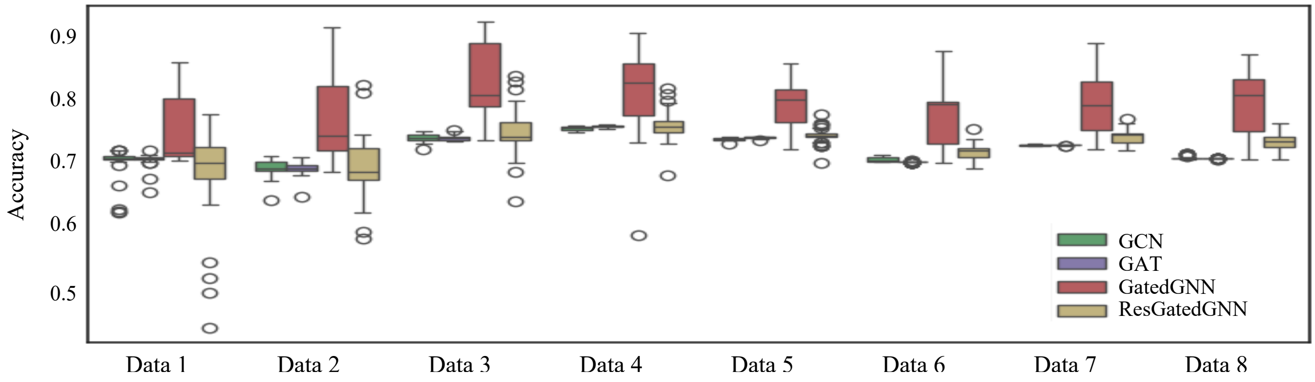


Fig. 4. Data segmentation and distribution of efficient and inefficient classes in training, validation, and test sets

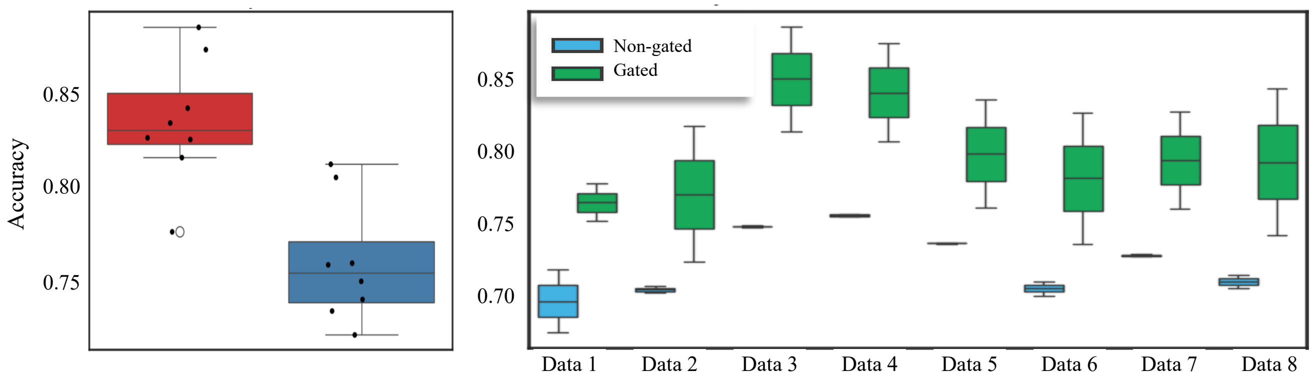
This highlights the model's reliability in identifying inefficient activities, which are critical for optimizing construction progress. The misclassification of efficient activities into the inefficient class can be attributed to the inherent class imbalance within the dataset. From an operational management perspective, predictions categorized as inefficient may be interpreted as probabilistic risk indicators associated with activity IDs, delivery-week schedules, WBS classifications, and confidence probabilities. These indicators support managerial decision-making by enabling prioritized interventions related to labor productivity assessment, resource allocation optimization, procurement constraint analysis, and validation of project progress assumptions. While this leads to a slight over-flagging of efficient activities, it offers a practical advantage by prompting project managers to revisit flagged progress plans, resource allocations, and activity schedules for further validation.

Developed GatedGNN activity progress prediction model consistently outperformed other models across all datasets, particularly in resolving class imbalance without relying on conventional rebalancing strategies. While ResGatedGNN also delivered competitive outcomes, its resilience under varying data distributions was comparatively weaker. Class imbalance, a well-established issue in real-world datasets [85], has prompted multiple interventions: synthetic minority generation ([54, 58], interpolation-based oversampling [55], adaptive weighting [56], and ensemble-based balancing [57]. External balancing loses efficacy in complex data. This study embeds gating into GNNs, internally mitigating class imbalance and overfitting. Table 2 shows strong performance across key metrics, confirming generalizability.

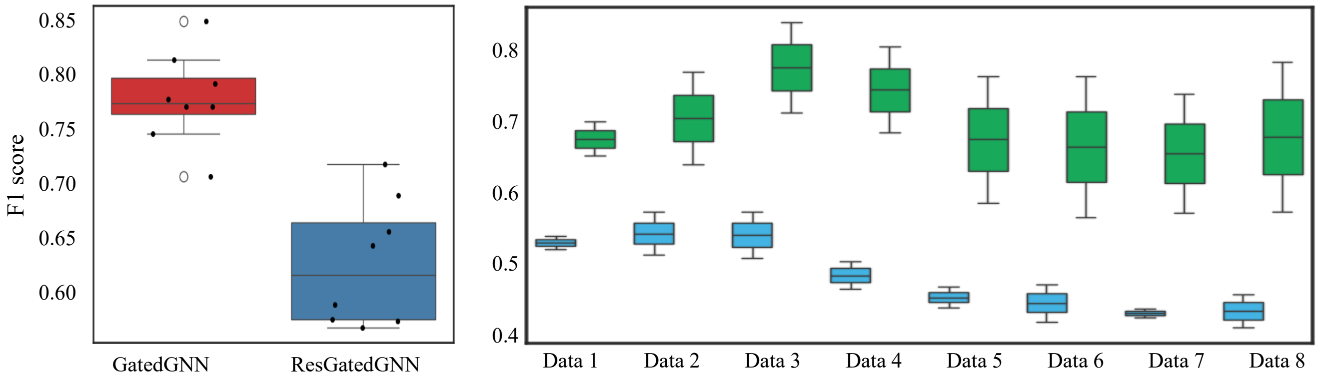
Compared with prior studies that addressed class imbalance through external rebalancing strategies, such as synthetic minority generation, interpolation-based oversampling, adaptive weighting, and ensemble-based balancing, the present study demonstrates that gating can improve minority-class recognition through the model architecture itself. This finding is important because construction progress data are often sparse, noisy, and temporally dependent, making external augmentation or undersampling difficult to apply without distorting project-specific activity relationships. The superior F1 stability of the GatedGNN suggests that controlled message passing can preserve minority-class signals more effectively than conventional GCN and GAT architectures. Therefore, the novelty of this study lies not only in applying GatedGNNs to CPI classification, but also in showing that gated information flow can serve as an internal mechanism for improving prediction reliability in dense, imbalanced construction activity graphs.



(a) Accuracy performance of GNN models across when trained by different parameters.



(b) Accuracy performance of GNN models across when trained by different parameters.



(c) F1 score performance of GNN models across when trained by different parameters.

Fig. 5. Performance comparison of the GNN models with and without gated mechanisms across tested datasets

However, the relatively higher test loss suggests challenges in accurately predicting the underrepresented class, despite the model's robust performance in terms of accuracy and F1 score. The model's ability to effectively address class imbalance represents a significant improvement over baseline GNNs, yet further refinements, such as incorporating advanced rebalancing techniques or hybrid architectures, could further enhance its performance on minority classes. However, its black-box nature limits usability in construction contexts [86], necessitating better interpretability for reliable deployment.

6. Conclusions

The management of engineering projects often requires the accurate classification of activities based on their CPI, yet traditional GNNs struggle to capture the complex dependencies inherent in such datasets. Models such as GCN and GAT, though widely regarded as state-of-the-art, falter when faced with challenges such as class imbalance and high-granularity data. This study set out to address these issues by proposing the GatedGNN and its residual counterpart, the ResGatedGNN. Additionally, extensive validation was conducted using eight datasets derived from a mega project, with results evaluated across key metrics including accuracy, F1 score, and AUC-ROC.

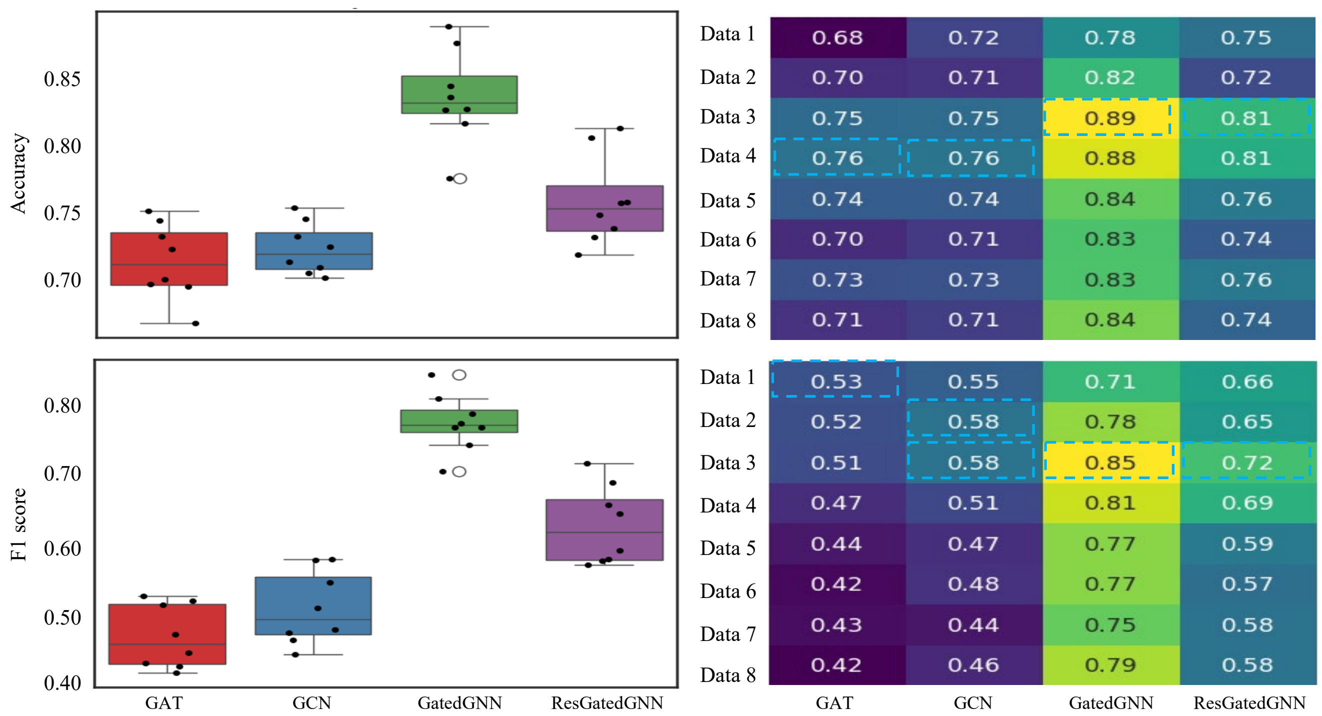
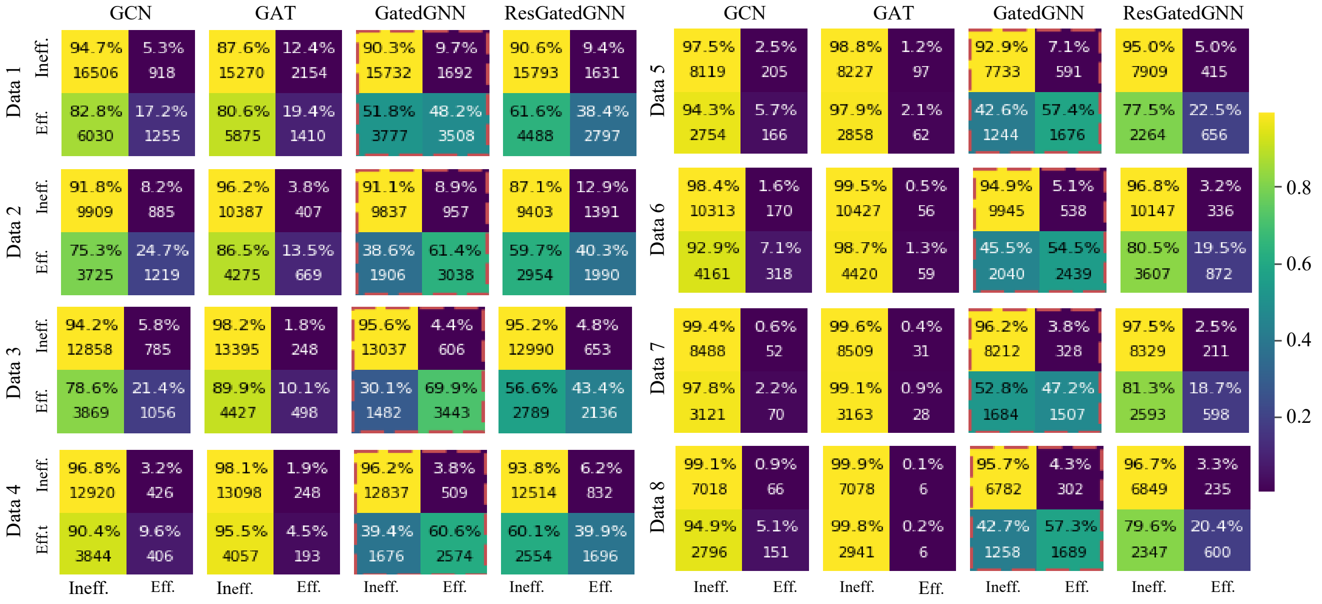


Fig. 6. Accuracy and F1 score performance of the proposed GatedGNN and benchmark GNN models



Ineff.: Inefficient; Eff.: Efficient

Fig. 7. Class prediction performance of GCN, GAT, ResGatedGNN, and GatedGNN across datasets

Table 2. Training, validation, and test performance of the proposed GatedGNN model across datasets

Data	Training time	Train accuracy	Test accuracy	Precision	Recall	F1 score	AUC	Train loss	Test loss	Validation loss	Validation accuracy
Data 1	9.60	0.80	0.78	0.74	0.69	0.71	0.78	0.45	0.50	0.47	0.80
Data 2	9.55	0.85	0.82	0.80	0.76	0.78	0.88	0.37	0.40	0.38	0.84
Data 3	9.32	0.87	0.89	0.87	0.83	0.85	0.92	0.32	0.31	0.30	0.89
Data 4	9.31	0.89	0.88	0.86	0.78	0.81	0.90	0.28	0.32	0.24	0.91
Data 5	9.35	0.91	0.84	0.80	0.75	0.77	0.87	0.23	0.39	0.29	0.88
Data 6	9.37	0.91	0.83	0.82	0.75	0.77	0.89	0.22	0.39	0.34	0.85
Data 7	9.39	0.90	0.83	0.83	0.72	0.75	0.89	0.24	0.37	0.37	0.84
Data 8	9.41	0.88	0.84	0.85	0.77	0.79	0.88	0.28	0.45	0.32	0.85

The findings demonstrate the superiority of the GatedGNN over existing models. Across datasets, the GatedGNN achieved accuracy improvements ranging from 6% to 14% over GCN and GAT, with corresponding F1 score enhancements of 7% to 13%. In Dataset 3, the GatedGNN attained an accuracy of 95.6% for inefficient activities and 69.9% for efficient activities, outperforming GAT, which achieved only 82.4% accuracy and a 51.2% F1 score. Comparatively, the ResGatedGNN displayed 3% to 10% lower accuracy than the GatedGNN, though it still surpassed GCN and GAT, indicating its potential in deeper architectures.

The study underscores that the gated mechanism offers a significant advancement over attention mechanisms, allows selective information propagation and mitigates the sharp drop in F1 score observed in GCN and GAT due to class imbalance. While GCN and GAT experienced drops of 15% to 30% in F1 score compared to their accuracy, the GatedGNN maintained a stable performance with only a 7% to 10% deviation. From a practical standpoint, the GatedGNN provides a robust tool for flagging inefficient activities in construction progress plans, facilitating more informed decision-making for resource allocation and corrective measures. The study's findings suggest that the GatedGNN can be seamlessly integrated into existing engineering management systems to enhance predictive accuracy and address long-standing challenges in handling class-imbalanced data. Future research should explore the scalability of the ResGatedGNN for larger architectures and investigate ways to further improve the interpretability of these models to address the "black box" nature of deep learning in engineering applications.

Declarations

Conflict of Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research received no external funding.

Author Contributions

F. Mostofi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. O. B. Tokdemir: Methodology, Validation, Resources, Data Curation, Writing - Review & Editing, Supervision. V Toğan: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data Curation, Resources, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration.

Acknowledgments

Not applicable.

Data Availability Statement

Some or all of the data and code are available from the corresponding author upon reasonable request.

Ethics Committee Permission

Not applicable.

Use of Generative AI and AI-assisted Technologies

The authors used AI tools for language polishing and have reviewed and take full responsibility for the final content.

References

[1] Ochieng EG, Price ADF, Ruan X, Egbu CO, Moore D (2013) The effect of cross-cultural uncertainty and complexity within multicultural construction teams. Eng Constr Archit Manag 20:307–324. https://doi.org/10.1108/09699981311324023.

[2] dos Santos RB, Jungles AE (2016) Risk level assessment in construction projects using the schedule performance index. J Constr Eng 2016:1–8. https://doi.org/10.1155/2016/5238416.

[3] Feng K, Chen S, Lu W, Wang S, Yang B, Sun C, Wang Y (2023) Embedding ensemble learning into simulation-based optimisation: A learning-based optimisation approach for construction planning. Eng Constr Archit Manag 30:259–295. https://doi.org/10.1108/ECAM-02-2021-0114.

[4] Love PED (2024) A pragmatist research agenda for employing psychological heuristics in construction: Context, design, artificial intelligence, and performance evaluation. IEEE Trans Eng Manag 71:11183–11197. https://doi.org/10.1109/TEM.2024.3411656.

[5] Nurhendi RN, Khoiry MA, Hamzah N (2022) Conceptual framework factors affecting construction labour productivity. Jurnal Kejuruteraan 34:89–99. https://doi.org/10.17576/jkukm-2022-34(1)-08.

[6] Gholami Y (2023) Investigating Adoption of Digital Technologies in Construction Projects. Linköping University Electronic Press, Linköping. https://doi.org/10.3384/9789180750257.

[7] Yan X, Zhang H, Gao H (2022) Mutually coupled detection and tracking of trucks for monitoring construction material arrival delays. Autom Constr 142:104491. https://doi.org/10.1016/j.autcon.2022.104491.

[8] Flyvbjerg B, Ansar A, Budzier A, Buhl S, Cantarelli C, Garbuio M, Glenting C, Holm MS, Lovallo D, Lunn D, Molin E, Rønnest A, Stewart A, van Wee B (2018) Five things you should know about cost overrun. Transp Res Part A Policy Pract 118:174–190. https://doi.org/10.1016/j.tra.2018.07.013.

[9] Guo SJ, Chen JH, Chiu CH (2017) Fuzzy duration forecast model for wind turbine construction project subject to the

impact of wind uncertainty. Autom Constr 81:401–410. https://doi.org/10.1016/j.autcon.2017.03.009.

[10] Biswas A, Ghosh A, Kar A, Mondal T, Ghosh B, Bardhan PK (2021) The impact of COVID-19 in the construction sector and its remedial measures. J Phys Conf Ser 1797:012054. https://doi.org/10.1088/1742-6596/1797/1/012054.

[11] Mou X (2019) Construction Progress Measurement Using Deep Learning. Master Thesis, Texas A&M University.

[12] Project Management Institute (2021) A Guide to the Project Management Body of Knowledge (PMBOK Guide) -- Seventh Edition and The Standard for Project Management, 7th ed. Project Management Institute.

[13] Jia Y, Wang J, Shou W, Hosseini MR, Bai Y (2023) Graph neural networks for construction applications. Autom Constr 154:104984. https://doi.org/10.1016/j.autcon.2023.104984.

[14] Yao L, Mao C, Luo Y (2019) Graph Convolutional Networks for Text Classification. Proc AAAI Conf Artif Intell 33:7370–7377. https://doi.org/10.1609/aaai.v33i01.33017370.

[15] Lukovnikov D, Lehmann J, Fischer A (2020) Improving the Long-Range Performance of Gated Graph Neural Networks.

[16] Liu T, Gu X, Li S, Bai Y, Wang T, Yang X (2023) Static voltage stability margin prediction considering new energy uncertainty based on graph attention networks and long short-term memory networks. IET Renew Power Gener 17:2290–2301. https://doi.org/10.1049/rpg2.12731.

[17] Li Z, Liu H, Zhang C, Fu G (2024) Gated graph neural networks for identifying contamination sources in water distribution systems. J Environ Manag 351:119806. https://doi.org/10.1016/j.jenvman.2023.119806.

[18] Mostofi F, Toğan V, Başağa HB, Çıtıpıtıoğlu A, Tokdemir OB (2023) Multiedge Graph Convolutional Network for House Price Prediction. J Constr Eng Manag 149. https://doi.org/10.1061/JCEMD4.COENG-13559.

[19] Veličković P (2023) Everything is connected: Graph neural networks. Curr Opin Struct Biol 79:102538. https://doi.org/10.1016/j.sbi.2023.102538.

[20] Hamilton WL (2020) Graph Representation Learning Hamilton. Synth Lect Artif Intell Mach Learn 14:1–159. https://doi.org/10.2200/S01045ED1V01Y202009AIM046.

[21] Garcia J, Villavicencio G, Altimiras F, Crawford B, Soto R, Minatogawa V, Franco M, Martínez-Muñoz D, Yepes V (2022) Machine learning techniques applied to construction: A hybrid bibliometric analysis of advances and future directions. Autom Constr 142:104532. https://doi.org/10.1016/j.autcon.2022.104532.

[22] Lin MC, Tserng HP, Ho SP, Young DL (2012) A novel dynamic progress forecasting approach for construction projects. Expert Syst Appl 39:2247–2255. https://doi.org/10.1016/j.eswa.2011.07.093.

[23] Liu D, Dai Q, Tang X, Zhang R, Lu T, Chen J (2025) An Improved Random Forest–Based Operation Duration Prediction of Long-Distance Tunnel Construction Considering Geological Uncertainty. J Comput Civ Eng 39. https://doi.org/10.1061/JCCEE5.CPENG-6041.

[24] Bosché F, Ahmed M, Turkan Y, Haas CT, Haas R (2015) The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. Autom Constr 49:201–213. https://doi.org/10.1016/j.autcon.2014.05.014.

[25] Liang H, Yeoh JKW, Chua DKH (2024) Premapping of Dynamic Indoor Construction Sites Using Prior Construction Progress Video to Enhance UGV Path Planning. J Comput Civ Eng 38. https://doi.org/10.1061/JCCEE5.CPENG-5819.

[26] Hu Q, Lin W, Tang M, Jiang J (2022) MBHAN: Motif-Based Heterogeneous Graph Attention Network. Appl Sci 12:5931. https://doi.org/10.3390/app12125931.

[27] Zhang Y (2020) Graph Deep Learning Models for Network-based Spatio-Temporal Data Forecasting: From Dense to Sparse. PhD Thesis, University College London.

[28] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: A review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001.

[29] Corso G, Stark H, Jegelka S, Jaakkola T, Barzilay R (2024) Graph neural networks. Nat Rev Methods Primers 4:17. https://doi.org/10.1038/s43586-024-00294-7.

[30] Avci HF (2024) The Bid/no-bid Framework Utilizing Node2vec for International Contractors. Master Thesis, Istanbul Technical University.

[31] Pan Y, Fu X, Zhang L (2022) Data-driven multi-output prediction for TBM performance during tunnel excavation: An attention-based graph convolutional network approach. Autom Constr 141:104386. https://doi.org/10.1016/j.autcon.2022.104386.

[32] Mostofi F, Bahadır Ü, Tokdemir OB, Toğan V, Yepes V (2025) Enhancing strategic investment in construction engineering projects: A novel graph attention network decision-support model. Comput Ind Eng 203:111033. https://doi.org/10.1016/j.cie.2025.111033.

[33] Khemani B, Patil S, Kotecha K, Tanwar S (2024) A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. J Big Data 11:18. https://doi.org/10.1186/s40537-023-00876-4.

[34] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021) A Comprehensive Survey on Graph Neural Networks. IEEE Trans Neural Netw Learn Syst 32:4–24. https://doi.org/10.1109/TNNLS.2020.2978386.

[35] Mostofi F, Tokdemir OB, Toğan V, Arditi D (2024) Predicting the Cost of Rework in High-Rise Buildings Using Graph Convolutional Networks. J Constr Eng Manag 150:1–17. https://doi.org/10.1061/JCEMD4.COENG-14739.

[36] Mostofi F, Toğan V (2024) Predicting Construction Accident Outcomes Using Graph Convolutional and Dual-Edge Safety Networks. Arab J Sci Eng 49:13315–13332. https://doi.org/10.1007/s13369-023-08609-8.

[37] Jing W, Song X, Di D, Song H (2021) geoGAT: Graph Model Based on Attention Mechanism for Geographic Text Classification. ACM Trans Asian Low-Resour Lang Inf Process 20:1–18. https://doi.org/10.1145/3434239.

[38] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph Attention Networks. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings 39–41. https://doi.org/arXiv:1710.10903.

[39] Mostofi F, Toğan V (2023) Construction safety predictions with multi-head attention graph and sparse accident networks. Autom Constr 156. https://doi.org/10.1016/j.autcon.2023.105102.

[40] Ruiz L, Gama F, Ribeiro A (2020) Gated Graph Recurrent Neural Networks. IEEE Trans Signal Process 68:6303–6318. https://doi.org/10.1109/TSP.2020.3033962.

[41] Chen T (2023) Prediction of heart disease using graph neural networks. In: Xiong Z, He R (eds) International Conference on Artificial Intelligence and Industrial Design (AIID 2022). SPIE, p. 42. https://doi.org/10.1117/12.2673160.

[42] Wang C, Liu H, An X, Gong Z, Deng F (2025) DCNCrack: Pavement Crack Segmentation Based on Large-Scaled Deformable Convolutional Network. J Comput Civ Eng 39. https://doi.org/10.1061/JCCEE5.CPENG-5926.

[43] Wang L, Cao H, Xu H, Liu H (2022) A gated graph convolutional network with multi-sensor signals for remaining useful life prediction. Knowl Based Syst 252:109340. https://doi.org/10.1016/j.knosys.2022.109340.

[44] Yang B, Kang Y, Zhang L, Li H (2021) GGAC: Multi-relational image gated GCN with attention convolutional binary neural tree for identifying disease with chest X-rays. Pattern Recognit 120:108113. https://doi.org/10.1016/j.patcog.2021.108113.

[45] Khan AA, Chaudhari O, Chandra R (2024) A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. Expert Syst Appl 244:122778. https://doi.org/10.1016/j.eswa.2023.122778.

[46] Lin WC, Tsai CF, Hu YH, Jhang JS (2017) Clustering-based undersampling in class-imbalanced data. Inf Sci 409–410:17–26. https://doi.org/10.1016/j.ins.2017.05.008.

[47] Liu XY, Wu J, Zhou ZH (2009) Exploratory Undersampling for Class-Imbalance Learning. IEEE Trans Syst Man Cybern B Cybern 39:539–550. https://doi.org/10.1109/TSMCB.2008.2007853.

[48] de Oliveira WDG, Berton L (2023) A systematic review for class-imbalance in semi-supervised learning. Artif Intell Rev 56:2349–2382. https://doi.org/10.1007/s10462-023-10579-0.

[49] de Giorgio A, Cola G, Wang L (2023) Systematic review of class imbalance problems in manufacturing. J Manuf Syst 71:620–644. https://doi.org/10.1016/j.jmsy.2023.10.014.

[50] Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G (2017) Learning from class-imbalanced data: Review of methods and applications. Expert Syst Appl 73:220–239. https://doi.org/10.1016/j.eswa.2016.12.035.

[51] Rezvani S, Wang X (2023) A broad review on class imbalance learning techniques. Appl Soft Comput 143:110415. https://doi.org/10.1016/j.asoc.2023.110415.

[52] Leevy JL, Khoshgoftaar TM, Bauder RA, Seliya N (2018) A survey on addressing high-class imbalance in big data. J Big Data 5:42. https://doi.org/10.1186/s40537-018-0151-6.

[53] Saini M, Susan S (2023) Tackling class imbalance in computer vision: a contemporary review. Artif Intell Rev 56:1279–1335. https://doi.org/10.1007/s10462-023-10557-6.

[54] Li WZ, Wang CD, Xiong H, Lai JH (2023) GraphSHA: Synthesizing Harder Samples for Class-Imbalanced Node Classification.

[55] Sun Y, Wang Y, Wang S (2024) Nia-GNNs: neighbor-imbalanced aware graph neural networks for imbalanced node classification. Appl Intell 54:7941–7957. https://doi.org/10.1007/s10489-024-05590-z.

[56] Mohammadizadeh M, Mozhdehi A, Ioannou Y, Wang X (2024) Meta-GCN: A Dynamically Weighted Loss Minimization Method for Dealing with the Data Imbalance in Graph Neural Networks. https://doi.org/10.21428/594757DB.0041F830.

[57] Fofanah AJ, Chen D, Wen L, Zhang S (2024) Addressing imbalance in graph datasets: Introducing GATE-GNN with graph ensemble weight attention and transfer learning for enhanced node classification. Expert Syst Appl 255:124602. https://doi.org/10.1016/j.eswa.2024.124602.

[58] Mostofi F, Behzat Tokdemir O, Toğan V (2024) Generating synthetic data with variational autoencoder to address class imbalance of graph attention network prediction model for construction management. Adv Eng Inform 62:102606. https://doi.org/10.1016/j.aei.2024.102606.

[59] Ping G, Chen J, Pan T, Pan J (2019) Degradation feature extraction using multi-source monitoring data via logarithmic normal distribution based variational auto-encoder. Comput Ind 109:72–82. https://doi.org/10.1016/j.compind.2019.04.013.

[60] Lu J, Pan B, Ren W, Liu Q, Liu P, Oeser M (2023) Regeneration of pavement surface textures using M-sigmoid-normalized generative adversarial networks. Comput-Aided Civ Infrastruct Eng 38:2225–2241. https://doi.org/10.1111/mice.12987.

[61] Liu J, Luo H, Liu H (2022) Deep learning-based data analytics for safety in construction. Autom Constr 140:104302. https://doi.org/10.1016/j.autcon.2022.104302.

[62] Gao Y, Chen R, Qin W, Wei L, Zhou C (2023) Learning from explainable data-driven tunneling graphs: A spatio-temporal graph convolutional network for clogging detection. Autom Constr 147:104741. https://doi.org/10.1016/J.AUTCON.2023.104741.

[63] Yan X, Ai T, Yang M, Yin H (2019) A graph convolutional neural network for classification of building patterns using spatial vector data. ISPRS J Photogramm Remote Sens 150:259–273. https://doi.org/10.1016/j.isprsjprs.2019.02.010.

[64] Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-Based Recommendation with Graph Neural Networks. Proc AAAI Conf Artif Intell 33:346–353. https://doi.org/10.1609/aaai.v33i01.3301346.

[65] Pan X, Zhong B, Wang Y, Shen L (2022) Identification of accident-injury type and bodypart factors from construction accident reports: A graph-based deep learning framework. Adv Eng Inform 54. https://doi.org/10.1016/J.AEI.2022.101752.

[66] Zhang X, Zhang X, Liu J, Wu B, Hu Y (2023) Graph features dynamic fusion learning driven by multi-head attention for large rotating machinery fault diagnosis with multi-sensor data. Eng Appl Artif Intell 125:106601. https://doi.org/10.1016/j.engappai.2023.106601.

[67] Wang Y, Zhao Y, Shah N, Derr T (2021) Imbalanced Graph Classification via Graph-of-Graph Neural Networks.

[68] Devi D, Biswas SK, Purkayastha B (2020) A Review on Solution to Class Imbalance Problem: Undersampling Approaches. In: Proceedings of International Conference on Computational Performance Evaluation (ComPE). IEEE, pp. 626–631. https://doi.org/10.1109/ComPE49325.2020.9200087.

[69] Bi J, Zhang C (2018) An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowl Based Syst 158:81–93. https://doi.org/10.1016/j.knosys.2018.05.037.

[70] Czakon M, Heymes D, Mitov A (2015) High-precision differential predictions for top-quark pairs at the LHC. https://doi.org/10.1103/PhysRevLett.116.082003.

[71] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The Graph Neural Network Model. IEEE Trans Neural Netw 20:61–80. https://doi.org/10.1109/TNN.2008.2005605.

[72] Liang Z, Du J, Shao Y, Ji H (2021) Gated Graph Neural Attention Networks for abstractive summarization. Neurocomputing 431:128–136. https://doi.org/10.1016/j.neucom.2020.09.066.

[73] Geng Z, Xu J, Wu R, Zhao C, Wang J, Li Y, Zhang C (2024) STGAFormer: Spatial–temporal Gated Attention Transformer based Graph Neural Network for traffic flow forecasting. Inf Fusion 105:102228. https://doi.org/10.1016/j.inffus.2024.102228.

[74] Kipf TN, Welling M (2017) Semi-Supervised Classification with Graph Convolutional Networks. ArXiv Preprint 1–14. https://doi.org/10.48550/arXiv.1609.02907.

[75] Cui L, Bai L, Bai X, Wang Y, Hancock ER (2024) Learning Aligned Vertex Convolutional Networks for Graph Classification. IEEE Trans Neural Netw Learn Syst 35:4423–4437. https://doi.org/10.1109/TNNLS.2021.3129649.

[76] Özcan A (2022) Applying Graph Convolution Networks to Recommender Systems based on graph topology. DÜMF Mühendislik Dergisi. https://doi.org/10.24012/dumf.1081137.

[77] Chen H, Deng Z, Xu Y, Li Z (2021) Non-Recursive Graph Convolutional Networks. http://arxiv.org/abs/2105.03868.

[78] Tian D, Li M, Han S, Shen Y (2022) A Novel and Intelligent Safety-Hazard Classification Method with Syntactic and Semantic Features for Large-Scale Construction Projects. J Constr Eng Manag 148. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002382.

[79] Ogoke F, Meidani K, Hashemi A, Farimani AB (2020) Graph Convolutional Neural Networks for Body Force Prediction. Mach Learn Sci Technol 2. https://doi.org/10.1088/2632-2153/ac1fc9.

[80] Xu B, Wang N, Chen T, Li M (2015) Empirical Evaluation of Rectified Activations in Convolutional Network. http://arxiv.org/abs/1505.00853.

[81] Agarap AF (2018) Deep Learning using Rectified Linear Units (ReLU). ArXiv Preprint. https://doi.org/10.48550/arXiv.1803.08375.

[82] Li Y, Zemel R, Brockschmidt M, Tarlow D (2016) Gated graph sequence neural networks. 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings. https://doi.org/10.48550/arxiv.1511.05493.

[83] Beck D, Haffari G, Cohn T (2018) Graph-to-Sequence Learning using Gated Graph Neural Networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 273–283. https://doi.org/10.18653/v1/P18-1026.

[84] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958. https://doi.org/10.5555/2627435.2670313.

[85] Yen SJ, Lee YS (2006) Under-Sampling Approaches for Improving Prediction of the Minority Class in an Imbalanced Dataset. In: Intelligent Control and Automation. Springer Berlin Heidelberg, pp. 731–740. https://doi.org/10.1007/978-3-540-37256-1_89.

[86] Love PED (2025) Smart Heuristics in Civil Engineering: Unearthing the Unexploited. ASCE ASME J Risk Uncertain Eng Syst A Civ Eng 11:1–4. https://doi.org/10.1061/AJRUA6.RUENG-1440.