Parallel Encoding Method for Critical Node Identification in Transportation Networks

Xiaoxin Li, Ming Xu, Qiang Ai†, *Member, IAENG,* Wenhui Cao

***Abstract*—To address the challenge of evaluating critical nodes in transportation networks, we propose a parallel encoding method for accurately encoding nodes. This method involves parallel processing of transportation network graph data, utilizing a Dynamic Adaptive Attention Network (DAAN) and a Static Feature Embedding method (SFE) to generate dynamic and static feature embeddings for each node. The DAAN module focuses on encoding dynamic features such as traffic flow and average speed, while the SFE module primarily handles static features of road segments, including length, lane count, and speed limits. These two embeddings are then concatenated and input into a Multilayer Perceptron (MLP) to generate the final encoding for each road segment node. Using these embeddings, the system performs node classification in the classification module to determine whether a node is a critical node. Experiments were conducted based on a simulation dataset, which was divided into training and validation sets, with multiple control experiments carried out according to the ratio of critical nodes in the training set. The experimental results show that our method significantly outperforms other methods in various metrics, highlighting its practicality and effectiveness in complex transportation network structures.**

***Index Terms*—Transportation Network, Critical Node Identification, Parallel Encoding, DAAN, Static Feature Embedding, MLP**

I. INTRODUCTION

As urbanization accelerates, the complexity of urban transportation systems has been increasing, posing significant challenges for transportation decision-making bodies in effectively managing and optimizing city road networks. Advances in science and technology have provided strong support in addressing these challenges, particularly in the fields of traffic simulation and traffic flow prediction, where these technologies have become critical auxiliary tools [1]. In this context, the ranking of the criticality of the nodes of the road network plays a crucial role in ensuring smooth traffic flow and effective emergency response. This technology not only impacts the rational distribution of traffic flow but also directly relates to urban emergency management and the optimal allocation of resources [2].

Manuscript received September 10, 2024; revised December 26, 2024.

This work was supported in part by the 2024 Postgraduate Innovation and Entrepreneurship Project of Liaoning Technical University 2024-38.

Xiaoxin Li is a Postgraduate Student at the Software College, Liaoning Technical University, Huludao 125105, China (phone: +8613358819070; e-mail: lixiaoxin1999@aliyun.com).

Ming Xu is an Associate Professor at the Software College, Liaoning Technical University, Huludao 125105, China (phone: +8615810261581; e-mail: xum.2016@tsinghua.org.cn).

Qiang Ai is a Postgraduate Student at the College of Computer, Qinghai Normal University, Xining 810008, China (corresponding author to provide phone: +8617813140425; e-mail: qiang.ai@outlook.com).

Wenhui Cao is a Postgraduate Student at the Software College, Liaoning Technical University, Huludao 125105, China (e-mail: wenhui.cao@aliyun.com).

The concept of road segment criticality originates from research on network vulnerability. Taylor et al. [3] proposed that critical road segments should possess two characteristics: first, a vulnerability indicating a higher likelihood of failure, and second, the degree to which their failure would significantly impact the entire network. Currently, the vulnerability of critical road segments is typically determined by assessing the probability of their failure, while the determination of a segment's importance is more complex and is usually based on the impact of the segment on the network's traffic before and after its failure. Finally, by combining the results of vulnerability and importance assessments, the criticality of the road segment is determined [4].

However, traditional network analysis methods, while capable of initially identifying critical nodes, often overlook the impact of dynamic changes in the road network and the complex relationships between nodes, thereby limiting their effectiveness in practical applications. With the advancement of technology, researchers have gradually introduced graph embedding representations and Graph Neural Networks (GNNs) [5] into tasks related to transportation networks to enhance the accuracy and practicality of the analysis.

Khoshrafta et al. [6] pointed out that although traditional graph embedding techniques effectively map graph data into low-dimensional spaces and reveal their global structural characteristics, these methods often face challenges when generalizing to unseen nodes. Since they rely on the global information of the graph, the addition of new nodes typically requires recalculating the mappings, which is particularly inconvenient in dynamically changing networks. In contrast, Graph Neural Networks (GNNs) exhibit significant advantages in generalizing to new nodes [7]. GNNs learn to aggregate information from the neighborhoods of nodes, allowing for direct feature inference of newly added nodes without the need to retrain the entire model [8]. Consequently, GNNs demonstrate high flexibility and adaptability in dynamic network scenarios, such as real-time traffic flow prediction and network expansion [9].

To achieve an understanding of the dynamic evolution characteristics of road network nodes, this paper proposes a novel network model named Dynamic Adaptive Attention Network (DAAN) to capture the semantic information of the dynamic features of road network nodes. DAAN leverages the advantages of the attention mechanism [10], enabling dynamic adjustment of node representations over the time dimension to better reflect their states and roles at different time steps. Additionally, to fully utilize the static features of road segments in the network, we propose a Road Segment Static Feature Embedding (RSE) method, which encodes the static features of nodes into embedding representations to

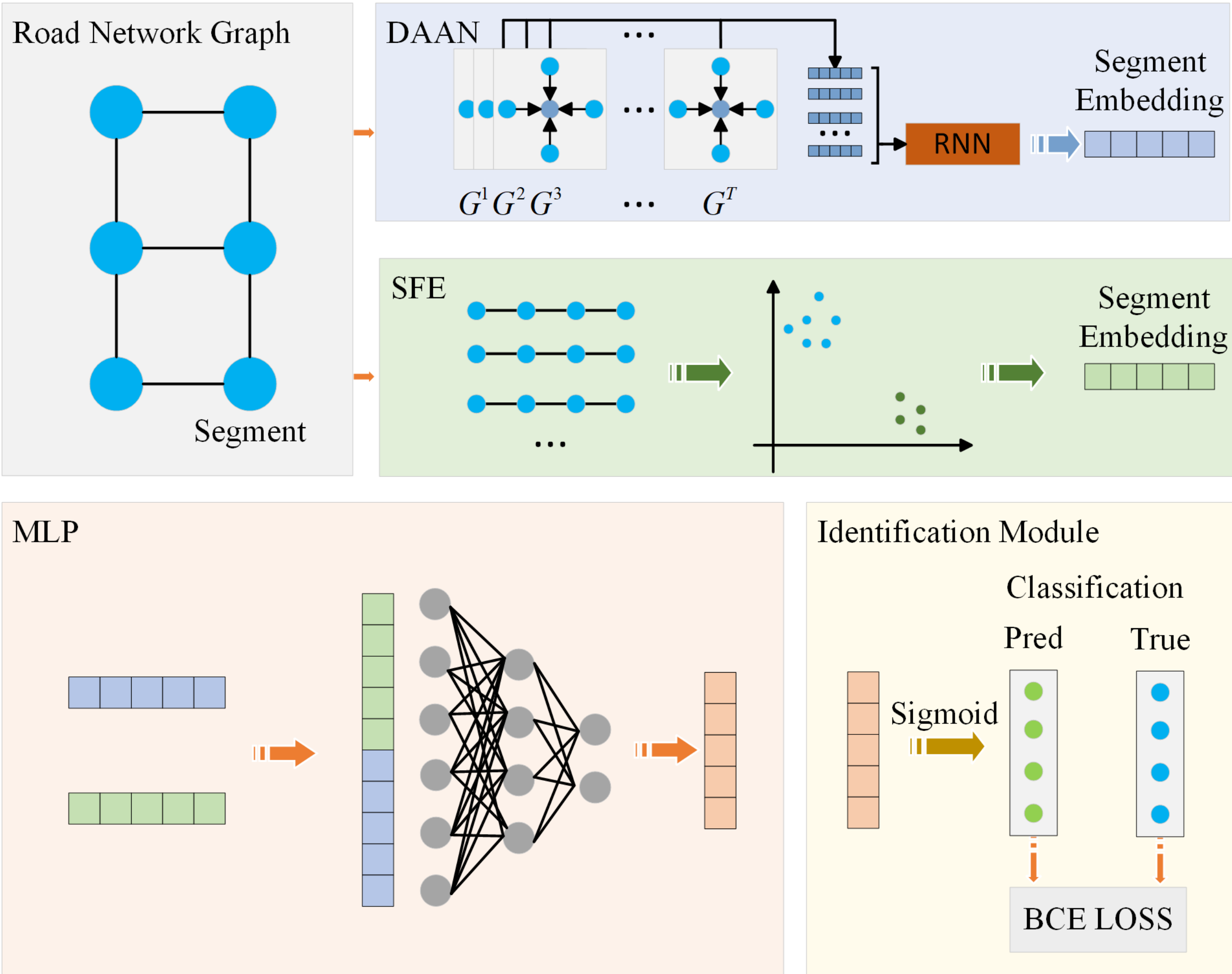

Fig. 1: Network Structure

capture the intrinsic structural information of the network. Finally, the two sets of features are aggregated through a feature fusion module and then fed into a ranking learning module to learn the importance ranking among road segments.

In summary, this study combines traditional graph embedding techniques with the feature vector encoding methods of Graph Neural Networks (GNNs) to propose a novel model for ranking the criticality of road network nodes. The model's high accuracy and adaptability in practical applications have been validated through supervised training on a simulation dataset [11] generated by the SUMO software [12].

II. METHODOLOGY

Our method encodes road network graph data in parallel by first feeding the graph data into two independent modules. As Fig. 1 shows. One module utilizes a Dynamic Adaptive Attention Network (DAAN) to generate node encodings based on neural networks, with DAAN primarily encoding the dynamic features of nodes, such as traffic flow and average speed. The other module employs a traditional embedding method-based static feature embedding (SFE) to generate graph structure-based embeddings, with RSE primarily encoding static features, such as road segment length, lane count, and speed limits. These two encodings are then concatenated and fed into a Multilayer Perceptron (MLP) to produce the final encoding for each road segment node. Using these encodings, we perform pairwise comparisons of all road segment nodes in the ranking module to generate a ranking list of node importance, which can be used to assess critical nodes within the road network.

A. *Dynamic Adaptive Attention Network (DAAN)*

This paper proposes a novel network embedding model named DAAN, specifically designed to capture the dynamic features of road network nodes. The model effectively generates the final representation for each road segment through an adaptive attention mechanism and a recursive aggregation strategy.

Given a sequence of dynamic road network graphs $\{G^1, G^2, \ldots, G^T\}$, where $G^t = (V, E^t)$ represents the road network graph at time step t, V is the set of nodes, and E^t is the set of edges present at time t. The initial representation of each node $v \in V$ at time step t is denoted as $\mathbf{x}_v^t$, which includes all the dynamic features of the node.

1) Structural Adaptive Layer: At each time step t, the objective of the Structural Adaptive Layer is to compute the representation of node v in the current graph G^t by aggregating information from its neighboring nodes. The computation in this layer is defined by the following equation:

$$\mathbf{h}_v^t = \sigma\left(\sum_{u \in N(v)} \alpha_{vu}^t \mathbf{W}_s \mathbf{x}_u^t\right) \quad (1)$$

where $\mathbf{h}_v^t$ is the structural representation of node v at time step t, and $\mathbf{W}_s$ is the weight matrix of the Structural Adaptive Layer. The attention weight α_{vu}^t of node v towards its neighbor u is calculated as follows:

$$e_{vu}^t = \exp\Big(\text{LeakyReLU}\big(\mathbf{a}^T\left[\mathbf{W}_s\mathbf{x}_v^t | \mathbf{W}_s\mathbf{x}_u^t\right]\big)\Big) \quad (2)$$

$$\alpha_{vu}^t = \frac{e_{vu}^t}{\sum\limits_{k \in N(v)} e_{vk}^t} \quad (3)$$

2) Temporal Adaptive Layer: The Temporal Adaptive Layer is designed to capture the dynamic evolution features of nodes across different time steps. The input to this layer is the sequence of structural representations of nodes at various time steps. The computation in the Temporal Adaptive Layer is defined by the following equation:

$$\mathbf{z}_v^t = \sum_{k=1}^{t} \beta_{vk}^t \mathbf{W}_t \mathbf{h}_v^k \quad (4)$$

where $\mathbf{z}_v^t$ is the temporal representation of node v at time step t, and $\mathbf{W}_t$ is the weight matrix of the Temporal Adaptive Layer. The attention weight β_{vk}^t of node v at time step t towards its historical time step k is calculated as follows:

$$e_{vk}^t = \exp\left(\frac{\mathbf{W}q\mathbf{h}v^t \cdot \mathbf{W}k\mathbf{h}v^k}{\sqrt{d}}\right) \quad (5)$$

$$\beta_{vk}^t = \frac{e_{vk}^t}{\sum\limits_{j=1}^{t} e_{vj}^t} \quad (6)$$

3) Recurrent Aggregation: To generate the final representation of the road segments, we employ a recurrent aggregation strategy. By using a Recurrent Neural Network (RNN), the temporal representations $\{\mathbf{z}_v^1, \mathbf{z}_v^2, \ldots, \mathbf{z}_v^T\}$ are processed sequentially to obtain the final representation of the node $\mathbf{z}_v$:

$$\mathbf{z}_v = \text{RNN}(\mathbf{z}_v^1, \mathbf{z}_v^2, \ldots, \mathbf{z}_v^T) \quad (7)$$

During the recurrent aggregation process, the RNN captures dependencies between time steps through its inherent memory mechanism, thereby generating a final representation that integrates information from all time steps.

The final output of the DAAN module is the embedding representation $\mathbf{z}_v$ of node v, which integrates both the local structural information and the temporal evolution features of the node.

B. Static Feature Embedding (SFE)

Given a road network graph $G = (V, E)$, where V is the set of nodes representing road segments, and E is the set of edges representing intersections. The node static feature matrix is denoted as $T \in \mathbb{R}^{n \times s}$, where n is the number of nodes, and s is the dimension of the static features for each node.

1) Node Similarity: To measure the similarity between node pairs in the embedding space, this paper defines a similarity metric function $F(i, j)$ based on the Sigmoid function, which is expressed as follows:

$$F(i,j) = 2\sigma(\mathbf{e}_i\mathbf{e}_j^T) - 1 = 2 \times \left(\frac{1}{1 + e^{-\mathbf{e}_i\mathbf{e}_j^T}}\right) - 1 \quad (8)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the static feature embedding vectors of nodes i and j, respectively.

2) Static Feature Embedding Matrix: To maximize the similarity between nodes with similar static features, the embedding matrix U is generated using the Skip-gram with Negative Sampling (SGNS) technique [13], and can be expressed as follows:

$$U_i \; = \; f\,(T,\; G) \quad (9)$$

where U_i is the embedding representation vector of node i. The goal is to obtain the embedding matrix U by optimizing the following equation, which reflects the intrinsic relationships between nodes:

$$\min_{U \geq 0} L(U) = \|M - UU^T\|_F^2 \quad (10)$$

where M is the node structural embedding matrix, which captures the latent relationships between nodes, and U is the desired node embedding matrix.

The final output of the module is the embedding representation matrix $U \in \mathbb{R}^{n \times d}$, where d is the embedding dimension, and each row represents the final embedding of a node in the embedding space.

C. Vector Fusion Module

This paper employs a MLP for the fusion of embedding vectors and neural network encoded vectors. In the MLP fusion strategy, the embedding vectors and neural network encoded vectors are concatenated and then fed into a MLP to generate the fused representation vector $\mathbf{h}$. The formula is as follows:

$$h_i = \text{MLP}([\mathbf{u_i} \| \mathbf{z_i}]) \quad (11)$$

where is a neural network composed of multiple fully connected layers, used to non-linearly combine the input embedding vectors and encoded vectors.

D. Discriminator and Loss Function

In this method, the discriminator is used to identify important categories. After processing node features through a MLP, a Sigmoid activation function is applied to determine the probability that a node belongs to a specific category.

Specifically, let the input features be $\mathbf{h_i}$, which is passed through the Sigmoid activation function to determine whether the node belongs to an important category. The Sigmoid function is defined as:

$$\sigma(h_i) = \frac{1}{1 + e^{-h_i}} \quad (12)$$

The output $\sigma(h_i)$ ranges between $(0, 1)$, representing the probability that the node belongs to an important category.

The discriminator's output is compared with the node labels in the data, using binary cross-entropy loss as the evaluation metric. By minimizing the loss, the model's predictive capability is optimized. The binary cross-entropy loss is defined as follows:

$$\mathrm{BCE}(x,y) = -\frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} y_i \log(\sigma(x_i)) \\ + (1-y_i)\log(1-\sigma(x_i)) \end{pmatrix} \quad (13)$$

where N represents the number of samples, x_i is the model output, σ is the Sigmoid function, and y_i is the true label.

III. VALIDATION AND RESULTS ANALYSIS

A. Datasets

The dataset used in this study [11] is based on the real road network of Tiexi District, Shenyang, China, and was constructed using the traffic simulator SUMO. This road network includes 1,004 road segments and 377 intersections. To simulate traffic distribution, the dataset integrates origin-destination (OD) information and points of interest (POI), and models traffic flow during peak hours. Road segments near POIs, such as subway stations, schools, and shopping centers, were selected as origins or destinations. A total of 7,200 vehicles were introduced into the simulated road network, modeling traffic flow over a one-hour period. Road segments without trajectory data were excluded from the dataset.

To evaluate the importance of road segment nodes, the dataset simulates road segment failures by reducing the traffic capacity of each segment to 10% and measuring the impact of these failures on the overall traffic efficiency of the road network. The method for obtaining the node importance ranking is as follows: first, the impact of a failure is determined by monitoring changes in the average speed within the network after a segment failure. If the average speed on a segment falls below 10% of its speed limit, the segment is considered to have failed. Next, the congestion level of each segment and its propagation capability within the network are calculated across different time periods. These are then weighted and summed using a decay factor γ to account for the importance across different time periods. Finally, the importance scores of the segments are obtained, and the nodes are ranked accordingly.

B. Experimental Environment and Evaluation Metrics

To quantitatively evaluate our model framework, this study utilizes the following four metrics:

- **Accuracy (Acc)**: Accuracy represents the percentage of correctly classified samples out of the total samples. It is calculated by dividing the sum of true positives and true negatives by the total number of samples.
- **Precision (Pre)**: Precision refers to the proportion of samples predicted as positive that are actually positive. It is computed as the number of true positives divided by the total number of samples predicted as positive.
- **Recall (Re)**: Recall indicates the percentage of actual positive samples that are correctly identified as positive. It is calculated by dividing the number of true positives by the total number of actual positive samples.
- **F1 Score (F1)**: The F1 score is the weighted harmonic mean of precision and recall, providing a balanced evaluation of these two metrics.

All methods were developed using Python 3.8 and executed on a server equipped with an Intel Xeon(R) Platinum 8370 CPU and an RTX 3090 24G GPU. The server runs the Ubuntu 20.04 operating system, with the deep learning framework PyTorch 1.8.1, and relies on CUDA 11.2 for acceleration.

C. Experimental Design

First, to demonstrate the effectiveness of the proposed strategy, we compare the algorithm introduced in this paper with various existing algorithms for identifying critical nodes within a road network. In the experiments, the training and testing sets are set at a 6:4 ratio, with different proportions of critical nodes included in the training set. The comparison algorithms are as follows:

- **Random Selection (RD)**: This method does not consider the connection patterns or attribute information of nodes; instead, it relies solely on random sampling to select nodes.
- **Degree Centrality (DG)** [14]: This approach assumes that a node's importance is proportional to the number of its connections, meaning nodes with more connections are deemed more important.
- **Betweenness Centrality (BT)** [15]: This method measures a node's importance by calculating the frequency with which it acts as an intermediary in the shortest paths throughout the network. Nodes with higher frequencies are considered more critical.
- **Closeness Centrality (CS)** [16]: This method evaluates a node's importance based on its distance to other nodes in the network, assuming that nodes closer to others are more significant.
- **PageRank (PR)** [17]: This algorithm assesses the importance of a node by considering both its connectivity structure and the influence of its neighbors. Nodes connected to more important nodes are themselves considered more important.

Second, we designed ablation experiments, where we evaluated the performance by using only DAAN, only SFE, and both DAAN and SFE without processing through MLP.

D. Comparative Experimental Results

In the comparative experiment of this paper, we compared the performance of algorithms under different training set key node ratios. The experimental results demonstrate the superiority of the proposed model across various ratios. The detailed comparison results are shown in the table I, and the trend of F1 values is shown in the Fig. 2.

Specifically, when the key node ratio of the training set is 10%, the proposed method still maintains a leading position in all metrics. Although the indicators of all algorithms have increased, the proposed method shows a particularly significant improvement, especially in recall rate and F1 value, which are increased by 17.50, 19.80, and 19.73 compared to RD, DG, BT, CS, and PR, respectively.

At the 15% key node ratio in the training set, the proposed method again demonstrates its advantage in F1 value,

TABLE I: Comprehensive Comparison of Algorithms with Different Critical Node Ratios

Training Set Critical Node Ratio	Method	Acc	Pre	Re	F1
10%	RD	87.20	3.00	8.00	4.48
	DG	87.43	3.30	8.90	4.94
	BT	87.28	4.00	10.00	5.56
	CS	86.50	4.50	12.80	6.00
	PR	87.62	3.50	9.50	4.98
	OUR	88.15	6.00	22.50	8.74
15%	RD	87.40	3.50	9.50	5.00
	DG	87.58	4.00	10.50	5.82
	BT	87.45	4.50	11.50	6.22
	CS	86.70	5.00	13.50	6.67
	PR	87.81	4.20	10.80	5.73
	OUR	88.25	6.50	23.00	8.95
20%	RD	87.35	4.00	10.00	5.71
	DG	87.55	4.50	11.00	6.39
	BT	87.42	5.00	12.50	7.14
	CS	86.62	5.50	14.00	7.90
	PR	87.74	4.80	11.50	6.77
	OUR	88.47	7.00	25.00	10.94
25%	RD	87.56	4.50	11.50	6.47
	DG	87.71	5.00	12.50	7.14
	BT	87.60	5.50	14.00	7.90
	CS	86.85	6.00	15.00	8.57
	PR	87.95	5.20	12.00	7.26
	OUR	88.65	7.50	27.00	11.74

reaching 8.95, compared to RD's 5.00. Precision and recall rate have also significantly improved, further verifying the effectiveness of the proposed method.

As the training set key node ratio gradually increases to 20% and 25%, the performance of the proposed method continues to improve, especially in recall rate and F1 value. Even at higher ratios, the proposed method still maintains strong robustness, outperforming all baseline algorithms, which validates the stability and superiority of the proposed method across different key node ratios.

In summary, the model proposed in this paper demonstrates excellent performance under different key node ratios in the training set and surpasses existing baseline algorithms in the four main metrics: accuracy, precision, recall, and F1 value, fully proving the effectiveness and advantages of the proposed method.

E. Ablation Experiment Results

In this ablation study, we evaluated the impact of different model components on model performance, as observed from the Table. II, with training set critical node ratios set at 10%, 15%, 20%, and 25%. We analyzed the roles of three key components: DAAN, SFE, and MLP. The experimental results show that with the progressive addition of each component, the model's performance metrics and F1 scores improved.

Firstly, the addition of DAAN played a crucial role in enhancing performance. Without DAAN, the model's performance was relatively low. For example, at a 10% critical node ratio, when DAAN was added, the F1 score increased from 8.09 to 8.74, and as other components were added, the F1 score continued to rise, highlighting the importance of DAAN in capturing and aggregating dynamic attention. Specifically, at a 25% critical node ratio, DAAN improved the F1 score from 9.26 to 11.74, showing its significant impact on model performance, especially at higher critical node ratios.

Secondly, the inclusion of SFE also had a positive impact on the model's performance. At the 10% ratio, adding SFE improved the F1 score from 8.09 to 8.51. As the critical node ratio increased, the effect of SFE became more pronounced. At a 25% ratio, the F1 score increased from 9.26 to 9.50 and then to 9.75, demonstrating SFE's advantage in enhancing spatial feature representation, enabling the model to better identify key nodes in the data.

As the final component, MLP also played a crucial role in improving model performance. At the 10% ratio, the F1 score increased from 8.72 to 8.74, though the improvement was modest. However, at higher ratios, the addition of MLP significantly boosted the F1 score. For example, at the 25% ratio, the F1 score increased from 9.75 to 11.74, indicating that MLP could further improve the model's ability to identify subtle patterns and enhance classification accuracy through non-linear transformations.

Overall, as the critical node ratio in the training set increased, the model's performance gradually improved across all metrics. Specifically, the F1 score showed significant improvement through the combined effects of DAAN, SFE, and MLP. Notably, at the 25% critical node ratio, the F1 score reached 11.74, demonstrating the importance of the collaborative effect of all components in optimizing model performance. Therefore, the combination of dynamic attention mechanisms, static feature encoding, and multi-layer perceptrons effectively enhances the model's performance in

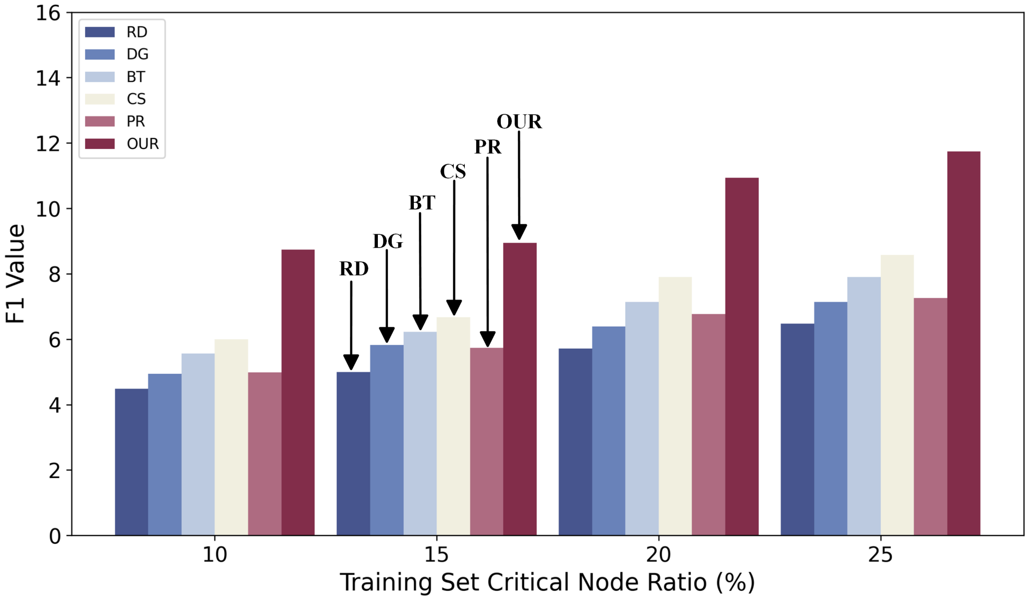

Fig. 2: F1 Values Comparison of Algorithms with Different Critical Node Ratios

TABLE II: Ablation study results.

Training Set Critical Node Ratio	DAAN	SFE	MLP	Acc	Pre	Re	F1
10%	✓			87.50	5.20	20.00	8.09
		✓		87.75	5.50	21.00	8.51
	✓	✓		88.00	5.80	21.50	8.72
	✓	✓	✓	88.15	6.00	22.50	8.74
15%	✓			87.50	5.50	21.00	8.53
		✓		87.75	5.80	22.00	8.76
	✓	✓		88.00	6.00	22.50	8.90
	✓	✓	✓	88.25	6.50	23.00	8.95
20%	✓			87.60	6.00	23.00	9.10
		✓		87.85	6.30	24.00	9.42
	✓	✓		88.10	6.50	24.50	9.65
	✓	✓	✓	88.47	7.00	25.00	10.94
25%	✓			87.80	6.30	24.50	9.26
		✓		88.00	6.50	25.00	9.50
	✓	✓		88.30	6.80	26.00	9.75
	✓	✓	✓	88.65	7.50	27.00	11.74

key node recognition and prediction tasks.

IV. CONCLUSION

The parallel encoding method proposed in this paper effectively integrates the dynamic and static features of nodes in transportation networks by combining the DAAN and the RSE. Experimental results demonstrate that the proposed method significantly improves performance in the task of critical node identification, particularly in terms of accuracy, generalization ability, and stability, compared to existing methods. Through comparative experiments and ablation analysis, we further validate the importance and synergistic effect of each module within the overall architecture. The resulting node importance ranking list provides a scientific basis for the management and optimization of transportation networks, proving the substantial potential of the proposed method in practical applications.

REFERENCES

[1] G. Wang, W. Cao, and Q. Ai, “Sw-stp: A spatio temporal pattern based model for detecting traffic flow data anomalies.” *Engineering Letters*, vol. 32, no. 11, pp. 2090–2098, 2024.

[2] C. Mylonas, E. Mitsakis, and K. Kepaptsoglou, “Criticality analysis in road networks with graph-theoretic measures, traffic assignment, and simulation,” *Physica A: Statistical Mechanics and its Applications*, vol. 629, p. 129197, 2023.

[3] M. A. Taylor, “Critical transport infrastructure in urban areas: impacts of traffic incidents assessed using accessibility-based network vulnerability analysis,” *Growth and Change*, vol. 39, no. 4, pp. 593–616, 2008.

[4] E. Jenelius and L.-G. Mattsson, “Road network vulnerability analysis of area-covering disruptions: A grid-based approach with case study,” *Transportation research part A: policy and practice*, vol. 46, no. 5, pp. 746–760, 2012.

[5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[6] S. Khoshraftar and A. An, “A survey on graph representation learning methods,” *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 1, pp. 1–55, 2024.

[7] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He *et al.*, “A survey of graph neural networks for recommender systems: Challenges, methods, and directions,” *ACM Transactions on Recommender Systems*, vol. 1, no. 1, pp. 1–51, 2023.

[8] Z. Liu, T.-K. Nguyen, and Y. Fang, “Tail-gnn: Tail-node graph neural networks,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1109–1119.

[9] S. F. Ahmed, S. A. Kuldeep, S. J. Rafa, J. Fazal, M. Hoque, G. Liu, and A. H. Gandomi, “Enhancement of traffic forecasting through graph neural network-based information fusion techniques,” 2024.

[10] V. Ashish, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, p. I, 2017.

[11] M. Xu and J. Zhang, “Mgl2rank: Learning to rank the importance of nodes in road networks based on multi-graph fusion,” *Information Sciences*, vol. 667, p. 120472, 2024.

[12] P. Alvarez Lopez, A. Banse, M. Barthauer, M. Behrisch, B. Couéraud, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, R. Nippold, J. Rummel, and P. Wagner, “Simulation of urban mobility (sumo),” May 2024. [Online]. Available: https://doi.org/10.5281/zenodo.11120728

[13] T. Mikolov, “Distributed representations of words and phrases and their compositionality,” *arXiv preprint arXiv:1310.4546*, 2013.

[14] T. Opsahl, “Degree centrality in a weighted network,” *Phys. Rev. E*, vol. 77, no. 4, pp. 1–10, 2008.

[15] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[16] K. Okamoto, W. Chen, and X.-Y. Li, “Ranking of closeness centrality for large-scale social networks,” in *International workshop on frontiers in algorithmics*. Springer, 2008, pp. 186–195.

[17] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

Xiaoxin Li is a postgraduate student at the Software College, Liaoning Technical University, Huludao, China. Her research interests include: data-driven traffic simulation and intelligent transportation system.

Ming Xu received his Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015. From 2016 to 2019, he was a Postdoctoral Fellow at Tsinghua University, Beijing, China. He is currently a professor at the Software College, Liaoning Technical University. He has published more than 20 papers in journals and conferences, including TITS, BIBM, and TCSS. His research work has reported by MIT Technology Reviews. He is the recipient of the ’2020 World Artificial Intelligence Conference Youth Outstanding Paper’ (WAICYOP) award.

His research interests include graph learning, spatio-temporal data mining, reinforcement learning, and data-driven traffic simulation.

Qing Ai(M’24) is currently a postgraduate student at the College of Computer, Qinghai Normal University. He received his bachelor’s degree from Software College, Liaoning Technical University. He became a Member (M) of IAENG in 2024. He has been worked at the Institute of Information Engineering, Chinese Academy of Sciences, and now he is the Principal Investigator of Urban Computing Laboratory, Aisess (Dalian) Computer Services Co., Ltd, while pursuing his Master’s degree at Qinghai Normal University.

His research interests include deep learning, combinatorial networks, urban computing, and intelligent transport.

Wenhui Cao is a postgraduate student at the Software College, Liaoning Technical University, Huludao, China. Her reasearch interests include anomaly detection, deep learing and spatio-temporal data mining.