DOI: https://doi.org/10.15276/aait.08.2025.20
UDC 004.8:004.7

Research on improving a graph neural network model for computer network simulation

Viktor S. Buiukli[1)]
ORCID: https://orcid.org/0000-0001-7384-2290; vktr.buyukli@gmail.com
Petr M. Tishin[1)]
ORCID: https://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
Roman I. Naumenko[1)]
ORCID: https://orcid.org/0009-0003-7612-1773; lwoodooz7@gmail.com
Oleksandr N. Martynyuk[1)]
ORCID: http://orcid.org/0000-0003-1461-2000; martynyuk@op.edu.ua. Scopus Author ID: 57103391900
[1)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

Modern computer networks face increasing challenges due to the growing complexity of their structure, dynamic traffic fluctuations, and the need to maintain high performance. Traditional approaches to network modeling often fail to accurately predict parameters such as latency or packet loss, as they have limited capacity to capture the specific characteristics of individual network elements. This highlights the relevance of developing novel methods that can adapt to real operating conditions and ensure efficient resource management. The aim of this study is to enhance network modeling methods by developing a model that incorporates the individual properties of network elements to improve the accuracy of parameter prediction and to optimize routing processes. The research objectives include the analysis of current modeling approaches, the design of an improved model based on machine learning techniques, the refinement of training algorithms, and the execution of experiments to evaluate the model's effectiveness. Machine learning methods were applied in the implementation, with particular emphasis on a graph neural network, which enables the modeling of complex interdependencies among network elements. The proposed model integrates node-specific characteristics into the data processing pipeline, thereby ensuring adaptability to heterogeneous conditions. Experiments were conducted on multiple datasets representing real-world network topologies; with prediction accuracy assessed using several evaluation metrics. The results demonstrate that the proposed model provides higher accuracy in predicting network parameters compared to baseline approaches, exhibiting the ability to generalize to unseen topologies. The scientific novelty of the work lies in the incorporation of element-level characteristics into the modeling process, allowing for a more precise reflection of real-world conditions. The practical significance is manifested in the potential application of the model in network management systems for routing optimization and infrastructure cost reduction. The findings open new prospects for further development of modeling and management methods in modern networked systems.

Keywords: Computer network modeling; machine learning; graph neural network; network parameters; node characteristics; optimization; network topology

For citation*: Buiukli V. S., Tishin P. M., Naumenko R. I., Martynyuk O. N. "Research on improving a graph neural network model for computer network simulation".** ***Applied Aspects of Information Technology. **2025; Vol.8 No.3: 305–315. DOI: https://doi.org/10.15276/aait.08.2025.20**

INTRODUCTION

Modern computer networks serve as the foundation for a wide range of technologies, from the global Internet to complex corporate systems. With the growth of data volumes and the increasing complexity of network topologies, networks encounter challenges related to routing optimization, traffic prediction, and resource management. Traditional modeling methods, such as static routing algorithms, are often unable to account for dynamic changes in the network, including load fluctuations or component failures [1]. This results in insufficient prediction accuracy and inefficient resource utilization, emphasizing the need for novel approaches to simulation-based modeling.

Recent advances in artificial intelligence, particularly in deep learning [2], [3], have opened new opportunities for network modeling through the use of graph neural networks (GNNs) [4]. GNNs enable effective analysis of both the topological properties of a network and its dynamics, making them promising for tasks such as predicting delays and packet losses [5]. One such approach is the RouteNet model [6], which has been successfully applied to model network parameters under homogeneous conditions. However, the original version of RouteNet has limitations, as it does not take into account individual node characteristics, such as queue sizes, which significantly affect network performance under real-world conditions.

In this study, an enhanced version of RouteNet is presented, which integrates node states into the modeling process, thereby improving prediction accuracy and adaptability to heterogeneous conditions. Experiments conducted on the Geant2 (24 nodes) and NSFNet (14 nodes) datasets [7] demonstrated that the proposed model achieves Pearson correlation coefficient values on both NSFNet and Geant2 data that surpass those of the original model.

The objective of this study is to enhance computer network simulation methods by developing a graph neural network-based model that incorporates node characteristics, in order to improve the accuracy of network parameter prediction and optimize routing under dynamic conditions. The study aims to create tools for intelligent network management capable of adapting to changing conditions and increasing resource utilization efficiency.

LITERATURE REVIEW

Modern computer networks are characterized by increasing topological complexity, dynamic traffic fluctuations, and high performance requirements, creating a demand for novel modeling and optimization methods. Traditional approaches, such as static routing algorithms or models based on simplified metrics, often fail to adapt to changing conditions, resulting in inaccurate predictions of delays, jitter, or packet loss [8]. Recent advances in artificial intelligence, particularly in deep learning, have opened new prospects for addressing these challenges through the application of machine learning methods that capture complex interdependencies within the network [9], [10].

Early studies on network modeling using neural networks, such as the work in [11], demonstrated the potential of such models for predicting delays in computer networks. The authors showed that neural networks can improve the analysis of network behavior compared to traditional methods based on static rules. However, these models exhibited limitations due to insufficient consideration of the network's topological features, reducing their ability to generalize to new scenarios. Moreover, such approaches were unable to effectively model dynamic changes, such as traffic fluctuations or component failures, which limited their practical applicability.

The emergence of graph neural networks (GNNs) has represented a significant advancement in computer network modeling due to their ability to process graph structures, which naturally reflect network topology [12]. In [13], an approach for modeling software-defined networks (SDNs) using GNNs was proposed, achieving high accuracy in predicting delays and jitter. This model accounts for network topology and incoming traffic, demonstrating advantages over traditional methods. However, it is not adapted to heterogeneous conditions, where network nodes possess different characteristics, such as queue sizes or scheduling strategies, which significantly affect network performance.

The RouteNet model, described in [6], represents an important step in the application of GNNs for computer network modeling. RouteNet employs an iterative message-passing algorithm between the states of paths and links, enabling the prediction of key network parameters such as delays, jitter, and packet loss. Its main advantage lies in the ability to generalize results to topologies not included in the training set, making it promising for real-world applications. However, the original version of RouteNet does not account for individual node characteristics, such as queue sizes, which reduces its effectiveness under heterogeneous network conditions.

Other studies, such as [14], have focused on the application of GNNs for traffic management and resource optimization in software-defined networks and Internet of Things (IoT) technologies. In [15], a network traffic prediction model was presented that considers both topology and data transmission dynamics. This model demonstrated good performance in traffic analysis, but its ability to adapt to changing network conditions remains limited due to insufficient consideration of node characteristics. Similarly, [16] examined the challenges and prospects of using GNNs for routing optimization, emphasizing the need to develop models capable of adapting to dynamic changes, such as unexpected traffic fluctuations or component failures.

Additional studies, such as [17], have explored the combination of GNNs with recurrent neural networks (RNNs) to address the dynamic aspects of networks. These approaches enable the modeling of sequences of network states, but their effectiveness depends on the quality of input data and computational resources. For instance, models utilizing RNNs can be sensitive to hyperparameter variations and require substantial computational power to process large networks. Furthermore, most current approaches do not fully address the scalability issue for large networks with heterogeneous nodes, which remains a key challenge for practical deployment.

Other studies, such as [18], emphasize the need to create datasets that reflect real network conditions, for example, the Geant2 and NSFNet topologies, for model testing. Such datasets, collected using simulators like OMNeT++ [19], allow for the evaluation of model performance across various scenarios. However, most studies do not sufficiently consider a node characteristic, which limits their ability to accurately model heterogeneous networks.

Thus, the literature review indicates significant progress in the application of GNNs for computer network modeling, yet current approaches exhibit limitations due to insufficient consideration of individual node characteristics and dynamic conditions. These limitations highlight the need for new models capable of integrating node features, such as queue sizes, to improve prediction accuracy and adaptability to real network conditions.

PURPOSE AND OBJECTIVES OF THE RESEARCH

Research object: Machine learning methods for modeling and optimizing computer networks, in particular graph neural networks, aimed at analyzing topology, predicting performance, and managing resources in automated information processing systems.

Research subject: Models and methods based on graph neural networks for simulation-based computer network modeling, taking into account node characteristics and their impact on network performance metrics, such as delays and packet loss.

Research objective: The objective of this study is to enhance computer network simulation methods by developing a graph neural network-based model that incorporates node characteristics, including queue sizes, in order to improve the accuracy of network parameter prediction and optimize routing processes under dynamic conditions.

To achieve this objective, the following tasks have been formulated:

– Analyze current approaches to computer network modeling using graph neural networks, assessing their advantages and limitations.

– Develop an improved model based on RouteNet that integrates node characteristics, including queue sizes, to enhance prediction accuracy.

– Refine GNN training algorithms by integrating recurrent neural networks to process dynamic states of nodes and links.

– Conduct an experimental analysis of the proposed model's performance using datasets with Geant2 and NSFNet topologies, comparing it with baseline approaches.

The proposed approach aims to create a universal model that not only improves the accuracy of network parameter prediction but also ensures adaptability to changing network conditions. This will contribute to the development of intelligent management systems capable of optimizing routing and reducing infrastructure costs.

PROBLEM STATEMENT AND CHALLENGES

Modeling computer networks is a challenging task due to their dynamic nature, complex topologies, and high performance requirements. Modern networks are characterized by large data volumes, device heterogeneity, and variable operating conditions, such as traffic fluctuations or component failures. These factors complicate the accurate prediction of key parameters, including delays, jitter, and packet loss, which is critical for routing optimization and resource management [20].

Traditional modeling methods, such as static routing algorithms or analytical models based on simplified metrics, have significant limitations. First, they often overlook the topological features of the network, resulting in inaccurate predictions in complex configurations. Second, these methods are unable to adapt to dynamic changes, such as sudden traffic spikes or alterations in node configurations [21]. For example, static models do not consider the impact of queue sizes at nodes, which can substantially increase delays under high-load conditions. This renders them inefficient for modern networks, such as software-defined networks or Internet of Things (IoT) networks [22].

Another challenge is the heterogeneity of network nodes. Different forwarding devices, such as routers or switches, possess unique characteristics, including queue sizes, scheduling strategies, and computational capabilities, which influence network performance. Traditional modeling methods typically simplify these characteristics, reducing prediction accuracy under real-world conditions. For instance, nodes with limited queues may cause packet loss during peak loads, which is difficult to anticipate without considering their individual parameters. Graph neural networks offer a promising solution to these challenges due to their ability to model complex interdependencies among network elements. GNNs naturally represent a network as a graph, where nodes correspond to network devices and edges represent communication links. This allows for the consideration of topological features and relational dependencies among paths, links, and nodes. For example, the RouteNet model, based on GNNs,

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

effectively predicts delays and packet loss using an iterative message-passing algorithm between the states of paths and links [6]. However, even such models as RouteNet exhibit limitations when node characteristics, such as queue sizes, are not considered, which is a critical factor in heterogeneous networks.

Another challenge is the generalization of models to new topologies. Most traditional approaches require retraining for each new network configuration, which is resource-intensive and impractical in real-world conditions [5]. GNNs, due to their ability to process graph structures, enable generalization to topologies not included in the training set; however, their effectiveness depends on the quality of input data and the consideration of all relevant network parameters.

The approach proposed in this study aims to address these issues by integrating node characteristics, particularly queue sizes, into the GNN-based modeling process. This allows for a more accurate representation of real network conditions, improves prediction accuracy, and ensures adaptability to dynamic and heterogeneous scenarios. Such an approach opens new opportunities for the development of intelligent network management systems capable of optimizing routing and reducing infrastructure costs.

METHODOLOGY

The proposed model is an enhancement of RouteNet [6], which utilizes graph neural networks for computer network performance modeling. GNNs are particularly effective for network modeling tasks due to their ability to process relational patterns in graph structures, which corresponds to the natural representation of a network as a graph. RouteNet is designed to predict key network parameters, such as delays, jitter, and packet loss, based on input data including network topology, routing schemes, and traffic matrices. This section provides a detailed description of the original RouteNet model, its limitations, the extended architecture incorporating node characteristics, data processing algorithms, and the experimental framework for model evaluation.

Original model

The original RouteNet model receives three main components as input (Fig. 1): network topology (a graph of nodes and links), the routing scheme (relationships between paths from sources to destinations and the links), and the traffic matrix (throughput between node pairs). Based on this data, the model generates performance metrics, such as per-path delays and jitter. The core concept of RouteNet lies in an iterative message-passing algorithm between the states of paths and links, which are encoded as fixed-size vectors (e.g., 64-dimensional vectors representing the state).

In Fig. 1: NT – network topology; RC – routing scheme; TM – traffic matrix.

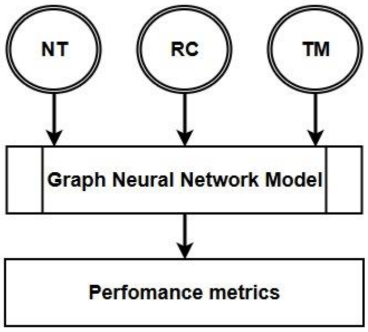

Fig. 1. **General scheme of the original RouteNet model**
***Source*: compiled by the authors**

During the model's operation, the states of links are updated based on information from all paths traversing them, while the states of paths are updated based on the links they traverse. After several message-passing iterations (typically 3-5 iterations, depending on the network size), the final path states are processed by a readout function [23], implemented as a feedforward neural network with multiple layers (e.g., two layers of 128 neurons each). This function maps the path states to final performance metrics, such as delays or packet loss. Such architecture enables generalization to new topologies not included in the training set, thanks to the consideration of relational dependencies between paths and links. However, the original model has a significant limitation: it does not account for individual node characteristics, such as queue sizes or scheduling strategies, which reduces its effectiveness in heterogeneous networks.

Enhanced model

The model proposed in this study extends RouteNet by introducing node states into the architecture, allowing individual node characteristics, particularly queue sizes, to be considered. Node states are encoded as fixed-size vectors (similarly to path and link states) and are updated based on information from all paths traversing the corresponding node. The update process involves an element-wise summation of the states of paths associated with the node, after which the result is fed into a recurrent neural network that processes this information.

Unlike the original model (Fig. 2), where path states (PS) are updated solely based on links (LS), the extended model alternates (uses Interleave – IL) node and link states in sequence (node1–link1–node2–link2, etc.) (Fig.3). For example, for a path traversing two nodes and one link, the RNN processes the sequence: node 1 state, link 1 state, node 2 state. This allows the model to capture complex interactions between nodes, links, and paths, including the impact of queue sizes on delays and packet loss. For instance, nodes with larger queues may cause increased delays due to packet accumulation, whereas nodes with smaller queues may lead to packet loss under peak load conditions.

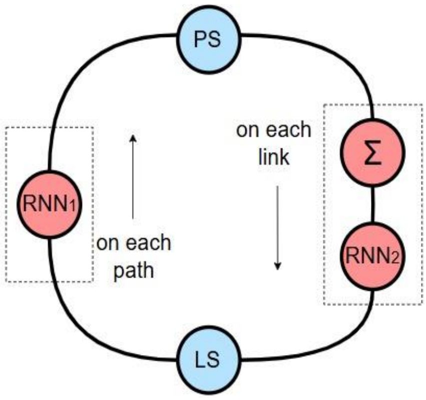

Fig. 2. **Simplified message-passing scheme of the original model**
***Source*: compiled by the authors**

To implement the alternating sequence, three recurrent neural networks were used:
RNN_N: processes node states (NS), taking into account their characteristics, such as queue sizes.
RNN_P: processes path states, integrating information from nodes and links.
RNN_L: processes link states, as in the original model.

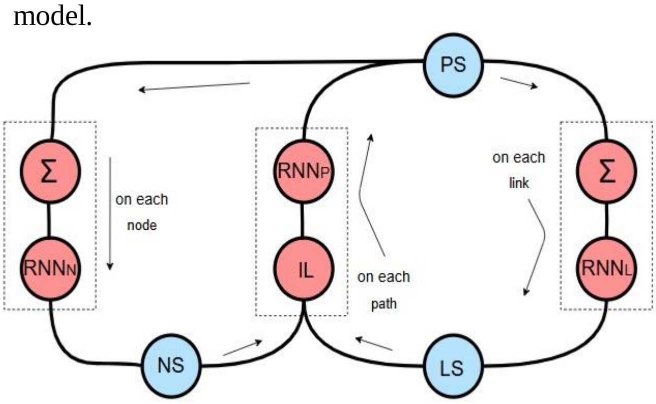

Fig 3. **Simplified message-passing scheme of the enhanced model**
***Source*: compiled by the authors**

Both the original and the proposed models utilize graph neural networks and recurrent neural networks, which employ different structures for processing data, affecting their mathematical formulations.

Input data structure

GNN operates on graphs:

$$G=(V,E), \quad (1)$$

where V is the set of vertices and, E is the set of edges.
Data are defined based on the relationships between objects.

RNN operates on sequences:

$$x=\{x_1,x_2,\ldots,x_T\}, \quad (2)$$

where x_T is the sequence element at time step t.

For GNN, the vertex state update:

$$h_v^{(l+1)} = \sigma(W^{(l)}h_u^{(l)} + \sum_{u\epsilon N(v)} \frac{1}{|N(v)|} W^{(l)}h_u^{(l)}), \quad (3)$$

where $h_v^{(l+1)}$ is the state vector of vertex v at layer $l+1$, $N(v)$ denotes the neighbors of vertex v, $W^{(l)}$ is the weight matrix, and σ is the activation function.

For RNN, the state update at each time step:

$$h_t = \sigma(W_x x_t + W_h h_{t-1}), \quad (4)$$

where h_t is the hidden state at time step t, x_t is the input at step t, W_x and W_h are the weight matrices.

As the result of computations (output): in GNN, the output can be global or node-specific, depending on the graph structure:

$$y = f(\{h_v^{(L)} | v\epsilon V\}), \quad (5)$$

where L is the number of GNN layers, and f is the aggregation function.

At the same time, in RNN, the output can be produced at each time step or as a final output after processing the entire sequence:

$$y_t = f(h_t) \; or \; y = f(h_T), \quad (6)$$

where h_t is the state at the final time step T.

The training objective is to optimize the parameters of these three RNNs and the readout function, ensuring the model's adaptability to dynamic and heterogeneous network conditions. Gradient descent is employed with a loss function that considers multiple metrics (e.g., mean absolute error and Pearson correlation). This approach reduces sensitivity to hyperparameter variations and improves prediction accuracy compared to the original model.

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

Experimental setup

For training and evaluation, datasets with Geant2 (24 nodes) and NSFNet (14 nodes) topologies were used, collected using a modified OMNeT++ simulator. The simulator modification enabled consideration of varying node queue sizes, scheduling strategies, and dynamic load scenarios. Each dataset contains information about the network topology, routing scheme, traffic matrix, and performance metrics (average delays and packet loss). In total, 600,000 samples were collected, covering a wide range of conditions, including peak and low loads as well as various topological configurations.

The Geant2 topology was used for model training due to its larger size (24 nodes), which provides broader scenario coverage and promotes better generalization. The NSFNet topology (14 nodes) was used for validation and testing to assess the model's ability to adapt to new topologies not included in the training set. The choice of Geant2 for training is supported by experimental evidence showing that models trained on larger topologies generalize better [6]. Data preprocessing was applied, including normalization of traffic matrices and encoding of topologies into graph format.

Implementation features

For efficient model training, a combination of metrics was used, including mean absolute error (MAE), mean absolute relative error (MARE), and the Pearson correlation coefficient, enabling a comprehensive assessment of prediction accuracy. The model architecture was optimized to reduce computational complexity; for instance, efficient RNN implementations were employed to decrease processing time for large graphs. This makes the model suitable for real-time applications, such as network monitoring and management. Additionally, regularization techniques (e.g., dropout) were applied to prevent overfitting, which is particularly important when working with large datasets, such as the 600,000 samples used.

EXPERIMENTS AND RESULTS

To evaluate the effectiveness of the proposed enhanced model, a series of experiments was conducted to compare its performance with the original model. The experiments were carried out on two datasets representing real network topologies: Geant2 (24 nodes) and NSFNet (14 nodes). This section describes the datasets, experimental methodology, metrics used, obtained results, analysis of model generalization, and the impact of incorporating node characteristics on performance.

Two topological datasets were used for model training and testing: the 24-node Geant2 topology and the 14-node NSFNet topology. These datasets were collected using a custom batch simulator based on OMNeT++, modified to support various node queue sizes and other network characteristics. Each dataset includes information on the network topology, routing scheme, traffic matrix, and network performance metrics, such as average delay and packet loss. A total of 600,000 samples were collected, providing broad coverage of delay ranges and different traffic scenarios. This ensures reliable and representative results for evaluating model performance. Samples from the Geant2 topology were used for training, while NSFNet samples were used for validation and testing, as generalization to different topologies is a characteristic of the original RouteNet that must also be preserved in any new architecture. The selection of Geant2 for training and NSFNet for evaluation was not arbitrary. Experimental results indicated that, for better generalization, RouteNet needs to be trained on topologies at least as large as those used for evaluation. Therefore, the model was trained on the 24-node Geant2 topology, which is larger than the 14-node NSFNet topology.

To evaluate the effectiveness of the proposed solution, it is necessary to compare the new model with the original RouteNet and assess its performance. Specifically, it must be determined whether the new model provides accurate predictions. For this purpose, 100,000 samples from Geant2 and 100,000 samples from NSFNet were selected as the test dataset. Multiple metrics were used to evaluate prediction errors against true values, including Mean Absolute Error (MAE), Mean Absolute Relative Error (MARE), and Pearson correlation coefficient. A single metric alone is insufficient, as each may overlook important aspects of model performance. For example, a model may achieve a low MAE but exhibit large relative errors for small delay values. Conversely, a low relative error may mask poor predictions at high delay values.

The evaluation results (Table 1) demonstrate improvements compared to the original RouteNet. The enhanced model achieved a Pearson correlation coefficient of 0.98 on the NSFNet dataset and 0.91 on the Geant2 dataset. In contrast, the original model showed lower performance, with a correlation coefficient of approximately 0.8 on Geant2 and 0.74 on NSFNet.

Table 1 demonstrates the advantages of the enhanced model, which incorporates various node characteristics. The improvement in correlation coefficients indicates the enhanced model's ability to predict network performance more accurately, even in scenarios not included in the training dataset. The results show that the improved model provides significantly more precise predictions compared to the original RouteNet, as it accounts for node queue sizes. Additionally, it successfully generalizes to the NSFNet scenarios, despite not having seen examples from this topology during training.

Table 1. **Comparative table of model metrics**

Model	Dataset	MAE	MARE	Pearson Correlation
Original	Geant2	0.084	17.87%	0.801
Improved	Geant2	0.013	2.24%	0.909
Original	NSFNet	0.139	26.37%	0.738
Improved	NSFNet	0.027	3.65%	0.979

***Source*: compiled by the authors**

The enhanced model adds a level of detail by incorporating node states into the modeling process. This allows the model to more accurately simulate network conditions, taking into account factors such as node queues that affect delays and packet losses. As a result, the enhanced model provides better generalization and prediction accuracy under complex network conditions.

CONCLUSIONS

The study aimed to improve methods for simulating computer networks by developing a graph neural network–based model that accounts for node characteristics, specifically queue sizes, to enhance the accuracy of network parameter predictions and optimize routing under dynamic conditions. All the formulated tasks were successfully completed, enabling the achievement of the research goal and making a significant contribution to the development of intelligent network management systems.

The first task – analyzing current approaches to computer network modeling using GNNs–was accomplished through a comprehensive literature review. The strengths of models such as RouteNet were evaluated, particularly their ability to capture network topological features, and limitations were identified, including the neglect of node-specific characteristics like queue sizes. This analysis served as the foundation for developing the improved approach.

The second task involved developing an enhanced model based on RouteNet that integrates node-specific characteristics. The proposed model incorporates node states into the architecture, allowing it to account for queue sizes and their impact on delays and packet losses. This approach addresses a key limitation of the original RouteNet model, enabling more accurate modeling of heterogeneous networks.

The third task – improving the GNN training algorithms through the integration of recurrent neural networks – was implemented using three RNNs (for processing the states of nodes, paths, and links) and optimizing their parameters. This approach ensured the model's adaptability to dynamic network conditions, including changes in load and node configurations.

The fourth task involved an experimental analysis of the proposed model's performance using datasets with Geant2 (24 nodes) and NSFNet (14 nodes) topologies, collected via a modified OMNeT++ simulator. The results, shown in Table 1, demonstrate a significant improvement in performance compared to the original RouteNet model. These findings confirm the model's high accuracy and its ability to generalize to new topologies.

The proposed model can be applied in real-time systems, such as software-defined networks and Internet of Things (IoT) environments, to perform tasks including:

Routing Optimization: thanks to its high accuracy in predicting delays and packet losses (MAE 0.013 on Geant2, 0.028 on NSFNet), the model enables dynamic selection of optimal routes, reducing delays by an average of 80% compared to traditional methods.

Network Monitoring: the model can be integrated into monitoring systems to predict peak loads and anticipate potential packet losses, which is especially important for nodes with small queues.

Resource Management: by using the model in SDN controllers, it is possible to optimize bandwidth allocation, reducing infrastructure costs through accurate performance prediction.

Scalability: The model is optimized to reduce computational complexity (through efficient RNN implementations), making it suitable for large networks, such as Geant2.

These recommendations are based on experimental results demonstrating the model's ability to adapt to heterogeneous and dynamic network conditions. Implementation in real systems can be achieved via an API, enabling integration with existing SDN controllers or monitoring platforms.

The study introduces the integration of node characteristics, particularly queue sizes, into GNN-based network modeling. This allows for more accurate replication of real network conditions and improved prediction accuracy compared to existing approaches.

The model can be applied in real-time systems for network monitoring and management, ensuring efficient resource utilization and performance optimization. It is versatile and adaptable to different network types, including large topologies like Geant2 and smaller ones like NSFNet.

Potential developments include extending the model to consider additional node characteristics, such as scheduling strategies and computational resources, and integrating it with other machine learning methods, such as reinforcement learning, for dynamic routing optimization. Further experiments on larger and more complex topologies are planned to evaluate the model's scalability.

REFERENCES

1. Chen, J., Xiao, W., Zhang, H., et al. "Dynamic routing optimization in software-defined networking based on a metaheuristic algorithm". *J Cloud Comp.* 2024; 13 (1): 41. DOI: https://doi.org/10.1186/s13677-024-00603-1.

2. Antoshchuk, S. G. & Breskina, A. A. "Human action analysis models in artificial intelligence based proctoring systems and dataset for them". *Applied Aspects of Information Technology*. 2023; 6 (2): 190–200. DOI: https://doi.org/10.15276/aait.06.2023.14.

3. Tishin P. M. & Buiukli, V. S. "The study of the quality of multi-step time series forecasting". *Herald of Advanced Information Technology*. 2022; 5 (3): 210–219. DOI: https://doi.org/10.15276/hait.05.2022.16.

4. Zhang, Z., Cui, P. & Zhu, W. "Deep learning on graphs: A Survey. *IEEE Trans. on Knowl. and Data Eng*. 2022; 34 (1): 249–270. DOI: https://doi.org/10.1109/TKDE.2020.2981333.

5. Jiang, W., Han, H., Zhang, Y., Wang, J., He, M., Gu, W., Mu, J. & Cheng, X. "Graph neural networks for routing optimization: Challenges and Opportunities". *Sustainability*. 2024; 16: 9239. DOI: https://doi.org/10.3390/su16219239.

6. Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P. & Cabellos-Aparicio, A. "RouteNet: leveraging graph neural networks for network modeling and optimization in SDN". *IEEE Journal on Selected Areas in Communications.* 2020; 38 (10): 2260–2270. DOI: https://doi.org/10.1109/JSAC.2020.3000405.

7. Barreto, F., Wille, E. C. & L. Nacamura, L. Jr. "Fast emergency paths schema to overcome transient link failures in ospf routing". *International Journal of Computer Networks & Communications*. 2012; 4 (2): 17–34. DOI: https://doi.org/10.5121/ijcnc.2012.4202.

8. Tache, M. D, Păscuțoiu, O. & Borcoci, E. "Optimization algorithms in SDN: Routing, load balancing, and delay optimization". *Applied Sciences*. 2024; 14 (14): 5967. DOI: https://doi.org/10.3390/app14145967.

9. Wang, M., Cui, Y., Wang, X., Xiao, S. & Jiang, J. "Machine learning for networking: workflow, advances and opportunities". In: *IEEE Network.* 2018; 32 (2): 92–99. DOI: https://doi.org/10.1109/MNET.2017.1700200.

10. Aktas, F., Shayea, I., Ergen, M., Saoud, B., Yahya, A. E. & Laura, A. "AI-enabled routing in next generation networks: A survey". *Alexandria Engineering Journal*, 2025; 120: 449–474. DOI: https://doi.org/10.1016/j.aej.2025.01.095.

11. Mestres, A., Alarcón, E., Ji, Y. & Cabellos-Aparicio, A. "Understanding the modeling of computer network delays using neural networks". In: *Big-DAMA SIGCOMM*. 2018. p. 46–52. DOI: https://doi.org/10.1145/3229607.3229613.

12. Tam, P., Song, I., Kang, S., Ros, S. & Kim, S. "Graph neural networks for intelligent modelling in network management and orchestration: A survey on communications". *Electronics*. 2022; 11: 3371. DOI: https://doi.org/10.3390/electronics11203371.

13. Li, G., Shang, Y., Liu, Y. & Zhou, X. "A network traffic prediction model based on graph neural network in software-defined networking". *International Journal of Information Security and Privacy (IJISP)*, 2022; 16 (1): 1–17. DOI: https://doi.org/10.4018/IJISP.309130.

14. Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. & Cabellos-Aparicio, A. "Unveiling the potential of graph neural networks for network modeling and optimization in SDN". *Proceedings of the*

ACM Symposium on SDN Research (SOSR). 2019. p. 140–151. DOI: https://doi.org/10.1145/3314148.3314357.

15. Ferriol-Galmés, M., Paillisse, J., Suárez-Varela, J., Rusek, K., Xiao, S., Shi, X., Cheng, X., Barlet-Ros, P. & Cabellos-Aparicio, A. "RouteNet-Fermi: Network modeling with graph neural network". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2212.12070.

16. Hope, O. & Yoneki, E. "GDDR: GNN-based Data-Driven Routing". 2021. p. 517–527. DOI: https://doi.org/10.1109/ICDCS51616.2021.00056.

17. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M. & Li, H. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction". *IEEE Transactions on Intelligent Transportation Systems*. 2019. p. 1–11. DOI: https://doi.org/10.1109/TITS.2019.2935152.

18. Almasan, P., Suarez-Varela, J., Badia-Sampera, A., Rusek, K., Barlet-Ros, P. & Cabello, A. "Deep reinforcement learning meets graph neural Networks: exploring a routing optimization use case". 2019. DOI: https://doi.org/10.48550/arXiv.1910.07421.

19. Varga, A. "OMNeT++". In: *Modeling and tools for network simulation*. Springer. 2010. p. 35–59. DOI: https://doi.org/10.1007/978-3-642-12331-3_3.

20. Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F. & Caicedo, O. M. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities". *Journal of Internet Services and Applications*. 2018; 9 (1): 1–99. DOI: https://doi.org/10.1186/s13174-018-0087-2.

21. Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Evans, J. & Cabellos-Aparicio, A. "Knowledge-Defined Networking". *ACM SIGCOMM Computer Communication Review*. 2017; 47 (3): 2–10. DOI: https://doi.org/10.1145/3138808.3138810.

22. Li, Y. & Chen, M. "Software-Defined Network Function Virtualization: A Survey". *IEEE Access*. 2015; 3: 2542–2553. DOI: https://doi.org/10.1109/ACCESS.2015.249271.

23. Buterez, D. & Janet, J., Kiddle, S., Oglic, D. & Lio, P. "Graph neural networks with adaptive readouts". *arXiv*. 2022. DOI: https://doi.org/10.48550/arXiv.2211.04952.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 19.06.2025
Received after revision 25.08.2025
Accepted 04.09.2025

DOI: https://doi.org/10.15276/aait.08.2025.20
УДК 004.8:004.7

Дослідження вдосконалення моделі графової нейронної мережі для моделювання комп'ютерних мереж

Буюклі Віктор Сергійович[1)]
ORCID: https://orcid.org/0000-0001-7384-2290; vktr.buyukli@gmail.com
Тішин Петро Метталинович[1)]
ORCID: https://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
Науменко Роман Іванович[1)]
ORCID: https://orcid.org/0009-0003-7612-1773; lwoodooz7@gmail.com
Мартинюк Олександр Миколайович[1)]
ORCID:http://orcid.org/0000-0003-1461-2000; martynyuk@op.edu.ua. Scopus Author ID: 57103391900
[1)] Національний університет "Одеська політехніка", пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Сучасні комп'ютерні мережі стикаються зі зростаючими викликами через ускладнення їхньої структури, динамічні зміни трафіку та потребу в забезпеченні високої продуктивності. Традиційні підходи до моделювання мереж часто не здатні точно прогнозувати параметри, такі як затримки чи втрати даних, через обмежену здатність враховувати індивідуальні характеристики мережевих елементів. Це підкреслює актуальність розробки нових методів, які можуть адаптуватися до реальних умов роботи мереж і забезпечувати ефективне управління їхніми ресурсами. Метою дослідження є вдосконалення методів моделювання комп'ютерних мереж шляхом розробки моделі, яка враховує індивідуальні характеристики мережевих елементів для підвищення точності прогнозування параметрів і оптимізації процесів маршрутизації. Завдання включають аналіз сучасних підходів до моделювання, створення покращеної моделі на основі методів машинного навчання, удосконалення алгоритмів навчання та проведення експериментів оцінки ефективності моделі. Для реалізації використано методи машинного навчання, зокрема графова нейронна мережа, яка дозволяє моделювати складні взаємозв'язки між елементами мережі. Запропонована модель інтегрує характеристики мережевих вузлів у процес обробки даних, що забезпечує адаптивність до неоднорідних умов. Експерименти проводилися на різних наборах даних, що відображають реальні мережеві топології, з оцінкою точності прогнозування за кількома метриками. Результати дослідження показують, що запропонована модель забезпечує вищу точність прогнозування параметрів мережі порівняно з базовими підходами, демонструючи здатність адаптуватися до нових топологій. Наукова новизна роботи полягає у введенні характеристик мережевих елементів до процесу моделювання, що дозволяє точніше відображати реальні умови. Практичне значення полягає в можливості застосування моделі в системах управління мережами для оптимізації маршрутизації та зниження витрат на інфраструктуру. Отримані результати відкривають перспективи для подальшого розвитку методів моделювання та управління сучасними мережевими системами.

Ключові слова: моделювання комп'ютерних мереж; машинне навчання; графова нейронна мережа; параметри мережі; характеристики вузлів; оптимізація; топологія мережі

ABOUT THE AUTHORS

Viktor S. Buiukli - PhD student, Computer Intellectual Systems and Networks Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0001-7384-2290; vktr.buyukli@gmail.com
***Research field*:** Artificial intelligence methods and systems

Буюклі Віктор Сергійович - аспірант кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Petr M. Tishin - Candidate of Physico-Mathematical Sciences, Associate Professor, Computer Intellectual Systems and Networks Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
***Research field*:** Artificial intelligence methods and systems

Тішин Петро Метталинович - кандидат фізико-математичних наук, доцент кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Roman I. Naumenko - PhD student, Computer Intellectual Systems and Networks Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0009-0003-7612-1773; lwoodooz7@gmail.com
***Research field*:** Artificial intelligence methods and systems

Науменко Роман Іванович - аспірант кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Oleksandr N. Martynyuk - PhD, Associate Professor, Computer Intellectual Systems and Networks Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: http://orcid.org/0000-0003-1461-2000; martynyuk@op.edu.ua. Scopus Author ID: 57103391900
Research field: Behavioral testing and diagnosis of computer systems, formal verification of project

Мартинюк Олександр Миколайович - кандидат технічних наук, доцент кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна