JSLHU JOURNAL OF SCIENCE OF LAC HONG UNIVERSITY
ISSN: 2525 - 2186
Tạp chí Khoa học Lạc Hồng, **2025**, 20, 025-030

MÔ HÌNH HÓA THÀNH CÔNG CỦA NGƯỜI DÙNG TRONG MẠNG XÃ HỘI TRỰC TUYẾN BẰNG CÁC KIẾN TRÚC GCNNS

Mai Trung Thành[1*], Phạm Minh Triết[2], Nguyễn Thanh Thư[3]
[1]*Trường Đại học Quốc tế Hồng Bàng, số 215 Điện Biên Phủ, quận Bình Thạnh, TP. Hồ Chí Minh, Việt Nam*
[2]*Phòng Hậu Cần, Công an tỉnh Bình Dương, số 681, đường Cách Mạng Tháng Tám, phường Chánh Nghĩa, thành phố Thủ Dầu Một, tỉnh Bình Dương, Việt Nam*
[3]*Trường Đại học Vinh, số 182 Lê Duẩn, phường Trường Thi, thành phố Vinh, tỉnh Nghệ An, Việt Nam*
* Tác giả liên hệ: *thanhmtbdu@gmail.com*

THÔNG TIN BÀI BÁO

Ngày nhận: 11/2/2025
Ngày hoàn thiện: 6/3/2025
Ngày chấp nhận: 14/3/2025
Ngày đăng: 15/3/2025

TỪ KHÓA

Mạng xã hội trực tuyến;
Thành công chuyên nghiệp;
Mạng nơ-ron tích chập đồ thị;
Biểu đồ xã hội.

TÓM TẮT

Các mạng xã hội trực tuyến (OSNs) cung cấp lượng dữ liệu phong phú, phản ánh tính cách, sở thích và các mối quan hệ xã hội của người dùng. Nghiên cứu này khám phá cách các mạng nơ-ron tích chập đồ thị (Graph Convolutional Neural Networks - GCNNs) có thể phân tích dữ liệu từ mạng xã hội VKontakte để dự đoán sự thành công trong nghề nghiệp của người dùng. Bằng cách sử dụng các đặc trưng như hồ sơ cá nhân và mối quan hệ xã hội, nghiên cứu đã đánh giá nhiều kiến trúc GCNN, bao gồm GCNConv, SAGEConv và GINConv. Kết quả cho thấy lớp mạng Graph Isomorphism Network (GIN) đạt độ chính xác cao nhất (0.88). Nghiên cứu này nhấn mạnh hiệu quả của các mạng nơ-ron tiên tiến trong việc phân tích các chỉ số thành công nghề nghiệp trên các mạng xã hội trực tuyến.

MODELING USER SUCCESS IN ONLINE SOCIAL NETWORKS USING ADVANCED GNN ARCHITECTURES

Mai Trung Thanh[1*], Pham Minh Triet[2], Nguyen Thanh Thu[3]
[1]*Hong Bang International University, No. 215 Dien Bien Phu Street, Binh Thanh District, Ho Chi Minh City, Vietnam.*
[2]*Logistics Department, Binh Duong Provincial Police, No. 681, Cach Mang Thang Tam Street, Chanh Nghia Ward, Thu Dau Mot City, Binh Duong Province, Vietnam.*
[3]*Vinh University, No. 182 Le Duan Street, Truong Thi Ward, Vinh City, Nghe An Province, Vietnam.*
*Corresponding Author: *thanhmtbdu@gmail.com*

ARTICLE INFO

Received: Feb 11st, 2025
Revised: Mar 6th, 2025
Accepted: Mar 14th, 2025
Published: Mar 15th, 2025

KEYWORDS

Online social network;
Professional success;
Graph convolutional neural networks;
Social graph.

ABSTRACT

Online social networks (OSNs) provide extensive data reflecting users' personalities, interests, and social connections. The study explores how graph convolutional neural networks (GCNNs) can be used to analyze data from the VKontakte social network to predict users' professional success. Using features like user profiles and social connections, it evaluates various GCNN architectures, including GCNConv, SAGEConv, and GINConv. The Graph Isomorphism Network (GIN) layer achieved the highest accuracy (0.88). This research highlights the effectiveness of advanced neural networks in understanding professional success metrics in online social networks. Hong Bang International University, Ho Chi Minh City, Vietnam.

Doi: https://doi.org/10.61591/jslhu.20.624
Available online at: https://js.lhu.edu.vn/index.php/lachong

1. INTRODUCTION

In recent years, online social networks (OSNs) have become an integral part of daily life, gaining immense popularity as platforms for connecting, communicating, and sharing information. Within these networks, user behavior and interpersonal interactions are shaped by both virtual communication and the various artifacts of virtual activity, such as personal profile metrics. These metrics—encompassing elements like audio recordings, video content, avatars, friends, posts, reposts, and likes—serve as reflections of a user's real-world behavior, encapsulating characteristics of their online activities. As a result, these metrics can act as predictors, enabling researchers to infer real-life behaviors based on observed patterns in OSNs [1]. Thus, OSNs enable users to share content, connect with others, and stay informed, making them valuable for research across disciplines [2]. Machine learning and deep neural networks are increasingly applied to analyze OSN data [3]. Accurate analysis requires incorporating both user profile metrics and the structural relationships of OSNs, modeled as graphs with users as vertices and connections as edges. Graphs, widely used in various sciences, enable tasks like node classification, link prediction, and clustering.

Graph neural networks (GNNs), state-of-the-art tools for processing graph-structured data, use neighborhood aggregation to iteratively update node representations. This extends convolutional neural networks (CNNs) to graphs, forming graph convolutional neural networks (GCNNs), which effectively analyze OSN structures [4]. Moreover, GNNs are gaining prominence in social network analysis. For example, [5] used GNNs to predict social influence by integrating network structures and user data, while [6] developed a heterogeneous neural network to model social influence using event and user relationships.

OSNs are also key for misinformation detection and personalized news. In [7], a GNN tracked misleading posts, and [8] employed an attention graph convolutional network for personalized news recommendations based on user profiles. Additionally, OSNs offer rich user data—photos, opinions, interactions—enabling insights into behavior and lifestyle. In [9], a GNN-based model used local and global user features to predict social roles and statuses.

Recent work on graph-level representation learning for classification has advanced through multi-task GNN architectures [10], DP-SGD methods for privacy [11], and hypercomplex GNNs for efficiency [12]. Social network analysis has been applied to personality traits [13], academic success [14], [15], education effects [16], recruitment [17], and job recommendations [18]. Previously, we developed a model to predict professional success using VKontakte data [19].

User success prediction forecasts professional success using VKontakte profile data linked with HeadHunter classifications. Users were labeled as "successful" or "unsuccessful," and neural network models, including GNNs, were trained using profile features and social connections.

This study highlights GNNs' superiority over feed-forward neural networks for analyzing social networks, leveraging their ability to handle graph-structured data. Using VKontakte's API, local connectivity graphs were built as input, achieving a significant accuracy improvement from 0.77 to 0.88 [19].

The results demonstrate GNNs' effectiveness in classifying user profiles and their potential for efficient recruitment systems and advanced social network analysis methods.

2. METHODOLOGY

2.1 Data collection

Job seeker profiles from the HeadHunter platform were used to create a labeled dataset for professionally successful and unsuccessful individuals. Fifteen parameters were initially collected, but four key features—requested salary, employment duration, professional area, and job hunting region—were selected for labeling. Parameters were scaled to account for regional and professional variations, categorizing values into five groups (1–5), where 1 represented the least successful and 5 the most successful individuals. This study focused on the first (lowest success) and fifth (highest success) groups [19].

The VKontakte API (https://vk.com/dev/SDK) was utilized to collect user profile data for analysis, requiring authorization via the Implicit Flow method to obtain an access_token for secure access. Using Python's requests library, the users.get API method retrieved quantitative data for each user, including the number of friends, subscribers, interesting pages, audio recordings, and video recordings in Fig. 1 and Fig. 2. This approach provided an efficient way to gather necessary data for the study directly from VKontakte's database.

```python
import requests

response = requests.get(
    url="https://api.vk.com/method/users.get",
    params={
        "user_ids": "USER_ID",  # Replace with the desired user ID(s)
        "fields": "counters",   # Specify the fields to retrieve
        "access_token": "ACCESS_TOKEN",  # Replace with your access token
        "v": "API_VERSION"  # Replace with the API version (e.g., "5.131")
    }
)
```

Figure 1. *Example of VKontakte API-based users.get Request*

```python
import requests

response = requests.get(
    url="https://api.vk.com/method/friends.get",
    params={
        "user_id": "USER_ID",  # Replace with the specific user ID
        "count": "NUMBER_OF_FRIENDS",  # Specify the number of friends to retrieve
        "fields": "FIELDS",  # Specify additional fields to retrieve
        "access_token": "ACCESS_TOKEN",  # Replace with a valid access token
        "v": "API_VERSION"  # Specify the API version (e.g., "5.131")
    }
)
```

Figure 2. *Example of VKontakte API friends.get Request to Retrieve User's Friend List and Additional Details*

To build user graphs, friendship data (friends and friends of friends) were collected using the friends.get method of the VKontakte API, alongside numerical profile metrics.

HeadHunter data were used to label users as professionally successful or unsuccessful, and users were identified on VKontakte by name and date of birth. VKontakte data served as input features, while HeadHunter labels were used as target values for neural network training.

2.2 Graph Building

Each user's local network was modeled as an undirected graph $G = (V, E)$, where nodes V represent users and edges E represent friendships [20]. The graph structure was encoded in an adjacency matrix A, and each node was enriched with a feature vector Xi containing profile metrics retrieved via the VKontakte API. Graphs were constructed for each user in the dataset, with vertices representing the user, their friends, and friends-of-friends, and edges representing friendships. The NetworkX library (https://networkx.org/), a Python tool for creating and analyzing graphs, was used for graph construction. An example graph for four users is illustrated in Fig. 3.

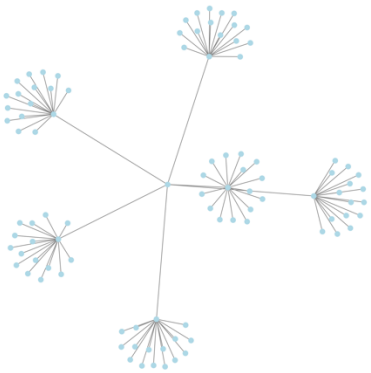

Figure 3. *Example Graph Visualizations of a VKontakte User's Friends Network, Centered Around the Analyzed User*

2.3 Graph Neural Network Architecture

Graph neural networks (GNNs) extend neural networks to handle graph-structured data by propagating node states iteratively until equilibrium is reached [21]. Convolutional propagation rules introduced [22] in further enabled graph-level classification, with later studies categorizing GNN components and outlining future research directions [23].

In this study, a GNN model with four graph convolutional layers and global mean pooling (GMP) was used to produce fixed-size graph representations. The GMP outputs were fed into a linear layer for classification, trained with a binary cross-entropy loss. Implementation utilized PyTorch Geometric, leveraging GPU-accelerated convolution layers such as GCNConv, SAGEConv, GraphConv, GATConv, TransformerConv, and GINConv. The model was trained end-to-end using gradient descent to aggregate node information and classify graphs efficiently.

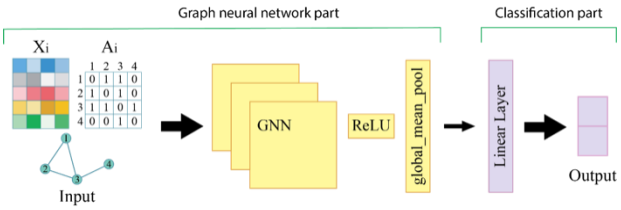

Figure 4. *General Architecture of a GNN-Based Classification Model*

To address the classification of social network users, each user is represented as an attributed graph $G_i = (X_i, A_i)$, where Xi contains numerical features derived from the user's profile and Ai is the adjacency matrix encoding connections with other users (see Fig. 4). Given a labeled dataset $D = \{Gi = (X_i, A_i), y_i\}$, where y_i represents the professional success label for each user, the goal is to train a classifier $g : (X_i, A_i) \rightarrow y_i$. that predicts the label for each user.

GCNConv Layer

GCNConv, introduced in [24], is a convolutional layer for graph neural networks that employs an efficient layer-wise propagation rule. This rule is derived from a first-order approximation of spectral convolutions on graphs and is expressed mathematically as:

$$H^{I+1} = \sigma\left(D^{\frac{1}{2}}A'D^{\frac{1}{2}}H^{l}W^{l}\right), \quad (1)$$

where $A' = A + I$ represents the adjacency matrix of an undirected graph G, enhanced with self-loops for each node. D is the degree matrix corresponding to A', with diagonal elements, W^l denotes the trainable weight matrix specific to layer, H^l indicates the activation matrix at layer l, containing feature representations of all nodes.

Each user is modeled as an attributed graph, defined by a node feature vector Xi and an adjacency matrix Ai. The input graph is processed through multiple graph convolutional layers, followed by a global mean pooling (GMP) layer to aggregate vertex features. The pooled representation is then fed into a traditional linear layer for binary classification with two output classes.

SAGEConv Layer

SAGEConv is a general inductive framework designed to efficiently generate node embeddings by leveraging node feature information, as introduced in [25]. The layer learns embeddings using the following formula:

$$x_i' = W_1 x_i + W_2 * \text{ mean}_{j \in N_i} x_j \quad (2)$$

where x_i is the feature vector of the i-th node, W_1 and W_2 are the trainable weight matrices, N_i denotes the set of nodes connected to node i, and $\text{mean}_{j \in N_i} x_j$ is the element-wise average of the feature vectors of the neighboring nodes.

GraphConv Layer

The GraphConv convolution layer, introduced in [26], is a generalization of graph neural networks based on the k-Weisfeiler-Lehman (k-WL) graph isomorphism test. The propagation rule for generating the representation of a node in the next layer is defined as follows:

$$x_i' = W_1 x_i + W_2 \sum_{j \in N(i)} e_{j,i} * x_j \quad (3)$$

where $e_{j,i}$ represents the weight of the edge between the source node j and the target node i, with a default value 1.

GATConv layer

GATConv, introduced in [27], is an attention-based architecture designed for node classification in graph-structured data. The attention-based architecture operates efficiently, since it is parallelizable and can be applied to graph nodes with different degrees by specifying arbitrary weights for the neighbors.

This layer computes hidden representations for each node by applying a self-attention mechanism over its neighbors, using the following formula:

$$x_i' = a_{i,i} W x_i + \sum_{j \in N(i)} a_{i,j} W x_j \quad (4)$$

where $a_{i,j}$ are the attention coefficients that define the importance of node j to node i, $a_{i,j} = \frac{\exp(\text{LeakyReLU}(a^T[Wx_i \| Wx_j]))}{\sum_{k \in N(i) \cup \{i\}} \exp(\text{LeakyReLU}(a^T[Wx_i \| x_k]))}$, in which, a is an attention mechanism that shows the importance of vertex j for vertex i, T denotes matrix transposition and $\|$ is a concatenation operation, LeakyReLU is a non-linear activation function.

TransformerConv layer

TransformerConv, introduced in [28], is a Unified Message Passing Model (UniMP) that integrates feature and label propagation using a Graph Transformer network. It takes feature embeddings and label embeddings as input for propagation. For a given layer, label propagation follows the rule:

$$x_i' = W_i x_i + \sum_{j \in N(i)} a_{i,j} W_2 x_j, \quad (5)$$

Here, $a_{i,j}$ is a multi-head attention coefficient, calculated as $a_{i,j} = \text{softmax}\left(\frac{(W_3 x_i)^T (W_4 x_j)}{\sqrt{d}}\right)$, where W_3, W_4 are trainable weight matrices, softmax(...) is a multivariate logistic function, d is the size of each attention head. To mitigate overfitting when propagating self-loop label information, the model employs a masked label prediction strategy. In this approach, a random subset of input labels is masked, and the network is trained to predict these labels, enhancing its generalization ability.

GINConv Layer

GINConv, based on the Graph Isomorphism Network (GIN) [29], is a simple neural architecture with discriminative power equivalent to the Weisfeiler-Lehman graph isomorphism test. The label propagation in a GINConv layer is defined by:

$$x_i' = MLP\left((1 + \epsilon) * x_i + \sum_{j \in N(i)} x_j\right) \quad (8)$$

where ϵ is a trainable parameter or fixed scalar, x_i is the -th node feature vector, MLP represents a multilayer perceptron applied to the aggregated features.

3. RESULTS AND DISCUSSION

Graph neural network models with various convolution layers were trained on a dataset of 5,446 records, split into 70% training (3,812 records) and 30% testing (1,634 records). Each experiment was repeated 10 times, and the average accuracy was used for evaluation. The balanced dataset ensured accuracy was a reliable performance metric.

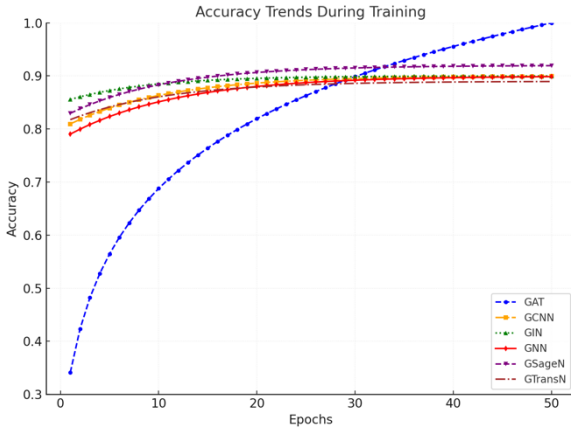

Figure 5. *Accuracy Analysis and Model Performance*

Models were trained for 50 epochs using the Adam optimizer (learning rate: 0.00001, batch size: 128). Most models, except the GAT-based model, converged quickly, with accuracy stabilizing after the 10th epoch, as shown in Fig. 5.

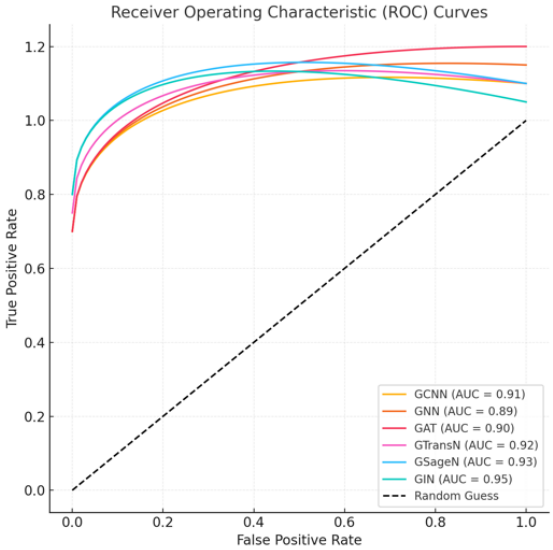

Figure 6. *Receiver Operating Characteristic curves of the trained models*

The evaluation of test accuracy across different convolutional layers indicated that the graph neural network with the GINConv layer achieved the best performance, with an accuracy of 89 percent, as shown in Figure 6. Experiments with varying numbers of layers, including one, two, three, and four layers, revealed that the optimal configuration for GINConv was three layers, with additional layers offering no further improvements.

The average training time per epoch was reported to estimate total training time easily. Additionally, the receiver operating characteristic curve for each model, presented in Figure 6, demonstrated that the GINConv-

based model achieved the highest area under the curve value of 0.93, confirming its superior classification capability compared to other models.

A confusion matrix was generated for the training set to visualize and evaluate the performance of the classification algorithm using the graph neural network with the GINConv layer, as shown in Table 1.

Table 1. *Confusion matrix for the graph neural network trained with GINConv convolution layers*

Predicted	**Actual**	**Actual**
TP	675	166
PP	36	757

In addition to the Accuracy metric, the confusion matrix was used to calculate Sensitivity, Specificity, Precision, and F1-score for the graph neural network with the GINConv convolution layer. These metrics provide a more comprehensive evaluation of the neural network's performance, as summarized in Table 2.

Table 2. *Evaluation Metrics for the Graph Neural Network Trained Using GINConv Convolution Layers*

Metrics	**Values**
Accuracy	0.88
Misclassification	0.12
Sensitivity	0.95
Specificity	0.8
Precision	0.8
F1-score	0.87

Graph neural networks (GNNs) achieved higher accuracy (0.89) compared to classical neural networks (0.77) from our previous study [30], as GNNs incorporate user connections within the VKontakte social graph. This inclusion significantly improved classification performance.

Future work will focus on the application of explainability methods for graph convolutional neural networks, as deep models often lack interpretability [31]. While explainability methods have achieved notable results in text and image analysis, their application to graph neural networks is an emerging field. Techniques such as GNNExplainer and XGNN [32] have been proposed to make GNNs more interpretable. These methods can help elucidate why the network produces specific outputs and identify key graph structures that influence the classification of social network users into different categories of social success.

4. CONCLUSION

This study investigated professional success prediction using graph neural networks (GNNs) with various architectures, including GCNConv, SAGEConv, GraphConv, GATConv, TransformerConv, and GINConv layers. The best accuracy of 88 percent was achieved with a three-layer GINConv-based GNN, outperforming feed-forward networks due to GNNs' ability to process graph-structured data. The findings support the development of psychometric models incorporating "Social Success" metrics, such as relationship quality, social influence, and leadership, to predict life activity and success.

5. REFERENCES

[1] Buettner R. Predicting user behavior in electronic markets based on personality-mining in large online social networks. *Electron. Mark* **2017**, 27(3), pp. 247–265.

[2] Martono G. H., et al. An extended approach of weight collective influence graph for detection influence actor *Int. J. Adv. Intell. Informatics* **2022**, 8(1), pp. 1–11, **2022**.

[3] Ulizulfa A., et al. Temperament detection based on Twitter data: classical machine learning versus deep learning. Int. J. Adv. Intell. Informatics, 8(1), pp. 45–57, **2022**.

[4] Alzubaidi L. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data **2021**, 8(1), pp. 1–74.

[5] Qiu J., et al. DeepInf: Social influence prediction with deep learning. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 2110–2119, **2018**.

[6] Gao L., et al., HetInf: Social Influence Prediction with Heterogeneous Graph Neural Network. Front. Phys., 9, p. 729, **2022**.

[7] Shang L., et al. FauxWard: a graph neural network approach to fauxtography detection using social media comments. Soc. Netw. Anal. Min., 10(1), pp. 1–16, **2020**.

[8] P. Zhu et al. SI-News: Integrating social information for news recommendation with attention-based graph convolutional network. Neurocomputing, 494, pp. 33–42, **2022**.

[9] C. Zhang, et al. Inferring Users' Social Roles with a Multi-Level Graph Neural Network Model. Entropy 2021, 23(11), p. 1453, **2021**.

[10] Y. Xie, et al. A Multi-Task Representation Learning Architecture for Enhanced Graph Classification. Front. Neurosci., 13, pp. 1395, **2020**.

[11] Mueller T. T. et al. Differentially Private Graph Classification with GNNs **2022**.

[12] T. Le, et al., "Parameterized Hypercomplex Graph Neural Networks for Graph Classification," Lect. Notes Comput. Sci, 12893, pp. 204–216, **2021**.

[13] D. Quercia, et al., "The personality of popular facebook users," Proc. ACM Conf. Comput. Support. Coop. Work. CSCW, pp. 955–964, **2012**.

[14] M. Saqr, U. Fors, and J. Nouri, "Using social network analysis to understand online Problem-Based Learning and predict performance," PLoS One, 13(9), p. e0203590, **2018**.

[15] M. Saqr and A. Alamro, "The role of social network analysis as a learning analytics tool in online problem based learning,"

BMC Med. Educ., 19(1), pp. 1–11, **2019**.

[16] A. M. Bhandarkar, A. K. Pandey, R. Nayak, K. Pujary, and A. Kumar, "Impact of social media on the academic performance of undergraduate medical students," Med. J. Armed Forces India, 77, pp. S37–S41, **2021**.

[17] K. Toteva and E. Gourova, "Social network analysis in professional e-recruitment," Adv. Intell. Soft Comput., 101, pp. 75–80, **2011**.

[18] S. Chala and M. Fathi, "Job seeker to vacancy matching using social network analysis," Proc. IEEE Int. Conf. Ind. Technol., pp. 1250–1255, **2017**.

[19] F. M. Gafarov, et al., "A Complex Neural Network Model for Predicting a Personal Success based on their Activity in Social Networks," Eurasia J. Math. Sci. Technol. Educ., 17(10), p. em2010, **2021**.

[20] E. Kay, J. A. Bondy, and U. S. R. Murty, "Graph Theory with Applications," Oper. Res. Q., 28(1), p. 237, **1977**.

[21] F. Scarselli, et al., "The graph neural network model," IEEE Trans. Neural Networks, 20(1), pp. 61–80, **2009**.

[22] J. Zhou et al., "Graph neural networks: A review of methods and applications," AI Open, 1, pp. 57–81, **2020**.

[23] Y. Zhou, et al., "Graph Neural Networks: Taxonomy, Advances, and Trends," ACM Trans. Intell. Syst. Technol., 13(1), pp. 1, **2022**.

[24] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, **2016**.

[25] N. K. Ahmed *et al.*, "Inductive Representation Learning in Large Attributed Graphs," in Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035, **2017**.

[26] C. Morris *et al.*, "Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks," *Proc. AAAI Conf. Artif. Intell.*, 33(1), pp. 4602–4609, **2019**.

[27] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph Attention Networks," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, **2017**.

[28] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification," *IJCAI Int. Jt. Conf. Artif. Intell.*, 2, pp. 1548–1554, **2021**.

[29] D. Bacciu, et al., "A gentle introduction to deep learning for graphs," *Neural Networks*, 129, pp. 203–221, **2020**.

[30] F. M. Gafarov, et al., "A Complex Neural Network Model for Predicting a Personal Success based on their Activity in Social Networks," *Eurasia J. Math. Sci. Technol. Educ.*, 17(10), p. em2010, **2021**.

[42] P. E. Pope, et al., "Explainability methods for graph convolutional neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10764–10773, **2019**.

[43] H. Yuan, et al. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Trans. Pattern Anal. Mach. Intell* **2020**.