НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2026 Том 26 № 2 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
March–April 2026 Vol. 26 No 2 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

doi: 10.17586/2226-1494-2026-26-2-331-336

Ego-net link prediction with GNN

Evgeny I. Zamyatin✉

ITMO University, Saint Petersburg, 197101, Russian Federation
ezamyatin@itmo.ru✉, https://orcid.org/0009-0002-7334-2856

Abstract
The task of link prediction is one of the key challenges in the field of social network analysis. The common way to build such systems is based on the idea of decomposing a task into two levels. At the first level, links within ego-nets are predicted; at the second, the results are aggregated to form the final predictions. The accuracy of such systems depends on the first-level model. Heuristic methods are usually used here. The focus of this work is on developing a new supervised model to improve the quality of link prediction within ego-nets. The heterogeneity of the edge attributes, the absence of node features, and the dynamic nature of ego-nets distinguish this task from others. The proposed method belongs to the class of graph neural networks. Its key feature is the ability to effectively consider the topology of the graph along with the attributes of the edges, without relying on the properties of the nodes. This effect is achieved by modeling the hidden state of node pairs, rather than the state of each node individually. The iterative nature of the model makes it possible to propagate knowledge about the relationships between nodes, increasing the complexity of the structures considered with each step. To measure the accuracy of the model, the Ego-VK dataset was used. This dataset consists of a set of ego-nets from a subsample of users of the VKontakte social network. The model is compared with the classical Adamic-Adar method as well as modern approaches based on graph neural networks. Experiments show that the proposed model is significantly superior to the baselines with respect to NDCG@5 ranking quality metric. The results demonstrate the high effectiveness of the proposed model, and the possibility of integration into distributed systems makes it widely applicable in the industry.

Keywords
information retrieval, recommender systems, graph neural networks, social networks analysis, ego-nets

Acknowledgements
The research was carried out under the Project No. 625135 “Development of software modules for effective training and inference of deep learning models”.

For citation: Zamyatin E.I. Ego-net link prediction with GNN. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2026, vol. 26, no. 2, pp. 331–336. doi: 10.17586/2226-1494-2026-26-2-331-336

УДК 004.896

Предсказание связей в эго-графах с GNN

Евгений Игоревич Замятин✉

Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
ezamyatin@itmo.ru✉, https://orcid.org/0009-0002-7334-2856

Аннотация
Введение. Задача предсказания связей в графе — одна из ключевых задач в области анализа социальных сетей. В основе одного из распространенных способов построения таких систем лежит идея декомпозиции задачи на два уровня. На первом уровне формируются предсказания возникновения связей внутри эго-графов, на втором — агрегация результатов и формирование итоговой выдачи. Точность таких систем определяется моделью первого уровня. Обычно здесь используют эвристические методы. Основное внимание в данной работе уделено разработке и исследованию новой модели с учителем для улучшения качества предсказаний связей внутри эго-графов. Неоднородность свойств ребер, отсутствие признаков вершин, а также динамическая природа эго-графов выделяют эту задачу среди остальных. **Метод.** Предлагаемый метод относится к классу графовых нейронных сетей. Его отличительная особенность в способности эффективно учитывать топологию

графа вместе со свойствами ребер, при этом не опираясь на признаки вершин. Такой эффект удается достичь за счет моделирования скрытого состояния именно пар вершин, а не каждой вершины в отдельности. Итеративная сущность модели позволяет распространять знание о взаимосвязях вершин, с каждым шагом увеличивая сложность учитываемых структур. **Основные результаты.** Для замеров эффективности модели была использована база данных Ego-VK, состоящая из набора эго-графов подвыборки пользователей социальной сети «ВКонтакте». Проведено сравнение с классическим методом предсказания связей Adamic-Adar, а также с современными подходами на основе графовых нейронных сетей. Эксперименты показали, что предлагаемая модель значительно превосходит бейзлайны с точки зрения метрики качества ранжирования NDCG@5. **Обсуждение.** Полученные результаты свидетельствуют о высокой эффективности предложенной модели, а возможность интеграции в декомпозированные системы делает ее широко применимой в индустрии.

Ключевые слова

информационный поиск, рекомендательные системы, графовые нейросети, анализ социальных графов, эго-графы

Благодарности

Работа выполнена в рамках проекта № 625135 «Разработка программных модулей для эффективного обучения и применения моделей глубокого обучения».

Ссылка для цитирования: Замятин Е.И. Предсказание связей в эго-графах с GNN // Научно-технический вестник информационных технологий, механики и оптики. 2026. Т. 26, № 2. С. 331–336 (на англ. яз.). doi: 10.17586/2226-1494-2026-26-2-331-336

Introduction

The link prediction task is one of the fundamental problems of graph analysis. It is widespread in the industry, especially in social networks. Many of their long-term metrics depend on the users' social activity, and recommender systems help support and promote this activity. The social graph of a modern service represents a heterogeneous continuous-time dynamic graph. Various connections between users appear and disappear, including friendships, messages, likes, profile visits and others. Many of them are characterized by timestamps and intensities. The task of building methods that can effectively consider all the available information about user relationships is very important, as the solutions quality can directly affect business performance.

One of the specifics of working with social networks is the lack of meaningful features that describe users. General attributes such as sex, age or interests are often available, but they are not sufficient to model a user's social dynamics. Some bioinformatics problems can be seen as examples of tasks with rich node features where units have natural descriptions that precisely define the principles of link formation. In the case of social networks, the structural information of the graph is of particular importance.

Despite active research [1, 2] on Graph Neural Networks (GNNs), heuristics-based approaches still play a significant role in the industry. Methods, such as Common Neighbors, Adamic-Adar [3], Preferential Attachment [3], etc., are widely used. Many recent papers refer to them as strong baselines [4–6], and the field of research aimed at finding ways to generalize them shows high potential [4, 5, 7]. Most of the heuristics do not rely on node features, which is an important property for the task of link prediction in social networks. The scalability, simplicity and efficiency of such methods make them attractive for industry use.

One of the most informative and convenient structures for analyzing local neighborhoods is the ego-net. The ego-net of a node is a subgraph containing all the neighbors of the node and the links between them. On the one hand, ego-nets provide a rich description of local neighborhoods, and their small size makes it possible to use complex methods for their analysis. On the other hand, ego-nets can still be processed within large graphs, as there is an efficient way [5, 8] to construct them using MapReduce. The maximum volume of all ego-nets within a graph is bound by $O(|E|^{1.5})$ [9], which is acceptable for industrial graphs. All the above makes ego-nets a potentially attractive structure for working with graphs that contain billions of nodes.

In this work, I address the problem of link predictions in ego-nets of the Friendship Score [5] approach the main principle of which is to reduce the problem of link prediction in a whole graph to a series of tasks within ego-nets and the aggregation of their results. The basic method for link prediction within ego-net is based on cluster analysis. Here I consider this task as a supervised graph-level link prediction problem in heterogeneous dynamic graphs without node features. This is yet another challenge for GNNs when it comes to inductively determining the laws of edge dynamics in graphs. The lack of meaningful node attributes in the social network domain makes the graph structure the only available source of information. The graph structure, on the other hand, is highly descriptive, as there are many possible connections between users in social networks.

The combination of these properties makes our task challenging and novel. First-order GNNs fail on this task because they strongly depend on the quality attributes of the nodes [10]. Higher-order GNNs are theoretically capable of solving the problem, but their application to heterogeneous graphs with rich attributes of edges is not sufficiently covered in the literature. To address this problem, I introduce the WalkGNN model — a second-order GNN which constructs representations for pairs of nodes and can efficiently account for different types of links between them and their numerical characteristics. Its key component is the WalkConv layer, which transforms each edge into an information filter and passes through it the corresponding state of the relationship of a pair of nodes.

Our research is based on data from VK[1], the largest social network in Russia with 100M monthly active users. To measure the models quality on our task, I use the Ego-

[1] Available at: https://vk.com/about (accessed: 17.02.2026).

VK[1] dataset, which represents the set of ego-nets of VK users. Each ego-net has at most 300 nodes and 4 possible types of connections between them. One of the edge types is the friendship's age, and the three other edges describe other types of relationships. The task is to predict new friendships in each ego-net.

Related works

Traditionally, the problem of link prediction in large-scale graphs is solved by one of two types of methods. One is based on local neighborhood analysis using heuristics, while the other is based on graph embeddings construction. Approaches that use neural networks, in most cases, turn out to be too complex to scale and are not directly applicable to large-scale graphs. However, given that in this work I reduce the large-scale problem to a series of low-scale tasks, the work related to building complex models for the link prediction problem is also relevant here.

Heuristic Methods

Despite the advanced age of most graph heuristics, they are still applicable due to their efficiency, simplicity and scalability. In contrast to most complex models, these methods do not require the presence of node features. The most popular among them are first- and second-order heuristics. Higher-order heuristics are much more difficult to scale on large graphs, so they are much less commonly used in the industry. Most used first-order heuristics are Common Neighbors and Preferential Attachment [3]. Adamic-Adar [3] and Resource Allocation [3] are examples of second-order heuristics.

Epasto et al. [5] proposed an original idea of analyzing local communities within ego-nets to improve the Common Neighbors heuristic. Their friendship score similarity measures the co-occurrences of two nodes in different ego-net clusters. This approach shows high accuracy and scales well to industrial graphs, but still relies on heuristic approaches. Taking the idea of link prediction using ego network analysis as a starting point, our work builds on it by exploring the use of supervised methods to obtain more accurate solutions.

Graph Neural Nets

Generally, complex transductive models do not scale well to large graphs due to the graphs huge number of parameters and nonlinear structure. In this paper, I reduce the problem of link prediction in a complete graph to the low-scale graph-level problems on ego-nets, which is why I can use GNNs.

At present, the most popular way of constructing GNNs is the message-passing framework [11]. Its main idea is to hierarchically build representations of nodes through convolutions of neighbors. The GraphSAGE [12], TGAT [13], GAT [14], and GCN [15] models are all different variations of Message Passing GNNs (MP-GNNs). They show good results on several problems but fail at tasks with no natural node attributes.

A lot of research is aimed at analyzing the expressive power of GNNs. Xu et al. [16] proved that many popular message-passing models have significant limitations when it comes to learning from graph structures. They proposed a Graph Isomorphism Network (GIN) model, the representational power of which is equal to the power of the 1-WL test [17].

To overcome the theoretical limitations on structure learning of GNNs, researchers are turning their attention to higher-order models. Moris et al. [18] proposed *k*-GNN, which associates each *k*-tuple of nodes with a vector representation. They showed that *k*-GNN is, at most, as powerful as the *k*-WL test. This property allows models to better understand the graph structure, but the computational complexity of such methods grows exponentially with *k*. Maron et al. [19] proposed PPGN, a 2-GNN based model which has a provable 3-WL expressive power. Zhu et al. [7] proposed the NBFNet framework for building such models, inspired by the generalized Bellman-Ford algorithm.

Although high-order GNNs show a high possibility of learning from topology, the question of their application to heterogeneous graphs is not well studied. In this work, I propose a new second-order WalkGNN model whose architecture allows us to efficiently account for different types of edges together with their numerical values.

Method

In this section, I will present a model I call WalkGNN, whose main task is to efficiently predict links within ego-nets. This model is inspired by the graph walk counting algorithm, and unlike most popular GNNs, builds representations of node pairs instead of nodes. Its key component is the novel neural network layer WalkConv which generates a state transition matrix for each edge and propagates the corresponding node pair states through it.

WalkGNN

The WalkGNN model is inspired by the walks counting algorithm. This algorithm is a simple example of a dynamic programming problem. To calculate the number of walks of length k between all pairs of nodes in a graph, we must first solve the same problem for length k–1 and multiply the result by the adjacency matrix: $\mathbf{W}_k = \mathbf{W}_{k-1} \times \mathbf{A}$, where $\mathbf{W}_k$ is a k-length walks count matrix; $\mathbf{A}$ is the adjacency matrix. The initial state is an identity matrix.

Instead of a scalar value of the walks count, we will store a vector of size d, which is an embedding that describes the relationship between a pair of nodes. To predict the probability of forming links, we can train a supervised classifier based on them. To construct such embeddings, we need to define an initial state and a transition function. We can use a diagonal matrix with constant values or features of nodes as the initial state. The transition function will be defined in the next subsection.

Algorithm 1. WalkConv

Input: current state matrix $\mathbf{S}_k$, edge list E
Output: new state matrix $\mathbf{S}_{k+1}$
$\mathbf{T}_k^{u,v} \leftarrow EdgeMLP_k(\mathbf{e}), (u, v, \mathbf{e}) \in E$

$$\mathbf{W}_{k+1}^{u,v,t} \leftarrow \frac{1}{d}\mathbf{S}_k^{u,q,c}\mathbf{T}_k^{q,v,c,t}$$

return $\mathbf{S}_k + MLP_k(\mathbf{W}_{k+1})$

[1] Available at: https://cups.online/ru/training/8/tasks/1445 (accessed: 17.02.2026).

WalkConv

Suppose we have relation representations for each pair of nodes u and v at the k-th iteration for a graph with n nodes. Given the state matrix $\mathbf{W}_k$ of dimensions $[n \times n \times d]$, we need to compute $\mathbf{W}_{k+1}$. A key principle of our architecture is that we treat each edge as an information filter through which we propagate the state. Remember that each edge is parameterized by different attributes, such as the age of the friendship, the number of messages, likes or profile visits, and so on. For each edge a vector of its attributes $\mathbf{e}$ is given. On top of it, we can build a multilayer perceptron (MLP), which would transform $\mathbf{e}$ into a d^2-dimensional vector. It can later be reshaped to a $[d \times d]$ dimensional matrix. The obtained matrix is the linear filter used for propagating the state. The transition formula is as follows:

$$\mathbf{W}_{k+1}^{u,v,t} = \frac{1}{d} \sum_{(t,v,\mathbf{e}) \in E} \left(\mathbf{W}_k^{u,t} \times EdgeMLP_k(\mathbf{e}) \right),$$

where d is the dimension of the state; E is a set of edges; $EdgeMLP_k$ is an MLP that takes the attributes of an edge as input and returns a matrix of dimension $[d \times d]$, *OutMLP* is a final MLP that transforms each state to a relevance prediction. This expression can also be represented in Einstein notation, as presented in Algorithm 1. The complexity of WalkConv is $O(n^3 \times d^2)$, where n is the number of nodes; d is the number of hidden units. This is the most computationally intensive part of our model, but it can be implemented efficiently as a dense multiplication of two matrices of dimensions $[n \times n \times d]$ and $[n \times n \times d \times d]$.

The WalkGNN model itself is a stack of WalkConv layers. To improve learning stability, I also use a residual connection after each layer (see Algorithm 1 and 2 for pseudocode).

Algorithm 2. WalkGNN

Input: edge list E, number of layers l
Output: pairwise relevance matrix
$\mathbf{S}_0^{u,v} \leftarrow \mathbf{I}_{u=v}$
for $k \leftarrow 1$ **to** l **do**
  $\mathbf{S}_k \leftarrow WalkConv(\mathbf{S}_{k-1}, E)$
end for
return $OutMLP(\mathbf{S}_l)$

Experiment

Dataset

I evaluate our model on two different datasets. The first dataset, Ego-VK, is a random sample of ego-nets from users of the VK social network for a certain date. The friendships formed on the next day are used as the ground truth edges; ego-nets with no new friendships during this period are ignored. Ground truth edges are undirected, i.e., any direction of the edge is considered as the right prediction. The base edges inside ego-nets are directed and can be of four types. Each type of edge has numerical characteristics describing the age of the friendship or three other kinds of activity. If there is no friendship between a pair of nodes, the age value is –1. It is guaranteed that there are no base edges of any type between pairs of nodes from the ground truth set in the original ego-net. The training, validation and test set are divided randomly. Natural node attributes are not used in this task. As attributes of nodes, I use attributes of edges with ego nodes in forward and backward directions. The edges with ego node at one end are omitted. In each ego-net, the ego node has id 0 and all other nodes are numbered with ordinal numbers, so all ego-nets are completely independent. All ego-nets are limited by 300 nodes by the activity of interactions with ego.

To evaluate the applicability of our model to graphs of other domains, I also measure its accuracy on the Yeast [20] dataset. This dataset is a set of small molecules, where each molecule is described as a graph with labeled nodes and edges. The nodes are described by one of 74 labels, and the edges are of three types. Originally, this dataset represents a graph-level classification task. To evaluate the quality of the link prediction task, I adapted the dataset accordingly. I removed one edge from each graph and assigned it as a target for the model. Note that graph labels are not used in our task. The statistics of both datasets are provided in Table 1.

Evaluation

The main metric used to evaluate the link-prediction accuracy is Normalized Discounted Cumulative Gain at 5 (NDCG@5) [21] which is the ratio of the cumulative gain of the top five predictions to the cumulative gain of the ideal five predictions. In each ego-net, the 5 pairs of nodes with best scores, other than those already present, are used to score the model. The hyperparameters of the models were selected to maximize accuracy on the validation dataset. Accuracies of the models on the test sample together with 95 % confidence intervals are presented in Table 2.

Baselines

I compare WalkGNN[1] against classical heuristics and the most popular GNNs that are known to be able to learn from the graph structure. As a heuristics baseline, I use Adamic-Adar and its weighted version. I use the state-of-the-art GIN model and its GINE modification as the baselines which are known in the literature as the most expressive MP-GNN. Xu et al. [16] and Errica et al. [10] argue that using node degree as a feature with GIN improves model accuracy. Other first-order models, such as GCN, GraphSage, and GAT, failed on the task. I use PPGN [19] as a high-order GNN baseline which has

[1]Available at: https://github.com/ezamyatin/walk_gnn (accessed: 17.02.2026).

Table 1. Datasets statistics

Dataset	Graphs, units	Avg. Nodes	Avg. Edges	Avg. Labels
Ego-VK	61,808	210.4	2,916.6	1.5
Yeast	79,601	21.5	22.8	1.0

Table 2. NDCG@5 comparison on test set

Algorithm	Ego-VK-No-Attr	Ego-VK	Yeast
Adamic-Adar	0.028 ± 0.003	0.055 ± 0.004	0.0032 ± 0.0007
GIN	0.015 ± 0.003	0.037 ± 0.003	0.4055 ± 0.0006
GIN-Deg	0.015 ± 0.003	0.037 ± 0.003	0.4073 ± 0.0007
PPGN	0.034 ± 0.003	0.075 ± 0.004	0.6423 ± 0.0006
WalkGNN	0.035 ± 0.003	0.090 ± 0.004	0.7201 ± 0.0006

strong theoretical expressive power, high accuracy in link prediction problems, a simple and clear architecture, and the ability to incorporate edge attributes. Higher-order GNNs are computationally expensive and are not suitable for our problem. To correctly compare expressive power, I consider a setting without edge attributes as well.

Results

The results of the model evaluation on the test set are presented in Table 2. WalkGNN outperforms the next-best solution by 20 % and 3 % in the setting with and without edge attributes on the Ego-VK dataset. It is worth noting that Adamic-Adar is still a strong baseline in the social networks domain and outperforms first-order GNN in terms of accuracy, which is explained by the lack of dependence on the node features. The 156 % gap between WalkGNN versions with and without edge attributes shows the model high degree of attention to graph structure and edge properties. On featureless tasks, the superiority of WalkGNN over Adamic-Adar, which is widely used in social networks, demonstrates the model ability to efficiently learn the natural laws of edge formation. The evaluation results on the Yeast dataset also show the superiority of our model over baselines. The low result of the Adamic-Adar heuristic shows that the laws of link formation in the social domain are inapplicable to chemical bond analysis. Despite this, WalkGNN performs well and significantly outperforms other baseline models in terms of accuracy.

Conclusion

In this paper, the graph-level problem of predicting friendships within ego-nets is addressed. The WalkGNN model was developed that works effectively within heterogeneous temporal graphs without node attributes, as demonstrated by our offline experiments.

To measure the quality metrics relevant to our task, the Ego-VK dataset was used which represents a supervised graph-level link-prediction task. Believe that our dataset could be useful for further research addressing the expressive power of graph models.

An interesting direction for future work is to investigate more complex local relevance aggregation methods that can be extended from scalar values to multidimensional vectors. Another option is to develop a way to efficiently incorporate node attributes into the WalkGNN architecture.

References

1. Keramatfar A., Rafiee M., Amirkhani H. Graph Neural Networks: a bibliometrics overview. *Machine Learning with Applications*, 2022, vol. 10, pp. 100401. https://doi.org/10.1016/j.mlwa.2022.100401
2. Abadal S., Jain A., Guirado R., López-Alonso J., Alarcón E. Computing Graph Neural Networks: a survey from algorithms to accelerators. *ACM Computing Surveys (CSUR)*, 2021, vol. 54, no. 9, pp. 1–38. https://doi.org/10.1145/3477141
3. Ragunathan K., Selvarajah K., Kobti, Z. Link prediction by analyzing common neighbors based subgraphs using convolutional neural network. *Frontiers in Artificial Intelligence and Applications*, 2020, vol. 325, pp. 1906-1913. https://doi.org/10.3233/faia200308
4. Zhang M., Chen Y. Link prediction based on graph neural networks. *Proc. of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 5171–5181.
5. Epasto A., Lattanzi S., Mirrokni V., Sebe I.O., Taei A., Verma S. Ego-net community mining applied to friend suggestion. *Proceedings of the VLDB Endowment*, 2015, vol. 9, no. 4, pp. 324–335. https://doi.org/10.14778/2856318.2856327
6. Perozzi B., Al-Rfou R., Skiena S. Deepwalk: online learning of social representations. *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710. https://doi.org/10.1145/2623330.2623732
7. Zhu Z., Zhang Z., Xhonneux L.-P., Tang J. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Proc. of the 35th International Conference on Neural Information Processing*, 2021, pp. 29476–29490.
8. Suri S., Vassilvitskii S. Counting triangles and the curse of the last reducer. *Proc. of the 20th International Conference on World Wide Web*, 2011, pp. 607–614. https://doi.org/10.1145/1963405.1963491

Литература

1. Keramatfar A., Rafiee M., Amirkhani H. Graph Neural Networks: a bibliometrics overview // Machine Learning with Applications. 2022. V. 10. P. 100401. https://doi.org/10.1016/j.mlwa.2022.100401
2. Abadal S., Jain A., Guirado R., López-Alonso J., Alarcón E. Computing Graph Neural Networks: a survey from algorithms to accelerators // ACM Computing Surveys (CSUR). 2021. V. 54. N 9. P. 1–38. https://doi.org/10.1145/3477141
3. Ragunathan K., Selvarajah K., Kobti, Z. Link prediction by analyzing common neighbors based subgraphs using convolutional neural network // Frontiers in Artificial Intelligence and Applications. 2020. V. 325. P. 1906-1913. https://doi.org/10.3233/faia200308
4. Zhang M., Chen Y. Link prediction based on graph neural networks // Proc. of the 32nd International Conference on Neural Information Processing Systems. 2018. P. 5171–5181.
5. Epasto A., Lattanzi S., Mirrokni V., Sebe I.O., Taei A., Verma S. Ego-net community mining applied to friend suggestion // Proceedings of the VLDB Endowment. 2015. V. 9. N 4. P. 324–335. https://doi.org/10.14778/2856318.2856327
6. Perozzi B., Al-Rfou R., Skiena S. Deepwalk: online learning of social representations // Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014. P. 701–710. https://doi.org/10.1145/2623330.2623732
7. Zhu Z., Zhang Z., Xhonneux L.-P., Tang J. Neural bellman-ford networks: A general graph neural network framework for link prediction // Proc. of the 35th International Conference on Neural Information Processing. 2021. P. 29476–29490.
8. Suri S., Vassilvitskii S. Counting triangles and the curse of the last reducer // Proc. of the 20th International Conference on World Wide Web. 2011. P. 607–614. https://doi.org/10.1145/1963405.1963491

9. Schank T. *Algorithmic Aspects of Triangle-Based Network Analysis*: Ph.D. dissertation. Universitat Karlsruhe, Fakultat fur Informatik, 2007.
10. Errica F., Podda M., Bacciu D., Micheli A. A fair comparison of graph neural networks for graph classification. *Proc. of the 8th International Conference on Learning Representations*, 2020.
11. Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry. *Proc. of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 1263–1272.
12. Hamilton W.L., Ying R., Leskovec J. Inductive representation learning on large graphs. *Proc. of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.
13. Xu D., Ruan Ch., Körpeoglu E., Kumar S., Achan K. Inductive representation learning on temporal graphs. *Proc. of the 8th International Conference on Learning Representations*, 2020.
14. Velickovic P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y. Graph attention networks. *Proc. of the 6th International Conference on Learning Representations*, 2018.
15. Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks. *Proc. of the 5th International Conference on Learning Representations*, 2017.
16. Xu K., Hu W., Leskovec J., Jegelka S. How powerful are graph neural networks? *Proc. of the 7th International Conference on Learning Representations*, 2019.
17. Leman A.A., Weisfeiler B. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia. Seriya 2*, 1968, no. 9, pp. 12–16.
18. Morris C., Ritzert M., Fey M., Hamilton W.L., Lenssen J.E., Rattan G., Grohe M. Weisfeiler and Leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 1. pp. 4602–4609. https://doi.org/10.1609/aaai.v33i01.33014602
19. Maron H., Ben-Hamu H., Serviansky H., Lipman Y. Provably powerful graph networks. *Proc. of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 2156–2167.
20. Morris C., Kriege N.M., Bause F., Kersting K., Mutzel P., Neumann M. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv*, 2020. arXiv:2007.08663. https://doi.org/10.48550/arXiv.2007.08663
21. Järvelin K., Kekäläinen J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 2002, vol. 20, no. 4, pp. 422–446. https://doi.org/10.1145/582415.582418

9. Schank T. Algorithmic Aspects of Triangle-Based Network Analysis. Ph.D. dissertation // Universitat Karlsruhe, Fakultat fur Informatik, 2007.
10. Errica F., Podda M., Bacciu D., Micheli A. A fair comparison of graph neural networks for graph classification // Proc. of the 8th International Conference on Learning Representations. 2020.
11. Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry // Proc. of the 34th International Conference on Machine Learning. 2017. V. 70. P. 1263–1272.
12. Hamilton W.L., Ying R., Leskovec J. Inductive representation learning on large graphs // Proc. of the 31st International Conference on Neural Information Processing Systems. 2017. P. 1025–1035.
13. Xu D., Ruan Ch., Körpeoglu E., Kumar S., Achan K. Inductive representation learning on temporal graphs // Proc. of the 8th International Conference on Learning Representations. 2020.
14. Velickovic P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y. Graph attention networks // Proc. of the 6th International Conference on Learning Representations. 2018.
15. Kipf T.N., Welling M. Semi-supervised classification with graph convolutional networks // Proc. of the 5th International Conference on Learning Representations. 2017.
16. Xu K., Hu W., Leskovec J., Jegelka S. How powerful are graph neural networks? // Proc. of the 7th International Conference on Learning Representations. 2019.
17. Leman A.A., Weisfeiler B. A reduction of a graph to a canonical form and an algebra arising during this reduction // Nauchno-Technicheskaya Informatsia. Seriya 2. 1968. N 9. P. 12–16.
18. Morris C., Ritzert M., Fey M., Hamilton W.L., Lenssen J.E., Rattan G., Grohe M. Weisfeiler and Leman go neural: Higher-order graph neural networks // Proceedings of the AAAI Conference on Artificial Intelligence. 2019. V. 33. N 1. P. 4602–4609. https://doi.org/10.1609/aaai.v33i01.33014602
19. Maron H., Ben-Hamu H., Serviansky H., Lipman Y. Provably powerful graph networks // Proc. of the 33rd International Conference on Neural Information Processing Systems. 2019. P. 2156–2167.
20. Morris C., Kriege N.M., Bause F., Kersting K., Mutzel P., Neumann M. Tudataset: A collection of benchmark datasets for learning with graphs // arXiv. 2020. arXiv:2007.08663. https://doi.org/10.48550/arXiv.2007.08663
21. Järvelin K., Kekäläinen J. Cumulated gain-based evaluation of IR techniques // ACM Transactions on Information Systems. 2002. V. 20. N 4. P. 422–446. https://doi.org/10.1145/582415.582418

Author

Evgeny I. Zamyatin — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, SC 57207471477, https://orcid.org/0009-0002-7334-2856, ezamyatin@itmo.ru

Автор

Замятин Евгений Игоревич — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, SC 57207471477, https://orcid.org/0009-0002-7334-2856, ezamyatin@itmo.ru

Received 25.09.2025
Approved after reviewing 15.01.2026
Accepted 21.03.2026

Статья поступила в редакцию 25.09.2025
Одобрена после рецензирования 15.01.2026
Принята к печати 21.03.2026