Записки научных
семинаров ПОМИ
Том 530, 2023 г.

**D. Shaikhelislamov, K. Lukyanov, N. Severin,
M. Drobyshevskiy, I. Makarov, D. Turdakov**

A STUDY OF GRAPH NEURAL NETWORKS FOR LINK PREDICTION ON VULNERABILITY TO MEMBERSHIP ATTACKS

Abstract. Graph neural networks (GNNs) have shown great promise in a variety of tasks involving graph data, including recommendation systems. However, as GNNs become more widely adopted in practical applications, concerns have arisen about their vulnerability to adversarial attacks. These attacks can lead to biased recommendations, potentially causing economic losses and safety risks. In this work, we consider an industrial application of recommendation systems for transport logistics and study their vulnerability to membership inference attacks. The dataset represents real train flows in Russia, published in the ETIS project. Experiments with three popular GNN architectures show that all of them can be successfully attacked even when the adversary has minimal background knowledge. Specifically, an attacker with access to only 1-2% of the actual data can successfully train their own GNN model to infer the membership of a shipper-consignee association in the training set with an accuracy over 94%. Our study also confirms that overfitting is the primary factor that influences the attack performance of recommendation systems.

§1. Introduction

The explosion of digital data and large amount of available information have made recommendation systems into an essential tool for many applications in different areas, such as e-commerce, social media, transport logistics industry, etc.

One of the major challenges in building recommendation systems is modeling the complex relationships between users and items based on the history of their interactions. Traditional collaborative filtering methods rely on matrix factorization techniques [11, 22], which can only capture linear relationships between these entities. However, in many real-world

Key words and phrases: membership inference attacks, recommendation systems, graph neural networks.

scenarios such relationships are non-linear, requiring more powerful models.

Recently, graph neural networks (GNNs) have emerged as a promising approach for tackling this problem. Within this approach, the data about historical interactions is represented as a graph, across which GNNs can learn to propagate information. However, as the practical applications of GNNs in recommendation systems become more widespread, study of their resistance to adversarial attacks is becoming increasingly important.

Poisoning attacks and membership attacks are two common types of adversarial attacks that can be launched against graph neural network (GNN) models [3]. In a poisoning attack, the attacker seeks to influence the training data used to train the GNN model. The goal of the attacker is to influence the model's behavior and cause it to make biased recommendations. For example, an attacker might add fake shipping records that exaggerate the performance of a particular shipper to bias the model towards recommending that shipper more frequently.

On the other hand, in a membership attack (MI), the attacker aims to infer whether a particular link, a shipping record in our case, was used in the training data for the GNN model. By doing so, an attacker might be able to infer sensitive information about a shipping company's customer base by determining whether certain shipping records were included in the training data. Also, the attacker can gain valuable insights into the model's training data and potentially use this information to launch more targeted attacks.

Both of these attacks can have serious consequences for the performance and security of GNN-based recommendation systems in the logistics industry. In this work, we focus on MI since they are underrepresented in literature.

Our main contributions are the following.

(1) We consider transportation logistics data as a directed graph between shippers and consignees and suggest a GNN-based recommendation system on it.
(2) We adopt MI attacks on GNNs to the link prediction task.
(3) We conduct a series of experiments to study the vulnerability of recommendation models to several MI attacks and discuss the implications.

The rest of the paper is organized as follows. We review related work on attacks on GNNs and recommendation systems in Section 2. Section 3 gives

the details about the dataset, models, and attack algorithms. In Section 4 we describe and discuss our experiments. Finally, Section 5 contains the conclusion.

§2. Related work

Recently, graph neural networks (GNNs) have become an increasingly popular approach for solving a variety of graph-based machine learning problems, including recommendation systems [18]. When applying GNNs, the recommendation task is usually defined as link weight regression [16], missing [17, 19] or future [20] link prediction problems.

In the context of transportation networks, GNNs have been used for various applications such as traffic flow prediction [4], route recommendation [9], and anomaly detection [25]. Early works in this area are focused on using traditional machine learning algorithms, such as logistic regression and random forests [12]. However, these methods are limited in their ability to capture the complex relationships between nodes and edges in transportation networks. More recently, deep learning-based methods, including GNNs, have been proposed for link prediction in networks [21, 29].

However, as the use of GNNs becomes more widespread, concerns have been raised about their vulnerability to attacks. Among the attacks, several different groups of attacks can be distinguished: poisoning, evasion, and membership inference. The articles [14, 15, 31, 34], and [35] show how the quality of the prediction of a particular class or the overall quality of the GNN can be lowered. The articles [28, 32] show how evasion attacks can be implemented on GNNs.

In [2] and [1], the link prediction adversarial attack problem was first defined, and an iterative gradient attack algorithm was suggested. The authors showed that graph autoencoders, as well as other considered deep learning models, were vulnerable to such attacks. In the work [13], a perturbation-based attack is suggested against the GNN model of link prediction. Their experiments show that the performance of the GNN can be substantially decreased.

In the work [8], a taxonomy is given to categorize all the papers of membership inference attacks (MI). They summarized most existing evaluation metrics, datasets, and open-source implementations of popular approaches.

The survey [33] summarizes current advancements and trends of trustworthy GNNs. The authors define trustworthy GNNs and compare different trustworthy GNNs from the aspects of robustness, explainability, privacy, fairness, accountability, and environmental well-being.

To the best of our knowledge, there are only a few studies that have investigated the use of GNNs for vulnerability analysis of transportation networks. In [7], the authors perform a comprehensive privacy risk assessment of GNNs through the lens of node-level membership inference attacks. They systematically defined the threat model along three dimensions, including shadow dataset, shadow model, and node topology, and proposed three different attack models. In [30], the vulnerability of GNNs to MI is investigated, and training-based and threshold-based attacks against various target GNN models are developed. Their results show that GNNs are indeed vulnerable to membership inference attacks, even with minimal background knowledge of an adversary, and overfitting is still the most significant factor that affects the attack performance.

§3. Dataset preparation, link prediction models, and attack methods

In this section we describe the dataset preparation process, three GNN models for link prediction, and several MI algorithms to attack the models.

3.1. Transportation data. The dataset represents an update of the train flows published in the European Transport Policy Information System (ETIS) project [24] and includes detailed information on the goods transported to or from Russia during the period from January 1 to January 5, 2012.

To analyze the dataset, a directed graph was formed between shippers and consignees, where the following attributes were used:

- Departure date: this attribute indicates the date when transportation began and was used to divide the dataset into a training set (January 1 to January 3) and a validation and testing set (January 4 and 5, respectively);
- Shipper and Departure station: these attributes were encoded in a numeric format and were used to determine the initial node of the directed graph;

- Consignee and Destination station: these attributes were also encoded in a numeric format and were used to define the final node of the directed graph;
- Departure and destination roads: these are disjoint groups of train directions that combine several stations in different regions; there are 17 groups in total;
- Subject of the Russian Federation of departure and destination.

The resulting graph represents the flow of trains between shippers and consignees in Russia during the specified time period. A directed edge is drawn between the nodes of the intersecting sets W and V, where W consists of nodes that correspond to pairs of shippers and stations, and V consists of pairs consignees and stations. To capture the possibility of a shipper appearing at different stations, we include a station as part of the node definition. It is worth noting that a consignee can also act as a shipper at the same stations, so shippers and consignees can be from the same set. This distinguishes the task from classic recommendation system tasks, where there are separate sets of users and items.

The graph is directed, and the edges are labeled with attributes such as roads, and the subject of the Russian Federation for both the departure and destination. To encode these attributes, we use a one-hot encoding scheme [6].

A giant connected component is extracted from the graph. The original graph is directed and comprised of 13 420 nodes and 16 734 edges with average degree 2.49.

3.2. Algorithms and the quality metric. We treat the shipping recommendation problem as a link prediction task between shippers and consignees based on known shipping records. Since the interaction graph under consideration is not bipartite, unlike the classical case, three of the most popular general-purpose GNNs were selected for analysis: GCN [10], GraphSAGE [5], and GAT [27].

GCN is a method that constructs node embeddings based on their local neighborhood. GCN is based on graph convolutions built by stacking multiple convolutional layers. Every layer starts off with a shared node-wise feature transformation (in order to achieve a higher-level representation), specified by a weight matrix. In order to construct neighbourhood of each node, GCN operates with full graph adjacency matrix at each layer.

On the other hand, GraphSAGE generalizes neighborhood aggregation, so that it samples only a subset of neighboring nodes at different depth

layers. On each layer, it aggregates the neighbors of the previous layers using an aggregator. Each aggregator function aggregates information from a different number of hops, or search depth, away from a given node. As a result, nodes incrementally gain more and more information about graph local structure.

Inspired by successes in natural language processing, GAT leverages masked self-attentional layers over the node features to define the importance of neighbours. Concatenation of output of several different heads enables method to specify different importances to different neighbours.

In all our experiments for link prediction task the outputs of GNNs were transmitted to 2-layered multilayer perceptron (MLP). Binary cross-entropy loss was used to optimize parameters of the models.

The models were evaluated in the inductive setting (predictions for a node unseen in the training phase) via the AUC-ROC quality metric measuring the performance for the classification of whether an edge between two nodes exists.

3.3. Attack algorithms. In this work we consider membership inference attacks. Most of such attacks are quite simple to implement. Moreover, this group of attacks can be applied even to the black box model in the absence of data leakage. The only requirement is that the model returns the degrees of confidence in its predictions.

The membership inference attack problem can be thought of as a binary classification problem. In the case of recommendation system, the task of the attack algorithm is to classify whether there was an edge between two given nodes in the training group.

A naive algorithm needs to make some requests to the target model. Next, it looks at the distribution of forecast confidence levels. And if it is possible to easily divide all forecasts into two groups, then a threshold is chosen according to which the forecasts are divided into two groups. Otherwise, one can either use the standard assumption of an 80-by-20 split, or assume that this information got into external sources and take an exact split boundary.

In a more complex attack scenario, a shadow model needs to be trained. As has been shown in the work [23], the 2-layer GCN model can be effectively used for this purpose. The general algorithm consists of the following steps (see Fig. 1).

(1) Imitate data leakage by excising a fraction of the dataset. The target model is trained on the rest of it.

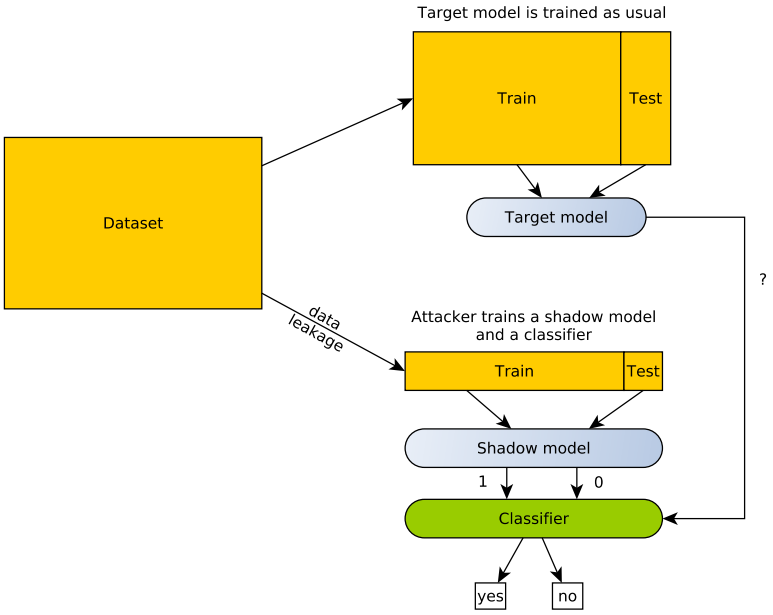

Figure 1. Shadow MI attack scheme.

(2) Build a shadow model based on GCN.
(3) Divide the part of the dataset into training and test set for the shadow model. Create the appropriate markup indicating which edges are included in the training sample, and which are not.
(4) Train the shadow model.
(5) Preserve the shadow model confidence levels for all edges in dataset.
(6) Train a classifier on shadow model confidence data.
(7) Attack the real model using the trained classifier.

§4. Experiments

Now we provide experimental results with two attack strategies, threshold MI and shadow MI, on our recommendation system models.

Throughout the experiments we used by default a two-layer GCN model trained on 5 epochs by default. Experiments were also carried out with

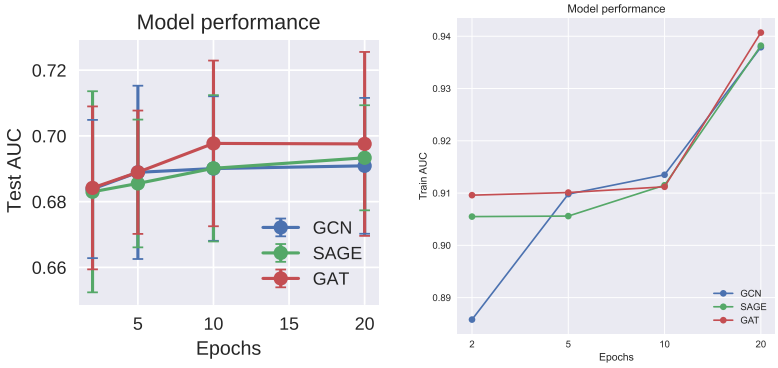

Figure 2. Performance (AUC-ROC) of 3 convolution types on the link prediction task is shown on the test dataset (left) and the training dataset (right).

other types of layers, but they did not show significant differences. 75% of edges were used for training, 5% for validation, and 20% for testing, splitted in chronological order. All the reported values were obtained by averaging over 20 runs.

SVC with an exponential kernel was used as the attacker's classifier.

4.1. Threshold MI attack. In the threshold MI experiments, test and train data were mixed, then all edges were evaluated by the target model. The attacker then predicts the top 75% of edges as train. To assess the attack quality, among the selected 75% of the data, it was estimated how many edges were actually in the training part — these are correct attack answers, and the rest are incorrect ones.

We studied the dependence of the attack quality on the number of training epochs of the target model and on the architecture of the model. As it can be seen from Fig. 2, the quality of the model on the training set continues to improve after 10 epochs, but there is no change in the quality on the test set. This indicates that the model is overfitting after 10 epochs.

Figure 3, with increasing training epochs and increasing model overfitting, the quality of the threshold MI attack increases. It is also worth noting that in spite of the fact that the quality of the model increases

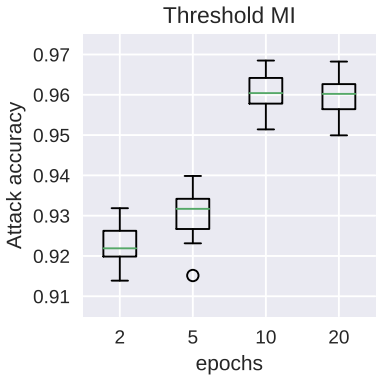

Figure 3. Threshold MI attack accuracy depending on the number of epochs the target model (GCN) was trained.

slightly with an increase in the number of epochs from 5 to 10, the quality of the attack increases significantly. That is, the model begins to score, on average, the edges from the training set higher than test edges, which indicates the overfitting.

Figure 4 shows that all three standard convolutions are equally susceptible to the attack and the architecture does not seem to correlate in any way with the attack accuracy.

4.2. Shadow MI attack. For the shadow MI attack, we used as a default shadow model a two-layer GCN trained on 5 epochs. The shadow dataset did not overlap with the data on which the target model was trained. The default shadow dataset has been set to 25% of the total data. Splitting the shadow dataset into train and test sets was the same to splitting the data when training the target model. After training the shadow model and the classifier, they were applied to dataset used with the target model. Scores were calculated for all the edges in this data, then fed to the input of the classifier. Finally, the quality was evaluated by the accuracy metric.

We studied the dependence of the quality of the attack on the number of training epochs of the target model, the architecture of the shadow and target models, and the amount of dataset leakage. As the results in Figures 5, 6, and 7 show, in all cases the variation of these parameters does

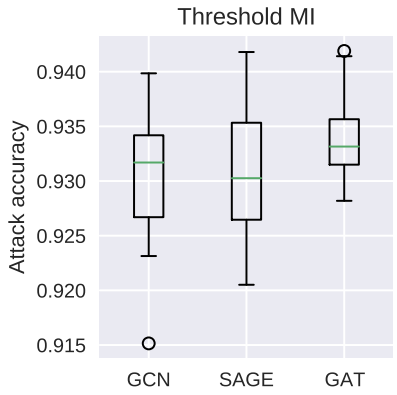

Figure 4. Threshold MI attack accuracy depending on the target model architecture.

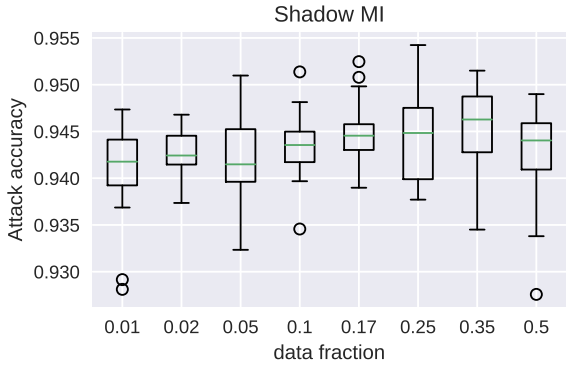

Figure 5. Shadow MI attack accuracy depending on shadow data fraction used for the attack.

not give significant changes in the quality of the attack, and the quality remains extremely high and shows a significant vulnerability of the models to this group of attacks.

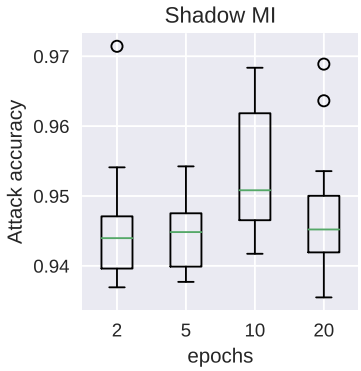

Figure 6. Shadow MI attacks accuracy depending on the number of epochs the target model was trained.

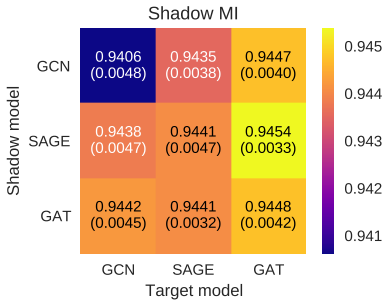

Figure 7. Shadow MI attack accuracy depending on shadow data fraction used for the attack.

An important point in evaluating the accuracy of attacks is that the quality of the attack depends on splitting the sample into train, validation, and test. Since in our case 75% of the dataset is the training set, then it is worth evaluating how much the metric value is more than 75 percent, in accordance with the splitting of the dataset.

It is also worth noting that one should not directly compare the quality of the attacks considered by MI, as they involve different use cases and make different assumptions about data leakage.

§5. Conclusion

Transportation logistics companies often collect a significant amount of data about their customers, such as delivery addresses and shipment details. However, this data can be sensitive and could reveal personal information about the clients. An attacker who gains access to such models

can use membership inference attacks to determine whether a specific individual's data was included in the training dataset, even if the model does not explicitly reveal this information [8, 26].

In this work, we have considered an industrial application of GNNs to recommendation systems in transportation logistics and analysed their susceptibility to membership inference attacks. Experiments with three GNN architectures shown that all of them can be successfully attacked even by a simple threshold-based classifier. Accuracy of training elements restoration increases as a target model gets overfitted. In our link prediction setting for transaction graph, the recommended number of epochs would be less than 5 to decrease the risks. In this way, a threshold MI attack could be used as an alternative method for overfitting detection.

Another risk is the leakage of training data which can lead to an attack with a shadow model. We showed that having 1-2% of the actual data, an attacker can successfully train their own GNN model to induce the membership of a shipper-consignee association in the training set with accuracy over 94%.

As a result, in order to preserve privacy the data owner should watch out for even small data leaks, while a machine learning specialist should be careful in choosing the model training steps.

In future works, our research will be focused on the study and development of effective methods for producing trustworthy machine learning models that are resilient against membership inference attacks.

Acknowledgments

The work of Ilya Makarov on Section 2 was prepared in the framework of the strategic project "Digital Business" within the Strategic Academic Leadership Program "Priority 2030" at NUST MISiS.

References

1. J. Chen, X. Lin, Z. Shi, and Y. Liu, *Link prediction adversarial attack via iterative gradient attack*, IEEE Transactions on Computational Social Systems **7** (2020), no. 4, 1081–1094.
2. J. Chen, Z. Shi, Y. Wu, X. Xu, and H. Zheng, *Link prediction adversarial attack*, arXiv preprint arXiv:1810.01110 (2018).
3. H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, *Adversarial attack on graph structured data*, International conference on machine learning, PMLR, 2018, pp. 1115–1124.

4. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, *Attention based spatial-temporal graph convolutional networks for traffic flow forecasting*, Proceedings of the AAAI conference on artificial intelligence, vol. 33, 2019, pp. 922–929.
5. L. Hamilton, W., R. Ying, and J. Leskovec, *Inductive representation learning on large graphs*, Proceedings of International Conference on NIPS (Red Hook, NY, USA), Curran Associates Inc., 2017, pp. 1025–1035.
6. D. Harris and S. L. Harris, *Digital design and computer architecture*, Morgan Kaufmann, 2010.
7. X. He, R. Wen, Y. Wu, M. Backes, Y. Shen, and Y. Zhang, *Node-level membership inference attacks against graph neural networks*, arXiv preprint arXiv:2102.05429 (2021).
8. H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, *Membership inference attacks on machine learning: A survey*, ACM Computing Surveys (CSUR) **54** (2022), no. 11s, 1–37.
9. L. Huang, Y. Ma, S. Wang, and Y. Liu, *An attention-based spatiotemporal lstm network for next poi recommendation*, IEEE Transactions on Services Computing **14** (2019), no. 6, 1585–1597.
10. T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, arXiv preprint arXiv:1609.02907 (2016).
11. M. Kula, *Metadata embeddings for user and item cold-start recommendations*, arXiv preprint arXiv:1507.08439 (2015).
12. T. J. Lakshmi and S. D. Bhavani, *Link prediction measures in various types of information networks: a review*, 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2018, pp. 1160–1167.
13. W. Lin, S. Ji, and B. Li, *Adversarial attacks on link prediction algorithms based on graph neural networks*, Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, 2020, pp. 370–380.
14. G. Liu, X. Huang, and X. Yi, *Adversarial label poisoning attack on graph neural networks via label propagation*, Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V, Springer, 2022, pp. 227–243.
15. J. Ma, S. Ding, and Q. Mei, *Towards more practical adversarial attacks on graph neural networks*, Advances in neural information processing systems **33** (2020), 4756–4766.
16. I. Makarov and O. Gerasimova, *Link prediction regression for weighted co-authorship networks*, Proceedings of the 15th International Work-Conference on Artificial Neural Networks (IWANN'19) (Berlin, Germany), Universitat Politecnica de Catalunya, Springer, July 12–14 2019, pp. 667–677.
17. ———, *Predicting collaborations in co-authorship network*, Proceedings of the 14th IEEE International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP'19) (New York, USA), Cyprus University of Technology, IEEE, June 09–10 2019, pp. 1–6.

18. I. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj, *Survey on graph embeddings and their applications to machine learning problems on graphs*, PeerJ Computer Science (2021), no. e357, 1–62.
19. I. Makarov, K. Korovina, and D. Kiselev, *Jonnee: Joint network nodes and edges embedding*, IEEE Access **9** (2021), 144646–144659.
20. I. Makarov, A. Savchenko, A. Korovko, L. Sherstyuk, N. Severin, D. Kiselev, A. Mikheev, and D. Babaev, *Temporal network embedding framework with causal anonymous walks representations*, PeerJ Computer Science **8** (2022), no. e858, 1–27.
21. H. Nguyen, L.-M. Kieu, T. Wen, and C. Cai, *Deep learning methods in transportation domain: a review*, IET Intelligent Transport Systems **12** (2018), no. 9, 998–1004.
22. X. Ning and G. Karypis, *Slim: Sparse linear methods for top-n recommender systems*, 2011 IEEE 11th ICDM, IEEE, 2011, pp. 497–506.
23. I. E. Olatunji, W. Nejdl, and M. Khosla, *Membership inference attack on graph neural networks*, 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, IEEE, 2021, pp. 11–20.
24. E. Szimba, M. Kraft, J. Ihrig, A. Schimke, O. Schnell, Y. Kawabata, S. Newton, T. Breemersch, R. Versteegh, J. Meijeren, H. Jin-Xue, C. de stasio, and F. Fermi, *Etisplus database content and methodology*.
25. J. Tang, J. Li, Z. Gao, and J. Li, *Rethinking graph neural networks for anomaly detection*, ICML, PMLR, 2022, pp. 21076–21089.
26. S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, *Towards demystifying membership inference attacks*, arXiv preprint arXiv:1807.09173 (2018).
27. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, *Graph attention networks*, 2017.
28. B. Wang, T. Zhou, M. Lin, P. Zhou, A. Li, M. Pang, C. Fu, H. Li, and Y. Chen, *Evasion attacks to graph neural networks via influence function*, arXiv preprint arXiv:2009.00203 (2020).
29. S. Wang, J. Cao, and P. Yu, *Deep learning for spatio-temporal data mining: A survey*, IEEE transactions on knowledge and data engineering (2020).
30. B. Wu, X. Yang, S. Pan, and X. Yuan, *Adapting membership inference attacks to gnn for graph classification: approaches and implications*, 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 1421–1426.
31. K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin, *Topology attack and defense for graph neural networks: An optimization perspective*, arXiv preprint arXiv:1906.04214 (2019).
32. H. Zhang, B. Wu, X. Yang, C. Zhou, S. Wang, X. Yuan, and S. Pan, *Projective ranking: A transferable evasion attack method on graph neural networks*, Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3617–3621.
33. H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, and J. Pei, *Trustworthy graph neural networks: Aspects, methods and trends*, arXiv preprint arXiv:2205.07424 (2022).
34. Z. Zhang, J. Jia, B. Wang, and N. Z. Gong, *Backdoor attacks to graph neural networks*, Proceedings of the 26th ACM Symposium on Access Control Models and Technologies, 2021, pp. 15–26.

35. H. Zheng, H. Xiong, J. Chen, H. Ma, and G. Huang, *Motif-backdoor: Rethinking the backdoor attack on graph neural networks via motifs*, arXiv preprint arXiv:2210.13710 (2022).

Поступило 6 сентября 2023 г.

Ivannikov Institute
for System Programming
of the Russian Academy of Sciences;
Moscow Institute of Physics and Technology
(National Research University);
HSE University, Moscow, Russia
E-mail: shaykhelislamov.ds@ispras.ru

Ivannikov Institute
for System Programming
of the Russian Academy of Sciences;
Moscow Institute of Physics and Technology
(National Research University);
ISP RAS Research Center
for Trusted Artificial Intelligence,
Moscow, Russia
E-mail: lukyanov@ispras.ru

HSE University,
Moscow, Russia
E-mail: nseverin@hse.ru

Ivannikov Institute
for System Programming
of the Russian Academy of Sciences;
Moscow Institute of Physics and Technology
(National Research University);
ISP RAS Research Center
for Trusted Artificial Intelligence,
Moscow, Russia
E-mail: drobyshevsky@ispras.ru

Ivannikov Institute
for System Programming
of the Russian Academy of Sciences;
HSE University;
ISP RAS Research Center
for Trusted Artificial Intelligence,
Moscow, Russia
E-mail: makarov@airi.net

E-mail: turdakov@ispras.ru