Universität des Saarlandes
MI Fakultät für Mathematik und Informatik
Department of Computer Science

Bachelorthesis

Link Stealing Attacks on Inductive Trained Graph Neural Networks

submitted by

Philipp Zimmermann

on January 01, 1970

Reviewers

Prof. Dr. Doktor Professer

Prof. Dr. Realy Intelligent

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Saarbrücken, January 01, 1970,

(Philipp Zimmermann)

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, January 01, 1970,

(Philipp Zimmermann)

Abstract

Since nowadays graphs are a common way to store and visualize data, Machine Learning algorithms have been improved to directly operate on them. In most cases the graph itself can be deemed confidential, since the owner of the data often spends much time and resources collecting and preparing the data. In our work, we show, that so called graph neural networks can reveal sensitive information about their training graph. We focused on extracting information about the edges of the underlaying graph by observing the predictions of the target model in so called link stealing attacks. In prior work, He et al. proposed the first link stealing attacks on graph neural networks, focusing on the transductive learning. More precisely, given a black box access to a graph neural network, they were able to predict, whether two nodes of a graph that was used for training, are linked or not. We now focus on the inductive setting. Specifically, given a black box access to a graph neural network, we aim to predict whether there exists a link between any two nodes of any graph, not only the one, the graph neural network was trained on. present results

Acknowledgements

Contents

Chapter 1

Introduction

1.1 Motivation

A graph is a datastructure which is used to model large data and the relationships between entities [1, 2]. It consists of nodes and edges and can be used to model data in almost every domain. For example in social networks, healthcare analytics or protein-protein interactions. In a social network, the nodes would be the users that are registered and the edges would represent whether the users know each other or not by connecting them or not. A graph itself can be deemed as intellectual property of the data owner, since she may spent lots of time and resources collecting and preparing the data. In most cases the graph is also highly confidential because it contains sensitive information like private social relationships between users in a social network or medical information about specific people in healthcare-analytic datasets. Since nowadays graphs are a common way to store and visualize data, Machine Learning algorithms have been improved to directly operate on them. These Machine Learning Models are called Graph Neural Networks (GNNs) [3, 4]. They can be used in different ways to operate on graphs. For example they can be trained to perform node classification [5]. More precisely, given a graph containing some labeled nodes the model is trained to predict the labels of the other unlabeled nodes in the graph. They can also be used to perform link prediction like in social networks where the friendship between two users is guessed [6].

A Graph Neural Network can be trained in different ways, depending on the purpose it will be used for. One way is to train them transductive [7–10]. Regarding the node classification problem that means, that test and evaluation node features are given during training. Only the labels are unknown. Nevertheless this training method is possible theoretically, it cannot be applied to real world problems like in social networks. That's why e.g. social networks keep evolving. Every day new users register and other user delete their accounts. For datasets like that GNNs can also be trained inductive [11–13]. Specifically, now not only the labels of the test and evaluation nodes is unknown but also their features and connections. That means, that the model is trained on one graph and will be evaluated on another one. In that way it is now possible to update the model on new nodes without retraining it over and over again on the full graph.

In our work, we show, that inductive trained Graph Neural Networks are very likely to leak sensitive information about the underlaying graph that was used for training by performing link stealing attacks on the target models.

1.2 Outline

write at the end

Chapter 2

Related Work

Ever since machine learning algorithms were developed, there have been new attacks against these models. In 2004, Dalvi et al. proposed simple evasion attacks to defeat linear classifiers that are used in spam filters [14]. Later in 2006, Barreno et al. outline a broad taxonomy of attacks against linear classifier in their paper *Can Machine Learning Be Secure?*[15]. After in 2012 Deep Neural Networks began to dominate different domains, attacks against these models were also found and further developed [16, 17]. Today it is well know, that machine learning models are vulnerable in a security and privacy manner and that there exist many attacks against Machine Learning Models. With *Membership Inference Attacks* [18–22] an adversary aims to distinguish whether a given data sample was part of the training dataset of the target model or not. Shokri et al. [20] proposed the first Membership Inference Attack on Machine Learning Models. Given a data record and black-box access to a model, they were able to determine if the record was in the target models training dataset. The authors used adversarial machine learning to train an adversary model, that recognizes differences in the target models prediction. They evaluated their experiments on realistic datasets like a hospital discharge, whose membership is sensitive from the privacy perspective and showed that these models can be vulnerable to membership inference attacks. To prevent this attacks, many defenses have been proposed [20, 23–25]. With *Model Inversion Attacks* [26–29], an adversary aims to learn sensitive attributes of the target models training dataset. The first model inversion attack has been proposed by Fredrikson et al. [26]. They showed, that given the target model and some demographic information about a patient, it is possible to predict the patient's genetic markers. The authors further investigate, that differential privacy mechanisms prevent their model inversion attacks, when the privacy budget is carefully selected. With *Model Extraction Attacks* [30–32], an adversary aims to steal the model internals and uses this information to gradually train a substitute model that immitates the behaviour of the target. Tramèr et al. [32] proposed simple model extraction attacks, which were able to steal target models with near-perfect fidality. A similar aproach was proposed by Wang and Gong [33], who were able to successfully steal the hyperparameters of target models. To mitigate these attacks, many defenses have been proposed [31, 34–36]. For Example Juuti et al. [31], showed that they were able to detect all prior model extraction attacks with no false positives by raising an alarm when the distribution of consecutive API queries deviates from benign behavior. Hu and Pang [34] proposed an effective defense against model extraction attacks on Generative Adversarial Networks [37], considering a trade-off between the utility and security of GANs.

Since many real world problems can be represented as graphs, it was urgent to develop machine learning algorithms to fully utilize graph data. Therefor, so called Graph Neural Networks have been developed and already used in various tasks [3, 5, 38, 39]. Although, recent work shows, that graph neural networks are vulnerable to adversarial attacks as well [40–42, 42, 43]. More precisely, an adversary can decrease the targets accuracy by manipulating the graph structure or node features. For example, Sun et al. [41] proposed node injection poisoning attacks, where adversarial nodes are injected into existing graphs to reduce the performance of classifying existing nodes. Zügner et al. [42] showed that even with only a few pertubations the accuracy of node classification significantly drops, while focusing on training and testing phase. Wang et al. [40] focused on adversarial collective classification. They formulate their attack as a graph-based optimization problem, solving which produces the edges that an attacker needs to manipulate to achieve its attack goal and als propose several techniques to solve the optimization problem. Lastly Jin et al. [43] categorized existing attacks and defenses, and reviewed the corresponding state-of-the-art methods. They also have developed a repository with representative algorithms. Our work is different, since we focus on stealing links from graph neural networks.

In recent work, He et al. proposed the first attacks on Graph Neural Networks to obtain information about the underlaying graph [44]. They call their attacks *Link Stealing Attacks*. Given a black box access to a graph neural network, they showed that an adversary is able to predict whether any two nodes of a graph, that was used for training, are linked or not. The attacks reveal serious concerns on the intellectual property, confidentiality and privacy of graphs, when they are used for training. Our work is different, since we focus on *Link Stealing Attacks* on inductive trained Graph Neural Networks. Specifically, given a black box access to a graph neural network, we aim to predict whether there exists a link between any two nodes of any graph, not only the one, the graph neural network was trained on.

Chapter 3

Background

3.1 Neural Networks

3.2 Graphs

As Graph we denote a data structure that contains nodes and edges. A node can have multiple attributes describing it and an edge describes the relationship between them. The most popular example where graphs are used are social networks. The nodes represent the users that have multiple attributes like location, gender, work place etc. In a directed graph user A will have an outgoing edge and user B an ingoing edge if A follows B and vice versa. In an undirected graph the edge won't have a direction. Which means that either A follows B, B follows A or both will lead to the same result, namely only one edge that is drawn, describing their relationship.

3.3 Graph Neural Networks

List of Figures

List of Tables

Appendix A

Additional Something

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Bibliography

[1] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *CoRR*, vol. abs/2005.00687, 2020. [Online]. Available: https://arxiv.org/abs/2005.00687

[2] D. J. Cook and L. B. Holder, *Mining graph data.* John Wiley & Sons, 2006.

[3] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," 2016.

[4] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2017.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[6] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," 2018.

[7] W. Liu and S.-F. Chang, "Robust multi-class transductive learning with graphs," pp. 381–388, 2009.

[8] Y. Zha, Y. Yang, and D. Bi, "Graph-based transductive learning for robust visual tracking," *Pattern Recognition*, vol. 43, no. 1, pp. 187–196, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320309002581

[9] Z. Wang, X. Zhu, E. Adeli, Y. Zhu, F. Nie, B. Munsell, and G. Wu, "Multi-modal classification of neurodegenerative disease by progressive graph-based transductive learning," *Medical Image Analysis*, vol. 39, pp. 218–230, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841517300749

[10] P. P. Talukdar and K. Crammer, "New regularized algorithms for transductive learning," in *Machine Learning and Knowledge Discovery in Databases*, W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 442–457.

[11] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Graphsaint: Graph sampling based inductive learning method," 2020.

[12] R. A. Rossi, R. Zhou, and N. K. Ahmed, "Deep inductive graph representation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 438–452, 2020.

[13] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," 2020.

[14] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 99–108. [Online]. Available: https://doi.org/10.1145/1014052.1014066

[15] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 16–25. [Online]. Available: https://doi.org/10.1145/1128817.1128824

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[17] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, p. 317–331, Dec 2018. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2018.07.023

[18] N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," 2019.

[19] Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, and H. Zhu, "Differentially private data generative models," 2018.

[20] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," pp. 3–18, 2017.

[21] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Towards demystifying membership inference attacks," 2019.

[22] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "Logan: Membership inference attacks against generative models," 2018.

[23] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications*

Security, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 259–274. [Online]. Available: https://doi.org/10.1145/3319535.3363201

[24] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," 2018.

[25] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in classification models," *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, Apr 2021. [Online]. Available: http://dx.doi.org/10.1145/3422337.3447836

[26] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," *Proceedings of the ... USENIX Security Symposium. UNIX Security Symposium*, vol. 2014, p. 17—32, August 2014. [Online]. Available: https://europepmc.org/articles/PMC4827719

[27] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, "Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes," pp. 115–11 509, 2017.

[28] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1322–1333. [Online]. Available: https://doi.org/10.1145/2810103.2813677

[29] S. Chen, R. Jia, and G.-J. Qi, "Improved techniques for model inversion attacks," 2020.

[30] B. G. Atli, S. Szyller, M. Juuti, S. Marchal, and N. Asokan, "Extraction of complex dnn models: Real threat or boogeyman?" 2020.

[31] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: Protecting against dnn model stealing attacks," 2019.

[32] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," 2016.

[33] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," pp. 36–52, 2018.

[34] H. Hu and J. Pang, "Model extraction and defenses on generative adversarial networks," 2021.

[35] H. Jia, C. A. Choquette-Choo, V. Chandrasekaran, and N. Papernot, "Entangled watermarks as a defense against model extraction," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/jia

[36] Y. Mori, A. Nitanda, and A. Takeda, "Bodame: Bilevel optimization for defense against model extraction," 2021.

[37] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[38] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018, accepted as poster. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[40] B. Wang and N. Gong, "Attacking graph-based classification via manipulating the graph structure," 03 2019.

[41] Y. Sun, S. Wang, X. Tang, T.-Y. Hsieh, and V. Honavar, "Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach," New York, NY, USA, p. 673–683, 2020. [Online]. Available: https://doi.org/10.1145/3366423.3380149

[42] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul 2018. [Online]. Available: http://dx.doi.org/10.1145/3219819.3220078

[43] W. Jin, Y. Li, H. Xu, Y. Wang, S. Ji, C. Aggarwal, and J. Tang, "Adversarial attacks and defenses on graphs: A review, a tool and empirical studies," 2020.

[44] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, "Stealing links from graph neural networks," *CoRR*, vol. abs/2005.02131, 2020. [Online]. Available: https://arxiv.org/abs/2005.02131