Privacy-Preserving Mechanisms for Machine Learning on Graph-Structured Data

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Information Systems

by

Chenhan Zhang

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

May 2024

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Chenhan Zhang*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

[Your Name]

DATE: 28th May, 2024

PLACE: Sydney, Australia

Abstract

Graphs provide a unique representation of real-world data. Graph neural networks (GNNs) have gained considerable attention for their proficiency in handling graph-structured data across multiple disciplines. Despite their laudable efficacy in a diverse array of applications, GNNs have been proven to be vulnerable to a multitude of security and privacy risks that can lead to unauthorized data exposure. The thesis aims to provide a rigorous exploration of privacy-preserving mechanisms specifically designed for GNNs. This thesis seeks to explore privacy-preserving mechanisms for graph-structured data involved in GNNs. Specifically, this thesis introduces three research works to this end:

- Research Work 1 addresses property inference attacks on graph data by utilizing the Information Bottleneck (IB) principle to modify graph structures, thereby reducing the leakage of sensitive property information in graph embeddings. This approach retains task-relevant information in the revised graphs, ensuring GNNs maintain predictive accuracy.
- Research Work 2 proposes the Subgraph-Out-of-Subgraph (SOS) approach in federated graph learning to prevent model inversion attacks by extracting task-relevant subgraphs using the IB principle. This approach minimizes sensitive information in GNN updates, incorporating a novel neural approach for mutual information estimation and a generation algorithm for optimized subgraphs.
- Research Work 3 explores federated graph learning in intelligent transportation systems for traffic forecasting. It introduces a novel framework with a differential privacy (DP)-based approach to protect topological information of data contributors. Additionally, it presents the attention-based spatial-temporal graph neural network for forecasting traffic speed, integrating it with the proposed framework for realizing privacy-preserving traffic speed prediction.

Comprehensive case studies employing benchmark GNNs and datasets serve as evaluative metrics, substantiating the practical relevance of the proposed mechanisms. The research aspires to make seminal contributions to the burgeoning field of secure and privacy-preserving GNNs, thereby enhancing both the theoretical underpinnings and practical implementations of these models in handling sensitive graph data.

ACKNOWLEDGMENTS

I would like to first express the deepest appreciation to my supervisor, Professor Shui Yu. Immense support and conscientious guidance from him to the research through my PhD project have been precious. My sincere thanks also go to my co-supervisor, Professor Bo Liu, especially for his help in this thesis.

Secondly, I wish to express my gratitude to the Graduate Research School of University of Technology Sydney and Australian Research Council (ARC DP200101374) for their financial support of my research. Their support has been instrumental in enabling the progression and completion of this work.

I would like to extend my gratitude to all the panel members of the candidature assessments, Professor Wei Liu, Professor Yulei Sui, Professor Yi Zhang, and others. Their equitable evaluations and constructive feedback have been invaluable in shaping the course of my research.

I also could not have undertaken this journey without my research teammates, Weiqi Wang, Zhiyi Tian, Mingjian Tang, Kaiyue Zhang, Andrew Vo, Christos Markos, Shuyu Zhang, and others for their moral support, especially those late-night feedback on my papers.

A huge thank you to my friends in Sydney, Shan Qin, Zhongmou Guan, Wenzheng Jin, and others. Thanks to them for being there and helping out during those down times.

Lastly, I would be remiss in not mentioning my family, especially my parents, Yan Kong and Xianjun Zhang, and my grandparents, Guirong Zhao and Qingqi Kong. Their unwavering support has kept for my spirits and motivation high during my research career.

Gratias ago tibi valde.

List of Publications

Journals:

Forgetting and Remembering are Both You Need: Balanced Graph Structure Unlearning.
Chenhan Zhang, Weiqi Wang, Zhiyi Tian, Shui Yu.
IEEE Transactions on Information Forensics and Security, Under Review.

Information Bottleneck-Based Subgraphs Defending Against Inference Attacks in Federated Graph Learning.
Chenhan Zhang, Shui Yu, James Jianqiao Yu.
IEEE Transactions on Dependable and Secure Computing, Under Review.

Semantic Communications Towards Graph Data.
Chenhan Zhang, Zhiyi Tian, Weiqi Wang, Shui Yu.
IEEE Wireless Communications, Major Revision.

Generative Adversarial Networks: A Survey on Attack and Defense Perspective.
Chenhan Zhang, Shui Yu, Zhiyi Tian, James Jianqiao Yu.
ACM Computing Surveys, 2024, DOI:10.1145/3615336.

SAM: Query-Efficient Adversarial Attacks Against Graph Neural Networks.
Chenhan Zhang, Shiyao Zhang, James Jianqiao Yu, Shui Yu.
ACM Transactions on Privacy and Security, 2023, DOI:10.1145/3611307.

Toward Large-Scale Graph-Based Traffic Forecasting: A Data-Driven Network Partitioning Approach.
Chenhan Zhang, Shuyu Zhang, Xiexin Zou, Shui Yu, James Jianqiao Yu.
IEEE Internet of Things Journal, 2022, DOI:10.1109/JIOT.2022.3218780.

A Communication-Efficient Federated Learning Scheme for IoT-Based Traffic Forecasting.
Chenhan Zhang, Lei Cui, Shui Yu, James Jianqiao Yu.
IEEE Internet of Things Journal, 2021, DOI:10.1109/JIOT.2021.3132363.

Towards Crowdsourced Transportation Mode Identification: A Semi-Supervised Federated Learning Approach.
Chenhan Zhang, Yuanshao Zhu, Chris Markos, Shui Yu, James Jianqiao Yu.
IEEE Internet of Things Journal, 2021, DOI:10.1109/JIOT.2021.3132056.

FASTGNN: A Topological Information Protected Federated Learning Approach For Traffic Speed Forecasting.
Chenhan Zhang, Shuyu Zhang, Shui Yu, James Jianqiao Yu.
IEEE Transactions on Industrial Informatics, 2021, DOI:10.1109/TII.2021.3055283.

SCU: An Efficient Machine Unlearning Scheme for Deep Learning Enabled Semantic Communications.
Weiqi Wang, Zhiyi Tian, **Chenhan Zhang**, Shui Yu.
IEEE Transactions on Information Forensics and Security, Under Review.

Inversion Triplet - A Contrastive Backdoor Mitigation Framework for Self-Supervised Learning.
Hiep Vo, Zhiyi Tian, **Chenhan Zhang**, Xi Zheng, Shui Yu.
IEEE Transactions on Neural Networks and Learning Systems, Under Review.

You Only Look at Yourself: Backdoor Sample Cleanse for Unlabeled Pre-training Dataset via Bootstrapped Dual Set Purification.
Luoyu Chen, Weiqi Wang, Zhiyi Tian, **Chenhan Zhang**, Shui Yu.
IEEE Transactions on Dependable and Secure Computing, Under Review.

Representation Forgetting Against Reconstruction Attacks on Machine Unlearning.
Weiqi Wang, **Chenhan Zhang**, Zhiyi Tian, Shushu Liu, Shui Yu.
IEEE Transactions on Dependable and Secure Computing, Under Review.

Machine Unlearning via Representation Forgetting with Parameter Self-Sharing.
Weiqi Wang, **Chenhan Zhang**, Zhiyi Tian, Shui Yu.
IEEE Transactions on Information Forensics and Security, 2023, DOI:10.1109/TIFS.2023.3331239.

An Asynchronous Multi-Task Semantic Communication Method.
Zhiyi Tian, Hiep Vo, **Chenhan Zhang**, Geyong Min, Shui Yu.
IEEE Network, 2023, DOI:10.1109/MNET.2023.3321547.

The Role of Class Information in Model Inversion Attacks against Image Deep Learning Classifiers.
Zhiyi Tian, Lei Cui, **Chenhan Zhang**, Shuaishuai Tian, Shui Yu, Yonghong Tan.
IEEE Transactions on Dependable and Secure Computing, 2023, DOI:10.1109/TDSC.2023.3306748.

Challenges and Future Directions of Secure Federated Learning: A Survey.
Kaiyue Zhang, Xuan Song, **Chenhan Zhang**, Shui Yu.
Frontiers of Computer Science, 2022, DOI:10.1007/s11704-021-0598-z.

Conferences:

Self Supervision Rejuvenates Similarity-Based Link Prediction.
Chenhan Zhang, Weiqi Wang, Zhiyi Tian, James Jianqiao Yu, Shui Yu.
IJCAI 2024, Jeju, Korea, 2024, Under Review.

Extracting Privacy-Preserving Subgraphs in Federated Graph Learning Using Information Bottleneck.
Chenhan Zhang, Weiqi Wang, James Jianqiao Yu, Shui Yu.
ACM ASIACCS 2023, Melbourne, Australia, 2023, DOI:10.1145/3579856.3595791.

Construct New Graphs using Information Bottleneck Against Property Inference Attacks.
Chenhan Zhang, Zhiyi Tian, James Jianqiao Yu, Shui Yu.
IEEE International Conference on Communications, Rome, Italy, 2023, DOI:10.1109/ICC45041.2023.10279148.

Graph-based Traffic Forecasting via Communication-efficient Federated Learning.
Chenhan Zhang, Shuyu Zhang, Shui Yu, James Jianqiao Yu.
IEEE Wireless Communications and Networking Conference, Austin, TX, US, 2022, DOI:10.1109/WCNC51071.2022.9771883.

Unlearning Effect Verification through Erased Data Reconstruction.
Weiqi Wang, Zhiyi Tian, **Chenhan Zhang**, Shui Yu, Weizhi Meng.

ACM CCS 2024, Salt Lake City, UT, US, 2024, Under Review.

BFU: Bayesian Federated Unlearning with Parameter Self-Sharing.
Weiqi Wang, Zhiyi Tian, **Chenhan Zhang**, An Liu, Shui Yu.
ACM ASIACCS 2023, Melbourne, Australia, 2023, DOI:10.1145/3579856.3590327.

CP-FL: Practical Gradient Leakage Defense in Federated Learning with Compressive Privacy.
Weiqi Wang, Shushu Liu, **Chenhan Zhang**, Mingjian Tang, Feng Wu, Shui Yu.
IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 2023, DOI:10.1109/GLOBECOM54140.2023.10436931.

FedMC: Federated Learning with Mode Connectivity Against Distributed Backdoor Attacks.
Weiqi Wang, **Chenhan Zhang**, Shushu Liu, Mingjian Tang, An Liu, Shui Yu.
IEEE International Conference on Communications, Rome, Italy, 2023, DOI:10.1109/ICC45041.2023.10278903.

GSMI: A Gradient Sign Optimization Based Model Inversion Method.
Zhiyi Tian, **Chenhan Zhang**, Lei Cui, Shui Yu.
Australasian Joint Conference (AI 2021), Sydney, Australia, 2022, DOI:10.1007/978-3-030-97546-3_6.

Safeguard the Original Data in Federated Learning via Data Decomposition.
Jing Lin, **Chenhan Zhang**, Shui Yu.
IEEE Global Communications Conference, Madrid, Spain, 2021, DOI:10.1109/GLOBECOM46510.2021.9685575.

Table of Contents

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**

1.1 Research Background 1
1.2 Research Problems 2
1.3 Research Contribution 3
1.4 Thesis Structure 4

2 Literature Review **7**

2.1 Security and Privacy Problems in Machine Learning 7
2.1.1 Attacks Against Machine Learning Systems 7
2.1.2 Defense Methods 9
2.2 Security and Privacy in Graph Data and Graph Neural Networks 10
2.2.1 Adversarial Attacks against Graph Neural Networks 10
2.2.2 Inference Attacks against Graph Neural Networks 10
2.3 Related Techniques and Frameworks Involved in the Thesis 11
2.3.1 Information Bottleneck in Graph Data 11
2.3.2 Federated Learning on Graph Data 12

3 Methodology **13**

3.1 Preliminary 13
3.1.1 Notations 13
3.1.2 General Taxonomy of Security and Privacy Problems in Machine Learning 14
3.1.3 Graph Data and Graph Learning Tasks 18
3.1.4 Graph Neural Networks 18

3.1.5 Federated Learning 21
3.1.6 Information Bottleneck 22
3.2 Relevance of Present Research Works 23
3.3 Research Work 1: Construct New Graphs using Information Bottleneck Against Property Inference Attacks 24
3.3.1 Research Background, Question, and Motivation 24
3.3.2 Threat Model 25
3.3.3 Method Overview 27
3.3.4 Graph Representation Learning based on Information Bottleneck 27
3.3.5 Construct New Graph Structure 28
3.3.6 Privacy and Utility Guarantee by the Information Bottleneck . . . 28
3.4 Research Work 2: Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck 30
3.4.1 Research Background, Question, and Motivation 30
3.4.2 Threat Model 31
3.4.3 Method Overview 33
3.4.4 IB-subgraph Publishing Mechanism for Subgraph-level Federated Learning Systems 33
3.4.5 Sub-GIB: Subgraph Generation with Information Bottleneck . . . 35
3.4.6 IB-subgraph Generation Algorithm 38
3.4.7 Discussion on Privacy and Utility of IB-subgraph 39
3.5 Research Work 3: A Topological Information Protected Federated Learning Approach For Traffic Speed Forecasting 42
3.5.1 Research Background, Question, and Motivation 42
3.5.2 Traffic Speed Forecasting on Transportation Networks 43
3.5.3 Federated Learning on Transportation Networks 43
3.5.4 Method Overview 44
3.5.5 Attention-based Spatial-Temporal Graph Neural Networks (AST-GNN) 44
3.5.6 Federated Learning Framework for ASTGNN (FASTGNN) 47

4 Case Studies and Results **53**
4.1 Case Studies of Research Work 1: Construct New Graphs using Information Bottleneck Against Property Inference Attacks 53
4.1.1 Experimental Setup 53

4.1.2 Resistance to Property Inference Attacks 55
4.1.3 Prediction Accuracy on Downstream Tasks 56
4.1.4 Hypereparameter Study on β: Tradeoff between Utility and Privacy 57
4.2 Case Studies of Research Work 2: Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck 58
4.2.1 Experimental Setup 58
4.2.2 Learning Performance 60
4.2.3 Comparison with Differential Privacy-based Defenses 64
4.2.4 Sensitivity Studies of Hyperparameters 64
4.3 Case Studies of Research Work 3: FASTGNN: A Topological Information Protected Federated Learning Approach For Traffic Speed Forecasting . . 66
4.3.1 Experimental Setup 66
4.3.2 Accuracy of Forecasting Traffic Speed 68
4.3.3 Ablation Study on FASTGNN 69
4.3.4 Performance Comparison of FASTGNN Under Different Organization Numbers 70
4.3.5 Generalization Ability 71

5 Summary **73**

Bibliography **77**

List of Figures

Figure **Page**

3.1 Taxonomy of security and privacy threats to machine learning. 15
3.2 (Research Work 1) Threat model: property inference attacks on graph embeddings. 25
3.3 (Research Work 1) Schematic of the proposed IB-based defense. 26
3.4 (Research Work 2) The central server only has a coarsely connected global graph. The workers have to manually fine-process the subgraph and hence own the intellectual property of such a subgraph structure. 31
3.5 (Research Work 2) Schematic of the proposed SOS scheme. 32
3.6 (Research Work 2) Architecture of Sub-GIB. GNN(·) and Readout(·) operations differ from the practically adopted GNN models in the FL systems. Negative samples for the holistic discriminator and regional discriminator are generated by row-wise shuffling the feature matrix X_i but keep the original adjacency matrix, i.e., $\tilde{G}_i = (A_i, \tilde{X}_i)$. 35
3.7 (Research Work 3) The framework of ASTGNN. 44
3.8 (Research Work 3) The framework of FASTGNN. 48
3.9 (Research Work 3) Adjacency matrix alignment. The red frame highlights the padding entries. The shadowed region highlights the entries entailing the connectivity among objective local-network and other local-networks. 50

4.1 (Case Study of Research Work 1) Comparison of graph classification accuracy. 55
4.2 (Case Study of Research Work 1) Sensitivity of β to graph classification accuracy and property inference attack accuracy on IMDB-B and COLLAB datasets with GCN model. 56
4.3 (Case Study of Research Work 2) Comparison of training time consumption per epoch. 61
4.4 (Case Study of Research Work 2) Comparison of prediction accuracy and MIA resistance between DP and the proposed scheme. GSAGE denotes GraphSAGE. 62

4.5 (Case Study of Research Work 2) The sensitivity of the proposed scheme to hyperparameter $\beta \in 0.01, 0.2, 0.5, 1, 5$, $\gamma \in 0.01, 0.1, 0.5, 1, 5$, and $\rho \in 0.01, 0.1, 0.3, 0.5, 0.7$ on Cora. AUC is obtained by averaging the MIA results of all four clients. . . 64
4.6 (Case Study of Research Work 3) Traffic speed forecasting curves in a day. (a) and (b) present results from two different sensor stations, respectively. 69
4.7 (Case Study of Research Work 3) Visualization of training process for 30 global epochs with different organization numbers. 71

List of Tables

Table **Page**

3.1 Glossary of key symbols in the thesis (unless other stated). 14

4.1 (Case Study of Research Work 1) Statistical summary of graph classification datasets. 54
4.2 (Case Study of Research Work 1) Comparison of property inference accuracy. 54
4.3 (Case Study of Research Work 2) Statistical summary of Cora, Citeseer, and PubMed datasets. 58
4.4 (Case Study of Research Work 2) Comparison of total training time consumption using GraphSAGE. 59
4.5 (Case Study of Research Work 2) Prediction accuracy comparison of overall federated learning system. 60
4.6 (Case Study of Research Work 2) Reconstruction performance of model inversion attacks. 60
4.7 (Case Study of Research Work 3) Comparison of traffic speed forecasting accuracy. 67
4.8 (Case Study of Research Work 3) Comparison of ablation tests on FASTGNN. 68
4.9 (Case Study of Research Work 3) The accuracy of FASTGNN with different organization numbers. 70
4.10 (Case Study of Research Work 3) Comparison of traffic speed forecasting accuracy on METR-LA. 72

CHAPTER 1

INTRODUCTION

1.1 Research Background

Graph data, characterized by its structured representation of entities and their interrelations, has become increasingly prevalent in various domains, such as social networks, biological networks, and knowledge graphs [115]. The intrinsic value of graph data is highlighted by its ability to model complex, interconnected systems in a naturally intuitive manner. For example, within social networks, graph data facilitates the understanding of social dynamics and influence propagation, providing insights into community structures and user interactions [17, 48, 66, 91]. In the realm of biological networks, graph data enables the elucidation of intricate life processes, mapping out the interactions among genes, proteins, and other biomolecular entities, thereby advancing the comprehension of biological functions and disease mechanisms [55, 65, 68, 75, 120, 136]. Knowledge graphs, on the other hand, leverage graph data to organize and integrate information in semantically rich forms, enhancing data interoperability, discovery, and reuse across various disciplines [4, 20, 54, 125, 143].

Nowadays, the predominant approaches to analyzing graph data are those based on machine learning (ML), with a particular emphasis on neural network-based approaches. However, traditional deep neural networks (DNNs), such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), face significant limitations when dealing with graph data, primarily due to their inherent design for grid-like, tabular data structures, which struggle to capture the complex, non-Euclidean nature of graphs.

This misalignment often fails to fully exploit the relational information and structural nuances inherent in graph data. In response to these challenges, graph neural networks (GNNs) have emerged as a powerful solution, specifically designed to operate on graph structures [113]. GNNs can handle the intricacies of graph data by directly incorporating the connectivity patterns and features of nodes and edges, thus offering a more natural and effective means of learning from graphs. For example, graph convolution networks (GCN) leverage graph convolutional operations to learn the underlying structure of the graph [50]. Graph Attention Networks (GAT) utilize attention mechanisms to dynamically weigh and aggregate neighbor features for each node in a graph [97]. This advancement significantly enhances the capability to model in capturing complex patterns and relationships between the nodes and edges, marking a pivotal shift in the approach to graph data analysis and application. GNNs have shown remarkable performance in various domains, such as recommendation systems [29, 41], drug discovery [44, 106], and traffic prediction [18, 86].

Graph data, encapsulating a wealth of private and sensitive information in many scenarios, presents significant privacy and security challenges. This complexity arises not only from the detailed intellectual properties and patents that organizations strive to protect from premature exposure but also from the personal information of individuals, such as health records or financial transactions, that demand confidentiality. The structural intricacies of graph data, where nodes and relationships can reveal patterns and insights about proprietary innovations or intimate personal details. For instance, in social networks, nodes could represent users, and edges could signify relationships or interactions. Revealing the structure or feature information of such graphs could lead to significant privacy breaches, exposing personal relationships and health information.

1.2 Research Problems

The applications of GNNs in the analysis of this sensitive graph data have brought to the forefront the vulnerabilities inherent in these advanced models [107]. It has been widely demonstrated that DNNs are intrinsically susceptible to a range of security and privacy attacks. From the security perspective, for instance, in the domain of adversarial attacks, slight modifications to an input image, imperceptible to the human eye, can cause a DNN-based classifier to misclassify the image with high confidence. This vulnerability not only compromises the integrity of the model's outputs but also raises questions about the reliability of DNNs in security-critical applications. On the privacy front,

attacks like model inversion and membership inference expose the underlying data or sensitive attributes about the individuals represented in the data. Model inversion attacks, for example, aim to reconstruct input data from model outputs, threatening the confidentiality of the information the model has learned. GNNs, inherent to the nature of deep neural networks, are also vulnerable to the above attacks [89]. Taking the above model inversion attacks as an example, they can also be extended to GNNs. In a model inversion attack on a GNN, an adversary aims to reverse-engineer the input graph on knowledge of the model's architecture and access to its predictions. This could involve deducing the features of individual nodes, the presence or absence of specific edges, or even entire subgraph structures that are characteristic of sensitive or private information. Given the GNN's ability to aggregate and leverage neighborhood information to generate predictions, these attacks can be especially effective, as the model's outputs often contain implicit details about the local graph structure and node attributes.

While there are an increasing number of studies on privacy attacks and defense issues within ML, current research mainly focuses on the domain of grid-like data (e.g., images) and traditional DNNs. The research effort on graph data and GNNs is at its infancy. Notably, given that graphs often manifest unique structural patterns, the safeguarding of privacy in graph data presents a nuanced challenge, fraught with multiple dimensions that are inherently complex to navigate [134]. For example, the efficacy of "stealing link", which refers to a membership inference attack aimed at a specific connection within the training graph, leverages the similarity between the outputs produced by two interconnected nodes [38]. Additionally, a model extraction attack [109] mimics the propagation of information within GNNs for the purpose of generating a surrogate graph. The attack methods targeting GNNs exhibit unique patterns compared to those directed at DNNs, prompting an interest research question: *how to devise efficient privacy-preserving mechanisms to counter these attacks?*

1.3 Research Contribution

The objective of this thesis is to conduct a thorough investigation of privacy-preserving mechanisms for graph data in the context of their use within GNNs, emphasizing the reduction of vulnerabilities that potential attacks may introduce. More importantly, this thesis places particular emphasis on navigating the intricate trade-off between maintaining data utility and achieving the desired privacy-preserving effect within our mechanisms. Recognizing that enhancing privacy often comes at the cost of reducing the

detail or usability of data, our approaches seek to balance these competing demands to ensure that the privacy measures do not unduly compromise the functional utility of the graph data.

Specifically, the contribution of this thesis can be summarized as below:

- This thesis proposes a novel method using the Information Bottleneck (IB) principle to create modified graph structures that obscure property information of the original graphs. This approach enhances the privacy of graph embeddings, making it more challenging for attackers to infer sensitive information. Please refer specifically to Research 1 and its case studies.
- This thesis proposes a defense scheme, Subgraph-Out-of-Subgraph (SOS), to protect against model inversion attacks (MIA) in federated graph learning systems by using the IB principle to extract task-relevant subgraphs for local GNN training. This method maintains prediction accuracy while making it more difficult for attackers to infer the original subgraph structures. Please refer specifically to Research 2 and its case studies.
- This thesis proposes a novel federated learning framework for privacy-preserving traffic forecasting in intelligent transportation systems. It introduces a differential privacy-based adjacency matrix preserving approach for the privacy protection of local transportation networks, alongside a GNN-based spatial-temporal predictor, which together form FASTGNN for privacy-preserving traffic speed forecasting. Please refer specifically to Research 3 and its case studies.

Through these endeavors, this thesis is dedicated to the development of secure and privacy-preserving GNNs, offering insights that bridge the gap between theoretical innovation and practical application in the handling of sensitive graph data.

1.4 Thesis Structure

This thesis is presented as a *thesis by compilation*. The structure is outlined as below.

- Chapter 2 presents the literature review.
- Chapters 3 introduces the preliminary knowledge and three published research works that collectively contribute to the this thesis[1].

[1]To comply with the requirements of a thesis by compilation and to avoid repetition, we have chosen to present only the most relevant publication for each research work.

- Research Work 1 (Section 3.3): Construct New Graphs Using Information Bottleneck Against Property Inference Attacks [129].

- Research Work 2 (Section 3.4): Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck [130].

- Research Work 3 (Section 3.5): FASTGNN: A Topological Information Protected Federated Learning Approach for Traffic Speed Forecasting [131].

- Chapter 4 presents experimental results of these research works, accompanied by a comprehensive discussion.

- Chapter 5 concludes the thesis alongside a discussion of potential avenues for future research.

CHAPTER 2

LITERATURE REVIEW

The current state of research on security and privacy in ML is rapidly expanding and evolving, with numerous studies being conducted. As such, it is important to take a step back and conduct a comprehensive review of the existing literature to gain a clear understanding of the current state of knowledge on the investigated subject. In Sections 2.1 and 2.2, we aim to critically examine the literature on security and privacy in ML, with a special emphasis on the research related to graph data and GNNs, aiming to offer an overview of the current state of knowledge on the topic of this thesis. Additionally, in Section 2.3, a review of the literature concerning related techniques mentioned in this thesis is provided to facilitate a smooth understanding.

2.1 Security and Privacy Problems in Machine Learning

Nowadays, malicious attacks against ML systems and their countermeasures constitute the primary focus of researches on security and privacy issues in ML [74]. This section will examine the literature related to attack methodologies and defense strategies.

2.1.1 Attacks Against Machine Learning Systems

There are two typical attack paradigms that mainly threatening the ML systems, namely, attacks at the training step and attacks at the inference step.

When the attacks are at the training step, the model will be targeted [12]. The adversary can carry out poisoning attacks to undermine the integrity of the training process or training data. The poisoning attacks can be purposed or stochastic. Purposed poisoning attacks from adversary target to induce models to yield specified labels [84], while stochastic poisoning attacks target to degenerate the model's prediction accuracy [58]. Furthermore, the poisoning attacks in the training step can be carried out on the model (i.e., model poisoning [7, 30]) or on the data (i.e., data poisoning [87]). Model poisoning attacks are especially common in federated learning (FL) scenarios, which aims at poisoning updates of model before transmitting to the central server or inserting a backdoor program inside [7]. Data poisoning attacks aim to poison the data involved in the training step to mislead the models, which can be classified as dirty-label poisoning [34] and clean-label poisoning [84]. Dirty-label poisoning brings a few data samples that attempts to mislead the classification of the wanted target labels in the training dataset. On the contrary, clean-label poisoning does not manipulate the training data's labels, since there exists a process for certifying that data belong to the right classes, and guarantees the imperceptibility of the poisoned samples. When the attacks are at the inference step, the adversary does not manipulate the model but attempts to mislead it to yield incorrect outputs or exploit models' features [8].

Attacks at the inference step, a.k.a., inference attacks, mainly target at the private data used for training or evaluating the ML models [85]. These attacks can be further categorized into white-box inference attacks [69] and black-box inference attacks [43]. The former means that the adversary has full authority to access the model, while that of the adversary in the latter is restricted. Another taxonomy includes membership inference attack, properties inference attack, class representatives inference attack, and input inference attack [61]. Particularly, the membership inference attack is the most common paradigm that the adversary attempts to infer whether there is the adoption of given data samples in the model training [85]. The class representative inference and input inference attacks are emerging with the development of DL techniques. For example, Reference [39] introduces a Generative Adversarial Networks (GAN)-based approach that the GAN can forge samples of the private training data. Reference [146] and [141] proposed an insightful work named Deep Leakage from Gradient (DLG), which is one type of input inference attack. In DLG, the adversary can develop both the training inputs and the corresponding labels with just a handful of iterations. Moreover, there is also a category of inference attacks targets the ML model itself, rather than the data, which is known as model extraction attacks.[92, 95, 119, 148]. Since ML models are

often the result of substantial investment and research, which are valuable intellectual properties. Model extraction attackers can illicitly acquire this intellectual property, thereby eroding competitive advantages and potential revenues [94].

2.1.2 Defense Methods

To mitigate or eliminate the adverse impact on the utility, performance, and privacy of model assets and stakeholders caused by the aforementioned malicious attacks, different strategies have been adopted or devised as countermeasures (defenses) against the attacks.

The conventional defense approaches against security and privacy attacks can be generally categorized into hardware-assisted approaches, cryptographic approaches, and differential privacy-based approaches. Hardware-assisted approaches such as trusted execution environment (TEE) [80] and dynamic root of trust measurement (DRTN) [28] are developed from underlying architecture, which involves developing separate hardware modules or operating systems for executing ML tasks. In this way, malicious actions can be blocked to a great extent and thus systems can offer strong security and privacy guarantees to all model assets. Notwithstanding, the requirement of specific hardware configurations impedes their applicability on different computing devices. Homomorphic Encryption (HE) [5] is a cryptographic technique widely adopted in data privacy-protection of ML that enables computation to be directly performed on encrypted data (i.e., ciphertext) with no need for decrypting the data, thereby making the computation's result maintain encrypted. Albeit the protection of data privacy, HE is computationally expensive, especially for decentralized systems where clients are usually edge devices (e.g., smartphones), and may significantly reduce the training efficiency of the shared model [128]. Differential Privacy (DP) [26] harnesses the addition of noise to the data or the use of generalization techniques to obfuscate certain sensitive attributes to the point where a third party cannot distinguish the individual, rendering the data unrecoverable. Compared with the above two branches of approaches, DP is considered a more feasible countermeasure to privacy attacks due to its ease of operation and negligible computational overhead. Nonetheless, recent research has also disclosed the limitations of DP [36]. For example, adding noise naturally depreciates the utility of the data for model training [42, 142].

2.2 Security and Privacy in Graph Data and Graph Neural Networks

In this section, the review of related literature will delve deeper into the methodologies of attacks and defenses concerning graph data and GNNs.

2.2.1 Adversarial Attacks against Graph Neural Networks

The seminal work of extending the notion of adversarial attack to the graph domain can be referred to [25, 149]. Reference [149] investigated both the poisoning and evasion attacks on node classifications by proposing a greedy incremental computation-based approach via modifying the graph structure to increase the loss. Comparatively, Reference [25] only considered the evasion attack and devised a reinforcement learning-based approach to downgrade the test accuracy. [13] proposed a poisoning attack against random walk-based models and demonstrated the transferability of the attack across different models. Reference [101] identified the limitation to attack multi-layer GNNs and proposed a corresponding optimization-based attack method. Reference [149] leveraged meta-learning and treated the graph structure as a hyperparameter to assist in seeking the graph perturbation. Reference [118] treated graph structural perturbation as continuous variables by using convex relaxation to optimize the perturbation.

2.2.2 Inference Attacks against Graph Neural Networks

Recent research efforts indicated that GNNs, as the extension of neural network models to the graph domain, are also vulnerable to inference attacks [137]. With respect to graph data privacy, inference attacks are nonnegligible threats therein. For inference attacks against graph data, Zhang *et al.* [139] demonstrated that gradients of a trained GNN can be utilized to reconstruct the graph in a white-box setting. He *et al.* [38] leveraged the outputs of GNN models to infer the graph data the GNN models were trained on in a black-box setting. Zhang *et al.* [137] utilized the leakage of the graph embeddings and designed a corresponding approach to model inversion attacks.

There is also an increasing research efforts on countermeasures against these attacks [89]. The existing studies mainly investigated defenses against adversarial attacks [111], while less attention has been paid to privacy attacks. Among the few research works studying the defense against inference attacks on GNNs, differential privacy (DP) techniques are mainly considered, which introduces noise to nodal attributes [81, 110,

137]. For example, Sajadmanesh and Gatica-Perez [81] used local differential privacy to preserve the nodal feature. However, as discussed above, research has demonstrated that neural network-based inference attacks can bypass the DP protection to a great extent [39, 77, 139]. Furthermore, the privacy preservation of nodal attributes is still far from the one of the graph structure.

This necessitates further investigation into identifying effective and balanced privacy-preserving mechanisms to counter inference attacks on GNNs, which is also the focus of this thesis.

2.3 Related Techniques and Frameworks Involved in the Thesis

2.3.1 Information Bottleneck in Graph Data

Information bottleneck (IB) was initially proposed to preserve the maximum *mutual information* (MI) of a piece of encoding data [93]. Alemi *et al.* [2] advanced IB to deep learning by the proposed variational information bottleneck (VIB). The capacity to extract condensed and significant representation makes it an effective tool for enhancing representation learning capacity among various learning tasks [102]. Nevertheless, conventional IB methods like VIB cannot be directly applied to graph data due to the intractability of MI calculation for irregular data. To address the problems, some earlier studies utilized MI maximization [40] to obtain graph representations [76, 88, 98].

Wu *et al.* [112] first conceptualized the graph representation learning with IB principle as graph information bottleneck (GIB), and leveraged Gaussian prior assumptions to sample neighbors for node representation learning. Yu *et al.* [126] leveraged neural network-based MI estimation to recognize a subgraph from an original graph. Sun *et al.* [90] proposed to learn new graph structures from original ones based on IB. Our notion of IB-based subgraph extraction is akin to the subgraph recognition pattern in [126]; however, the one in [126] is designed for graph classification tasks, which is far from node classification tasks. Furthermore, for most of GIB studies [76, 88, 90, 98, 126], privacy robustness of the IB-extracted graph representations is not involved. For example, Wu *et al.* [112] showed that the graph representation developed by their proposed GIB is robust to adversarial attack, the resistance to inference attacks is not studied in this work. Wang *et al.* [100] tried to solve the problem by leveraging meta-learning to make the central server learns a task-independent global model. The global model will be

distributed to clients and fine-tuned with the local training data.

2.3.2 Federated Learning on Graph Data

Federated learning (FL) is an emerging ML technique that enables distributed learning among multiple parties while protecting their data privacy [121]. While FL has been widely investigated with regard to Euclidean data (e.g., images and texts), and related research regarding irregular graph data is still in the early stage. For federated graph learning (FGL), the scenarios can be categorized into 1) Node-level FL [52], 2) Graph-level FGL [37], and 3) Subgraph-level FGL [134]. Notably, our work falls into the realm of subgraph-level FGL, where each client holds a subgraph that is part of a larger global graph. For FGL, the challenges of non-independent and identically distributed (Non-IID) datasets are more severe than those in conventional FL training. In addition to features and labels, heterogeneous distributions of graph structures have a nonnegligible impact on FGL training. Several research efforts have been devoted to this problem. Focusing on subgraph-level FGL, Zheng *et al.* [145] proposed to handle the distribution divergence among different clients' graph data by split learning. Zhang *et al.* [134] leveraged GraphSAGE [35] model to improve the inductiveness and scalability of graph mining. Furthermore, they attached importance to the missing connections between subgraphs and attempted to recover them since they can contribute to the representation learning on every single subgraph.

CHAPTER 3

METHODOLOGY

This thesis delves into investigating the privacy preserving mechanisms for graph data in graph learning. In particular, the thesis aims to elucidate the vulnerabilities and risks associated with deep learning models applied to graph data, while concurrently proposing and evaluating novel mechanisms that serve to mitigate these privacy concerns. To achieve the overarching objectives, this chapter will commence by introducing preliminary knowledge pertinent to the research works in Section 3.1. Then, this chapter details three published research works that correspond to three specific research questions from Section 3.2 to Section 3.5.

3.1 Preliminary

3.1.1 Notations

In this thesis, we include three research works, introducing a bulk of mathematical symbols and equations. To ensure that readers can understand the symbols easily, a summary of the commonly appeared symbols in this thesis is presented in Table 3.1. Note that the actual representation of each symbol may differ in research works.

Table 3.1: Glossary of key symbols in the thesis (unless other stated).

Symbol	Explanation
G	Graph.
$\mathcal{V}$	Node set of the graph.
v_i, i	Node i.
$\mathcal{N}(v_i)$	Neighbor node set of node v_i.
N	Number of nodes in the graph.
$\mathcal{E}$	Edge set of the graph.
$e_{i,j}$, (i,j)	The edge connecting node i and j.
A	Adjacency matrix.
D	Degree matrix.
L	Laplacian matrix.
∇_θ	Gradients of parameter θ.
$\mathscr{L}$	Loss function; objective function.
$\mathscr{R}$	Empirical risk function.
X	Node features (attribute).
Y	Labels (ground truth).
W, θ, ϕ	Weight matrix of neural networks.
σ	Activation function of neural networks.
T	Number of training epochs/rounds.
h, H, z, Z	Latent representation; hidden states.
$I(\cdot,\cdot)$	Mutual information.

3.1.2 General Taxonomy of Security and Privacy Problems in Machine Learning

The high-level approach is adopting the classical confidentiality, integrity, and availability (CIA) prism, which is inherited from the cybersecurity domain. In terms of ML, confidentiality is usually associated with the model assets (e.g., training data, model algorithm, and model parameter). Some of the confidentiality attacks seek to divulge the model's architecture or parameters, which may be recognized as significant intellectual property of the model holder [73]. Others attempt to expose the data used to train the model and may compromise the data source's privacy, e.g., the patients' clinical data used to train medical decision models is often of foremost privacy. Therefore, confidentiality attacks are more closely connected to privacy issues. Comparatively, integrity attacks jeopardize the model by false negatives, i.e., inducing deviated model output. Availability attacks try to deny access to valuable model outputs or the features of the system via false positives, i.e., denial-of-service (DoS) [103].

The more exclusive approach is identifying a threat model including viewing the

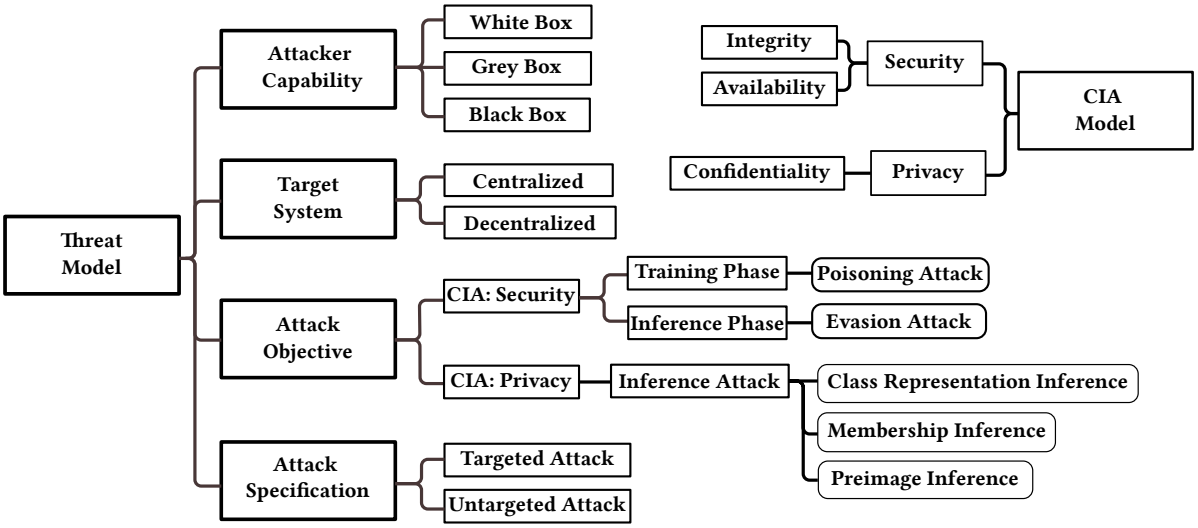


Figure 3.1: Taxonomy of security and privacy threats to machine learning.

threat surface (also known as the attack surface) of ML-based systems and understanding attacker's capabilities, goal, and specifications to gain an insight into when, where, and how an attacker will attempt to compromise the system, which can better help us understand the threats and strategies of security and privacy attacks against ML models. An illustration of the taxonomy is shown in Figure 3.1[1].

Attack Objective. According to the attackers' objectives, threat models can also be classified into poisoning attacks (also known as causative attack), evasion attacks, and inference attacks. Both poisoning attacks and evasion attacks attempt to undermine the integrity that makes ML models yield incorrect results (e.g., predicted label and information retrieval) given input, which can be categorized as security attacks through the CIA prism. In the realm of ML model surface, there are two main phases involved: inference phase and training phase. Attack at the inference phase (also known as exploratory attack) does not manipulate the model itself but attempts to either deviate it from yielding correct outputs (similar to the concept of integrity attack as aforementioned) or gather information about the model architecture or parameters. Attacks at the training phase seek to exploit or compromise the model itself.

This thesis mainly falls into the realm of inference attacks. Inference attack[2], also known as exploratory attack, refers to that the attacker can exploit leak information

[1]Please note that we only provide a general taxonomy in this section. The specific threat model for each research work will be detailed in the corresponding part.

[2]Note that "inference attack" is different from "attack at the inference phrase".

about the features of training data, which is considered as a threat to the confidentiality of model assets through CIA prism. With regards to different features the attacker intends to infer, inference attack can be further categorized into: preimage inference, class representation inference, and membership inference. Given a ML model, preimage inference (also known as model inversion) targets reconstructing training data from model parameters in which the requirement of inference accuracy is usually high (e.g., pixel-level [147]) [32]. A special variant of preimage inference targets at reconstructing model rather than data (also known as model extraction attack), which attempts to obtain an adversarial model that is functionally and statistically equivalent and statistically close to the target model. Comparatively, class representation inference does not aim to reconstruct actual training data but only class representatives. In our taxonomy, we set apart from the concepts of preimage inference and class representation inference due to their different threat levels. Membership inference attempts to predict whether or not an exact data point (e.g., an image) is contained in the model's training dataset [85]. From a view of the population, property inference aims to learn from the model properties of the training dataset seemingly independent of the model's actual goal [64]; we classify it as a variant of membership inference in this thesis.

Attacker's Capabilities. From the perspective of the attacker's capabilities (or attacker's observations), there are three different scenarios of attacks: white, grey, and black boxes. A while-box attack assumes that the attacker has full information about the model assets. On the contrary, a black-box attack supposes the attacker has no information about the model assets, but s/he can *query* from the victim model by API services which is usually provided by Machine Learning as a Service (MLaaS) platforms. Comparatively, a grey-box attack comes somewhere in between; the attacker knows partial information[3]. It is a common pattern that the attacker uses a surrogate model (shadow model) to mimic the target model locally and develop specific examples that can affect the target model due to the difficulty of directly manipulating the target model. White-box setting allow the attacker to construct an identical model to the target model, while in grey- and black-box settings the attacker can only employ an architecture- or function-analogous mode, or distill a model [116]. Accordingly, the dangerous level of the three attacks is: Black > Grey > White.

[3]Many studies merge the grey-box setting into the white box or black box settings [11, 15, 69]. For a rigorous definition, we differ the three terms by defining the white box and black setting as extreme situations.

Attack Specification. Furthermore, according to the specification of the attack, we can also categorize the threat model into targeted and untargeted attacks (also known as dodge attacks). A targeted attack means that the attack is forged towards an assigned and clear instance, while the untargeted attack is not. For an evasion attack that targets a multiple-classifier that classifies an animal image, a targeted evasion attack could be making the output label which is originally "dog" to "cat"; meanwhile, an untargeted attack only concerns if the output is correct and arbitrary incorrect labels are acceptable. For an inference attack, a targeted attack can be the attacker who wants to infer a specific class of data examples. Note that the attack specification is mainly consideration for classification task, and particularly, attack to binary classification task is naturally deemed as targeted. In addition, a membership inference attack is regarded as targeted as the "membership" is undoubtedly a specific objective.

Target System. The threat model can target centralized or decentralized ML systems. From the level of the system, the threat model could target at centralized or decentralized ML systems.

In traditional ML, the efficiency and accuracy of models are determined by computational power and training data available on a centralized computing device (e.g., a server). With the increase in data volume and model complexity, a centralized system limited by computational power is arduous to undertake such complicated computing tasks. In addition, centralized systems are also struggling with security and privacy issues. For one thing, data holders may be reluctant to contribute their data to a centralized system since their data may contain a mass of private and sensitive information. For another, a centralized system stores all sensitive information in a central custodian (usually a central server), presumably encountering single point failure.

The adversaries to ML systems can be conducted from both outside and inside. Outside attacks include those triggered by eavesdroppers during the data transmission process between DLaaS providers and users. Insider attacks incorporate the attacks conducted by DLaaS or related stakeholders when the task is crowdsourced, and by malicious users. Generally, from a security aspect, inside attacks are more surging than outside attacks since they considerably beef up the adversary. From a privacy aspect, outside attacks may lead to more serious consequences since the personal information of users can be divulged publicly. Inside attacks usually follow three paradigms: (1) single attack paradigm [7]: A non-colluding participant targets to lead the model to classify a couple of inputs erroneously; (2) Sybil attack [33]: the attackers can simulate a large

number of pseudonymous participant accounts to mount disproportionate attacks. (3) byzantine attack [123]: the behavior of byzantine saboteurs are likely to be absolutely arbitrary and give the output a distribution imitating the correct model's, which renders them difficult to be detected.

The adversaries can be categorized into two types per the intentions, namely, semi-honest and malicious. The semi-honest adversaries are regarded to be honest-but-curious or passive [10], who attempt to peep other participants' private information without compromising the communication protocol. Particularly, it is assumed that the passive adversaries merely look attentively at the training result rather than the data of honest participants. Thereagainst, the malicious adversaries intend to learn other participants' private information by infringing the communication protocol, e.g., manipulation, spoofing and colluding [21].

3.1.3 Graph Data and Graph Learning Tasks

Formally, let $G = (\mathcal{V}, \mathcal{E})$ denote an undirected and unweighted graph, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The adjacency matrix of G is $A = \{a_{ij}\} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ where we have the entry $a_{ij} = 1$ if edge $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise.

We study the semi-supervised node classification task in this work. The graph G is assumed to be attributed, and we denote the node attribute (feature) matrix by $X = \{x_i\} \in \mathbb{R}^{|\mathcal{V}|\times D}$ where D is the dimension of the node feature vector. Thus, the attributed graph G can be also denoted by $G = (A, X)$. Given the set of labeled nodes $\mathcal{V}_{\text{lab}}$ and the labels $Y \in \mathbb{R}^{|\mathcal{V}_{\text{lab}}|}$, the task aims to learn a GNN f that can map the class of each unlabeled node to the exact one.

3.1.4 Graph Neural Networks

GNN learns a representation for each node by two key computations: (1) AGGREGATE(): aggregate the non-linear-transformed vectors of neighbor nodes; (2) UPDATE(): update the node representation by non-linear transformation [70]. Let $\mathcal{N}(v)$ be the set of 1-hop neighbor nodes of node v, the canonical aggregation of GNNs is described as:

$$
\begin{aligned}
h^l_{\mathcal{N}(v)} &= \text{AGGREGATE}_l\left(\left\{h^{l-1}_u, \forall u \in \mathcal{N}(v)\right\}\right), \\
h^l_v &= \text{UPDATE}_l\left(h^l_{\mathcal{N}(v)}\right),
\end{aligned}
\tag{3.1}
$$

where h^l_u denotes the embedding of node u at layer l. Moreover, for developing graph-level representation, a graph pooling operation (e.g., max pooling and mean pooling) is

performed by aggregating the embeddings of all nodes:

$$h_g = \text{POOLING}(h_v, \forall v \in V). \tag{3.2}$$

3.1.4.1 Graph Convolution Networks

To process the graph data in the spectral domain, the normalized graph Laplacian matrix is first computed given the adjacency matrix A as

$$L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{3.3}$$

where $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix and $D = diag\left(\Sigma_j \phi_{ij}\right) \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix. Then, L is decomposed as

$$L = U \Lambda U^T, \tag{3.4}$$

where $U \in \mathbb{R}^{N \times N}$ is the matrix of eigenvectors of L; $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal matrix of κ with $\Lambda = diag(\kappa)$, and κ is the eigenvalues of L in descending order. Subsequently, a convolution operation is performed in the spectral domain of the graph and can be formulated as

$$g_\theta * X = g_\theta \left(U \Lambda U^T\right) X = U g_\theta(\Lambda) U^T X, \tag{3.5}$$

where X is the input data, $*$ denotes the graph convolutional operator, and g_θ is the kernel with a group of convolution parameters represented by $\theta \in \mathbb{R}^N$. Thus, an updated feature of the input X can be computed by the multiplication between g_θ and $U^T X$. Consequently, the spatial correlations among different nodes in the graph can be learned and stored in the new features.

The graph convolutional operation can be finally formalized as

$$X' = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} X \theta_g\right), \tag{3.6}$$

where X' is the update which contains updated nodal embeddings; $\widetilde{A} = A + I_N$ is the adjacency matrix with added self-connections, $\widetilde{D}$ is the diagonal degree matrix of $\widetilde{A}$, θ_g denotes all the shared parameters in this operation, and $\sigma(\cdot)$ is the sigmoid function.

3.1.4.2 Graph Attention Networks

GAT extends GCN by incorporating an explicit attention mechanism. Following a self-attention strategy [96], GAT learns the hidden features of each node by iteratively using

node feature for similarity computation. The key difference between GAT and GCN is on how to collect and accumulate the feature representations of neighbor nodes. In GCN, a standard convolution includes the standardized sum of the features of adjacent nodes as

$$h_i^{l+1} = \sigma\left(\sum_{j\in \mathrm{N}(i)} \frac{1}{c_{ij}} \phi^l h_j^l\right), \tag{3.7}$$

where $\mathrm{N}(i)$ is the set of adjacent nodes which are immediate neighbors of node i, σ is a non-linear activation function, c_{ij} is a standardized constant based on graph structure, l is the current layer, ϕ^l is the weight matrix for node feature transformation, h_i^{l+1} is the updated hidden feature of node i.

GAT replaces the above convolution operation in graph convolution with an attention mechanism. To better illustrate how the node features of layer l are updated to those of layer $l+1$, we first introduce the consitituting component of GAT, i.e, graph attentional layer. The input to a GAT layer is a set of node features, $h^l = \{h_1^l, h_2^l, ...h_N^l\}, h_i^l \in \mathbb{R}^F$ where N is the number of nodes and F is the number of features from each node. To transform the input features into higher-level features, a shared weight matrix, $\phi \in \mathbb{R}^{F'\times F}$, is used to cast the input to another feature space of F'-dimension. Then, a self-attention mechanism is defined and shared between along edges to calculate the attention coefficient of nodes and their neighbors:

$$e_{ij} = a\left(\phi h_i^l, \phi h_j^l\right), a : \mathbb{R}^F \times \mathbb{R}^{F'} \to \mathbb{R}, \tag{3.8}$$

where $a(\cdot,\cdot)$ is the attention mechanism, e_{ij} is the computed attention coefficient. Note that to retain topological information of the graph, only the attention coefficients of the node and its first-hop neighbors are computed. A softmax function is used to normalize the attention coefficients into a easily comparable form. Finally, a Leaky Rectified Linear Units (LeakyReLU) activation function [117] is applied the final normalized attention coefficients α_{ij} is obtained as

$$\alpha_{ij} = \mathrm{softmax}\left(\mathrm{LeakyReLU}\left(e_{ij}\right)\right). \tag{3.9}$$

Consequently, these coefficients are employed to update model features utilizing the GCN convolution rule [50]:

$$h_i^{l+1} = \sigma\left(\sum_{j\in \mathrm{N}(i)} \alpha_{ij} \phi^l h_j^l\right). \tag{3.10}$$

Multi-head Attention Mechanism Multi-head attention mechanism enables the model to learn an attention coefficient through multiple representation subspaces. In order to make the self-attention learning process robust, multi-head attention mechanism strategies are usually adopted [14, 96]. Specifically, take the adopted multi-head attention mechanism in [97] as an example, K independent attention mechanisms perform the above transformation across in K heads (i.e., K independent attention processes) and their resulting features are concatenated together to develop an output feature representation. Subsequently, the final output is obtained by averaging the concatenation of feature representation. This process is formally defined as

$$\begin{cases} h_i^{l+1} = \Big\Vert_{K=1}^{K} \sigma\left(\sum_{j\in \mathrm{N}(i)} \alpha_{ij}^K \phi^K h_j^l\right), \text{Concatenation} \\ h_i^{l+1} = \sigma\left(\frac{1}{K}\sum_{K=1}^{K}\sum_{j\in \mathrm{N}(i)} \alpha_{ij}^K \phi^K h_j^l\right), \text{Averaging} \end{cases}. \tag{3.11}$$

3.1.5 Federated Learning

To overcome these problems, decentralized systems have grown in popularity over the years due to their distributed storage and parallel computing natures — especially the recently emerged federated learning that has been a research hotspot due to its precise stroke on security and privacy issues of ML [121]. Federated learning (FL) is among the most widely-adopted decentralized ML system [63]. FL assumes a scenario that there are N data holders (in different tasks they may be named as "clients", "workers", or "participants"), all of whom hope to train a ML model by merging their data $\{X_1, X_2, \ldots, X_N\}$. However, privacy policies such as GDPR [99] do not allow any direct exposure of raw data, i.e., they cannot directly consolidate their data by $X_{\text{sum}} = X_1 \bigcup X_2, \ldots, \bigcup X_N$ and use to train a model f_{sum}. In such a condition, a FL system describes a learning process where data holders train a model f_{Fed} collaboratively without sharing their respective raw data, and f_{fed} should achieve an accuracy very close to that of f_{sum}. To this end, McMahan *et al.* [63] proposed an algorithm named FedAvg that allows the training data to be kept locally and learns a shared model[4] by aggregating locally-computed updates by a central server. The FedAvg algorithm can be described as

$$f_{\text{Fed}} = \frac{1}{N}\sum_{i=1}^{N} f_i, \tag{3.12}$$

[4] In this thesis, the shared model, global model, and federated model are used interchangeably in the context of decentralized ML system.

where f_i denotes the locally trained model. It is also worth mentioning that all participants of a FL system are actually operating in a grex-box setting, i.e., they can observe the changes of the shared model based on the received model parameters but nothing about the raw data. FL can only preserve data privacy to some extent since nobody could have a holistic view of all training data; notwithstanding, the ideal privacy-preserving effect can only be achieved when all participants are benign and there exists no malicious outsider, which is impractical. Therefore, security and privacy attacks are also nonnegligible for FL systems.

3.1.6 Information Bottleneck

The information bottleneck (IB) [93] principle provides a trade-off solution between the distortion (compression) and the utility of data based on mutual information (MI). Let $I(X,Z)$ denote the MI between the input X and the encoded representation Z, and $I(Y,Z)$ denote the MI between Z and the class label Y. IB principle introduces a distortion function that measures how well Y is predicted from a compressed representation Z compared to its direct prediction from X, and it thus optimizes a trade-off between $I(Y,Z)$ and $I(X,Z)$, that is

$$\min_{Z} \mathscr{L}_{\mathrm{IB}} = -I(Y,Z) + \beta I(X,Z), \tag{3.13}$$

where β is a Lagrange multiplier to control the compression (distortion) extent of Z.

3.2 Relevance of Present Research Works

In the forthcoming subsections, this thesis will introduce three related research efforts. This section as a preamble elucidates the relevance of presenting research works.

With in a generic research objective of the body knowledge, Research works 1 and 2 primarily concentrate on defending against model inversion attacks targeting the structural information of graphs. These two studies are conducted progressively. Research Work 1 investigates an adversary within a MLaaS system, focusing on graph property information, such as graph density. Research Work 2 examines a more complex scenario where the adversary aims to uncover the complete graph structure within the centralized server of a federated graph learning system. Both studies utilize the information bottleneck principle as the foundational approach for developing defense mechanisms.

In contrast, Research work 3 targets an application-level research objective, which addresses a realistic privacy-preserving challenge in a real-world application (the intelligent transportation system). It introduces an end-to-end solution for privacy-preserving traffic forecasting in an outsourced learning system, employing differential privacy techniques.

Collectively, all three studies are situated within contexts that involve graph data in ML, considering potential attackers who seek to access information about graph structural information. Specifically, they all involve the development of approaches tailored to graph data and GNNs. Together, these studies tackle emerging challenges, thereby contributing to the advancement of knowledge in the field.

3.3 Research Work 1: Construct New Graphs using Information Bottleneck Against Property Inference Attacks

Corresponding Publication
Zhang, C., Tian, Z., James, J. Q., & Yu, S. (2023, May). Construct new graphs using information bottleneck against property inference attacks. *In ICC 2023-IEEE International Conference on Communications (pp. 765-770). IEEE.*

3.3.1 Research Background, Question, and Motivation

Recent studies indicated that GNNs are vulnerable to *inference attacks* [72, 137, 138]. Training data samples leave *footprints* on the GNNs, which are recorded by the model gradients or learned embeddings. The attacker can easily trace relevant information of the training graph data using these footprints. It is often assumed that attackers would like to steal graph structures and nodal attributes as they are the fundamental components of a graph [138]. However, some statistical properties of the graph data, such as the number of nodes and the graph density, can also be private. The data curators may not intend to share these properties since they may reveal sensitive information such as business transaction frequency. Also, these properties imply intellectual property since collecting them is laborious. Therefore, the privacy of graph properties is an integral part to the privacy of graph data, which is worthy of in-depth study.

Stealing graph property information from graph embeddings is a realistic assumption — local graph embeddings can be shared to other parties for broad use, which gives access to *man-in-the-middle* [22] attackers. The paradigm of property inference attacks based on graph embeddings can be referred to [137]. The attack model mainly focuses on extracting information from the graph embeddings queried from the GNNs. Inference attacks on graph property are easy to carry out and have a high success rate compared to inference attacks on other targets. Thus, studying the defenses to property inference attacks against GNNs is essential, which has not been given much attention yet [24, 108, 132].

Differential privacy (DP) has been recognized as an effective measure of countering membership inference attacks as this type of attack focuses on the privacy of *individual* records [45]. DP adds controlled noise to target models' gradients or outputs, which can effectively impede the inference. However, graph properties, e.g., graph density and

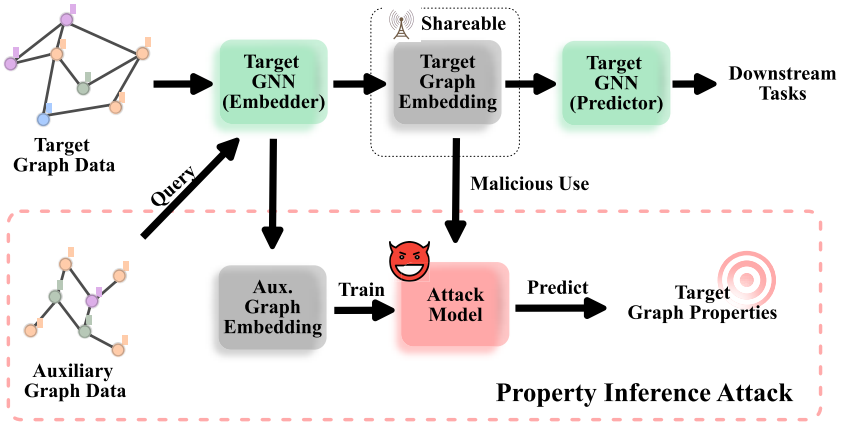


Figure 3.2: (Research Work 1) Threat model: property inference attacks on graph embeddings.

number of edges, are *global*. Previous studies have shown that DP-based defenses are not similarly effective to such property inference attacks [6]. Furthermore, the nature of adding noises makes DP cause an inevitable loss of data utility [138, 140]. This situation motivates us to find an effective way to defend against such attacks targeting global properties.

A possible solution is using *compressive privacy*, which compresses the data to juice out the private parts [51]. Information bottleneck (IB) [93] is a key technique of CP, which compresses the data by squeezing out task-irrelevant information while retaining task-relevant information, and it provides a tradeoff between the two parts. This technique drives us to wonder: *How about using the IB principle to squeeze out relevant information about the graph properties but include sufficient predictive information to achieve the privacy-utility tradeoff of graph data?*

In this work, we propose to leverage the IB principle to defend against the property inference attacks on graph embeddings. Specifically, we leverage IB to construct new graphs, which are predictive yet distorted from the original graph structures. The graph embeddings developed from the new graphs have less information corresponding to the original graph structures, making property inference attackers hard to extract the accurate graph property from them.

3.3.2 Threat Model

In this work, we propose a defense approach to property inference attacks against GNNs. In terms of our adversary, the property inference attacker, we generally follow the

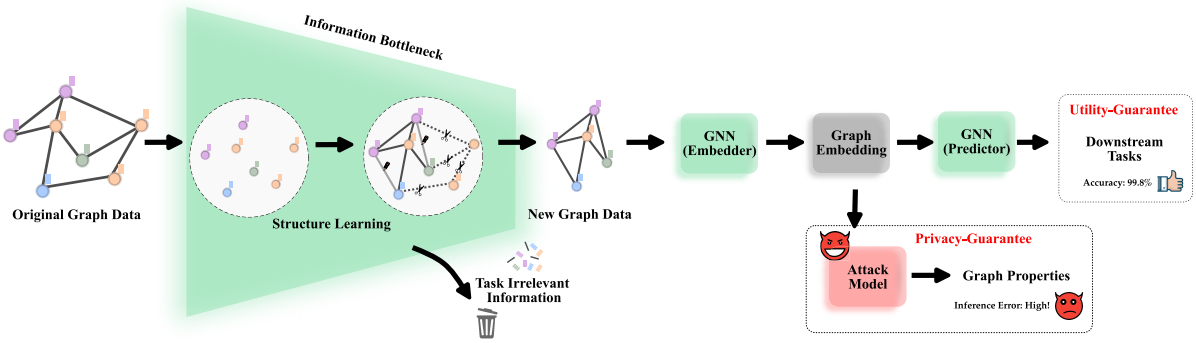


Figure 3.3: (Research Work 1) Schematic of the proposed IB-based defense.

assumption in [137].

Attacker's Knowledge We study a grey-box setting. The attacker can obtain the graph embedding by querying the target GNN model with an input graph. All other knowledge, such as training graphs, architectures and parameters of the target GNN model, is not accessible to the attacker.

Attacker's Goal and Capabilities The attacker aims to extract the property information of the target graph from the graph embedding. Graph embedding-based inference attacks are very realistic since local graph embeddings have been shared with other parties for further graph analysis or learning tasks [136], where data leakage or theft can happen.

We focus on *graph density* inference in this work. To this end, the attacker adopts an attack model A to input the graph embedding. A is a MLP that predicts the graph density [137]. The procedure of the attack can be referred to in Figure 3.2. There also exists an auxiliary dataset D_{aux} that the attacker can access to train the attack model. The graphs in the auxiliary dataset are assumed to be from the same distribution as the target graph. However, different from [137], which formed the property inference into a classification problem, we treat the property inference as a regression problem to evaluate more fine-grained inference results.

Denoting the graph embedding as an intermediate state output by the target GNN, the attacker is optimizing the following attack objective function, that is

$$\arg\min_{A} \mathop{\mathbb{E}}_{G_{\text{aux}} \in D_{\text{aux}}} \left[\sum \mathscr{L}\left(A\left(H_{G_{\text{aux}}}\right), P_{G_{\text{aux}}}\right)\right], \tag{3.14}$$

where P is the true value of graph density.

3.3.3 Method Overview

We propose a new defense approach that uses the information bottleneck for graph data to reconstruct the graph structures to mitigate property inference attacks. Figure 3.3 illustrates the pipeline of our defense approach. We first leverage the IB principle to learn new graphs given the original graphs. Then, we treat the new graphs as the input to train the GNNs. The graph embeddings developed by the GNNs trained on such new graphs will have less information about the original properties of the graphs, which is the core to resisting property inference attacks. We will elaborate on the technical details and analyze the privacy and utility guarantee of the proposed approach in the sequel.

3.3.4 Graph Representation Learning based on Information Bottleneck

Graph information bottleneck extends the IB principle to representation learning on graph data [90, 112, 126]. Given a graph $G \in \mathbb{G} = (X, A)$ and its label Y, the IB-optimized graph is formulated as

$$\arg\min_{G_{\mathrm{IB}}} -I(Y, G_{\mathrm{IB}}) + \beta I(G, G_{\mathrm{IB}}), \tag{3.15}$$

where G_{IB} is constituted of the task-relevant feature matrix X_{IB} and task-relevant adjacency matrix A_{IB}.

Resorting to variational IB [2], we can derive a variational bound which is tractable for Eq. (3.24), that is

$$\begin{aligned} & I(Y, G_{\mathrm{IB}}) - \beta I(G, G_{\mathrm{IB}}) \geq \\ & \quad \frac{1}{N} \sum_{i=1}^{N} \int p(G_{\mathrm{IB}} \mid G_i) \log q_\phi(Y_i \mid G_{\mathrm{IB}}) dG_{\mathrm{IB}} \\ & \qquad - \beta \mathrm{KL}\left(p\left(G_{\mathrm{IB},i} \mid G_i\right) \mid r(G_{\mathrm{IB}})\right), \end{aligned} \tag{3.16}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback Leibler (KL) divergence, $r(G_{\mathrm{IB}})$ is the variational approximation of $p(G_{\mathrm{IB}})$, and $q_\phi(Y \mid G_{\mathrm{IB}})$ and $q_\phi(G_{\mathrm{IB}} \mid G)$ are reparameterized variational approximations to $p(Y \mid G_{\mathrm{IB}})$ and $p(G_{\mathrm{IB}} \mid G)$, respectively.

However, it is hard to estimate $I(G, G_{\mathrm{IB}})$ as the irregularity of graph data. Neural network-based mutual information estimation has been demonstrated to be an available solution to this problem. Denoting the neural network-estimated graph representation as Z_{IB}, we have $I(G, G_{\mathrm{IB}}) \geq I(G, Z_{\mathrm{IB}})$ [9], which is in favor of the IB optimization. Particularly, we use the mutual information estimation adopted in [90] to optimize the IB.

Consequently, we obtain minimally sufficient Z_{IB} by optimizing this objective, which is less prone to overfitting and thus delivers better performance on downstream tasks. Interested readers can refer to [90] for the technical details.

3.3.5 Construct New Graph Structure

We then move forward to how to construct G_{IB} from the results of IB optimization. We leverage the notion of graph auto-encoder (GAE) [49] to construct the new graph structure. We first use MLP to get a latent representation of each node feature by:

$$Z(v) = \mathrm{MLP}(X_v). \tag{3.17}$$

For any two nodes v and u, we have the assignment probability ψ to determine whether the edge (v,u) should be included or not. We consider that two nodes with closing representation are more likely to have an edge. Therefore, we calculate ψ by applying the logistic sigmoid function to the inner product of $Z(v)$ and store the values in the adjacency matrix form, which is formulated as:

$$A_{\mathrm{assign}} = \psi(v,u) = \mathrm{sigmoid}(Z(v)Z(u)^T). \tag{3.18}$$

Subsequently, we follow [112] by employing Gumbel-softmax to make A_{assign} differentiable from Bernoulli distribution. Finally, we can determine the binary adjacency matrix $A_{\mathrm{IB}} = \{a_{u,v}\}$ by conducting a Bernoulli sampling from A_{assign}. We construct G_{IB} according to A_{IB}. We first identify the largest connected component (LCC) in A_{IB} as the new graph structure. For the node feature matrix X_{IB}, we keep the node feature of the nodes included in the new graph structure and discard others.

So far, new graph structure G_{IB} can be either used for neural mutual estimation to further optimize IB, or developing the corresponding graph embedding $H_{G_{\mathrm{IB}}}$ for downstream tasks by forwarding G_{IB} to GNN embedders. We consider graph embeddings $H_{G_{\mathrm{IB}}}$ are more privacy-guaranteed than the original ones when exposed to the threat model. We will justify our hypothesis in the case study.

3.3.6 Privacy and Utility Guarantee by the Information Bottleneck

For the target graph property P_G, the information transmission among P_G, G, and G_{IB} can be described by the Markov chain:

$$P_G \longrightarrow G \longrightarrow G_{\mathrm{IB}} \longrightarrow \hat{P}_G, \tag{3.19}$$

where $\hat{P}_G = A(H_{G_{\text{IB}}})$ denotes the graph property predicted by the attacker using $H_{G_{\text{IB}}}$. We can ensure that property inference attackers cannot derive more information from G_{IB} than G from Eq. (3.25) since 1) G subsumes G_{IB} and 2) G_{IB} is optimized to maximumly squeeze the mutual information with G. According to [9], we know that this assurance also holds for the graph embedding H_G and $H_{G_{\text{IB}}}$ since the amount of information loss from G_{IB} to $H_{G_{\text{IB}}}$ is close to the one from G to H_G when using the same embedder. Eq. (3.25) also implies that information transmission is diminishing, and we can obtain:

$$I(G, P_G) \geq I(G_{\text{IB}}, P_G). \tag{3.20}$$

Therefore, it is easy to derive that the upper bound of property inference attacks using $H_{G_{\text{IB}}}$ equivalent to the attacks using G. On this basis, we can conclude that, $H_{G_{\text{IB}}}$, which is from G_{IB}, will be much less informative in terms of the property inference.

For the label Y, the information transmission among Y, G, and G_{IB} in Eq. (3.24) can be described by the Markov chain:

$$Y \longrightarrow G \longrightarrow G_{\text{IB}} \longrightarrow \hat{Y}. \tag{3.21}$$

We assume that G_{IRR} is the component of G, which is irrelevant to the target Y. We can derive an upper bound for mutual information between G_{IB} and G_{IRR} from [126], that is

$$I(G_{\text{IRR}}, G_{\text{IB}}) \leq I(G, G_{\text{IB}}) - I(Y, G_{\text{IB}}). \tag{3.22}$$

Consequently, optimizing Eq. (3.24) amounts to minimizing $I(G_{\text{IRR}}, G_{\text{IB}})$, making the optimized G_{IB} with less irrelevant information to target Y.

3.4 Research Work 2: Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck

Corresponding Publication
Zhang, C., Wang, W., Yu, J. J., & Yu, S. (2023, July). Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck. *In Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security (pp. 109-121)*.

3.4.1 Research Background, Question, and Motivation

Federated learning (FL) has emerged as a privacy-preserving solution to data-silo problems, where clients can train a shared model collaboratively under the coordination of a central server. FL has been extended to the graph domain to handle the aforementioned problem of isolated graph data, i.e., federated graph learning (FGL). Recent studies indicated that inference attacks, especially model inversion attacks (MIA), can be used to reconstruct private graph structures from leaked information of trained GNN models (e.g., gradients) or graph data (e.g., nodal attributes) [38, 139]. Such attacks are direct threats to the privacy of graph data. It is more challenging to defend FGL systems against MIA. On the one hand, the central server can legitimately acquire local GNN updates under the FL protocol — attackers at the central server side can secretly extract private data information by taking this privilege. On the other hand, in addition to feature attributes, FGL systems have to take great ingenuity to protect the structures of graph data (i.e., topology) as graph structures are also considered valued assets of data contributors. This requirement is beyond the traditional FL system: traditional FL systems only needs to consider the privacy of feature information [121].

Privacy and sensitivity of graph structures. Graph structures embody the intellectual property of data contributors since it is laborious and resource-consuming to collect the relationships among different nodal entities [38]. Furthermore, graph structures may record private social connections or commercial transactions among different nodal entities [78, 81].

In the context, the privacy-preserving capability of FGL systems is crucial [24]. However, the research efforts on the defenses against inference attacks for GNNs is few, and they mainly focus on the protection of nodal attributes rather than graph structures

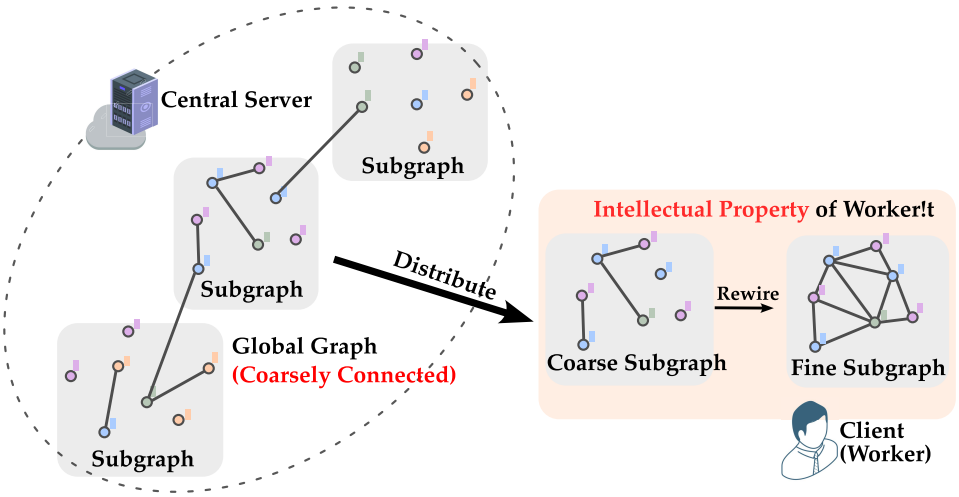


Figure 3.4: (Research Work 2) The central server only has a coarsely connected global graph. The workers have to manually fine-process the subgraph and hence own the intellectual property of such a subgraph structure.

[81, 110, 137]. Their adopted differential privacy (DP) techniques can have many limitation on achieving the utility-privacy tradeoff of data. Furthermore, existing solutions of FGL are usually designed to include more learning processes to improve accuracy. However, under the threats of inference attacks, this paradigm of *addition* can go against to the privacy-preservation since more information are exposed to the attackers. For example, Zhang *et al.* [134] considered the missing edges connecting different subgraphs should be included in a local GNNs' training. Therefore, they introduced a node generator, which requires each client to train the generator to further recover missing neighbors. This will naturally increase the attack surface of gradients exposed to MIA as more information pertaining to the original subgraph are learned. On the other hand, such paradigms inevitably add to the system's communication and computational load. These concerns motivate us to know *whether it is possible to design FGL schemes by doing "subtraction" rather than "addition."*

3.4.2 Threat Model

In this work, the threat model is assumed in a realistic crowdsourcing-based FGL scenario. The curator of the central server is the initial graph data owner, which splits the global graph data into several subgraphs and distributes them to each client (worker). Therefore, (S)he knows the nodal feature and labels of the distributed data. However, the original global graph only has a *coarse* connection pattern. The workers have to

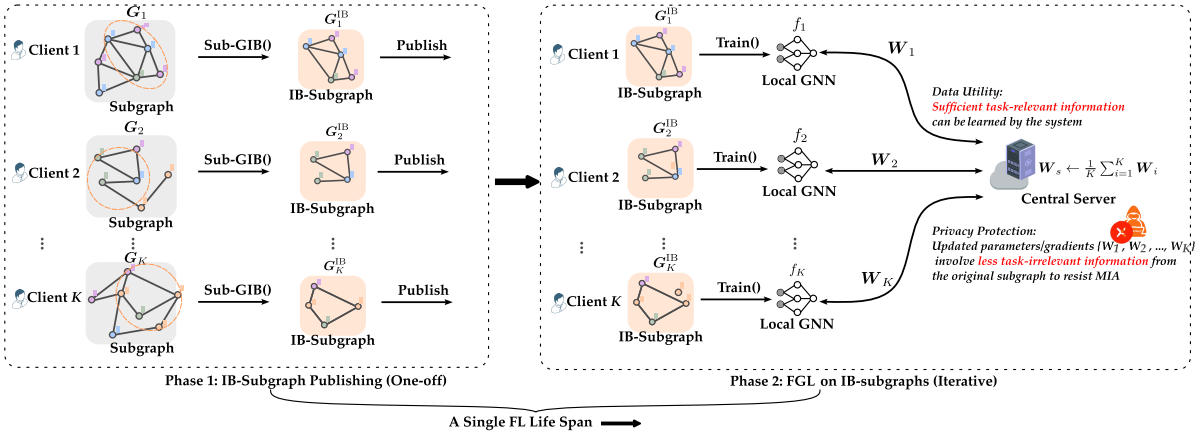


Figure 3.5: (Research Work 2) Schematic of the proposed SOS scheme.

further *rewire* the received subgraphs by collecting/identifying relationships among nodal entities to construct reasonable training sets for the local GNN models—*fine subgraphs*. An illustration is given in Figure 3.4. Since this process is laborious and resource-consuming for the workers, the graph structures of the subgraphs are naturally part of their intellectual properties. It is worth stating additionally that while some studies suggested that it can be more efficient for workers to learn new graph structures from coarse subgraphs [90, 135], the utility of graph data can be compromised. Therefore, we consider the manual collection of graph construction, as a traditional paradigm, to be of existential significance.

We assume that the attacker is the curator of the central server, which is curious about the subgraph structure A_i constructed by the workers. Since the curator can only access the model updates according to the FL protocol, (s)he attempts to leverage model inversion attacks (MIA) to reconstruct the subgraph structures based on the received model updates and the subgraph attributes (s)he has already known. We know that the model update W is associated with graph structure A, where the former is developed by the latter in local GNN training. Due to this correlation, releasing W to the central server will enable him or her to draw some inferences on A. Furthermore, we know that the MIA attack is in a white-box setting in our investigated case. The "malicious central server" assumption is akin to the one in [104]; however, as the attacker in our setting has more knowledge related to the target, the attacks will be much more threatening.

More specifically, such a reconstruction is subsumed to "link stealing attack" [38, 139]: the goal of the attacker is to reconstruct the target client b's graph structure by identifying whether there exists an edge between each node pair of the subgraph given

its local model updates (parameters) W_b, feature information X_b, and label information Y_b. The attack process can be formulated by

$$\max_{A} P(A \mid X_b, Y_b, W_b), \tag{3.23}$$

where P is the posterior possibility that the attacker aims to maximize by finding the adjacency matrix A.

To mitigate the inference threat on A, the client can release a distorted version of either (1) A before local GNN training or (2) W before model updating. In this paper, we adopt the first notion to construct our approach. Additionally, we also compare with a local differential privacy (LDP) method who adopts the second notion in our experiments.

3.4.3 Method Overview

We present the proposed SOS scheme in a top-down manner. We commence from a framework view by introducing the IB-subgraph publishing mechanism for subgraph-level FGL systems. Then, we define the graph information bottleneck in subgraph-level FGL and elaborate on the Sub-GIB approach from mutual information estimation to the final IB-subgraph generation. Lastly, we discuss the privacy and utility of IB-subgraphs.

3.4.4 IB-subgraph Publishing Mechanism for Subgraph-level Federated Learning Systems

As shown in Figure 3.5 and Algorithm 1, the proposed SOS scheme introduces two phases for subgraph-level FGL systems: (1) IB-subgraph publishing; (2) FGL on IB-sbugraphs. In reality, the global graph data can be either static or dynamic. We consider the graphs in a static condition as a single FGL lifespan. That is to say, if any of the subgraphs changes, the previous lifespan is over, and it will proceed to the subsequent FGL lifespan. The proposed approach is incorporated as an *in-processing* one catered for a *single* FGL lifespan.

In the IB-subgraph publishing phase, the system requires each client to generate an IB-subgraph out of the original subgraph. Specifically, we propose the subgraph generation with information bottleneck (Sub-GIB) approach for IB-subgraph generation, which will be introduced later. Given an attributed graph $G = (A, X)$, we name its subgraph developed by Sub-GIB as IB-subgraph denoted by G^{IB}.

Once all the IB-subgraphs are published, the system will proceed to the FGL phase. The clients will hold their original subgraphs and IB-subgraphs locally. Notably, for

Algorithm 1 Brief pipeline of SOS

Input: Number of clusters K, subgraphs $G_i = \{\mathcal{V}_i, \mathcal{E}_i, X_i \mid i \in [K]\}$, GNN model f, learning rate η, number of IB optimization epochs T_{IB}, number of global training epochs T, number of local training epochs T_l.

Output: $\mathscr{R}_s(F(W_s; G))$.

```
// Phase 1: IB-subgraphs Publishing
foreach i ∈ K in parallel do
    foreach epoch t = 1,2,...,T_IB do
        θ_h, θ_r, θ_g, θ_b ← Optimize NN_h, NN_r, NN_l, NN_b via Eq. (3.32)
    B_i ← NN_b(θ_b; GNN(G_i))  G_i^IB ← Construct IB-subgraph via the algorithm described
      in Section 3.4.6 with B_i
// Phase 2: Federated Graph Learning on IB-subgraphs
W_{i,0} ← Initialize GNN model f  foreach epoch t = 1,2,...,T do
    # Updates local GNN model's weights  foreach i ∈ K in parallel do
        foreach epoch e = 1,2,...,T_l do
            W_{i,t} ← W_{i,t-1} − η · ∇L(f(G_i^IB), Y_i)
    # Updates aggregation at central server (FedAvg) and broadcasting  W_{s,t} ←
      (1/K) Σ_{i=1}^{K} W_{i,t}
return R_s(F(W_s; G))
```

privacy concerns, the clients will train their local GNN model on the IB-subgraphs instead of the original subgraph. Thus, original subgraphs will not be involved in the FL training. This design embodies the advantage of the proposed approach: as the local GNN model is trained on IB-subgraph data, less "footprint" associated with the original subgraph will be left on the local model updates. As thus, if the central server or any outsider (if there is a gradient leakage) intends to infer the original subgraph information (cf. the model inversion attack scenario described in Section 3.4.2), the inference effect would be impeded to a great extent. Another advantage is that according to the FL protocol, one cannot know whether local models were trained on the original subgraphs or any processed subgraphs. Even if they successfully reconstruct the training graphs (i.e., IB-subgraphs), they are not identical to the original subgraphs.

Furthermore, the IB-subgraph publishing phase is designed to be *one-off* in a FGL lifespan — it will only be executed in the initializing stage for one time. As the subgraphs are all static in a lifespan, it is of no necessity to publish their IB-subgraphs iteratively along with the training epochs. One may notice that the FGL phase in the proposed scheme remains the same as the naive FedAvg algorithm, it does not introduce any additional actions to both the central server and clients. These designs endow the

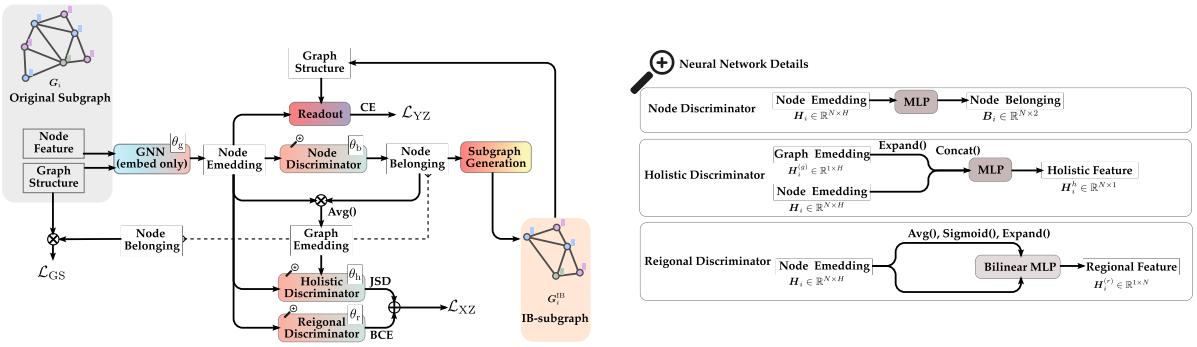


Figure 3.6: (Research Work 2) Architecture of Sub-GIB. GNN(·) and Readout(·) operations differ from the practically adopted GNN models in the FL systems. Negative samples for the holistic discriminator and regional discriminator are generated by row-wise shuffling the feature matrix X_i but keep the original adjacency matrix, i.e., $\tilde{G}_i = (A_i, \tilde{X}_i)$.

proposed scheme with some advantages. First, the algorithmic complexity of the proposed scheme is close to that of vanilla FGL, making it computationally and communicationally sparing. Second, The proposed scheme is both model-agnostic and FL algorithm-agnostic, which can be integrated with different FL algorithms. Moreover, the proposed scheme is orthogonal to many of the existing subgraph-level FGL approaches, such as the splitting learning-based approach in [145], which can be orchestrated to improve the FGL performance further.

3.4.5 Sub-GIB: Subgraph Generation with Information Bottleneck

Subgraph Information Bottelneck in Graph Data Generally, Sub-GIB extends the IB principle, which casts about for most predictive but compressed G^{IB} by: (1) minimizing the MI between G^{IB} and G, i.e., $I\left(G, G^{\mathrm{IB}}\right)$; (2) maximizing the MI between G^{IB} and Y, i.e., $I\left(Y, G^{\mathrm{IB}}\right)$. The optimization objective of Sub-GIB can be formulated as

$$\min_{G^{\mathrm{IB}} \in \mathbb{G}^{\mathrm{IB}}} \mathscr{L}_{\mathrm{GIB}} = -I\left(Y, G^{\mathrm{IB}}\right) + \beta I\left(G, G^{\mathrm{IB}}\right), \tag{3.24}$$

where $\mathbb{G}^{\mathrm{IB}}$ denotes the subgraph search space of G.

We generalize the Sub-GIB for the learning of a centralized GNN to the subgraph-level FGL scenario. In contrast to centralized systems, the FGL system involves multiple clients, each of which already owns a smaller subgraph of a global graph. In such a federated scenario, Sub-GIB seeks to further recognize a subgraph of the subgraph owned by these clients — we name this process as *Subgraph-Out-of-Subgraph* (SOS). Each IB-subgraph is expected to embody *minimal-sufficient* information. *Sufficient*

requires that IB-subgraph is as informative as possible regarding the target to develop accurate predictions. *Minimal* promotes the IB-subgraph to be distorted from the original subgraph, and the information that is irrelevant to the prediction can be juiced out as much as possible. In this work, we in particular leverage the *Minimal* feature to achieve the privacy preservation of the IB-subgraph.

Furthermore, we only consider the graph structural compression in this work. That is to say that the encoded G_i^{IB} will have a new and smaller graph structure; however, the nodal feature will not be encoded. As shown in Figure 3.5, the attributes of the nodes that are eliminated after the compression will be naturally discarded.

Neural Network-Powered Mutual Information Estimation In this work, we commence by using VIB [2] to optimize the Sub-GIB for each client's IB-subgraph generation. Describing the information transmission among Y, G, and G^{IB}, the Markov chain in VIB is

$$Y \overset{p(G|y)}{\longrightarrow} G \overset{q_{\theta_2}(G^{\mathrm{IB}}|G)}{\longrightarrow} G^{\mathrm{IB}} \overset{q_{\theta_1}(y|G^{\mathrm{IB}})}{\longrightarrow} \hat{Y}, \tag{3.25}$$

where $q_{\theta_1}\left(y \mid G^{\mathrm{IB}}\right)$ and $q_{\theta_2}(G^{\mathrm{IB}} \mid G)$ are reparameterized variational approximations to $p\left(y \mid G^{\mathrm{IB}}\right)$ and $p\left(G^{\mathrm{IB}} \mid G\right)$, respectively.

Resorting to the deduction in [126], a tractable variational lower bound for the first term of Eq. (3.24) for each client i can be derived as

$$\begin{aligned} I\left(Y_i, G_i^{\mathrm{IB}}\right) &= \int p\left(y, G_i^{\mathrm{IB}}\right) \log p\left(y \mid G_i^{\mathrm{IB}}\right) dy\, dG_i^{\mathrm{IB}} + H(Y_i) \\ &\geq \int p\left(y, G_i^{\mathrm{IB}}\right) \log q_{\theta_1}\left(y \mid G_i^{\mathrm{IB}}\right) dy\, dG_i^{\mathrm{IB}}, \end{aligned} \tag{3.26}$$

where the entropy of labels $H(Y_i)$ can be neglected as it is independent of the optimization. In the practical optimization, Eq. (3.26) can be transformed into the node classification loss between the real labels Y_i and the labels predicted on G_i^{IB}, which can be formulated as

$$\min_{G_i^{\mathrm{IB}}} \mathscr{L}_{\mathrm{YZ}} = \mathscr{L}_{\mathrm{ce}}(f_{\theta_{\mathrm{g}}}(G_i^{\mathrm{IB}}), Y_i), \tag{3.27}$$

where $f_{\theta_{\mathrm{g}}}(\cdot)$ is the adopted GNN-based classifier in the FGL system, and $\mathscr{L}_{\mathrm{ce}}(\cdot)$ denotes the cross entropy loss. As some of the nodes in G_i are eliminated in G_i^{IB}, the loss evaluation are only performed on the nodes included in G_i^{IB}.

The second term of Eq. (3.16) is tractable for the primitive VIB only if the data's empirical distribution information is known to further compute the MI. Nevertheless,

the discreteness and non-IID of graph-structured data renders its empirical distribution untraceable [112]. In other words, we cannot find a proper prior distribution $q_{\theta_2}(G^{\mathrm{IB}})$ (cf. $r(z)$ in Eq. (3.16)) for G^{IB}.

To make $I\left(G_i, G_i^{\mathrm{IB}}\right)$ tractable, we introduce a neural network-powered mutual information estimation approach to instantiate and minimize $I\left(G_i, G_i^{\mathrm{IB}}\right)$. We adapt ideas from [40] to consider both *holistic* and *regional* features[5]. We first introduce a neural network-based extractor $f_h(\cdot)$ to learn implicit graph-level representation (holistic feature). Basically, $f_h(\cdot)$ consists of an encoder and a discriminator $f_{\theta_{\mathrm{h}}}(\cdot)$. The encoder here shares the same architecture and parameters with $f_{\theta_{\mathrm{g}}}(\cdot)$ in Eq. (3.27). Thus, we have $f_h = f_{\theta_{\mathrm{g}}} \circ f_{\theta_{\mathrm{h}}}$. Particularly, we define the holistic feature developed by $f_h(\cdot)$ as the graph-level information — summarizing the graph feature as a whole. Different from prior work [90, 126], we propose to use Jensen-Shannon divergence (JSD) to evaluate MI, which requires a smaller number of negative samples and demonstrates better stability in practice. Based on a JSD-based MI estimation[71], we can formulate the objective specific to the holistic feature as

$$\begin{aligned} \mathscr{L}_h = &- \log \mathbb{E}_{\tilde{G}_i \in p(G_i, G_i^{\mathrm{IB}})} \left(1 + e^{f_h(\tilde{G}_i)}\right) \\ &- \log \mathbb{E}_{G_i \in p(G_i)} \left(1 + e^{-f_h(G_i)}\right), \end{aligned} \tag{3.28}$$

where $\tilde{G}_i$ represents the negative samples from the joint distribution $p\left(G_i, G_i^{\mathrm{IB}}\right)$, which is instantiated by row-wise shuffling the feature matrix X_i but keep the original adjacency matrix, i.e., $\tilde{G}_i = (A_i, \tilde{X}_i)$.

We consider regional features as specific node-level representations contributing to the node classification. Correspondingly, let $f_r(\cdot)$ be the neural network-based extractors for the regional feature. Similar to $f_h(\cdot)$, $f_r(\cdot)$ adopts $f_{\theta_{\mathrm{g}}}(\cdot)$ as the encoder but with a specific discriminator $f_{\theta_{\mathrm{r}}}(\cdot)$, i.e., $f_r = f_{\theta_{\mathrm{g}}} \circ f_{\theta_{\mathrm{r}}}$. The details of $f_{\theta_{\mathrm{r}}}(\cdot)$ and $f_{\theta_{\mathrm{r}}}(\cdot)$ are illustrated in Figure 3.6. Incorporating the negative samples, we use binary cross-entropy (BCE) to evaluate the loss of regional MI, which can be formulated as

$$\mathscr{L}_r = \mathscr{L}_{\mathrm{ce}}(f_r(G_i), \mathbf{1}) + \mathscr{L}_{\mathrm{ce}}(f_r(\tilde{G}_i), \mathbf{0}), \tag{3.29}$$

where $\mathbf{1}$ and $\mathbf{0}$ are the all-one and all-zero vectors respectively representing the positive and negative labels. Combining Eq. (3.28) and (3.29), we can obtain the objective estimating $I\left(G_i, G_i^{\mathrm{IB}}\right)$:

$$\min_{\theta_{\mathrm{h}}, \theta_{\mathrm{r}}} \mathscr{L}_{\mathrm{XZ}} = \mathscr{L}_h + \gamma \mathscr{L}_r, \tag{3.30}$$

[5] we use "holistic" and "regional" to describe "global" and "local" here to avoid abuse.

where γ is a multiplier to control the tradeoff between the two parts. This design is different from the one in [126] which only considers the holistic feature for the graph classification task. We believe that involving holistic features and regional features will further improve the quality of generated IB-subgraphs, and the hypothesis is justified in our experiments.

To ensure that G_i^{IB} has a compact graph structure with sufficient feature smoothness, we additionally introduce a loss term with respect to the generated graph structure based on the one in [139], which is formulated as

$$\min_{\theta_{\mathrm{h}},\theta_{\mathrm{r}}} \mathscr{L}_{\mathrm{GS}} = \mathrm{Tr}(B_i^T L_i B_i), \tag{3.31}$$

where L_i is the Laplacian adjacency matrix of G_i, B_i is a node belonging of G_i^{IB} (the computation will be detailed in Section 3.4.6), and $\mathrm{Tr}(\cdot)$ represents the trace of a matrix.

Combining Eq. (3.27), (3.30), and (3.31), we can obtain the final objective function of Sub-GIB, that is

$$\min_{\theta_{\mathrm{h}},\theta_{\mathrm{r}},\theta_{\mathrm{g}},\theta_{\mathrm{b}}} \mathscr{L}_{\text{Sub-GIB}} = \mathscr{L}_{\mathrm{YZ}}(\theta_{\mathrm{g}},\theta_{\mathrm{b}}) + \beta \mathscr{L}_{\mathrm{XZ}}(\theta_{\mathrm{h}},\theta_{\mathrm{r}}) + \mathscr{L}_{\mathrm{GS}}(\theta_{\mathrm{g}},\theta_{\mathrm{b}}). \tag{3.32}$$

In practice, Eq. (3.32) is optimized in a bi-level manner: we first optimize $\mathscr{L}_{\mathrm{XZ}}(\theta_{\mathrm{h}},\theta_{\mathrm{r}})$ and fixed θ_{h} and θ_{r} to further optimize Eq. (3.32) as a whole. Corresponding to Eq. (3.25), we have $\theta_1 = \{\theta_{\mathrm{g}}\}$ and $\theta_2 = \{\theta_{\mathrm{h}},\theta_{\mathrm{r}},\theta_{\mathrm{b}}\}$

3.4.6 IB-subgraph Generation Algorithm

We design an algorithm as the last step of Sub-GIB to generate the G_i^{IB} for publication. When Sub-GIB is optimized, we first use a node discriminator (denoted by $\mathrm{NN}_{\mathrm{b}}(\cdot)$) to generate a node belonging to each node, as shown in Figure 3.6. The node belonging evaluates each node by two score: $\mathrm{IN} \in [0,1]$ and $\mathrm{OUT} = 1 - \mathrm{IN}$ which is the probability that the node should be included or not included in G_i^{IB}, respectively.

Then, we use Top-K algorithm to sort out all the nodes' IN scores to decide the ones to be reserved in the IB-subgraph. Let ρ be the ratio controlling the number of the nodes that are reserved in G_i^{IB}, we have $N_i^{(\min)} = \mathrm{int}(\rho N_i)$. Thus, the first $N_i^{(\min)}$ points with the largest IN score values in subgraph G_i will be retained. We define a downsampling function $\mathrm{Ds} \in \{-1,+1\}$ to perform this process. The downsampling function is defined as

$$\mathrm{Ds}(j) = \begin{cases} 1 & \text{if } j \in [N_i^{(\min)}] \\ -1 & \text{if } j \notin [N_i^{(\min)}] \end{cases}. \tag{3.33}$$

We then perform the downsampling operation on the adjacency matrix, which is formulated as

$$A_i^* = \{a_j^*\} = \{\frac{1}{2}(1 + \mathrm{Ds}(j))a_j\}, \tag{3.34}$$

where $a_j \in A_i$ and a_j^* are the entry of node j in the adjacency matrix before and after the downsampling, respectively. Then, we can use the updated adjacency matrix $A_i^* \in \mathbb{R}^{N_i^{(\min)} \times N_i^{(\min)}}$ and the corresponding feature matrix $X_i^* \in \mathbb{R}^{N_i^{(\min)} \times d}$ to construct IB-subgraph $G_i^{\mathrm{IB}} = (A_i^*, X_i^*)$.

Such an algorithm ensures that at least $N_i^{(\min)}$ nodes can be retained, eliminating the possibility that no nodes are reserved. The edges connecting the discarded nodes and the retained nodes will be naturally dropped — if some nodes supposed to be retained, however, become singletons after these edges' dropping, they will be discarded as well.

3.4.7 Discussion on Privacy and Utility of IB-subgraph

Privacy Analysis Let W_i and W_i^{IB} be the local model updates developed by the original subgraph and IB-subgraph, respectively. The privacy leakage can be measured by the mutual information between the local model updates trained on IB-subgraphs and the target private subgraph structure, i.e., $I(W_i^{\mathrm{IB}}, A_i)$. According to the Markov chain stated in Eq. (3.25), it can be ensured that W_i^{IB} cannot contain more information about the original subgraph structure A_i than W_i since $G_i = (A_i, X_i)$ subsumes $G_i^{\mathrm{IB}} = (A_i^{\mathrm{IB}}, X_i^{\mathrm{IB}})$. We can further derive the diminishing mutual information with A_i from W_i to W_i^{IB}, i.e.,

$$I(W_i, A_i) \geq I\left(W_i^{\mathrm{IB}}, A_i\right), \tag{3.35}$$

where the inequality is irreversible. We can deduce that the upper bound of MIA using W_i^{IB} is equivalent to MIA using W_i. By the same token, the upper bound of MIA using $(W_i^{\mathrm{IB}}, X_i, Y_i)$ is equivalent to MIA using (W_i, X_i, Y_i) As G_i^{IB} is optimized to maximumly juice out the task-irrelevant MI with G_i, MIA using $(W_i^{\mathrm{IB}}, X_i, Y_i)$ will be much less effective.

Moreover, most of the existing privacy-targeted FL approaches modify the model updates to protect data privacy [3, 144]. While these approaches can help defend against MIA, if the local datasets are actively hacked and data privacy will also be compromised. An inherent advantage of our proposed IB-subgraph is that the publishing mechanism isolates the original subgraphs, which are the privacy carriers. For example, the original subgraphs can be further stored in a trusted execution environment [80] at the client

side to guarantee privacy protection. This advantage enables the privacy guarantee to hold in broadening threat scenarios.

Utility Analysis Assume that G_i^{irr} is the subgraph of G_i which is irrelevant to the target Y_i. Following [126], an upper bound for MI between G_i^{IB} and G_i^{irr} is derived as

$$I\left(G_i^{\text{irr}}, G_i^{\text{IB}}\right) \leq I\left(G_i, G_i^{\text{IB}}\right) - I\left(Y_i, G_i^{\text{IB}}\right). \tag{3.36}$$

Eq. (3.36) proved that G_i^{IB} is dependent on G_i^{irr}. Thus optimizing Eq. (3.24) will be equivalent to minimize $I\left(G_i^{\text{irr}}, G_i^{\text{IB}}\right)$, making the optimized G_i^{IB} would be with less irrelevant information to target Y_i.

Additionally, since we apply the proposed approach in federated scenarios, two concerns pop up. 1) *Is the new dataset compatible with the original GNN model due to the change in dataset size?* 2) *Will the down-sampling process on the local graph datasets affect the overall system training effect?*

For the first concern, we treat the learning process of GNN models in an inductive way, therefore, the change in graph size will not cause any incompatibility problem.

For the second concern, we would like to mention that the paradigm of the proposed approach is akin to dropout-related FL approaches [16, 62], where some training elements (e.g., participated clients, neuron links, or model weights) are dropped out in the training procedure. The difference is that the majority of dropout-related FL approaches, such as Federated Dropout [16] select a subset of the *shared model* to locally train and update while we select a subset (IB-subgraph) of the *local dataset* (original subgraph) to locally train the shared model. On the one hand, the performance of federated aggregation algorithms (e.g., FedAvg) has been demonstrated to be robust to even benefit from these dropout operations [16]. On the other hand, compared with the change in model weights or participated clients, which directly influences the model aggregation on the server side, the change in local training data samples will have less influence on the model aggregation. Our scheme is similar to dropping out some local training data samples in the FL systems on regular data (e.g., images). While the data samples in node classification tasks, i.e., nodes, may strongly correlate to each other due to the existence of edges, the IB principle enables the proposed Sub-GIB method to preserve useful edges in the process of preserving task-relevant information. Once local models can effectively learn predictive information from these IB-subgraphs by optimizing local empirical risk $\mathscr{R}_i$, model aggregation algorithms are capable of handling these model updates to develop a generalized and accurate global model [56, 133]. Meanwhile, one

limitation is that the influence of non-IIDness between different IB-subgraphs on the FGL performance is not specifically investigated in this work, which can be a possible direction for future work.

3.5 Research Work 3: A Topological Information Protected Federated Learning Approach For Traffic Speed Forecasting

Corresponding Publication
Zhang, C., Zhang, S., James, J. Q., & Yu, S. (2021). FASTGNN: A topological information protected federated learning approach for traffic speed forecasting. *IEEE Transactions on Industrial Informatics, 17(12), 8464-8474.*

3.5.1 Research Background, Question, and Motivation

With the emerging Federated Learning (FL) technology, the collaborative problems have been vastly resolved [121]. FL serves as a learning framework for multiple data providers, allowing providers to build an effective model collaboratively while keeping their data locally. Comprehensive and successful cases have demonstrated that FL can trade off between model performance and privacy [53, 60]. While existing FL frameworks have been successfully applied to many deep learning-based approaches, we found few cases involving GNN-based models. There are two main challenges to combining FL and GNN-based models. First, unlike regular deep learning-based models, GNN-based models need to handle not only the input data feature but also topological information. Existing FL aggregation algorithms are not capable of handling topological information, which may limit their use in GNN scenarios. Second, the conventional FL framework can only protect the privacy of the data feature. In intelligent transportation systems (ITS), the topological information privacy is also important since the topological information may contain sensitive information (e.g., the relationships among mobile data contributors, the number of deployed sensor stations).

To address the above two problems, we propose a FL framework named Federated Attention-based Spatial-Temporal Graph Neural Networks (FASTGNN). The proposed framework integrates a novel FL strategy towards topological information protection and a GNN-based model named Attention-based Spatial-Temporal Graph Neural Networks (ASTGNN) for traffic speed forecasting. Specifically, in the proposed fl strategy, we introduce a differential privacy (DP)-based local-network adjacency matrix preserving approach, and it enables each organization's topological information in the FL framework can be well-preserved. A local-network topological information aggregation mechanism is also devised, which allows the local models to take advantage of a DP-processed

global topological information to guarantee its performance. In the proposed ASTGNN model, a graph attention mechanism and Gated Recurrent Units networks are adopted, and they make ASTGNN possess excellent spatial-temporal feature learning capacity for developing accurate network-wide traffic speed predictions. In such configuration, FASTGNN can develop promising traffic speed forecasting without compromising privacy.

3.5.2 Traffic Speed Forecasting on Transportation Networks

A transportation network can be represented by an undirected graph, $G = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is the set of nodes which we define each node as a road segment and $\mathcal{E}$ is the set of edges, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of G where N is the number of nodes in G. $\forall\ v_i, v_j \in \mathcal{V}$, if v_i and v_j are connected, $(v_i, v_j) \in \mathcal{E}$ and entry $A_{ij} = 1$ (otherwise $A_{ij} = 0$). Denote the traffic speed observed on G as a graph-wide feature matrix $X \in \mathbb{R}^{N \times Q}$ where Q is the number of incorporated features of each node. Let vector $X^t \in \mathbb{R}^N$ denote the traffic speed observation at time t, the problem can be thus defined as learning a function $f(\cdot)$ to develop traffic speed predictions $\hat{X}^{t+1}, \hat{X}^{t+2}, \ldots, \hat{X}^{t+s}$ in the following s time stamps, given historical traffic speed observations of T stamps $X^{t-T+1}, X^{t-T+2}, \ldots, X^t$.

3.5.3 Federated Learning on Transportation Networks

In this work, we construct the FL framework for traffic speed forecasting on the transportation network. We define a "global-network" G as the entire transportation network of an area. This area is divided and conquered by several organizations (e.g., companies, governments). Let $\mathcal{O} = \{O_1, O_2, \ldots, O_p\}$ denote the organization set where p is the number of organizations. Thus, each organization operates a local-network G_i^* of G. Let $\mathcal{G}^* = \{G_1^*, G_2^*, \ldots, G_p^*\}$ denote the local-network set. The respective databases of these organizations are D_i, which collect traffic speed data from their operated local-networks. Particularly, we have $D_i = (X_i^*, Z_{G_i^*})$ where X_i^* and $Z_{G_i^*}$ are the historical traffic speed data and topological information (e.g., road connectivity) collected from local-network G_i^*, respectively. Additionally, this study is based on the assumption that the organizations do not have overlapping regions and data with each other, i.e., for any two organizations i and j, $D_i \cap D_j = \emptyset$. This is a common assumption among the literature, see [19, 31, 59] for some examples. Our goal is to train a powerful model in the cloud that can predict the global-network-wide traffic speed with local traffic speed data from D_i. Nonetheless, due to privacy concerns, these organizations are prohibited from sharing the raw traffic

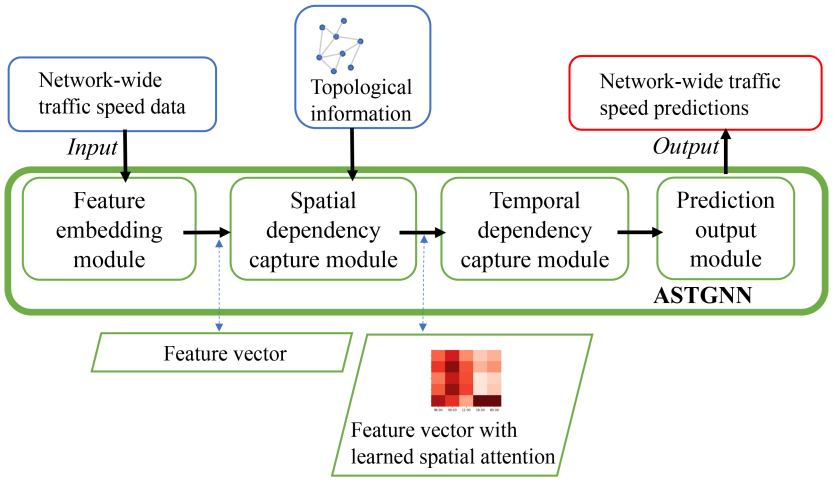


Figure 3.7: (Research Work 3) The framework of ASTGNN.

data and the topological information of their operated local-networks (i.e., they can only access their respective local-networks).

To achieve our goal under the aforementioned privacy constraints, it is required to adopt a Secure Parameter Aggregation Mechanism (SPAM) [121] in the FL framework. Particularly, the graph-based deep learning model M_i constructed by each organization O_i computes a group of updated model parameters ϕ_i utilizing the local training data from D_i and the topological information of the corresponding local-network G_i^*. After all the organizations complete the parameters' updating, their respective parameters are uploaded to the cloud. The global model is finally developed by aggregating these uploaded parameters. SPAM guarantees that no traffic speed data leakage happens among the organizations.

3.5.4 Method Overview

We present the proposed Attention-based Spatial-Temporal Graph Neural Networks (ASTGNN) as the local graph deep learning-based model for traffic speed forecasting. Then, we elaborate on our proposed FL framework FASTGNN (Federated-ASTGNN).

3.5.5 Attention-based Spatial-Temporal Graph Neural Networks (ASTGNN)

For the network-wide traffic speed forecasting problem, we propose ASTGNN as the local forecasting model. As illustrated in Fig. 3.7, ASTGNN consists of four modules: feature

embedding module, spatial dependency capture module, temporal dependency capture module, and prediction output module.

Feature Embedding Module Feature embedding module transforms the input time-series data into feature vectors, which can be processed by the spatial dependency capture module afterward. Specifically, given a sequence (length = T) of network-wide time-series speed values $X^1, X^2, \ldots, X^T$, each feature vector can be formulated as

$$h^t = [X^{t-F+1}, X^{t-F+2}, \ldots, X^t], \tag{3.37}$$

where $h^t \in \mathbb{R}^{F \times N}$ is the network-wide feature vector at time t; F is the dimension of the vector whose physical meaning is equivalent to the past window size (i.e., T). That means that we actually embed a sequence of speed data whose length is the same as the past window size into a feature vector. In this way, we can obtain a sequence of feature vectors $h^1, h^2, \ldots, h^T$.

Spatial Dependency Capture Module Spatial Dependency Capture Module is used to exploit the spatial dependency among different road segments (nodes) in the transportation network (graph). We construct this module by following Graph Attention Networks (GAT) [97], which utilizes the attention mechanism to obtain the spatial correlations. The operational steps of this module can be described as the following steps:

i. We commence by computing the attention score. For any ordered pair of nodes $(v_i, v_j) \in \mathcal{V}$, the attention score v_i perceive from v_j can be formulated as

$$Att_{v_i \leftarrow v_j} = a^T \cdot \text{concat}(Wh_i^t, Wh_j^t), \tag{3.38}$$

where $Att_{v_i \leftarrow v_j}$ denotes the attention score, h_i^t and h_j^t are feature vector of node v_i and v_j at time t respectively, $W \in \mathbb{R}^{F^h \times F}$ is a weight matrix which can transform feature vector into a higher-level dimension F^h, concat$(\cdot)$ denotes the concatenation operation, $a \in \mathbb{R}^{2F^h}$ is a weight vector, and $\cdot^T$ denotes the transposition operation.

ii. Subsequently, we use activation functions to normalize the attention score and obtain the attention efficient, which can be expressed as

$$\alpha_{v_i \leftarrow v_j}^t = \text{softmax}\left(\text{LeakyReLU}\left(e_{ij}\right)\right), \tag{3.39}$$

where $\alpha_{v_i \leftarrow v_j}^t \in [0, 1]$ denotes the attention coefficient, LeakyReLU$(\cdot)$ denotes the Leaky Rectified Linear Units activation function, and softmax$(\cdot)$ denotes the softmax activation function.

iii. Next, we filter the obtained attention coefficient to survive the attention coefficients only for connected node pair, which can be formulated as

$$\hat{\alpha}^t_{v_i \leftarrow v_j} = \alpha^t_{v_i \leftarrow v_j} \odot A_{ij}, \tag{3.40}$$

where A_{ij} is the entry for node v_i and v_j in the adjacency matrix A, and $\odot$ denotes the Hadamard product. We can deduce that when $A_{ij} = 1$, the attention coefficient survives, otherwise be discarded (i.g., equal to 0).

iv. Finally, the attention coefficients are employed to update the feature vector of node v_i, which can be formulated as

$$\hat{h}^t_i = \sigma\left(\sum_{j \in \mathrm{N}(i)} \hat{\alpha}^t_{v_i \leftarrow v_j} W^o h^t_j\right), \tag{3.41}$$

where $\hat{h}^t_i$ is the updated feature vector of node v_i at time t which is regarded as the output of this module, $\mathrm{N}(i)$ is the set of immediately adjacent nodes of node v_i, W^o is a weight matrix.

Temporal Dependency Capture Module The temporal dependency capture module is designed to learn the potential temporal dependency of data. We employ two layers of GRU neural networks in this module. GRU introduces a collection of gating units and cell states to process the input information, which can solve the gradient vanishing problem in the learning process. The gating units have two types, i.e., reset gate r and update gate z. Given the input data x^t[6], the hidden layer output h^t_g can be computed by

$$z^t = \sigma\left(W^{(z)} x^t + U^{(z)} h^{t-1}_g\right), \tag{3.42}$$

$$r^t = \sigma\left(W^{(r)} x^t + U^{(r)} h^{t-1}_g\right), \tag{3.43}$$

$$\tilde{h}^t_g = \tanh\left(W x^t + r^t \odot U h^{t-1}_g\right), \tag{3.44}$$

$$h^t_g = z^t \odot h^{t-1}_g + (1 - z^t) \odot \tilde{h}^t_g, \tag{3.45}$$

where $W^{(z)}$, $W^{(r)}$, $U^{(z)}$, $U^{(r)}$ are the weight matrices connecting x^t and h^{t-1}_g to two gates, $\tilde{h}^t_g$ is the intermediate candidate activation.

[6] x^t is the output of the spatial module, i.e., $\hat{h}^t$, we use x^t here to avoid confounding notations

Prediction Output module A fully-connected layer is employed in this module to produce the traffic speed of future s time stamps. Such a linear transformation conducted by a full-connected layer is formulated as

$$\hat{X}^{t+1}, \hat{X}^{t+2}, \ldots, \hat{X}^{t+s} = W^{(fc)} h_g^t + b, \tag{3.46}$$

where $W^{(fc)} \in \mathbb{R}^{C \times s}$ is a weight matrix that maps the hidden output of GRU in the temporal module to s prediction output, and b is the bias.

3.5.6 Federated Learning Framework for ASTGNN (FASTGNN)

In the previous subsection, we introduce the proposed ASTGNN model for traffic speed forecasting. In this subsection, we introduce the proposed FL framework for ASTGNN, namely, FASTGNN. As illustrated in Fig. 3.8), each organization operates an ASTGNN as the local model, whose input is traffic speed data and topological information from its local traffic database. The DP-based adjacency matrix preserving algorithm is implemented at the organization end to protect the local topological information. The cloud server is in charge of aggregating the preserved local topological information and ASTGNN model parameters and broadcast the aggregated ones. The detailed elaboration of related algorithms can be seen in the following.

FASTGNN Communication Protocol Each organization can only access its own traffic data and local-network topological information for local models' training. One concern with training local models using only local-network topological information is that local-networks do not contain all the essential topological information for computing attention coefficients by ASTGNN. This issue may lead to the final low learning effect. Thus it is requisite to feed the topological information of the global-network to the local models for obtaining better results. To achieve this without compromising the privacy of local-network topological information, we propose a FL communication protocol as following:

i. The organizations apply a privacy-preserving algorithm to its local-network topological information and obtain preserved topological information $Z_{G_i^*}^{(pp)}$.

ii. The organizations upload $Z_{G_i^*}^{(pp)}$ to the cloud server, the latter aggregate the uploaded $Z_{G_i^*}^{(pp)}$ and develop one of the global-network $Z_G^{(pp)}$.

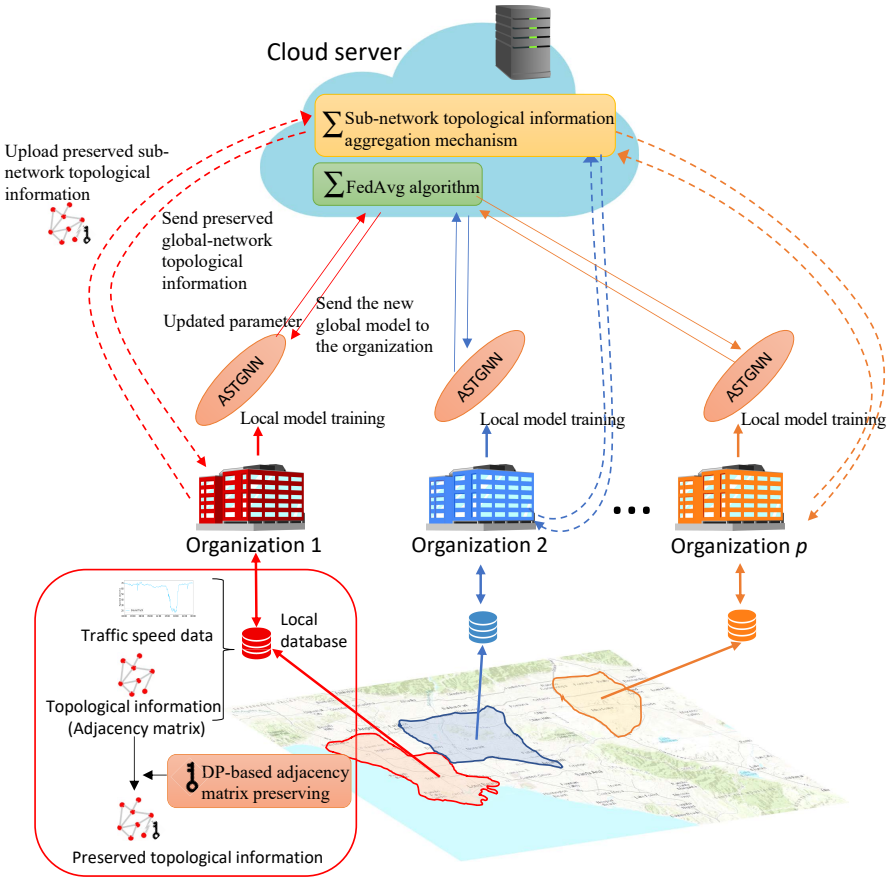


Figure 3.8: (Research Work 3) The framework of FASTGNN.

iii. The cloud distributes copies of the global model and $Z_G^{(pp)}$ to all organizations, and each organization trains its copies using local data.

iv. Each organization uploads the learned model parameters ϕ_i to the cloud. Since private data and topological information are not shared in the entire process, the privacy preservation is guaranteed.

v. The cloud server aggregates ϕ_i by SPAM as introduce before to build a new global model. Subsequently, the new model is distributed to the organizations.

We then detail the adopted privacy-preserving algorithm for topological information, the local-network topological information aggregation mechanism, SPAM, and the entire FL process.

Differential Privacy-based Adjacency Matrix Preserving In this work, we regard the adjacency matrix of local-network as the carrier of topological information. We introduce a differential privacy (DP)-based approach to provide privacy-preserving to the adjacency matrix while keeping its utility in the learning process of ASTGNN. This approach is based on [1], which leverages the theories of DP and random matrix to the adjacency matrix privacy-preserving. Specifically, given the to be preserved adjacency matrix $A \in \mathbb{R}^{N\times N}$, the algorithm is presented below:

i. Generate two Gaussian random matrices $R^{(p)} \in \mathbb{R}^{N\times M}$ and $R^{(q)} \in \mathbb{R}^{M\times M}$ where M is the number of random projection [82] that have $M \ll N$. In this way, each entry of $R^{(p)}$ and $R^{(q)}$ are independently sampled from Gaussian distribution $N_1(0, 1/M)$ and $N_2(0, \sigma^2)$[7]

ii. Compute the projection matrix $A^{(p)} \in \mathbb{R}^{N\times M}$ by $A^{(p)} = AR^{(p)}$. By doing this, each row of A is projected from a high dimension $\mathbb{R}^N$ into a low dimension $\mathbb{R}^M$.

iii. Perturb $A^{(p)}$ with the Gaussian random matrix $R^{(q)}$ by $\tilde{A}^{(p)} = A^{(p)} + R^{(q)}$. We then project $\tilde{A}^{(p)}$ back to the dimension $\mathbb{R}^{N\times N}$ by $\tilde{A} = \tilde{A}^{(p)}(R^{(q)})^T$. Matrix $\tilde{A}$ is the output of the algorithm.

The perturbed matrix $\tilde{A}$ is regarded as the preserved one of the original adjacency matrix A. The top eigenvectors of the adjacency matrices are mainly utilized in GNN-based models to compute the spatial correlations [124]. The adoption of random projection as described in Step i preserves the top eigenvectors of A, which provides a guarantee for the effectiveness of the preserved adjacency matrix in the subsequent ASTGNN predictor. Furthermore, this algorithm enables us to involve a small amount of random perturbation, which further improves the utility of the perturbed matrix. In the case studies of this work, we empirically set $M = 10$ and $\sigma = 0.5$. Regarding the mathematical analysis of this algorithm, interested readers can refer to [1].

Local-network Topological Information Aggregation Mechanism Step ii of the FASTGNN communication protocol requires the cloud server to aggregate the uploaded $Z_G^{(pp)}$ (i.e., the preserved adjacency matrix). Thus, we propose an adjacency matrix aggregation mechanism. Given a group of uploaded preserved local-network adjacency matrices $\{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_p\}$ where p is the number of involved local-networks, their corresponding sizes are $\{N_1, N_2, \ldots, N_p\}$. Since the sizes of these matrices are different, we first

[7]With abuse of notation, σ in this subsection exclusively denotes the variance of distribution N_2.

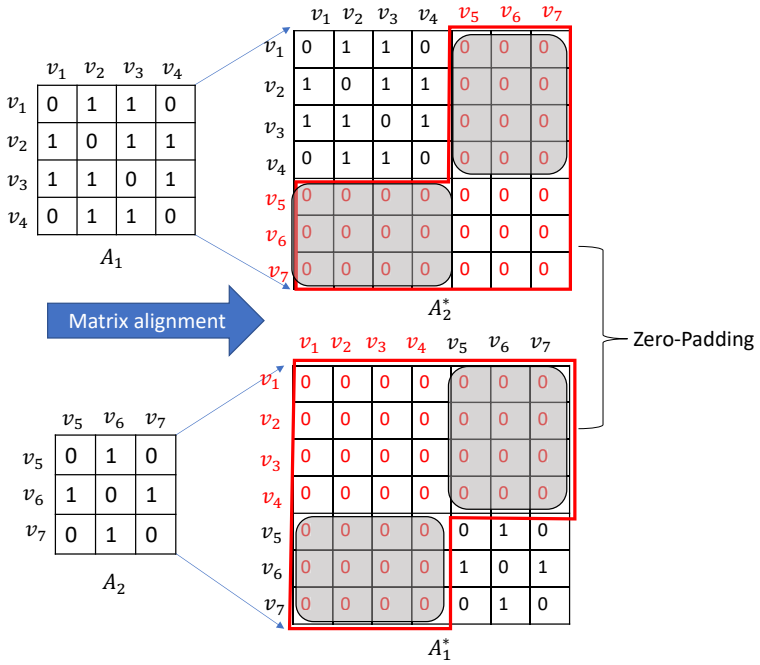


Figure 3.9: (Research Work 3) Adjacency matrix alignment. The red frame highlights the padding entries. The shadowed region highlights the entries entailing the connectivity among objective local-network and other local-networks.

use a matrix alignment approach to make them possess the same size while keeping their own topological information. Specifically, as shown in Fig. 3.9, we align the dimensions of them to the size of the global-network (i.e., N) by using zero-padding and thus obtain a group of aligned matrices $\{\tilde{A}_1^{(a)}, \tilde{A}_2^{(a)}, \ldots, \tilde{A}_p^{(a)}\}$ where $\forall \tilde{A}_i^{(a)}, \tilde{A}_i^{(a)} \in \mathbb{R}^{N \times N}$. Further, considering the significance of the connectivity (i.e., edges) among different local-networks (as the shadowed region shown in Fig. 3.9) for learning the attention, we construct a random connection for them. Specifically, we generate a Gaussian random matrix with the same size as the shadowed regions using the approach as introduced before and symmetrically replace the original parts. Finally, we obtain the aggregated preserved adjacency matrix by adding the aligned matrices together, which can be formulated as

$$\tilde{A}^{(aggre)} = \sum_{i}^{p} \tilde{A}_i^{(a)}. \tag{3.47}$$

Particularly, denote $[\tilde{A}^{(aggre)}]$ as the entry of $\tilde{A}^{(aggre)}$, we threshold its value by

$$\forall \left| [\tilde{A}^{(aggre)}] \right| < \frac{p}{M}, [\tilde{A}^{(aggre)}] \leftarrow 0. \tag{3.48}$$

In FASTGNN, we use FedAvg [63] algorithm as SPAM to aggregate the uploaded

Algorithm 2 Learning Paradigm of FASTGNN.

Input: Organizations $\mathcal{O} = \{O_1, O_2, \ldots, O_p\}$; The number of rounds (i.g., global epochs), E; The preserved adjacency matrix of global-network, $\tilde{A}^{(aggre)}$; The size of local mini-batch, S; The number of local epochs, E_l; The learning rate, η; The gradient optimizer for ASTGNN, $\mathscr{L}(\cdot,\cdot)$.

Output: Parameter ϕ_i.

// Server (k,ω):

initialize global model parameters ϕ_g^0 **foreach** $t = 1,2,\ldots,\ t \in E$ **do**

  foreach *organization* $O \in \mathcal{O}$ ***in parallel*** **do**

    $\phi_{(g)}^{t+1} \leftarrow$ LocalModelUpadate $(O, \tilde{A}^{(aggre)}, \phi_{(g)}^{t})$

  $\phi_{(g)}^{t+1} \leftarrow \frac{1}{p}\sum_{i=1}^{p} \phi_i$

// LocalModelUpadte $(O, \tilde{A}^{(aggre)}, \phi_{(g)}^{t})$:

$\mathcal{B} \leftarrow$ (divide X_i^* in to batches of size B) **foreach** *epoch* $e = 1,2,\ldots,\ e \in E_l$ **do**

  for *each batch* $b = 1,2,\ldots,\ b \in \mathcal{B}$ **do**

    $\phi_i \leftarrow \phi_i - \eta \cdot \mathscr{L}(\tilde{A}^{(aggre)}, \phi_i)$

return ϕ_i to cloud server

parameters and develop a new global model. The FedAvg algorithm can be formulated as

$$\phi_{(g)}^{t+1} = \frac{1}{p}\sum_{i=1}^{p} \phi_i, \tag{3.49}$$

where ϕ_i is the parameter of local model, p is the number of organizations (i.g., the number of local models), and $\phi_{(g)}^{t+1}$ is the aggregated parameter for the new global model. FedAvg algorithm can help train high-quality global with a small cost of communication.

Finally, the entire learning process of each round in FASTGNN consists of three steps:

i. The cloud server broadcasts the global model with initial parameters

   $\phi_g^0 = (W, W^o, W^{(z)}, W^{(r)}, U^{(z)}, U^{(r)}, W^{(fc)})$

   and the preserved adjacency matrix of global-network $\tilde{A}^{(aggre)}$ to the organizations.

ii. Each organization O_i trains its local data X_i^* using $\tilde{A}^{(aggre)}$ and updates the initial local model parameter ϕ_i^t for E_l epochs of an optimizer with mini-batch size B to obtain ϕ_i^{t+1}.

iii. The cloud server aggregates each organization's ϕ_i^{t+1} through FedAvg algorithm and obtains a new global model with the aggregated parameter $\phi_{(g)}^{t+1}$.

Theoretical Discussion of DP-based Adjacency Matrix Preserving on Model Performance Many existing studies have demonstrated that the noise added by DP algorithms to the data may lead to degenerated learning and further affect the model performance [27, 105]. In our proposed approach, the noises are added to the adjacency matrices rather than the data. In the learning process of each local model, the adopted aggregated DP-processed global adjacency matrix $\tilde{A}^{(aggre)}$ is used to only filter the attention coefficients as described in (3.40). Since $\tilde{A}^{(aggre)}$ approximates a binary matrix (i.e., (0,1)-matrix) after DP processing and aggregation, the values of attention coefficients will not be affected significantly. Thus, promising final model performance can be guaranteed. Furthermore, the existing performance loss is due to the disparity between the original global topology and the new global topology after DP processing and aggregation on the adjacency matrices.

CHAPTER 4

CASE STUDIES AND RESULTS

This thesis incorporates empirical studies to evaluate the proposed methods. In this chapter, case studies including experimental results corresponding to the research works introduced in Chapter 3 are presented. Besides numerical outcomes, a comprehensive discussion is also provided, highlighting several significant findings and conclusions.

4.1 Case Studies of Research Work 1: Construct New Graphs using Information Bottleneck Against Property Inference Attacks

4.1.1 Experimental Setup

Dataset In our experiments, we employ three real-world graph-structured social network datasets in terms of graph classification tasks, namely, IMDB-B, IMDB-M, and COLLAB [79]. The statistical information of the three datasets is summarized in Table 4.3. Following [137], the ratios of the training set (for the training of target GNN), auxiliary set (for the training of attack model), testing set (for testing of both target GNN and attack model), and testing sets are 40%, 40%, and 20%, respectively. Additionally, we apply data augmentation to the auxiliary set by adding random edges to ensure sufficient training samples for the attack model.

Table 4.1: (Case Study of Research Work 1) Statistical summary of graph classification datasets.

Dataset	# Graphs	Avg. Nodes	Avg. Edges	# Classes
IMDB-B	1000	19.77	96.53	2
IMDB-M	1500	13.00	65.94	3
COLLAB	5000	74.49	2457.21	3

Table 4.2: (Case Study of Research Work 1) Comparison of property inference accuracy.

Inference Accuracy		COLLAB			IMDB-B			IMDB-M		
		RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
GCN	Original	10.26	9.07	13.91	18.92	15.87	13.17	112.91	110.46	29.24
	DP ($\epsilon = 1$)	16.56	15.93	**27.46**	**35.22**	**32.99**	**29.20**	116.92	111.57	30.39
	DP ($\epsilon = 5$)	15.92	15.33	26.24	33.13	29.79	26.36	114.81	110.97	30.27
	DP ($\epsilon = 10$)	15.84	15.69	25.68	32.43	28.87	25.23	114.22	110.96	30.18
	IB (Ours)	**19.19**	**18.91**	26.24	26.77	26.54	25.99	**122.26**	**121.22**	**35.28**
GAT	Original	11.19	8.81	12.54	17.54	14.42	11.91	112.20	110.43	29.97
	DP ($\epsilon = 1$)	17.24	15.81	23.07	**30.28**	28.77	25.61	18.33	16.73	23.96
	DP ($\epsilon = 5$)	17.46	15.73	23.16	26.76	24.10	20.52	18.60	17.73	25.87
	DP ($\epsilon = 10$)	16.37	14.28	20.63	24.63	21.87	18.39	18.07	16.58	25.07
	IB (Ours)	**18.52**	**18.43**	**27.58**	30.24	**29.94**	**27.99**	**122.90**	**121.24**	**35.21**

Model and Hyperparameters We incorporate two GNN models in our case studies: graph convolution networks (GCN) [50] and graph attention networks (GAT) [97]. We empirically set them to 2-layer and with embedding size 16. Unless other stated, we adopt the following settings. To train the attack model, we set the epoch number to 50 and the mini-batch size to 20. To train target GNN models, we set the epoch number to 200 and the mini-batch size to 100. For the proposed approach, the Lagrange multiplier β is an important hyperparameter to control the distortion level of the new graph structure. We set $\beta = 1 \times 10^{-5}$ as default and conduct a related hyperparameter test later. The learning rates for training GNNs and the attack model are all set to 1×10^{-3}.

Baseline Since we are among the pioneering work focusing on the defenses against property inference attacks, there are no baselines dedicated to this problem. We denote the proposed approach as "IB". We first consider the "Original" case, i.e., the one without any defense mechanism. Furthermore, we identify that differential privacy (DP) is the most widely-adopted defense strategy against inference attacks in the existing works [136–138]. Therefore, we mainly compare our proposed approach with DP. Specifically, we apply DP-based noises to the target graph embedding to defend the property inference attacks by $\tilde{H}_G = H_G + \text{Lap}(0, \frac{s}{\epsilon})$, where the noises are from the Laplacian distribution

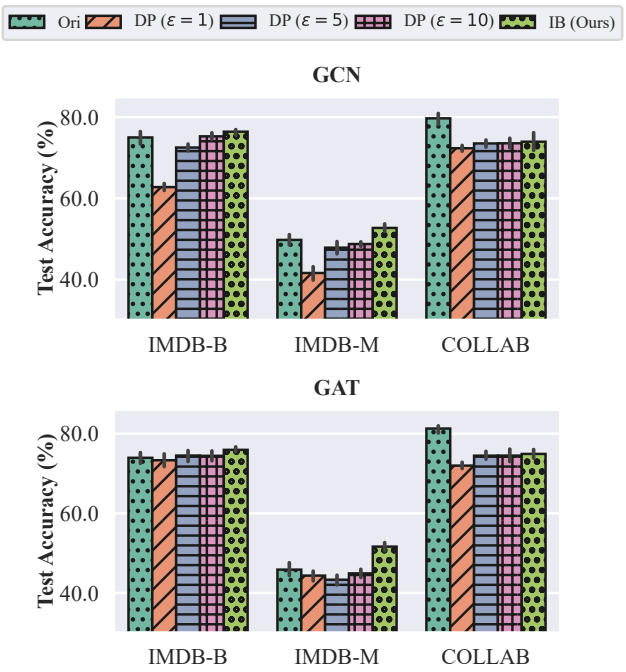


Figure 4.1: (Case Study of Research Work 1) Comparison of graph classification accuracy.

Lap$(0, \frac{s}{\epsilon})$ with mean 0 and scale $\frac{s}{\epsilon}$. ϵ and S denote the *privacy budget* and *sensitivity*, respectively. We set $s = 1$ with different $\epsilon = \{1, 5, 10\}$ to evaluate the performance with difference scales. In general, smaller values of ϵ provide more privacy preservation and vice versa.

Metrics To evaluate the resistance to property inference attacks, root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are used as the metrics to evaluate the accuracy of inference. The lower accuracy indicates a better resistance of the approach; otherwise, worse. In particular, according to common practice, MAPE is considered preferable. To measure the data's utility, we consider the graph classification accuracy of the GNN model. Higher classification accuracy indicates a better utility of the learned graph embeddings.

4.1.2 Resistance to Property Inference Attacks

We first compare the proposed approach and baselines' resistance to property inference attacks. The results are shown in Table 4.2 where the best ones are highlighted in **bold**. Obviously, the proposed approach is effective in resistance to property inference

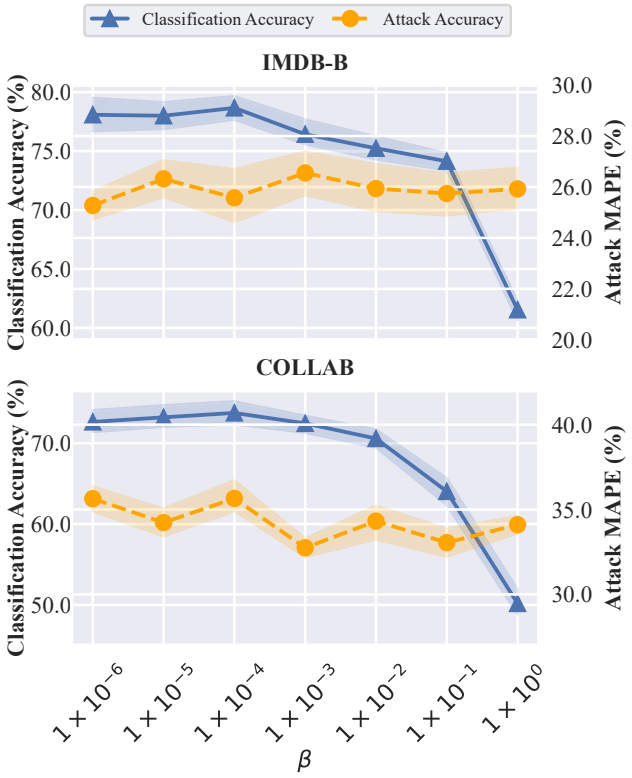


Figure 4.2: (Case Study of Research Work 1) Sensitivity of β to graph classification accuracy and property inference attack accuracy on IMDB-B and COLLAB datasets with GCN model.

attacks against GNNs. Compared with the original setting where no defense strategy is adopted, the inference accuracy (MAPE) is dropped from 5.24% to 16.08% in different configurations. Additionally, we recognize that the effectiveness of the proposed approach varies under different GNN models and datasets. This is due to that the fact that attack model performs differently on the graph embeddings from these different settings. The proposed approach outperforms DP-based defenses in most cases, even compared with DP with small values of ϵ. Particularly, for those DP-based defenses with good resistance performance, the data utility suffers an obvious loss meantimes, which will be discussed later.

4.1.3 Prediction Accuracy on Downstream Tasks

The prediction accuracy developed by the GNN is a significant metric of the data (graph embeddings) utility guarantee provided by the defense approaches. We compare the

graph classification accuracy between different baselines as shown in Figure 4.1. We find that the proposed approach obtains satisfactory classification accuracy — reaches and even outperforms the results of the original setting. It means that the new structure is informative of prediction. Compared with DP-based approaches, especially the ones with small ϵ, the proposed approach shows superiority. As aforementioned, DP-based defenses are successful in protecting privacy in smaller ϵ settings, but at the expense of data utility.

In a nutshell, we can conclude that the IB principle enables the proposed approach to achieve the tradeoff between privacy and data utility to a great extent.

4.1.4 Hypereparameter Study on β: Tradeoff between Utility and Privacy

Moreover, we investigate the sensitivity of the proposed approach to the Lagrange multiplier β as it is a pivotal hyperparameter for the proposed approach in controlling the distortion of learned graph structures, which is shown in Figure 4.2. In terms of classification accuracy, we observe that large β usually develops inferior performance. This is due to large β can lead to over-distorted new graph structures, rendering the loss of predictive information. In terms of the resistance to attacks, we find different result patterns on the two datasets. For IMDB-B, there is a slight uptrend of the attack accuracy with larger β; however, it shows a decline on COLLAB. There may exist some non-trivial influence on the attack model's performance due to the difference in graph size, topology, etc. We will conduct further investigation into this phenomenon in the future.

4.2 Case Studies of Research Work 2: Extracting Privacy-Preserving Subgraphs in Federated Graph Learning using Information Bottleneck

4.2.1 Experimental Setup

In this section, we carry out comprehensive case studies on three real-world datasets and three popular GNN models to evaluate the efficacy of the proposed scheme. Specifically, we first assess the utility of IB-subgraphs generated by the proposed scheme by comparing the trained GNN models' accuracy. Then the resistance of the proposed scheme to inference attacks is compared with that of other baselines. Subsequently, a sensitivity test is conducted to evaluate the impact of hyperparameter selection on the proposed scheme. Finally, a graph spectrum analysis is made to investigate some potential factors of IB-subgraphs' utility and privacy.

Table 4.3: (Case Study of Research Work 2) Statistical summary of Cora, Citeseer, and PubMed datasets.

Data	# Node	# Edge	Density	# Class	# Feature
Cora	2708	5278	$14.3e-4$	7	1433
Citeseer	3312	4536	$8.2e-4$	6	3703
PubMed	19717	44338	$2.2e-4$	3	500

Experiment Preparation In our experiments, we employ three real-world and widely-adopted graph-structured datasets, namely, Cora [83], Citeseer [83], and PubMed [67]. The statistical information of the three datasets is summarized in Table 4.3. Following the previous work [134], the ratios of training, validating, and testing sets are 60%, 20%, and 20%, respectively.

To simulate the Non-IID characteristics of graph data distributed across different clients, we adopt Metis partitioner [46] to partition the global graph into K subgraphs for corresponding K clients and further construct their datasets. Metis is known to ensure a balanced distribution of nodes between different graph partitions. Taking an example from our case studies, the numbers of nodes of the four partitioned subgraphs of Cora are 696, 661, 688, and 663.

Unless otherwise stated, the number of clients in the FGL system is set to $C = 4$. We set the global training epoch $T = 50$. The number of local training epochs is set to 10 for

Table 4.4: (Case Study of Research Work 2) Comparison of total training time consumption using GraphSAGE.

Times (s)	Cora	Citeseer	PubMed
No-IB	15.6	16.8	27.1
FedSage+	133.9	223.1	292.3
IB	21.3	20.3	54.7

Cora and Citeseer and 2 for PubMed. The local GNN models are optimized by Adam with a learning rate $\eta = 0.001$. For the proposed scheme, the hyperparameters are set as: $\beta = 0.2$, $\rho = 0.5$, and $\gamma = 0.5$ — the influence of different settings will be investigated later. The epoch of IB optimization is set as $T_{\text{IB}} = 150$. To demonstrate the model-agnostic of the proposed scheme, graph convolutional networks (GCN), graph attention networks (GAT) [97], and GraphSAGE [35] are adopted as the GNN model to be trained in the FGL system. We employ a 2-layer of all adopted GNNs with the hidden space size of 16 as did in most existing works. Particularly, for the GNN used in Sub-GIB, the hidden space size (i.e., H in Figure 3.6) is set to 512.

We denote the proposed scheme as **IB**. As we are the first to investigate the novel yet significant scenario — subgraph-level FGL against MIA, there are no completely corresponding baselines from existing works. Therefore, we first construct two baselines: 1) **No-IB**: No IB-Subgraph, equal to the vanilla FGL as introduced before, which just uses the original subgraphs for local training; 2) **IB-HCW**: High Card Win, this is a variant of the proposed scheme where the larger of scores IN and OUT decides the retention of the node, that is: reserve the node if it has IN $\geq$ OUT and discarded otherwise. Moreover, we recognize that one of the state-of-the-art FGL methods, FedSage [134], can be a proper performance benchmark, and thus incorporate it in the comparison. Notably, we adopt **FedSage+**, which is the best-performance one in the referenced literature. For a fair comparison, the setting of FedSage+ is adjusted according to our investigated scenario.

The proposed scheme and the baseline approaches are implemented with PyTorch using half-precision (i.e., float16). All experiments are conducted on a computing server with two Intel Xeon E5 CPUs, and eight nVidia GTX 2080 Ti GPUs are employed for neural networks' computing acceleration. To alleviate the randomness, experiments for each setting are run over five repetitions.

Table 4.5: (Case Study of Research Work 2) Prediction accuracy comparison of overall federated learning system.

Accuracy		GCN			GraphSAGE			GAT		
		Cora	Citeseer	PubMed	Cora	Citeseer	PubMed	Cora	Citeseer	PubMed
$C=2$	No-IB	79.3%	79.9%	80.8%	93.2%	85.3%	81.7%	90.3%	82.8%	85.1%
	FedSage+	–	–	–	84.2%	85.7%	86.6%	–	–	–
	IB-HCW	79.1%	82.0%	81.3%	88.8%	83.6%	89.9%	76.7%	76.3%	84.3%
	IB	78.6%	78.6%	81.3%	91.2%	84.6%	88.6%	80.9%	77.3%	85.7%
$C=4$	No-IB	84.8%	82.2%	75.0%	89.5%	83.6%	76.6%	84.8%	80.6%	83.5%
	FedSage+	–	–	–	85.4%	86.2%	82.6%	–	–	–
	IB-HCW	83.7%	81.8%	73.3%	81.0%	79.8%	89.6%	73.0%	73.3%	80.2%
	IB	84.2%	81.6%	74.7%	87.9%	83.5%	88.8%	74.2%	75.9%	80.7%
$C=8$	No-IB	81.6%	74.7%	84.3%	85.1%	80.7%	81.4%	80.8%	78.1%	83.2%
	FedSage+	–	–	–	85.4%	73.6%	86.2%	–	–	–
	IB-HCW	76.4%	80.1%	81.4%	76.7%	79.7%	89.3%	73.0%	79.8%	84.2%
	IB	78.4%	74.4%	84.2%	85.8%	83.1%	89.6%	76.6%	81.0%	81.3%

Table 4.6: (Case Study of Research Work 2) Reconstruction performance of model inversion attacks.

		GCN						GraphSAGE						GAT					
		Cora		Citeseer		PubMed		Cora		Citeseer		PubMed		Cora		Citeseer		PubMed	
		AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
G_1	No-IB	60.6%	66.1%	57.2%	67.4%	59.2%	60.2%	51.1%	56.6%	48.9%	56.3%	52.8%	57.7%	64.6%	61.0%	53.1%	63.0%	61.7%	63.8%
	FedSage+	–	–	–	–	–	–	59.8%	64.5%	57.4%	67.3%	60.1%	60.9%	–	–	–	–	–	–
	IB	**49.8%**	**49.8%**	**55.8%**	**53.7%**	**49.7%**	**49.8%**	**44.6%**	**50.3%**	**40.2%**	**50.8%**	**51.8%**	**55.1%**	**61.2%**	**58.5%**	**52.6%**	**62.6%**	**61.5%**	**63.4%**
G_2	No-IB	61.8%	65.6%	56.0%	68.4%	57.3%	61.8%	50.1%	61.9%	46.0%	58.6%	50.6%	56.8%	62.2%	68.1%	54.1%	66.2%	**56.5%**	61.3%
	FedSage+	–	–	–	–	–	–	61.3%	66.8%	55.9%	69.8%	56.9%	61.4%	–	–	–	–	–	–
	IB	**53.4%**	**51.6%**	**54.7%**	**52.4%**	**54.0%**	**52.2%**	**47.2%**	**52.3%**	**39.2%**	**52.3%**	**47.5%**	**54.0%**	**60.4%**	**64.2%**	**53.5%**	**66.2%**	56.6%	**61.3%**
G_3	No-IB	53.6%	57.3%	61.3%	65.5%	55.4%	59.5%	49.7%	56.3%	51.1%	56.8%	48.5%	54.6%	63.4%	68.6%	57.8%	62.9%	56.2%	62.6%
	FedSage+	–	–	–	–	–	–	51.8%	54.7%	61.2%	65.5%	55.3%	60.6%	–	–	–	–	–	–
	IB	**41.5%**	**46.1%**	**51.9%**	**50.6%**	**51.9%**	**50.6%**	**46.6%**	**53.0%**	**45.8%**	**52.3%**	**44.5%**	**50.0%**	**60.1%**	**65.2%**	**56.6%**	**61.7%**	**56.1%**	**62.3%**
G_4	No-IB	64.3%	70.4%	61.6%	66.3%	58.4%	61.6%	53.6%	59.7%	45.8%	55.9%	50.1%	54.4%	62.8%	68.9%	**61.3%**	67.3%	61.7%	66.2%
	FedSage+	–	–	–	–	–	–	62.7%	68.9%	62.0%	68.2%	58.9%	62.7%	–	–	–	–	–	–
	IB	**50.7%**	**50.2%**	**53.3%**	**51.7%**	**49.3%**	**49.6%**	**44.7%**	**51.6%**	**44.0%**	**50.6%**	**48.8%**	**53.1%**	**61.8%**	**68.2%**	**60.7%**	**67.1%**	**60.3%**	**64.5%**

4.2.2 Learning Performance

Prediction Accuracy As FGL systems aim at training a powerful global model, the prediction accuracy of the global model is the most important metric for FGL systems. It can also evaluate the utility of the IB-subgraphs developed by the proposed scheme. Therefore, we first evaluate the developed global GNN model's prediction accuracy, and compare that between the proposed scheme and baselines. Particularly, as FedSage+ is a GraphSAGE-based approach, we only compare it with others under the setting of using GraphSAGE as the GNN.

From the results presented in Table 4.5, we observe that IB develops a very close accuracy performance to the No-IB, even though the IB-subgraphs used to train have less information than the original subgraph. The results demonstrate that the developed IB-subgraphs can provide the GNN model with sufficient predictive information, which further justifies the effectiveness of the proposed scheme: an informative subgraph in

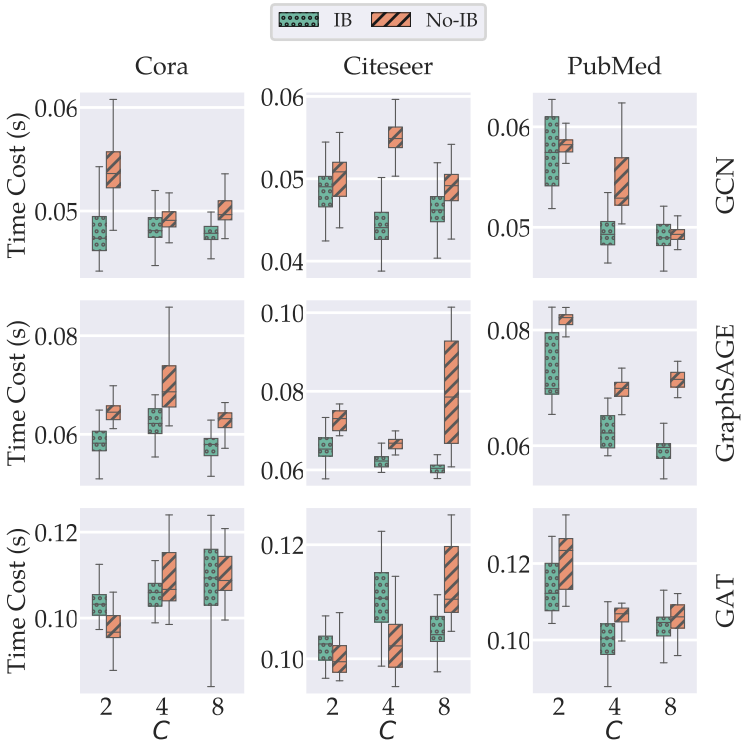


Figure 4.3: (Case Study of Research Work 2) Comparison of training time consumption per epoch.

terms of prediction can be extracted. The variant IB-HCW can also obtain a satisfactory prediction accuracy; however, the no-node-reserved situations occur in our offline tests. In this aspect, the stability of IB is greater than IB-HCW, thanks to the protection mechanism by the Top-K algorithm. While FedSage+ demonstrates a remarkable prediction accuracy, the success is on the premise of sacrificing computational efficiency as the involved missing link prediction consumes huge amounts of time. As shown in Figure 4.3, the training time consumption of FedSage+ can be ten times as much as that of the proposed scheme.

The proposed scheme generally demonstrates exemplary performance on all three GNN models. Comparatively, the performance drop when using GAT is relatively notable. This is due to the difference in the graph convolution method between GAT and the other two, where GAT is spatial-based and can be more sensitive to neighbors' absence when convolution.

Additionally, we compare prediction accuracy under FGL settings with different numbers of clients, i.e., $C = 2, 4, 8$. Generally, the results on $C = 2, 4, 8$ demonstrate a

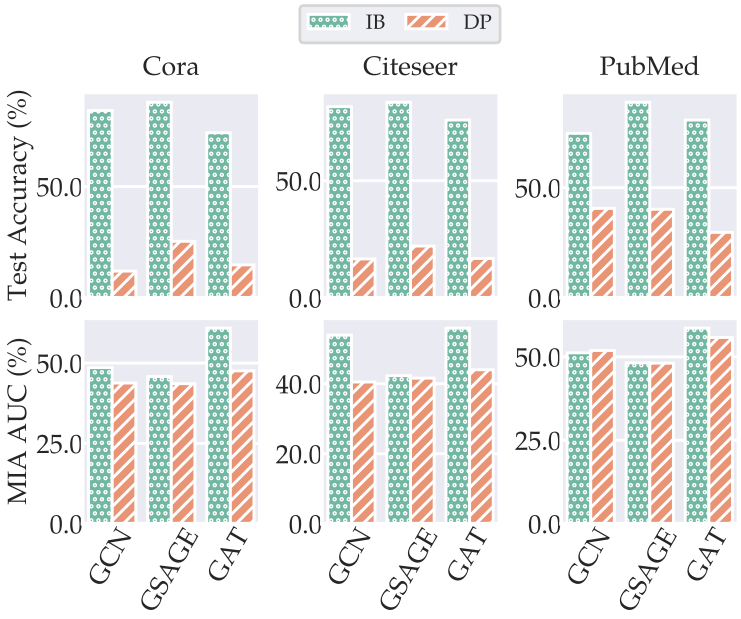


Figure 4.4: (Case Study of Research Work 2) Comparison of prediction accuracy and MIA resistance between DP and the proposed scheme. GSAGE denotes GraphSAGE.

similar pattern. The only exception occurs when we set $C = 8$ on Cora, where there is an accuracy drop from 81.6% to 78.4%; however, this drop is within the acceptable range.

Training Efficiency The proposed scheme can develop IB-subgraphs that are smaller in size than the original subgraphs. These "compressed" subgraphs naturally enable less training time on it and develop more efficient training.

To demonstrate this advantage of the proposed scheme, we compare training time consumptions on original subgraphs and IB-subgraphs. Particularly, under different configurations of dataset and model, we randomly sample 50 training epochs in a FGL life span and record the corresponding training time cost. As shown in Figure 4.3, we find the training time consumption on IB-subgraphs is less than that on the original subgraphs in most cases. We can envision that such an advantage will be more noticeable when the number of training epochs is large.

Resistance to Model Inversion Attack To validate the effectiveness of our proposed scheme against MIA, we conduct a case study of the resistance to MIA following the scenario introduced before. Particularly, we introduce a state-of-the-art MIA method, GraphMI [139]. For achieving MIA, GraphMI integrates projected gradient descent and

graph autoencoder techniques, which can be regarded as a more threatening adversary than traditional methods. Our simulation configures GraphMI at the central server side to attack targeted clients. The setting of GraphMI (including architecture and parameters) follows the recommended one in the original literature with minuscule non-algorithmic changes in the practical implementation. We adopt two metrics to evaluate the attack, namely, area under the ROC curve (AUC) and average precision (AP), as did in [38] and [139]. The larger the AUC or AP, the more successful the MIA. Furthermore, as the subgraph structures vary from the client, we also show the different clients' results to provide a fairer demonstration.

From the results shown in Table 4.6, we can find that the AUC and AP of MIA on the proposed scheme are generally lower than those on No-IB on three datasets and three GNNs. That is to say, the effect of MIA is reduced by the proposed scheme. The results highlight the advantage of the proposed scheme: as the IB-subgraphs developed by the proposed scheme are considerably distorted from the original subgraphs, less information regarding the original subgraphs can be learned by the GNN model. This condition further prevents MIA from reasoning relevant information about the structure of the original subgraph from the model weights. However, a particular case occurs when using GAT where the drop of the proposed scheme is unfavorably not significant. According to the accuracy performance shown in Section 4.2.2, we have known that the accuracy drop when using GAT is more prominent than others. Such an inconsistency with the intuition of trade-off between utility and privacy is due to the MIA method involved in this work. GraphMI utilizes the gradient of GNN to infer the graph structure, and the attention mechanism makes GAT's gradients contain more relevant information to graph structures. This observation suggests the referenced MIA method's inference capacity varies from different attacked GNN models. In light of this, we call for further research on studying the corresponding concerns.

Additionally, it can be observed that the robustness brought by the IB-subgraphs (i.e., the downgrade of MIA performance) shows differently for different clients. This phenomenon can be explained by the fact that GraphMI predicts the graph structure by a graph autoencoder, where the neural network nature renders randomness in the prediction process. Generally, it can be concluded that the proposed scheme can effectively mitigate MIA.

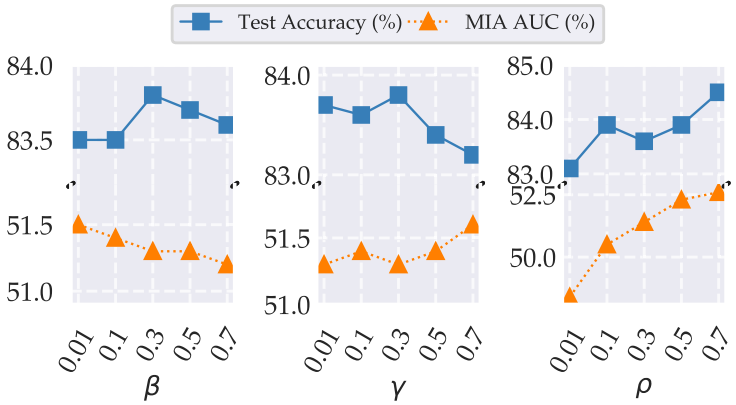


Figure 4.5: (Case Study of Research Work 2) The sensitivity of the proposed scheme to hyperparameter $\beta \in 0.01, 0.2, 0.5, 1, 5$, $\gamma \in 0.01, 0.1, 0.5, 1, 5$, and $\rho \in 0.01, 0.1, 0.3, 0.5, 0.7$ on Cora. AUC is obtained by averaging the MIA results of all four clients.

4.2.3 Comparison with Differential Privacy-based Defenses

Local differential privacy (LDP) [47] is among the most common strategies for data privacy-preservation in FL systems, which introduces noise addition to clients' responses before sending them to the central server. We evaluate the effectiveness of FGL systems when the LDP strategy is used, and compare it with the proposed scheme. Particularly, we empirically adopt the Laplacian mechanism. On the baseline No-IB, we add Laplacian-distributed noise onto the local GNN updates in each training epoch of FGL.

Figure 4.4 presents the comparison of prediction accuracy and MIA resistance between DP and the proposed scheme. Our offline fine-tuning suggests that when the exponential decay of Laplacian distribution is set to 1 (i.e., results in the figure), we can obtain a set of noise enabling a comparable MIA resistance with the proposed scheme. However, under this noise level, the prediction accuracy deteriorates considerably. This phenomenon implies a huge utility drop on the GNN models' updates when applying such noises. Generally, applying DP directly cannot achieve a tradeoff between utility and privacy in the investigated scenario.

4.2.4 Sensitivity Studies of Hyperparameters

The proposed Sub-GIB approach mainly incorporates three hyperparameters to control the optimization process, namely, the Larange multiplier that control the distortion extent of G_i^{IB} — β, the Larange multiplier that control the trade-off of contribution between the holistic feature and regional feature — γ, and the ratio controlling the lowest

number of the nodes reserved in $G_i^{\mathrm{IB}} - \rho$. They play a pivotal role in determining the finally generated IB-subgraphs. Therefore, we conduct hyperparameter sensitivity tests.

In Figure 4.5, the results of the one with Cora dataset and GCN model is presented. We can draw several conclusions from the results. The larger the ρ, the higher the AUC of MIA and the lower the accuracy. This is because the larger ρ makes more nodes in the original subgraph reserved in the IB-subgraph, making the GNN model learn sufficient node interaction and develop more accurate predictions. Conversely, the information recorded in the GNN model can also leave exploitable loopholes for MIA. This is why the AUC of MIA is higher when more nodes are reserved in the IB-subgraphs. For β, a larger one means the MI with the original subgraph will be less considered in the developed IB-subgraph. This makes the developed IB-subgraph reserve less information about the original subgraph. Therefore, the larger β can render a lower AUC of MIA. In terms of γ, a larger one will take less influence of the regional feature into account when doing MI estimation; the results show that this will degenerate the prediction performance. This implies the significance of regional features in the Sub-GIB optimization, which will lead to IB-subgraphs containing more predictive features regarding structure.

Generally, the proposed scheme is more sensitive to the change of ρ as it directly controls the scale of finally generated IB-subgraphs while β and γ mainly control the optimization of Sub-GIB. While the sensitivity of the proposed scheme to the change of β and γ is less remarkable, appropriate fine-tuning can contribute to developing more reasonable IB-subgraphs to affect the final performance.

4.3 Case Studies of Research Work 3: FASTGNN: A Topological Information Protected Federated Learning Approach For Traffic Speed Forecasting

In this work, we propose FASTGNN as a FL framework to address the traffic speed forecasting problem with privacy-preserving concern. To fully assess the performance of the proposed framework, we carry out three comprehensive case studies on a real-world traffic dataset. First, we investigate the accuracy of forecasting speed using the proposed framework and the comparison with baselines. Subsequently, an ablation study is conducted to evaluate the critical components of FASTGNN. Lastly, we exhibit the performance of FASTGNN under different organization numbers.

4.3.1 Experimental Setup

Dataset Description and Pre-processing PeMSD7 is the experimental dataset in this work, which is a public dataset collected from Caltrans Performance Measurement System (PeMS) in District 7 of California. We select 228 out of 39000 sensor stations in PeMSD7 to construct the final dataset as a tailored one for our case studies. The time interval of speed data is set to 5 minutes, and the period of the dataset is from May 1st to June 30th of 2012[1]. Linear interpolation is employed to recover the missing data when there exist missing data points. We apply Z-score to normalize the data before input to the models. The training, validation, and test sets are correspondingly constructed for supervised learning, each of which contains 60%, 20%, and 20% of all data, respectively.

To simulate the distributed training scenario of FASTGNN, we first construct the adjacency matrix of the entire traffic network (i.e., global-network) A by

$$[A_{ij}] = \begin{cases} 1, \text{ if } i \neq j \text{ and } \exp\left(-\dfrac{\text{dist}(v_i, v_j)}{\varsigma^2}\right) \geqslant \varepsilon \\ 0, \text{ otherwise.} \end{cases}, \tag{4.1}$$

where $[A_{ij}]$ is the entry of A that denotes the connectivity between node v_i and node v_j, which is decided by their Euclidean space distance $\text{dist}(v_i, v_j)$; ε and ς^2 are the user-controlled parameters that control the density of graph, and we set their values to 0.5 and 10, respectively. Note that since we define the network as an undirected graph, the adjacency matrix is symmetrical, i.e., $[A_{ij}] = [A_{ji}]$. Then, we partition the

[1]Only weekdays' data is contained to avoid atypical traffic, which is in accordance with the literature. See [23, 124] for examples

Table 4.7: (Case Study of Research Work 3) Comparison of traffic speed forecasting accuracy.

Approach	Accuracy			Graph-based	Privacy-preserving
	RMSE	MAE	MAPE (%)		
HA	7.20	4.01	10.61	–	–
ARIMA	9.45	6.33	16.10	–	–
LSVR	8.28	4.53	11.49	–	–
DCRNN	7.14	4.11	9.92	✓	–
Graph WaveNet	6.23	3.51	9.03	✓	–
STGCN	**5.80**	**3.47**	8.56	✓	–
FASTGNN	5.83	3.50	**8.36**	✓	✓

global-network into p sub-networks for corresponding p organizations randomly. Let $\mathcal{V}_u, \mathcal{V}_v$ denote any two sub-networks' node sets, we have $\mathcal{V}_u \cap \mathcal{V}_v = \emptyset$. We can thus obtain sub-networks' adjacency matrices $\{A_1, A_2, \ldots, A_p\}$.

Experiment Setting The proposed FASTGNN is implemented with PyTorch, and all tests are conducted on a computing server with an Intel(R) Xeon(R) E5-2620 v4 CPU and eight nVidia GeForce RTX 2080 Ti GPUs. When training FASTGNN, the objective dimension of the weight matrix W in (3.38) (i.e., F^h) is set to 144, and the numbers of neurons in the two GRU layers are set to 64 and 256, respectively. All the neural networks-based models are trained with Adam optimizer for 50 epochs, and the batch size and learning rate are set to 50 and $1e^{-3}$, respectively[2]. Unless otherwise stated, we simulate FASTGNN with the number of organizations $p = 4$. In terms of the traffic speed forecasting, the past time window is 60 minutes (i.e., 12 timestamps), and we use these to predict speed in the next 45 minutes (i.e., nine timestamps). With regards to accuracy comparison, we adopt RMSE, MAE, and MAPE as the metrics to evaluate the forecasting accuracy of all approaches. Particularly, MAPE is considered as the most referable one among the three metrics (see [122, 127] for examples), which can be defined as,

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - \hat{X}_i}{X_i}\right| \times 100\%, \tag{4.2}$$

where X_i and $\hat{X}_i$ are the observed and the forecasted traffic speeds at time i, respectively.

[2]For FASTGNN, it denotes that the global epoch size $E = 50$ and the size of local mini-batch $S = 50$.

Table 4.8: (Case Study of Research Work 3) Comparison of ablation tests on FASTGNN.

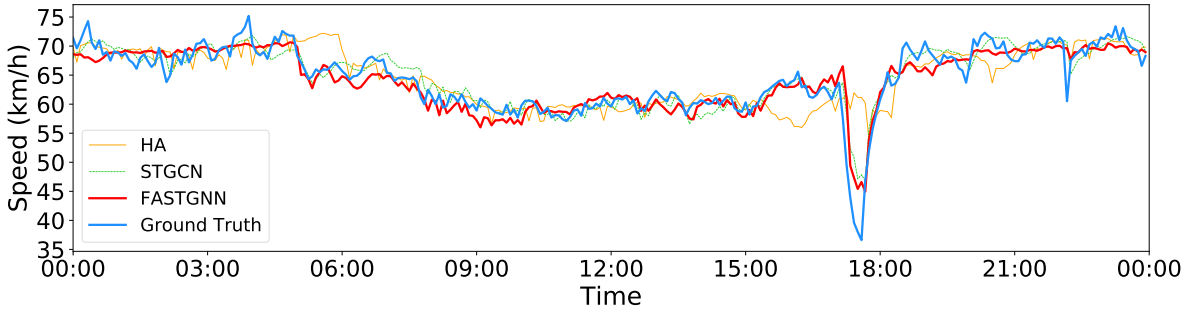
	FASTGNN	V1	V2	V3	ASTGNN
RMSE	5.83	5.51	9.29	7.98	5.33
MAE	3.50	3.20	7.27	6.10	3.21
MAPE (%)	8.36	8.03	16.10	12.33	7.84

4.3.2 Accuracy of Forecasting Traffic Speed

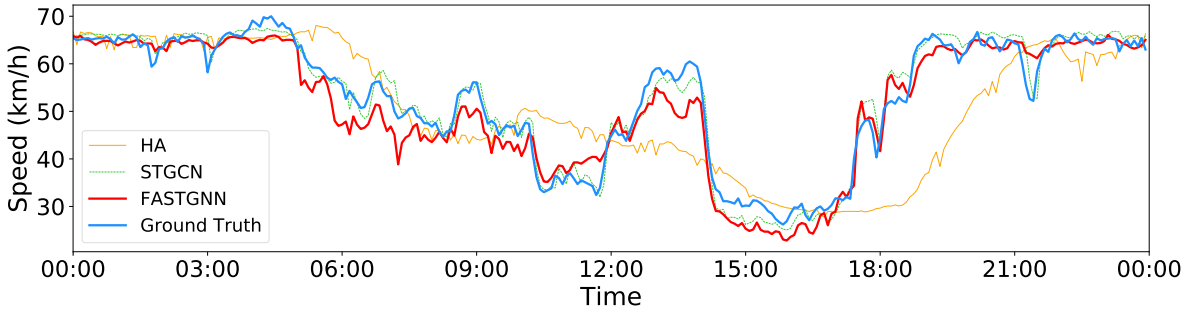
We first investigate the accuracy of forecasting traffic speed with PeMSD7 dataset. Specifically, FASTGNN is compared with the following baselines and state-of-the-art approaches: 1) Historical Average (HA), 2) Autoregressive Integrated Moving Average (ARIMA), 3) Linear Support Vector Regression (LSVR), 4) Diffusion Convolutional Recurrent Neural Network (DCRNN) [57], 5) Graph WaveNet [114], and 6) Spatio-Temporal Graph Convolutional Networks (STGCN) [124]. To make a fair comparison, we configure the baseline approaches with the default hyperparameters in their respective literature.

The forecasting results are presented in Table 4.7 for 45-min ahead traffic speed forecasting. From the simulation results, traditional approaches, i.e., HA, ARIMA, and LSVR, have the worst performance with relatively large forecasting errors, which implies their shortage in handling nonlinearity. Comparatively, the graph deep learning-based approaches, i.e., DCRNN, Graph WaveNet, STGCN, and FASTGNN, perform much better than the conventional approaches with an average improvement of 2.08 (RMSE), 1.34 (MAE), and 3.76% (MAPE). Particularly, the proposed FASTGNN can achieve the same performance level as STGCN, whose accuracy of MAPE even surpasses STGCN by 0.20%. This demonstrates the efficacy of the adopted technical scheme for spatial-temporal learning. Furthermore, FASTGNN is the only one among these approaches that can both deal with spatial information and achieve privacy-preserving through a decentralized training scheme in the proposed FL framework. It indicates that FASTGNN can achieve outstanding performance and privacy-preserving at the same time.

Besides, to better illustrate the forecasting performance of FASTGNN, we present and compare the forecasting curves developed by FASTGNN, HA, and STGCN. As shown in Fig. 4.6, FASTGNN can produce traffic speed prediction with a small deviation and accurately reflect the oscillation on ground truth.



(a)



(b)

Figure 4.6: (Case Study of Research Work 3) Traffic speed forecasting curves in a day. (a) and (b) present results from two different sensor stations, respectively.

4.3.3 Ablation Study on FASTGNN

To evaluate the several scheme designs in the proposed FASTGNN, we conduct ablation studies in this subsection. Specifically, we first transform FASTGNN into the following variants by adding particular constraints and compare their MAPE performance with FASTGNN:

- **FASTGNN-V1**: Without the differential privacy-based adjacency matrix preserving approach.
- **FASTGNN-V2**: Without local-networks aggregation, i.e., each local model of FASTGNN can only access the local-network other than the global-network for training.
- **FASTGNN-V3**: Without considering the connectivity among different local-networks when constructing the aggregated global-network.
- **ASTGNN**: Naive ASTGNN model.

Table 4.9: (Case Study of Research Work 3) The accuracy of FASTGNN with different organization numbers.

$p =$	2	4	6	8	12	16
RMSE	5.73	5.83	5.96	6.03	6.18	6.22
MAE	3.31	3.50	3.58	3.76	4.05	4.36
MAPE (%)	8.02	8.36	8.76	9.25	9.79	10.38

The results are presented in Table 4.8. Comparing FASTGNN and ASTGNN, it can be seen that performance degeneration due to the adoption of FL's decentralized training is not significant, where the accuracy only suffers from a 0.52% MAPE penalty. This indicates that the combo of adopted techniques can ensure the learning effect of ASTGNN in FL framework under privacy-preserving. Especially when we compare FASTGNN with FASTGNN-V1, the minuscule difference of accuracy performance implies that the adoption of differential privacy-based adjacency matrix preserving approach does not veritably weaken the topological information of the network and further affect the spatial learning effect of the model, which proves the effectiveness of this approach. By contrast, a large performance gap is observed between FASTGNN and FASTGNN-V2, where there is a 7.74% accuracy difference. Since in FASTGNN-V2 the local model can only access the local-network for training where the latter can only provide limited topological information for training a generalized model applicable to the global-network, this results in the striking performance degeneration. It can also shed light on the necessity of adopting a local-network aggregation mechanism to construct a shareable global network for each local training. A similar conclusion can be drawn when comparing FASTGNN and FASTGNN-V3. FASTGNN-V3 performs worse than FASTGNN by 3.97% MAPE. While in the setting of FASTGNN-V3, the local-networks are aggregated, the connectivity among them is absent. This results in the declined performance of FASTGNN-V3.

4.3.4 Performance Comparison of FASTGNN Under Different Organization Numbers

In the above tests, the default organization number is set as $p = 4$. Nonetheless, the number of organizations in real scenarios may vary a lot. It is interesting to investigate the impact of different organization numbers on the performance of FASTGNN. In this experiment, we set $p \in \{2, 4, 6, 8, 12\}$ for FASTGNN and compare the accuracy performance under this group of settings.

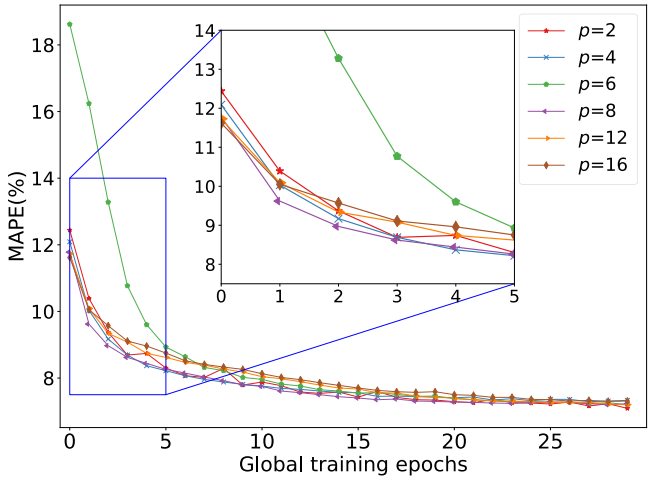


Figure 4.7: (Case Study of Research Work 3) Visualization of training process for 30 global epochs with different organization numbers.

As the results are shown in Table 4.9, we can observe the number of organizations has a negative correlation with the performance of FASTGNN. More organizations involve increasing groups of local topological information and model parameters, which makes it challenging for cloud sever to perform the aggregation algorithms. We can draw the same conclusion from the convergence curves as shown in Fig. 4.7, where the larger number of involved organizations, the more difficult the learning curves converge. It is worth mentioning that no matter how many organizations are involved in our simulation, their respective data and topological information are obtained by dividing the same global-network. This may make the results contrast not distinct. We will conduct refined tests in future work.

4.3.5 Generalization Ability

In the above case studies, we test the performance of FASTGNN on PeMSD7 dataset. To assess the generalization ability of FASTGNN, we adopt another dataset METR-LA to examine the forecasting accuracy of FASTGNN. METR-LA is a public dataset, which contains traffic data collected from 207 loop detectors in the highway of Los Angeles County. The experiment setting are configured as the same as it on PeMSD7 for the sake of fairness.

Table 4.10 presents the simulation results. For the results, we can observe that conventional machine learning approaches (i.e., ARIMA and LSVR) perform worse than

Table 4.10: (Case Study of Research Work 3) Comparison of traffic speed forecasting accuracy on METR-LA.

Approach	Accuracy		
	RMSE	MAE	MAPE (%)
HA	7.80	4.16	13.02
ARIMA	12.11	6.01	15.04
LSVR	12.01	5.92	14.81
DCRNN	7.24	3.41	9.67
Graph WaveNet	6.49	3.01	9.22
STGCN	**6.11**	**2.98**	**8.84**
FASTGNN	6.42	3.03	9.15

on PeMSD7. This implies that the data of METR-LA is more unstable and changeable than that of PeMSD7. In this context, FASTGNN can still obtain matched performance compared with the three state-of-the-art baselines, where the MAPE of FASTGNN is only 0.31% higher than that of STGCN. This indicates that FASTGNN is capable of handling data with different time-series fluctuation and topology.

CHAPTER 5

SUMMARY

This thesis has endeavored to advance the field of secure and privacy in machine learning through a rigorous examination of vulnerabilities and the subsequent development of tailored privacy-preserving mechanisms for graph-structured data. Graph data, with its rich structural representation of entities and their interrelations, has garnered significant prominence across diverse fields, including social networks, biological networks, and knowledge graphs. Graph neural networks (GNNs), as an advanced technical family of machine learning for graph data, continue to proliferate across various domains, the importance of safeguarding the privacy and security of the involved graph data cannot be overstated. Particularly, a great number of studies have revealed the vulnerabilities of GNNs to privacy attacks, such as model inversion and membership inference, which threaten the confidentiality of the graph data these models learn from. This vulnerability necessitates the development of mechanisms that can safeguard privacy while retaining the utility of the graph data for analytical purposes.

This study underscores the complex balance between retaining data utility and achieving the intended privacy-preserving outcomes within these mechanisms. Privacy-preserving often necessitates a compromise in the level of detail or utility of data. Therefore, the study endeavors to harmonize these competing imperatives, ensuring that the measures for privacy preservation do not significantly impede the functional value of the graph data. In pursuit of this objective, a meticulous design of methodologies and case studies is undertaken, utilizing benchmark GNNs and datasets as evaluative tools. Specifically, the methodological design incorporates principles of differential and

compressive privacy tailored to graph data, aiming to strike a balance between data utility and privacy. Furthermore, the investigation broadens to include real-world scenarios and applications, especially the federated learning scenarios, responding to the escalating demand for decentralized privacy-preserving techniques in machine learning. Through these efforts, this thesis is committed to fostering the development of secure and privacy-preserving GNNs, providing insights that bridge the divide between theoretical innovation and practical application in managing sensitive graph data.

Structured as a thesis by compilation, the narrative unfolds through a literature review, followed by the presentation of three published research works that collectively advance the thesis's objective.

- **Research work 1 focuses on defending against property inference attacks on graph data.** To this end, this work proposes to leverage the information bottleneck (IB) principle to construct new graph structures from the original graphs. The change in graph structures enables the new graphs to contain less information related to the property information of the original graphs, making it harder for attackers to infer property information of the original graphs from the graph embeddings. Meantime, the IB principle enables task-relevant information to be sufficiently contained in the new graph, enabling GNNs to develop accurate predictions.

- **Research work 2 concentrates on defending against model inversion attacks on graph data in federated graph learning systems.** This work identifies a realistic crowdsourcing-based FGL scenario where MIA from the central server towards clients' subgraph structures is a nonnegligible threat. This work proposes a defense scheme, Subgraph-Out-of-Subgraph (SOS), to mitigate such MIA and meanwhile, maintain the prediction accuracy. Following a similar strategy to Research work 1, this work leverages the information bottleneck (IB) principle to extract task-relevant subgraphs out of the clients' original subgraphs. The extracted IB-subgraphs are used for local GNN training and the local model updates will have less information about the original subgraphs, which renders the MIA harder to infer the original subgraph structure. Particularly, this work devises a novel neural network-powered approach to overcome the intractability of graph data's mutual information estimation in IB optimization. Additionally, this work designs a subgraph generation algorithm for finally yielding reasonable IB-subgraphs from the optimization results.

- **Research work 3 delves into a realistic scenario, federated graph learning-based intelligent transportation systems for privacy-preserving traffic forecasting.** Specific to the scenario, this work proposes a novel federated learning framework to tackle this problem. Specifically, this work introduces a differential privacy-based adjacency matrix preserving approach for protecting the topological information. This work also devises an adjacency matrix aggregation approach to allow local GNN-based models to access the global network for a better training effect. Furthermore, this work introduces a GNN-based model named Attention-based Spatial-Temporal Graph Neural Networks (ASTGNN) for traffic speed forecasting. Finally, this work integrates the proposed federated learning framework and ASTGNN as FASTGNN for traffic speed forecasting.

These studies concentrate on preserving the privacy of graph structures, unveiling the distinctive attributes of graph data. The methodologies and experimental outcomes of these research projects are thoroughly detailed in this thesis, showcasing the extensive research efforts undertaken.

Expanding on this thesis's contributions, future research directions will be traced out into three critical areas to advance privacy preservation in graph data analysis. The first one is deepening theoretical foundations. Future efforts should delve into theoretical research to refine privacy-preserving algorithms for GNNs, aiming for a deeper understanding of privacy-utility dynamics. The second one is the extension to multimodal machine learning: Exploring the application of privacy mechanisms in multimodal machine learning, where graph data is one among various data types, offers a path to address privacy across diverse data modalities. The third one is the combination with large language models (LLMs): In view of the powerful reasoning ability of LLMs, the integration of LLMs with GNNs presents an opportunity to enhance the interpretability in security and privacy diagnosis in graph data and GNNs.

BIBLIOGRAPHY

[1] F. AHMED, R. JIN, AND A. X. LIU, *A random matrix approach to differential privacy and structure preserved social network graph publishing*, arXiv preprint arXiv:1307.0475, (2013).

[2] A. A. ALEMI, I. FISCHER, J. V. DILLON, AND K. MURPHY, *Deep variational information bottleneck*, in Proc. International Conference on Learning Representations, 2017.

[3] D. ALISTARH, D. GRUBIC, J. LI, R. TOMIOKA, AND M. VOJNOVIC, *Qsgd: Communication-efficient sgd via gradient quantization and encoding*, Advances in Neural Information Processing Systems, 30 (2017), pp. 1709–1720.

[4] R. ANGLES AND C. GUTIERREZ, *Survey of graph database models*, ACM Computing Surveys (CSUR), 40 (2008), pp. 1–39.

[5] F. ARMKNECHT, C. BOYD, C. CARR, K. GJØSTEEN, A. JÄSCHKE, C. A. REUTER, AND M. STRAND, *A guide to fully homomorphic encryption.*, IACR Cryptol. ePrint Arch., 2015 (2015), p. 1192.

[6] G. ATENIESE, L. V. MANCINI, A. SPOGNARDI, A. VILLANI, D. VITALI, AND G. FELICI, *Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers*, International Journal of Security and Networks, 10 (2015), pp. 137–150.

[7] E. BAGDASARYAN, A. VEIT, Y. HUA, D. ESTRIN, AND V. SHMATIKOV, *How to backdoor federated learning*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.

[8] 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, 2006, pp. 16–25.

[9] M. I. BELGHAZI, A. BARATIN, S. RAJESHWAR, S. OZAIR, Y. BENGIO, A. COURVILLE, AND D. HJELM, *Mutual information neural estimation*, in Proc. International conference on machine learning, 2018, pp. 531–540.

[10] A. BEN-EFRAIM, Y. LINDELL, AND E. OMRI, *Optimizing semi-honest secure multiparty computation for the internet*, in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 578–590.

[11] S. BHAMBRI, S. MUKU, A. TULASI, AND A. B. BUDURU, *A survey of black-box adversarial attacks on computer vision models*, arXiv preprint arXiv:1912.01667, (2019).

[12] B. BIGGIO, B. NELSON, AND P. LASKOV, *Support vector machines under adversarial label noise*, in Asian conference on machine learning, PMLR, 2011, pp. 97–112.

[13] A. BOJCHEVSKI AND S. GÜNNEMANN, *Adversarial attacks on node embeddings via graph poisoning*, in Proc. International Conference on Machine Learning, 2019, pp. 695–704.

[14] D. BUSBRIDGE, D. SHERBURN, P. CAVALLO, AND N. Y. HAMMERLA, *Relational graph attention networks*, arXiv preprint arXiv:1904.05811, (2019).

[15] Z. CAI, Z. XIONG, H. XU, P. WANG, W. LI, AND Y. PAN, *Generative adversarial networks: A survey toward private and secure applications*, ACM Computing Surveys (CSUR), 54 (2022), p. 38.

[16] S. CALDAS, J. KONEČNY, H. B. MCMAHAN, AND A. TALWALKAR, *Expanding the reach of federated learning by reducing client resource requirements*, arXiv preprint arXiv:1812.07210, (2018).

[17] D. CAMACHO, A. PANIZO-LLEDOT, G. BELLO-ORGAZ, A. GONZALEZ-PARDO, AND E. CAMBRIA, *The four dimensions of social network analysis: An overview of research methods, applications, and software tools*, Information Fusion, 63 (2020), pp. 88–120.

[18] C. CHEN, K. LI, S. G. TEO, X. ZOU, K. WANG, J. WANG, AND Z. ZENG, *Gated residual recurrent graph neural networks for traffic prediction*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 485–492.

[19] M. CHEN, Z. YANG, W. SAAD, C. YIN, H. V. POOR, AND S. CUI, *A joint learning and communications framework for federated learning over wireless networks*, IEEE Transactions on Wireless Communications, (2020).

[20] X. CHEN, S. JIA, AND Y. XIANG, *A review: Knowledge reasoning over knowledge graph*, Expert Systems with Applications, 141 (2020), p. 112948.

[21] K. CHIDA, D. GENKIN, K. HAMADA, D. IKARASHI, R. KIKUCHI, Y. LINDELL, AND A. NOF, *Fast large-scale honest-majority mpc for malicious adversaries*, in Annual International Cryptology Conference, Springer, 2018, pp. 34–64.

[22] M. CONTI, N. DRAGONI, AND V. LESYK, *A survey of man in the middle attacks*, IEEE communications surveys & tutorials, 18 (2016), pp. 2027–2051.

[23] Z. CUI, R. KE, AND Y. WANG, *Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction*, arXiv preprint arXiv:1801.02143, (2018).

[24] E. DAI, T. ZHAO, H. ZHU, J. XU, Z. GUO, H. LIU, J. TANG, AND S. WANG, *A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability*, arXiv preprint arXiv:2204.08570, (2022).

[25] H. DAI, H. LI, T. TIAN, X. HUANG, L. WANG, J. ZHU, AND L. SONG, *Adversarial attack on graph structured data*, in Proc. International conference on machine learning, 2018, pp. 1115–1124.

[26] C. DWORK, *Differential privacy: A survey of results*, in International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.

[27] C. DWORK, A. ROTH, ET AL., *The algorithmic foundations of differential privacy.*, Foundations and Trends in Theoretical Computer Science, 9 (2014), pp. 211–407.

[28] K. ELDEFRAWY, G. TSUDIK, A. FRANCILLON, AND D. PERITO, *Smart: Secure and minimal architecture for (establishing dynamic) root of trust.*, in Ndss, vol. 12, 2012, pp. 1–15.

[29] W. FAN, Y. MA, Q. LI, J. WANG, G. CAI, J. TANG, AND D. YIN, *A graph neural network framework for social recommendations*, IEEE Transactions on Knowledge and Data Engineering, 34 (2020), pp. 2033–2047.

[30] M. FANG, X. CAO, J. JIA, AND N. GONG, *Local model poisoning attacks to byzantine-robust federated learning*, in 29th {USENIX} Security Symposium ({USENIX} Security 20), 2020, pp. 1605–1622.

[31] J. FENG, C. RONG, F. SUN, D. GUO, AND Y. LI, *Pmf: A privacy-preserving human mobility prediction framework via federated learning*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 4 (2020), pp. 1–21.

[32] 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*, in 23rd USENIX Security Symposium (USENIX Security 14), 2014, pp. 17–32.

[33] C. FUNG, C. J. YOON, AND I. BESCHASTNIKH, *Mitigating sybils in federated learning poisoning*, arXiv preprint arXiv:1808.04866, (2018).

[34] T. GU, B. DOLAN-GAVITT, AND S. GARG, *Badnets: Identifying vulnerabilities in the machine learning model supply chain*, arXiv preprint arXiv:1708.06733, (2017).

[35] W. HAMILTON, Z. YING, AND J. LESKOVEC, *Inductive representation learning on large graphs*, Proc. Advances in neural information processing systems, 30 (2017).

[36] M. U. HASSAN, M. H. REHMANI, AND J. CHEN, *Differential privacy techniques for cyber physical systems: a survey*, IEEE Communications Surveys & Tutorials, 22 (2019), pp. 746–789.

[37] C. HE, E. CEYANI, K. BALASUBRAMANIAN, M. ANNAVARAM, AND S. AVESTIMEHR, *Spreadgnn: Serverless multi-task federated learning for graph neural networks*, arXiv preprint arXiv:2106.02743, (2021).

[38] X. HE, J. JIA, M. BACKES, N. Z. GONG, AND Y. ZHANG, *Stealing links from graph neural networks*, in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2669–2686.

[39] B. HITAJ, G. ATENIESE, AND F. PEREZ-CRUZ, *Deep models under the gan: information leakage from collaborative deep learning*, in Proceedings of the 2017

ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 603–618.

[40] R. D. HJELM, A. FEDOROV, S. LAVOIE-MARCHILDON, K. GREWAL, P. BACHMAN, A. TRISCHLER, AND Y. BENGIO, *Learning deep representations by mutual information estimation and maximization*, in Proc. International Conference on Learning Representations, 2018.

[41] C. HUANG, H. XU, Y. XU, P. DAI, L. XIA, M. LU, L. BO, H. XING, X. LAI, AND Y. YE, *Knowledge-aware coupled graph neural network for social recommendation*, in Proceedings of the AAAI conference on artificial intelligence, vol. 35, 2021, pp. 4115–4122.

[42] B. JAYARAMAN AND D. EVANS, *Evaluating differentially private machine learning in practice*, in 28th {USENIX} Security Symposium ({USENIX} Security 19), 2019, pp. 1895–1912.

[43] 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, 2019, pp. 259–274.

[44] D. JIANG, Z. WU, C.-Y. HSIEH, G. CHEN, B. LIAO, Z. WANG, C. SHEN, D. CAO, J. WU, AND T. HOU, *Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models*, Journal of cheminformatics, 13 (2021), pp. 1–23.

[45] H. JIANG, J. PEI, D. YU, J. YU, B. GONG, AND X. CHENG, *Applications of differential privacy in social network analysis: a survey*, IEEE Transactions on Knowledge and Data Engineering, (2021).

[46] G. KARYPIS AND V. KUMAR, *Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, tech. rep., Computer Science & Engineering (CS&E) Technical Reports, 1997.

[47] S. P. KASIVISWANATHAN, H. K. LEE, K. NISSIM, S. RASKHODNIKOVA, AND A. SMITH, *What can we learn privately?*, SIAM Journal on Computing, 40 (2011), pp. 793–826.

[48] J. KIM AND M. HASTAK, *Social network analysis: Characteristics of online social networks after a disaster*, International journal of information management, 38 (2018), pp. 86–96.

[49] T. N. KIPF AND M. WELLING, *Variational graph autoencoders*, arXiv preprint arXiv:1611.07308, (2016).

[50] T. N. KIPF AND M. WELLING, *Semi-supervised classification with graph convolutional networks*, in Proc. International Conference on Learning Representations, 2017.

[51] S.-Y. KUNG, *Compressive privacy: From information\/estimation theory to machine learning [lecture notes]*, IEEE Signal Processing Magazine, 34 (2017), pp. 94–112.

[52] A. LALITHA, O. C. KILINC, T. JAVIDI, AND F. KOUSHANFAR, *Peer-to-peer federated learning on graphs*, arXiv preprint arXiv:1901.11173, (2019).

[53] B. LI, Y. WU, J. SONG, R. LU, T. LI, AND L. ZHAO, *Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems*, IEEE Transactions on Industrial Informatics, (2020), pp. 1–1.

[54] L. LI, P. WANG, J. YAN, Y. WANG, S. LI, J. JIANG, Z. SUN, B. TANG, T.-H. CHANG, S. WANG, ET AL., *Real-world data medical knowledge graph: construction and applications*, Artificial intelligence in medicine, 103 (2020), p. 101817.

[55] R. LI, X. YUAN, M. RADFAR, P. MARENDY, W. NI, T. J. O,ÄôBRIEN, AND P. M. CASILLAS-ESPINOSA, *Graph signal processing, graph neural network and graph learning on biological data: a systematic review*, IEEE Reviews in Biomedical Engineering, 16 (2021), pp. 109–135.

[56] X. LI, K. HUANG, W. YANG, S. WANG, AND Z. ZHANG, *On the convergence of fedavg on non-iid data*, arXiv preprint arXiv:1907.02189, (2019).

[57] Y. LI, R. YU, C. SHAHABI, AND Y. LIU, *Diffusion convolutional recurrent neural network: Data-driven traffic forecasting*, arXiv preprint arXiv:1707.01926, (2017).

[58] Y. LIANG, D. HE, AND D. CHEN, *Poisoning attack on load forecasting*, in 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), IEEE, 2019, pp. 1230–1235.

[59] Y. LIU, J. J. YU, J. KANG, D. NIYATO, AND S. ZHANG, *Privacy-preserving traffic flow prediction: A federated learning approach*, IEEE Internet of Things Journal, 7 (2020), pp. 7751–7763.

[60] Y. LU, X. HUANG, Y. DAI, S. MAHARJAN, AND Y. ZHANG, *Differentially private asynchronous federated learning for mobile edge computing in urban informatics*, IEEE Transactions on Industrial Informatics, 16 (2020), pp. 2134–2143.

[61] L. LYU, H. YU, AND Q. YANG, *Threats to federated learning: A survey*, arXiv preprint arXiv:2003.02133, (2020).

[62] Y. MAO, Z. ZHAO, G. YAN, Y. LIU, T. LAN, L. SONG, AND W. DING, *Communication-efficient federated learning with adaptive quantization*, ACM Transactions on Intelligent Systems and Technology (TIST), 13 (2022), pp. 1–26.

[63] B. MCMAHAN, E. MOORE, D. RAMAGE, S. HAMPSON, AND B. A. Y ARCAS, *Communication-efficient learning of deep networks from decentralized data*, in Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[64] L. MELIS, C. SONG, E. DE CRISTOFARO, AND V. SHMATIKOV, *Exploiting unintended feature leakage in collaborative learning*, in 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 691–706.

[65] S. K. MOHAMED, A. NOUNU, AND V. NOVÁČEK, *Biological applications of knowledge graph embedding models*, Briefings in bioinformatics, 22 (2021), pp. 1679–1693.

[66] S. A. MYERS, A. SHARMA, P. GUPTA, AND J. LIN, *Information network or social network? the structure of the twitter follow graph*, in Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 493–498.

[67] G. NAMATA, B. LONDON, L. GETOOR, B. HUANG, AND U. EDU, *Query-driven active surveying for collective classification*, in Proc. International Workshop on Mining and Learning with Graphs, vol. 8, 2012, p. 1.

[68] E. NASIRI, K. BERAHMAND, M. ROSTAMI, AND M. DABIRI, *A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding*, Computers in Biology and Medicine, 137 (2021), p. 104772.

[69] M. NASR, R. SHOKRI, AND A. HOUMANSADR, *Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning*, in 2019 IEEE symposium on security and privacy (SP), IEEE, 2019, pp. 739–753.

[70] M. NIEPERT, M. AHMED, AND K. KUTZKOV, *Learning convolutional neural networks for graphs*, in Proc. International Conference on Machine Learning, 2016, pp. 2014–2023.

[71] S. NOWOZIN, B. CSEKE, AND R. TOMIOKA, *f-gan: Training generative neural samplers using variational divergence minimization*, Proc. Advances in neural information processing systems, 29 (2016).

[72] I. E. OLATUNJI, W. NEJDL, AND M. KHOSLA, *Membership inference attack on graph neural networks*, in 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, 2021, pp. 11–20.

[73] N. PAPERNOT, P. MCDANIEL, A. SINHA, AND M. WELLMAN, *Towards the science of security and privacy in machine learning*, arXiv preprint arXiv:1611.03814, (2016).

[74] N. PAPERNOT, P. MCDANIEL, A. SINHA, AND M. P. WELLMAN, *Sok: Security and privacy in machine learning*, in 2018 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2018, pp. 399–414.

[75] G. A. PAVLOPOULOS, M. SECRIER, C. N. MOSCHOPOULOS, T. G. SOLDATOS, S. KOSSIDA, J. AERTS, R. SCHNEIDER, AND P. G. BAGOS, *Using graph theory to analyze biological networks*, BioData mining, 4 (2011), pp. 1–27.

[76] Z. PENG, W. HUANG, M. LUO, Q. ZHENG, Y. RONG, T. XU, AND J. HUANG, *Graph representation learning via graphical mutual information maximization*, in Proc. The Web Conference 2020, 2020, pp. 259–270.

[77] H. REN, J. DENG, AND X. XIE, *Grnn: generative regression neural network,Äîa data leakage attack for federated learning*, ACM Transactions on Intelligent Systems and Technology (TIST), 13 (2022), pp. 1–24.

[78] D. RON AND A. SHAMIR, *Quantitative analysis of the full bitcoin transaction graph*, in Proc. International Conference on Financial Cryptography and Data Security, Springer, 2013, pp. 6–24.

[79] R. ROSSI AND N. AHMED, *The network data repository with interactive graph analytics and visualization*, in Proc. AAAI conference on artificial intelligence, 2015.

[80] M. SABT, M. ACHEMLAL, AND A. BOUABDALLAH, *Trusted execution environment: what it is, and what it is not*, in 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, IEEE, 2015, pp. 57–64.

[81] S. SAJADMANESH AND D. GATICA-PEREZ, *Locally private graph neural networks*, in Proc. ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 2130–2145.

[82] T. SARLOS, *Improved approximation algorithms for large matrices via random projections*, in 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), IEEE, 2006, pp. 143–152.

[83] P. SEN, G. NAMATA, M. BILGIC, L. GETOOR, B. GALLIGHER, AND T. ELIASSI-RAD, *Collective classification in network data*, AI magazine, 29 (2008), pp. 93–93.

[84] A. SHAFAHI, W. R. HUANG, M. NAJIBI, O. SUCIU, C. STUDER, T. DUMITRAS, AND T. GOLDSTEIN, *Poison frogs! targeted clean-label poisoning attacks on neural networks*, arXiv preprint arXiv:1804.00792, (2018).

[85] R. SHOKRI, M. STRONATI, C. SONG, AND V. SHMATIKOV, *Membership inference attacks against machine learning models*, in 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 3–18.

[86] C. SONG, Y. LIN, S. GUO, AND H. WAN, *Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting*, in Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, pp. 914–921.

[87] J. STEINHARDT, P. W. KOH, AND P. LIANG, *Certified defenses for data poisoning attacks*, arXiv preprint arXiv:1706.03691, (2017).

[88] F.-Y. SUN, J. HOFFMAN, V. VERMA, AND J. TANG, *Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization*, in Proc. International Conference on Learning Representations, 2020.

[89] L. SUN, Y. DOU, C. YANG, K. ZHANG, J. WANG, P. S. YU, L. HE, AND B. LI, *Adversarial attack and defense on graph data: A survey*, IEEE Transactions on Knowledge and Data Engineering, (2022), pp. 1–20.

[90] Q. SUN, J. LI, H. PENG, J. WU, X. FU, C. JI, AND S. Y. PHILIP, *Graph structure learning with variational information bottleneck*, in Proc. AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4165–4174.

[91] L. TANG AND H. LIU, *Graph mining applications to social network analysis*, Managing and mining graph data, (2010), pp. 487–513.

[92] M. TASUMI, K. IWAHANA, N. YANAI, K. SHISHIDO, T. SHIMIZU, Y. HIGUCHI, I. MORIKAWA, AND J. YAJIMA, *First to possess his statistics: Data-free model extraction attack on tabular data*, arXiv preprint arXiv:2109.14857, (2021).

[93] N. TISHBY, F. C. PEREIRA, AND W. BIALEK, *The information bottleneck method*, in Proc. Annual Allerton Conference on Communication, Control and Computing, 1999, pp. 368–377.

[94] F. TRAMÈR, F. ZHANG, A. JUELS, M. K. REITER, AND T. RISTENPART, *Stealing machine learning models via prediction {APIs}*, in 25th USENIX security symposium (USENIX Security 16), 2016, pp. 601–618.

[95] J.-B. TRUONG, P. MAINI, R. J. WALLS, AND N. PAPERNOT, *Data-free model extraction*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4771–4780.

[96] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. U. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008.

[97] P. VELIČKOVIĆ, G. CUCURULL, A. CASANOVA, A. ROMERO, P. LIÒ, AND Y. BENGIO, *Graph attention networks*, in Proc. International Conference on Learning Representations, 2018.

[98] P. VELICKOVIC, W. FEDUS, W. L. HAMILTON, P. LIÒ, Y. BENGIO, AND R. D. HJELM, *Deep graph infomax*, in Proc. International Conference on Learning Representations, 2019.

[99] P. VOIGT AND A. VON DEM BUSSCHE, *The eu general data protection regulation (gdpr)*, A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10 (2017), pp. 10–5555.

[100] B. WANG, A. LI, H. LI, AND Y. CHEN, *Graphfl: A federated learning framework for semi-supervised node classification on graphs*, arXiv preprint arXiv:2012.04187, (2020).

[101] 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).

[102] R. WANG, X. HE, R. YU, W. QIU, B. AN, AND Z. RABINOVICH, *Learning efficient multi-agent communication: An information bottleneck approach*, in Proc. International Conference on Machine Learning, 2020, pp. 9908–9918.

[103] X. WANG, J. LI, X. KUANG, Y.-A. TAN, AND J. LI, *The security of machine learning in an adversarial setting: A survey*, Journal of Parallel and Distributed Computing, 130 (2019), pp. 12–23.

[104] Z. WANG, M. SONG, Z. ZHANG, Y. SONG, Q. WANG, AND H. QI, *Beyond inferring class representatives: User-level privacy leakage from federated learning*, in IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 2512–2520.

[105] K. WEI, J. LI, M. DING, C. MA, H. H. YANG, F. FAROKHI, S. JIN, T. Q. QUEK, AND H. V. POOR, *Federated learning with differential privacy: Algorithms and performance analysis*, IEEE Transactions on Information Forensics and Security, 15 (2020), pp. 3454–3469.

[106] O. WIEDER, S. KOHLBACHER, M. KUENEMANN, A. GARON, P. DUCROT, T. SEIDEL, AND T. LANGER, *A compact review of molecular property prediction with graph neural networks*, Drug Discovery Today: Technologies, 37 (2020), pp. 1–12.

[107] B. WU, Y. BIAN, H. ZHANG, J. LI, J. YU, L. CHEN, C. CHEN, AND J. HUANG, *Trustworthy graph learning: Reliability, explainability, and privacy protection*, in Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4838–4839.

[108] B. WU, J. LI, J. YU, Y. BIAN, H. ZHANG, C. CHEN, C. HOU, G. FU, L. CHEN, T. XU, ET AL., *A survey of trustworthy graph learning: Reliability, explainability, and privacy protection*, arXiv preprint arXiv:2205.10014, (2022).

[109] B. WU, X. YANG, S. PAN, AND X. YUAN, *Model extraction attacks on graph neural networks: Taxonomy and realisation*, in Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, 2022, pp. 337–350.

[110] C. WU, F. WU, Y. CAO, Y. HUANG, AND X. XIE, *Fedgnn: Federated graph neural network for privacy-preserving recommendation*, arXiv preprint arXiv:2102.04925, (2021).

[111] H. WU, C. WANG, Y. TYSHETSKIY, A. DOCHERTY, K. LU, AND L. ZHU, *Adversarial examples for graph data: deep insights into attack and defense*, in Proc. International Joint Conference on Artificial Intelligence, 2019, pp. 4816–4823.

[112] T. WU, H. REN, P. LI, AND J. LESKOVEC, *Graph information bottleneck*, Proc. Advances in Neural Information Processing Systems, 33 (2020), pp. 20437–20448.

[113] Z. WU, S. PAN, F. CHEN, G. LONG, C. ZHANG, AND S. Y. PHILIP, *A comprehensive survey on graph neural networks*, IEEE transactions on neural networks and learning systems, 32 (2020), pp. 4–24.

[114] Z. WU, S. PAN, G. LONG, J. JIANG, AND C. ZHANG, *Graph WaveNet for deep spatial-temporal graph modeling*, in Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 1907–1913.

[115] F. XIA, K. SUN, S. YU, A. AZIZ, L. WAN, S. PAN, AND H. LIU, *Graph learning: A survey*, IEEE Transactions on Artificial Intelligence, 2 (2021), pp. 109–127.

[116] C. XIAO, B. LI, J. Y. ZHU, W. HE, M. LIU, AND D. SONG, *Generating adversarial examples with adversarial networks*, in 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, International Joint Conferences on Artificial Intelligence, 2018, pp. 3905–3911.

[117] B. XU, N. WANG, T. CHEN, AND M. LI, *Empirical evaluation of rectified activations in convolutional network*, arXiv preprint arXiv:1505.00853, (2015).

[118] 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*, in Proc. International Joint Conference on Artificial Intelligence, 2019, pp. 3961–3967.

[119] H. YAN, X. LI, H. LI, J. LI, W. SUN, AND F. LI, *Monitoring-based differential privacy mechanism against query flooding-based model extraction attack*, IEEE Transactions on Dependable and Secure Computing, (2021).

[120] F. YANG, K. FAN, D. SONG, AND H. LIN, *Graph-based prediction of protein-protein interactions with attributed signed graph embedding*, BMC bioinformatics, 21 (2020), pp. 1–16.

[121] Q. YANG, Y. LIU, T. CHEN, AND Y. TONG, *Federated machine learning: Concept and applications*, ACM Transactions on Intelligent Systems and Technology (TIST), 10 (2019), pp. 1–19.

[122] H. YAO, F. WU, J. KE, X. TANG, Y. JIA, S. LU, P. GONG, J. YE, AND Z. LI, *Deep multi-view spatial-temporal network for taxi demand prediction*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[123] D. YIN, Y. CHEN, R. KANNAN, AND P. BARTLETT, *Byzantine-robust distributed learning: Towards optimal statistical rates*, in International Conference on Machine Learning, PMLR, 2018, pp. 5650–5659.

[124] B. YU, H. YIN, AND Z. ZHU, *Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting*, in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3634–3640.

[125] H. YU, H. LI, D. MAO, AND Q. CAI, *A relationship extraction method for domain knowledge graph construction*, World Wide Web, 23 (2020), pp. 735–753.

[126] J. YU, T. XU, Y. RONG, Y. BIAN, J. HUANG, AND R. HE, *Graph information bottleneck for subgraph recognition*, in Proc. International Conference on Learning Representations, 2020.

[127] J. J. Q. YU AND J. GU, *Real-time traffic speed estimation with graph convolutional generative autoencoder*, IEEE Transactions on Intelligent Transportation Systems, 20 (2019), pp. 3940–3951.

[128] C. ZHANG, S. LI, J. XIA, W. WANG, F. YAN, AND Y. LIU, {*BatchCrypt*}*: Efficient homomorphic encryption for* {*Cross-Silo*} *federated learning*, in 2020 USENIX Annual Technical Conference (USENIX ATC 20), 2020, pp. 493–506.

[129] C. ZHANG, Z. TIAN, J. JAMES, AND S. YU, *Construct new graphs using information bottleneck against property inference attacks*, in ICC 2023-IEEE International Conference on Communications, IEEE, 2023, pp. 765–770.

[130] C. ZHANG, W. WANG, J. J. YU, AND S. YU, *Extracting privacy-preserving subgraphs in federated graph learning using information bottleneck*, in Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security, 2023, pp. 109–121.

[131] C. ZHANG, S. ZHANG, J. J. YU, AND S. YU, *FASTGNN: A topological information protected federated learning approach for traffic speed forecasting*, IEEE Transactions on Industrial Informatics, (2021).

[132] 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).

[133] J. ZHANG, C. DE SA, I. MITLIAGKAS, AND C. RÉ, *Parallel sgd: When does averaging help?*, arXiv preprint arXiv:1606.07365, (2016).

[134] K. ZHANG, C. YANG, X. LI, L. SUN, AND S. M. YIU, *Subgraph federated learning with missing neighbor generation*, Advances in Neural Information Processing Systems, 34 (2021).

[135] Q. ZHANG, J. CHANG, G. MENG, S. XU, S. XIANG, AND C. PAN, *Learning graph structure via graph convolutional networks*, Pattern Recognition, 95 (2019), pp. 308–318.

[136] X.-M. ZHANG, L. LIANG, L. LIU, AND M.-J. TANG, *Graph neural networks and their current applications in bioinformatics*, Frontiers in genetics, 12 (2021).

[137] Z. ZHANG, M. CHEN, M. BACKES, Y. SHEN, AND Y. ZHANG, *Inference attacks against graph neural networks*, in Proc. USENIX Security Symposium, 2022, pp. 1–18.

[138] Z. ZHANG, Q. LIU, Z. HUANG, H. WANG, C.-K. LEE, AND E. CHEN, *Model inversion attacks against graph neural networks*, IEEE Transactions on Knowledge and Data Engineering, (2022).

[139] Z. ZHANG, Q. LIU, Z. HUANG, H. WANG, C. LU, C. LIU, AND E. CHEN, *Graphmi: Extracting private graph data from graph neural networks*, in Proc. International Joint Conference on Artificial Intelligence, 2021, pp. 3749–3755.

[140] Z. ZHANG, Q. LIU, Z. HUANG, H. WANG, C. LU, C. LIU, AND E. CHEN, *Graphmi: Extracting private graph data from graph neural networks*, in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Z.-H. Zhou, ed., International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 3749–3755.

[141] B. ZHAO, K. R. MOPURI, AND H. BILEN, *idlg: Improved deep leakage from gradients*, arXiv preprint arXiv:2001.02610, (2020).

[142] L. ZHAO, Q. WANG, Q. ZOU, Y. ZHANG, AND Y. CHEN, *Privacy-preserving collaborative deep learning with unreliable participants*, IEEE Transactions on Information Forensics and Security, 15 (2019), pp. 1486–1500.

[143] W. ZHAO AND X. WU, *Boosting entity-aware image captioning with multi-modal knowledge graph*, IEEE Transactions on Multimedia, (2023).

[144] Y. ZHAO, J. ZHAO, M. YANG, T. WANG, N. WANG, L. LYU, D. NIYATO, AND K.-Y. LAM, *Local differential privacy-based federated learning for internet of things*, IEEE Internet of Things Journal, 8 (2020), pp. 8836–8853.

[145] L. ZHENG, J. ZHOU, C. CHEN, B. WU, L. WANG, AND B. ZHANG, *Asfgnn: Automated separated-federated graph neural network*, Peer-to-Peer Networking and Applications, 14 (2021), pp. 1692–1704.

[146] L. ZHU AND S. HAN, *Deep leakage from gradients*, in Federated Learning, Springer, 2020, pp. 17–31.

[147] L. ZHU, Z. LIU, AND S. HAN, *Deep leakage from gradients*, Advances in Neural Information Processing Systems, 32 (2019).

[148] Y. ZHU, Y. CHENG, H. ZHOU, AND Y. LU, *Hermes attack: Steal {DNN} models with lossless inference accuracy*, in 30th USENIX Security Symposium (USENIX Security 21), 2021.

[149] D. ZÜGNER AND S. GÜNNEMANN, *Adversarial attacks on graph neural networks via meta learning*, in Proc. International Conference on Learning Representations, 2018.