Finance Fraud Detection by Advanced Deep Graph Learning

by

Enxia Li

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Australian Artificial Intelligence Institute (AAII)
The School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney

May 2026

Certificate of Original Authorship

I, Enxia Li, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the 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 was supported by an Australian Government Research Training Program (RTP) scholarship (doi.org/10.82133/C42F-K220).

Signed: Production Note:
Signature removed prior to publication.

Date: 10-Dec-2025

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Professor Ling Chen, for her invaluable guidance and unwavering support throughout my PhD study. She has devoted tremendous time and effort to supervising my research, from the early exploration of ideas to the final completion of this dissertation. With great foresight, she carefully selected fraud detection in financial as my research direction, fully recognizing my academic background in accounting and helping me bridge the gap between finance and data science with advanced graph neural network techniques. I have greatly benefited from her profound insight, rigorous attitude, and persistent encouragement. Deeply inspired by her passion for research and her pursuit of excellence, I am determined to continue exploring this important field beyond my graduation.

I would also like to express my sincere gratitude to my co-supervisor, Professor Lu Qin, for his invaluable guidance and support throughout my Ph.D. journey. His profound expertise in graph management and analytics has significantly elevated the quality and depth of my research. I am particularly grateful to my research partners: Dr. Sheng Xiang, Dr. Hanchen Wang, Jin Ouyang and Jianke Yu, especially for their help in the implementations of algorithms and models.

Last but not least, I would like to express my sincere gratitude to my husband, Dr Ying Zhang, for his unwavering love and steadfast support throughout my PhD journey. My deepest appreciation also goes to my parents and my family for their enduring love and encouragement. I would especially like to thank my mother for her exceptional support, unconditional care, and strength throughout these years. I hope to set a positive example for my nieces and nephew Ying, Ying and Heng, inspiring them to pursue their future careers with confidence and determination. And finally, a special mention to my little cat, Angel, who insisted on lying on my laptop for "quality control" and supervising my work—whether I asked for it or not.

Finance Fraud Detection by Advanced Deep Graph Learning

by

Enxia Li

Abstract

Graphs are powerful tools for modeling complex relationships in financial data, such as transaction networks, user-account associations, and inter-institutional connections. Detecting financial fraud, however, faces critical challenges: evolving fraudulent patterns, imbalanced data distributions, and the need for efficient analysis of large-scale financial systems. To address these issues, this thesis proposes advanced deep graph learning-based approaches for accurate and scalable financial fraud detection, aiming to fill academic gaps and provide practical solutions for financial institutions.

The first chapter lays the research foundation by outlining the background, significance, and objectives of financial fraud detection. It clarifies the core problems to be solved, including the limitations of traditional detection methods in handling complex financial relationships and the urgency of adapting to dynamic fraud tactics. This chapter also presents the overall structure of the thesis and summarizes key contributions. Chapter 2 systematically reviews existing research on financial fraud detection. It covers traditional machine learning methods, supervised/semi-supervised/unsupervised graph-based approaches, and recent deep graph learning techniques. The review identifies research gaps: insufficient exploration of semi-supervised learning for imbalanced fraud data, limited utilization of heterogeneous financial information, and inadequate attention to the robustness of detection models against adversarial attacks.

The third chapter focuses on semi-supervised credit card fraud detection. It proposes a semi-supervised deep graph learning framework that leverages a small amount of labeled fraud data and large-scale unlabeled transaction data. The method enhances the representation of suspicious patterns through graph contrastive learning, effectively addressing the data imbalance problem in credit card fraud detection and improving detection accuracy. Chapter 4 develops a heterogeneous

graph neural network (GNN) for fraud detection. It constructs a heterogeneous financial graph integrating multi-type entities (users, accounts, merchants) and relations (transactions, bindings), and designs a specialized aggregation mechanism to capture complex interactive patterns. This framework outperforms traditional homogeneous GNNs by fully utilizing heterogeneous information in financial systems. Chapter 5 explores black box attacks against credit card fraud detection models. It analyzes the vulnerability of existing detection systems to adversarial attacks and proposes a targeted black box attack method tailored to credit card fraud scenarios. This research provides insights for improving the robustness of fraud detection models and guides the design of more secure detection systems.

Chapter 6 summarizes the key contributions of the thesis, including the semi-supervised deep graph learning framework, heterogeneous GNN-based detection approach, and black box attack analysis for fraud detection. It discusses future research directions, such as integrating multimodal financial data, enhancing model interpretability for regulatory compliance, and developing real-time adaptive detection systems. This work advances the state of the art in financial fraud detection with deep graph learning and offers actionable references for both academic research and industrial applications.

Publications

Journal Articles

1. **Li E.**, Chen M., Xiang S., Chen L., "Graph-Learning-Empowered Financial Fraud Detection: Progress and Future Directions". *Intelligent Computing*, 2025

2. **Li E.**, Ouyang J., Xiang S., Qin L., Chen L., "Efficient Relation-Aware Heterogeneous Graph Neural Network for Fraud Detection", *World Wide Web Journal*, 2025

3. Li M., Wang Y., Zhang P., Wang H., Fan L., **Li E.**, Wang W., "Deep Learning for Approximate Nearest Neighbour Search: A Survey and Future Directions", *IEEE Transactions on Knowledge and Data Engineering*, 2023

4. Yu J., **Li E.**, Xiang S., Chen L., Qin L., Huang N., Wang K., Yang Z., "Black-box Attack for Credit Card Fraud Detection via Adaptive Perturbation", *Information Sciences* (Under Minor Revision)

Conference Papers

1. **Li E.**, Ouyang J., Xiang S., Qin L., Chen L., "Relation-Aware Heterogeneous Graph Neural Network for Fraud Detection", *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 2024

2. Xiang S., Zhu M., Cheng D., **Li E.**, Zhao R., Ouyang Y., Chen L., Zheng Y., "Semi-supervised Credit Card Fraud Detection via Attribute-Driven Graph Representation", *AAAI 37th Conference on Artificial Intelligence,*, 2023.

Contents

Declaration of Authorship ii

Acknowledgements iii

Abstract v

Publications vii

List of Figures xiii

List of Tables xv

1 Introduction 1
- 1.1 Preliminaries . . . 2
  - 1.1.1 Financial Fraud Detection via Machine Learning . . . 2
  - 1.1.2 Graph Empowered Financial Fraud Detection . . . 3
- 1.2 Motivation . . . 4
  - 1.2.1 Key Objectives . . . 5
  - 1.2.2 Thesis Structure . . . 6

2 Literature Review 9
- 2.1 Chapter Overview . . . 9
  - 2.1.1 Scope . . . 9
- 2.2 Unsupervised Approaches . . . 11
  - 2.2.1 Overview . . . 11
  - 2.2.2 Anti-Money Laundering . . . 12
    - 2.2.2.1 **FlowScope** [29]. . . . 12
    - 2.2.2.2 **AntiBenford Subgraph** by [30]. . . . 13
- 2.3 Semi-supervised Approaches . . . 14
  - 2.3.1 Overview . . . 14
  - 2.3.2 Credit Card Fraud Detection . . . 15
    - 2.3.2.1 **Semi-GNN** [32]. . . . 16
    - 2.3.2.2 FMLN [31]. . . . 17

2.3.2.3 GTAN [22]. 18
2.3.3 Insurance Fraud Detection 20
2.3.3.1 InfDetect [33] 20
2.3.4 Anti-Money Laundering 20
2.3.4.1 GNN-AML [34] 21
2.3.4.2 GraphSense [35] 22
2.4 Supervised Approaches 23
2.4.1 Overview 23
2.4.2 Credit/Loan Risk Assessment 24
2.4.2.1 Tem-GNN [36] 24
2.4.2.2 HGAR [37] 25
2.4.2.3 DGANN [38] 26
2.4.3 Loan Default Analysis 26
2.4.3.1 ST-GNN [39] 27
2.4.3.2 AMG-DP [40] 27
2.4.4 Insurance Fraud Detection 29
2.4.4.1 INS-GNN [41] 29
2.4.5 Credit Card Fraud Detection 30
2.4.5.1 PC-GNN [42]. 31
2.4.5.2 MAHINDER [43] 32
2.4.5.3 HACUD [44] 33
2.4.6 Anti-Money Laundering 35
2.4.6.1 GAGNN [21] 35
2.5 Discussions and Future Works 36
2.5.1 Financial Fraud Gang Replaces Traditional Isolated Fraud Event 36
2.5.2 The Sensitiveness and Complexity of Graph Data in Financial Fraud Detection 37
2.5.3 Interpretability and Robustness of Graph Neural Network-based Financial Fraud Detection Models 37
2.6 Conclusion 38

3 Semi-Supervised Credit Card Fraud Detection 39
3.1 Chapter Overview 39
3.2 Background of Credit Card Fraud Detection 39
3.3 Related Works 42
3.3.1 Credit Card Fraud Detection 42
3.3.2 Graph-Based Semi-supervised Learning 43
3.4 Proposed Method 43
3.4.1 Model Architecture 44
3.4.2 Attribute Embedding and Feature Learning 44
3.4.3 Gated Temporal Attention Networks 45
3.4.4 Risk Embedding and Propagation 46
3.4.5 Fraud Risk Prediction 47
3.4.6 Masking to Avoid Label Leakage 48

3.5 Experiments . . . 48
3.5.1 Experiment Settings . . . 49
3.5.1.1 Datasets . . . 49
3.5.1.2 Compared Methods. . . . 49
3.5.1.3 Evaluation Metrics . . . 51
3.5.2 Fraud Detection Performance . . . 52
3.5.3 Semi-supervised Experiment . . . 53
3.5.4 Ablation Study . . . 53
3.5.5 Parameter Sensitivity . . . 54
3.6 Conclusion . . . 55

4 Heterogeneous Graph Neural Network for Fraud Detection 57
4.1 Chapter Overview . . . 57
4.2 Background . . . 57
4.3 Preliminaries . . . 60
4.3.1 Problem Definition . . . 60
Heterogeneous Graph: . . . 61
Meta-path: . . . 61
Graph-based Fraud Detection: . . . 61
4.3.2 Related Works . . . 62
4.3.2.1 Non-Graph Based Fraud Detection . . . 62
4.3.2.2 Graph-Based Fraud Detection . . . 62
4.3.2.3 Heterogeneous Graph Neural Networks . . . 63
4.4 Proposed Method . . . 63
4.4.1 Model Architecture . . . 63
4.4.2 Computation Graph Pre-Process . . . 65
4.4.3 Heterogeneous Propagation . . . 66
4.4.4 Fraud Detection and Optimization . . . 67
4.5 Experiments . . . 68
4.5.1 Experiments Settings . . . 68
4.5.1.1 Datasets . . . 68
4.5.1.2 Compared Methods . . . 69
4.5.1.3 Parameter Setting and Evaluation Metrics . . . 69
4.5.2 Fraud Detection Experiment . . . 69
4.5.3 Ablation Study . . . 70
4.5.4 Parameter Sensitivity . . . 71
4.6 Conclusion . . . 72

5 Black Box Attack for Credit Card Fraud Detection 75
5.1 Chapter Overview . . . 75
5.2 Background . . . 75
5.3 Related Work . . . 79
5.4 Preliminaries . . . 81
5.5 Proposed Model . . . 82

5.5.1 Preprocessing stage 83
5.5.2 Ensemble-guided Adversarial Generation 84
5.5.3 Memory-based Adversarial Defense 86
5.5.4 Adversarial Learning Stage 88
5.6 Experiments 90
5.6.1 Experimental Setup 90
5.6.2 Effectiveness of Attack Algorithms 93
5.6.3 Ablation Study 96
5.6.4 Parameter Sensitivity Analysis 96
5.7 Conclusion 99

6 EPILOGUE **101**

Appendices **102**

Bibliography **103**

List of Figures

3.1 The illustration of typical credit card fraud detection process. The detection system of card issuer assesses each transaction with an online predictive model once it has passed account checking. 40
3.2 The illustration of the proposed model architecture and temporal graph attention mechanism. 42
3.3 The result of semi-supervised experiments with different ratios of labeled training data. 51
3.4 The ablation study results on three datasets. Gray bars represent the GTAN-A variant, blue bars represent the GTAN-R variant, and red bars represent the GTAN model. 52
3.5 Parameter sensitivity analysis with respect to (a) the number of GNN layers; (b) the number of temporal edges per node; (c) hidden dimension; and (d) the batch size. 54

4.1 An example of a heterogeneous graph in the financial field, showing multiple types of nodes/edges in real-world graphs. 59
4.2 Overall Framework of the proposed Relation-Aware Heterogeneous Graph Neural Network model (RHGNN). (1) Pre-computation Stage: To leverage the efficiency of hybrid propagation, the main framework of RHGNN consists of alternating propagation iterations, where we introduce a Stochastic Projection Reduction component to manage dimensionality reduction for updated vertex representations. To exploit the granularity of relation-specific information for low information loss, we introduce a hybrid propagation scheme, which alternates focus between 1-hop and 2-hop neighbors, collecting neighbor information based on different relational contexts separately. (2) Training Stage: After K iterations, the collected neighbor information (of target vertices) is used as the input for a heterogeneous graph encoder to generate the vertex representations of target vertices, enhancing the model's ability to detect fraudulent activities accurately and efficiently. 64
4.3 Ablation study results on three datasets. 70
4.4 Parameter sensitivity analysis for the RHGNN model. 71

5.1 Framework of BADA 83
5.2 SEIs results for Learning Rate 97
5.3 SEIs results for Max Iterations 98

List of Tables

2.1 Categories, names, applications, and datasets of graph learning for financial fraud detection. 10

3.1 Statistics of the three fraud detection datasets. 49
3.2 Fraud detection performance on three datasets, compared with popular benchmark methods. We report the results of the area under the roc curve (AUC), macro average of F1 score (F1-macro), and averaged precision (AP). 50

4.1 Notation Summary. 60
4.2 Fraud Detection Experimental Results. 70

5.1 Notation Table 82
5.2 Dataset statistics 90
5.3 SEIs of target models under attacks on dataset LC 93
5.4 SEIs of target models under attacks on dataset IEEE 95
5.5 SEIs of target models under attacks on dataset IEEE-80K 95
5.6 Ablation study on LC 97

Chapter 1

Introduction

The significant losses resulting from financial fraud have consistently drawn the focus of academia, industry, and regulatory communities. As discussed, fighting financial fraud is beneficial for sustainable economic growth, which is one of the United Nations' Sustainable Development Goals (SDGs) [1, 2]. For instance, the online payment system can help to cut 400 million metric tons of annual carbon emissions from physical currency, which is crucial in combating climate change. However, fraudulent behaviors against online payments and unauthorized credit card transactions, have inflicted financial harm upon online payment users and hindered the promotion of online payment [3]. In the financial area, fraud behavior is widely modeled and discussed. For instance, Tergiman and Villeval experimentally study the nature of lies and designed the 'Announcement Game' to evaluate how the introduction of reputation affects lies [4]. Also, Liu et al. study the involvement of AI in credit scoring for loan lenders and indicate that AI techniques will benefit different aspects of lending including fraud detection [5]. At the same time, more theoretical models in specific scenarios that are easily affected by fraud behaviors are studied, such as mobile app ranking fraud [6–9]. Designing efficient and effective finance fraud detection methods can reduce the operating costs of service providers, protect the property of bank users, and help to establish a more sustainable finance system.

Due to the importance of financial fraud detection, a number of methods have been proposed in existing literature, falling into two broad categories: rule-based approaches and machine-learning-based approaches. Generally speaking, rule-based methods rely on a variety of rules generated by

domain experts to identify suspicious behaviors. Though rule-based approaches are usually highly efficient and accurate in some scenarios, it is difficult for experts to identify proper rules when the context is complicated and dynamic. Moreover, fraudsters can learn the underlying logic of rules and then develop corresponding strategies to fool the system. In recent years, many machine-learning-based methods have been developed to train models from historical data such as logistic regression, decision tree, and support vector machine. To better capture the interrelation among the finance data (e.g., transactions) and identify potential fraud behaviors, graphs have been widely used to model finance activities. Moreover, in addition to the traditional machine learning models, deep learning techniques have been applied in various applications with outstanding performance such as Anti-money Laundering and credit card fraud detection, thanks to the advance of graph machine learning.

1.1 Preliminaries

In this section, we first introduce the background of machine learning approaches for financial fraud detection, with a focus on graph-empowered approaches. The fundamental concepts of graph learning are introduced in Section 1.1.2.

1.1.1 Financial Fraud Detection via Machine Learning

Current research in financial fraud detection can be broadly categorized into rule-based methodologies and machine learning-based methodologies. For instance, to extract association-based clues for detecting fraud, Seeja and Zareapoor proposed a rule-based method [10]. Balagolla et al. propose a blockchain-based approach to avoid fraudulent transactions [11]. Weinmann et al. utilize trace data to detect fraud by analyzing movements captured via a computer mouse [12]. Although rule-based approaches are highly efficient and accurate in some scenarios, it is difficult for experts to identify proper rules when the context is complicated and dynamic. Many machine-learning-based approaches have been proposed to help detect finance fraud. For instance, , Fiore et al. study feature extraction using neural networks and developed supervised classifiers to identify 362 fraudulent transactions [13]. Xiao et al. introduce a nonlinear optimization model to generate black-box attacks and evaluate the performance of different machine learning–based credit

card fraud detection models [14]. Xu et al. additional categories of behavioral features from P2P lending transaction data, aiming to enhance various performance metrics in fraud detection [15]. Additionally, anomaly detection via machine learning techniques can also provide insights for fraud detection in the financial area [16–20]. In recent years, advanced machine learning techniques, such as graph representation learning, have shown superior performance in financial fraud detection [21, 22]. Therefore, existing machine learning-based fraud detection solutions can be classified into non-graph machine learning-based approaches and graph representation learning approaches. For instance, Fu et al. further study the usage of automatic feature engineering in a convolutional neural network (CNN) [23]. Recently, finance data have been modeled by graphs and graph neural network techniques have been deployed to detect finance fraud behaviors. For example, Cheng et al. propose a graph neural network model via spatio-temporal attention mechanism for the detection of fraud credit card transactions [24]. And Jing et al. propose a learning method based on graph neural networks which can meet the effectiveness in credit card fraud detection with few data samples [25].

1.1.2 Graph Empowered Financial Fraud Detection

The realm of financial systems has witnessed profound evolution in both structure and complexity over the past few decades. A significant component of this evolution can be attributed to the intricate interconnectivity established between financial entities, transactions, and activities. Graph theory, a branch of mathematics dealing with networks of interconnected nodes and edges, offers a robust framework to represent and analyze this complex web of financial inter-dependencies. Research in graph theory and its application has been very active. For example, Zhou et al. propose a novel "reduce-solve-combine" search strategy that integrates a problem reduction mechanism to effectively tackle the common challenge of identifying critical nodes [26]. The surge in financial malpractices has necessitated the exploration of advanced techniques, such as graph neural networks (GNNs), to identify and combat fraudulent activities within these interconnected financial systems.

The following are two representative financial applications that can greatly benefit from the modeling and analytical capabilities of graph structures:

- **Networked-Loan Fraud Detection:** In a financial ecosystem, the manifestation of loans often entails a series of transactions and relationships between various entities, such as banks, borrowers, guarantors, companies, and intermediary organizations. By constructing a graph wherein entities are represented as nodes and transactions or relationships as edges, one can effectively capture the patterns and anomalies associated with potential fraudulent loan activities. For instance, a closed-loop of entities lending to each other without clear business justification could indicate a circular lending scheme, a common strategy to illicitly inflate assets or siphon off funds. As reported, companies participating in a real-world loan network are dynamic and complex [27]. Graph learning models, with their capacity to consider relational dependencies, offer a promising avenue for detecting such sophisticated networked-loan frauds that might remain elusive to conventional detection techniques.

- **Group-Based Money-Laundering Detection:** Money laundering, the act of disguising the origins of ill-gotten money, typically involves intricate transactional networks to obfuscate the trail of illicit funds. Group-based money laundering involves coordinated efforts by a consortium of individuals or entities to launder money in a manner that evades detection. Representing financial transactions as a graph enables the identification of suspicious clusters or subgraphs where the patterns of transactions deviate from legitimate behavior. Graph learning models can be trained to recognize these anomalous patterns, capturing the relational dependencies between entities and thereby spotlighting potential laundering groups as validated [21].

In summary, as financial frauds grow in complexity and cunning, leveraging the topological and relational properties of graphs, in conjunction with the power of graph learning models, offers a promising frontier in the fight against these malicious activities. This survey aims to delve deeper into the methodologies and breakthroughs surrounding this interdisciplinary confluence of finance and graph neural networks.

1.2 Motivation

Traditional isolated fraud events have evolved into intricate networks of fraudulent activities. The sophistication of these activities, coupled with the massive and scattered nature of financial data,

poses significant challenges to existing fraud detection methods. In recent years, graph learning has emerged as a potential solution for addressing financial fraud effectively. By modeling entities as nodes and relationships as edges, graph learning techniques can capture complex patterns and relationships in the data, making them particularly suited for detecting organized fraud activities. To our best knowledge, there is no literature review to comprehensively introduce the current progress of graph-learning-based financial fraud detection approaches. Besides, it is worthwhile to discuss the challenges in the use of graph learning in financial fraud detection, such as the sensitivity and complexity of graph data, as well as the interpretability and robustness of graph-learning-based models, with some promising research directions in this important area.

The main contributions of our literature review (chapter 2) can be summarized as follows.

- A comprehensive analysis of applications of graph learning in the field of financial fraud detection is presented, highlighting their advantages over traditional approaches. We stress the novelty and adaptability of introducing graph learning in fraud detection and how they can improve traditional machine-learning-based and rule-based methods.
- Graph-learning-based financial fraud detection methods are categorized, including unsupervised, semi-supervised, and supervised techniques. Mechanisms and applications of each method are discussed in detail to show their advantages and availability in different scenarios.
- Future research directions regarding graph-learning-based finance fraud detection are illustrated corresponding to their features discussed, including their dependability, scalability, and robustness, which are eagerly expected in modern financial fraud research trends.

1.2.1 Key Objectives

This thesis aims to address the critical gaps in graph-learning-based financial fraud detection by pursuing four core objectives: Systematically synthesize existing research on financial fraud detection, clarifying the advantages and limitations of graph learning techniques compared to traditional methods. Propose a semi-supervised graph learning framework tailored for credit card fraud detection, mitigating the data imbalance issue in real-world fraud scenarios. Develop a specialized

heterogeneous graph neural network to fully exploit multi-type financial entities and relationships, enhancing the detection of complex fraudulent patterns. Analyze the vulnerability of existing fraud detection models and design a targeted black box attack method, providing insights for improving model robustness.

1.2.2 Thesis Structure

To achieve the above objectives, this thesis is organized into six chapters with distinct focuses:

- Chapter 1 (Introduction): Establishes the research background, significance, and motivation of financial fraud detection. It clarifies the core challenges, outlines key research objectives, summarizes the main contributions of this work, and provides an overview of the thesis structure.
- Chapter 2 (Literature Review): Conducts a comprehensive review of existing graph learning-based financial fraud detection methods, including supervised/semi-supervised/unsupervised techniques. It categorizes relevant studies, identifies research gaps, and lays the theoretical foundation for subsequent innovations.
- Chapter 3 (Semi-supervised Credit Card Fraud Detection): Proposes a semi-supervised deep graph learning framework that leverages limited labeled fraud data and large-scale unlabeled transaction data. It addresses the data imbalance problem through graph contrastive learning, improving the accuracy of credit card fraud detection.
- Chapter 4 (Heterogeneous Graph Neural Network for Fraud Detection): Constructs a heterogeneous financial graph integrating multi-type entities (users, accounts, merchants) and relations (transactions, bindings). A specialized aggregation mechanism is designed to capture complex interactive patterns, outperforming traditional homogeneous graph models.
- Chapter 5 (Black Box Attack for Credit Card Fraud Detection): Analyzes the vulnerability of existing credit card fraud detection models to adversarial attacks. It proposes a scenario-adapted black box attack method, providing a benchmark for evaluating model robustness and guiding the design of more secure detection systems.

- Chapter 6 (Epilogue): Summarizes the key contributions and innovations of this thesis. It discusses the limitations of the current work and outlines promising future research directions, including multimodal data integration, model interpretability enhancement, and real-time adaptive detection systems.

Chapter 2

Literature Review

2.1 Chapter Overview

This chapter establishes a foundational framework for finance fraud detection with advanced deep graph learning, and classifies them in terms of the learning paradigm. The content has been previously published in [28]. Section 2.2 explores unsupervised graph-based methods for fraud detection, focusing on anomaly detection techniques that leverage network topology without labeled data. Section 2.3 investigates semi-supervised approaches that integrate limited labeled examples with extensive unlabeled data to improve model generalization in imbalanced financial datasets. Section 2.4 reviews supervised learning frameworks utilizing graph neural networks for fraud classification, emphasizing their ability to capture complex node and edge relationships. Section 2.5 conducts a comparative analysis of the three paradigms, highlighting their respective strengths, limitations, and applicability in different financial scenarios. Section 2.6 summarizes key findings from the literature review and identifies critical research gaps that motivate our proposed methodology.

2.1.1 Scope

This survey paper primarily focuses on the application of graph learning techniques in the domain of financial fraud detection. In order to structure our exploration and understand the evolving

Category	Model	Applications	Datasets
Unsupervised	Flowscope [29]	Anti-Money Laundering	CBank, CFD
	AntiBenford [30]	Anti-Money Laundering	Ethereum
Semi-supervised	GTAN [22]	Credit Card Fraud Detection	Amazon, YelpChi, FFSD
	FMLN [31]	Credit Card Fraud Detection	ECC, RA, SD and Vesta
	Semi-GNN [32]	Credit Card Fraud Detection	AliPay
	InfDetect [33]	Insurance Fraud Detection	Ant Financial Services
	GNN-AML [34]	Anti-Money Laundering	AMLSim
	GraphSense [35]	Anti-Money Laundering	Bitcoin Ransomware
Supervised	Tem-GNN [36]	Credit/Loan Risk Assessment	AliPay
	HGAR [37]	Credit/Loan Risk Assessment	Commercial Bank in Asia
	DGANN [38]	Credit/Loan Risk Assessment	Financial Institution in East Asia
	ST-GNN [39]	Loan Default Analysis	AliPay
	AMG-DP [40]	Loan Default Analysis	Ant Credit Pay
	INS-GNN [41]	Insurance Fraud Detection	Ant Finance
	PC-GNN [42]	Credit Card Fraud Detection	Amazon, YelpChi
	MAHINDER [43]	Credit Card Fraud Detection	AliPay
	HACUD [44]	Credit Card Fraud Detection	Ant Credit Pay
	GAGNN [21]	Anti-Money Laundering	Amazon, YelpChi, UnionPay

TABLE 2.1: Categories, names, applications, and datasets of graph learning for financial fraud detection.

landscape of graph-learning-based fraud detection methodologies, we adopt a taxonomy based on the nature of learning paradigms employed, as shown in Table 2.1.

- **Unsupervised graph learning Techniques:** Techniques under this category do not rely on labeled datasets for training. They aim to capture the intrinsic patterns and anomalies within the financial transaction graphs. By identifying deviations from normative behaviors, these models present avenues for uncovering novel, previously unidentified fraudulent strategies.
- **Semi-Supervised graph learning Techniques:** Situated between the supervised and unsupervised paradigms, these techniques employ a blend of labeled and unlabeled data. The primary advantage lies in their ability to leverage vast amounts of unlabeled data while utilizing a limited amount of labeled instances to guide the learning process. This often results in more generalizable models that can detect nuanced fraudulent patterns.
- **Supervised graph learning Techniques:** This category encompasses methodologies where the Graph-based models are trained with labeled data, comprising instances of both genuine and fraudulent financial activities. By leveraging the annotated datasets, these models aim to

learn patterns and relationships indicative of fraud, offering the potential for high precision in detecting malicious activities.

While the aforementioned taxonomy provides a clear and concise categorization, it is worth noting that the landscape of graph learning applications in fraud detection is vast and multifaceted. Several alternative taxonomies can be devised to characterize the diverse methodologies in this area. For instance:

- *Based on Graph Construction:* This can categorize methods depending on how they construct graphs, such as static vs. dynamic graphs, weighted vs. unweighted edges, or multi-layered vs. single-layered graphs.
- *By Fraud Domain Application:* Techniques could also be grouped based on specific financial fraud applications they target, such as credit card fraud, insider trading, or digital currency scams.
- *Considering Graph Learning Framework* A categorization can be based on the underlying graph learning framework, distinguishing between models like GraphSAGE, DeepWalk, or hybrid methods with tree-based machine learning methods.

As we venture into the depths of graph-learning-based methodologies for financial fraud detection, it is paramount to acknowledge the diversity and richness of the techniques and their underlying principles. This survey, while emphasizing the supervised vs semi-supervised vs unsupervised taxonomy, will touch upon the aforementioned categories, aiming to offer a comprehensive and holistic overview of the field.

2.2 Unsupervised Approaches

2.2.1 Overview

The unsupervised paradigm of graph learning techniques for financial fraud detection primarily focuses on the discovery of underlying patterns without the need for labeled data. By leveraging

intrinsic properties and anomalies within financial transaction graphs, unsupervised graph learning methods offer the potential to unearth novel and sophisticated fraudulent strategies that might elude supervised techniques.

A crucial application benefiting from these unsupervised techniques is anti-money laundering, where the absence of predefined fraudulent patterns demands adaptable and evolving detection mechanisms. Although in this review paper, the unsupervised approaches are for anti-money laundering, they can be applied to fraud detection or risk assessment as well.

2.2.2 Anti-Money Laundering

In this section, two main related works in this topic are discussed [29, 30]. Both of them introduces new algorithms to detect abnormal financial activities in graph data. They both introduce novel criterion into the task and show higher efficiency than all existing methods.

2.2.2.1 FlowScope [29].

The paper addresses the detection of money laundering in bank transfer networks, highlighting the lack of flow tracking in current methods and inadequate theoretical guarantees. It introduces the FlowScope algorithm, which models the money transfer network as a graph $G = (V, E)$, where $V = X \cup W \cup Y$ represents the set of accounts. Here, W denotes the inner bank accounts, while X and Y correspond to outer accounts receiving net inflows and outflows, respectively. Each edge $e_{ij} \in E$ represents the aggregate monetary transfer from account v_i to v_j. The algorithm tracks transaction flows from source to destination, focusing on patterns like large deposits, internal transfers, and withdrawals, which are harder for fraudsters to hide. FlowScope also aims to predict intermediary layers in laundering schemes, advocating for a maximum transaction step limit to mitigate risks. A novel density-based metric in FlowScope ensures efficient detection of dense money flows and caps the transferable amount by fraudsters. The algorithm proves more effective than existing benchmarks in various tests, with its computational complexity nearly linear to transaction volume.

- **Anomalousness of ML for k-partite subgraph**:

Anomalous flow is characterized from a node set A, through single or multiple layers of intermediate accounts M, to a different node set C. For an intermediary account v_i, let $f_i(S)$ and $q_i(S)$ denote the lower and upper bounds of the cumulative weighted out-degree and in-degree with respect to the node group $S \subseteq V$. The anomalousness metric is defined as:

$$g_k(S) = \frac{1}{|S|} \sum_{l=1}^{k-2} \sum_{v_i \in M_l} f_i(S) - \lambda(q_i(S) - f_i(S))$$

or equivalently,

$$g_k(S) = \frac{1}{|S|} \sum_{l=1}^{k-2} \sum_{v_i \in M_l} (1+\lambda) f_i(S) - \lambda q_i(S), \quad k \geq 3$$

where λ (the imbalance cost rate) quantifies the adversity fraudsters encounter per unit of retention or shortfall, representing the cost of concealment.

- **FlowScope algorithm**:

  FlowScope algorithm seeks the optimal subset S that maximizes $g(S)$. Initially, it constructs a priority hierarchy for nodes in S, where each node v_i is assigned a weight or priority, defined as:

$$w_i(S) = \begin{cases} f_i(S) - \frac{\lambda}{1+\lambda} q_i(S), & \text{if } v_i \in M_l \\ d_i(S), & \text{if } v_i \in A \cup C \end{cases}$$

  where $d_i(S)$ denotes a specified degree function for boundary accounts in $A \cup C$.

2.2.2.2 AntiBenford Subgraph by [30].

This paper proposes AntiBenford subgraphs, which is an unsupervised framework aiming to detect abnormal subgraphs in financial networks. Its goal is to identify a group of nodes within a large transaction or financial graph that significantly diverges from Benford's law, a principle that describes the expected distribution of the first digit in each number among data. The framework utilizes a dense subgraph discovery algorithm to identify these anomalous subgraphs, which are

characterized by both their edge density and the deviation of their edge weights' distribution of the first digit from Benford's law. With adequate experiments, AntiBenford subgraphs is proven to be able to detect anomalies in real-world networks that other anomaly detection strategies based on graph may not notice.

- **Anomaly Score**:

  For each vertex $v \in V$ in a financial network $G = (V, E)$, an anomaly score $s(v)$ is defined using a χ^2 score compared to Benford's distribution:

$$s(v) = \Sigma_{d=1}^{9} \frac{(X_d^u - \mathbb{E}(X_d^u))^2}{\mathbb{E}(X_d^u)}$$

  where X_d^u is the number of transactions, or edges in the graph, related to u whose first digit is d.

- **AntiBenford Subgraph Detection**:

  An antiBenford subgraph detection algorithm is designed in this work based on the above anomaly score. First, the score of each vertex is computed. Then edge $e(u, v)$ is weighted using function $f(u, v) = \sqrt{s(u) \cdot s(v)}$. Then, most of the densest subgraph problem is solved by minimizing the objective $max_{S \subset V} \frac{\Sigma f(u,v)}{|S|}$, which namely represents a subgraph with large average weight for each edge. While this paper gives another objective $max_{S \subset V} \frac{e(S)+\Sigma s(v)}{|S|}$ to avoid including vertices that have low scores themselves but with higher degree towards an anomalous set of nodes.

2.3 Semi-supervised Approaches

2.3.1 Overview

Semi-supervised graph learning approaches in the domain of financial fraud detection leverage the strength of both labeled and unlabeled data to uncover and predict fraudulent activities. This hybrid approach provides a robust framework for scenarios where labeled data is limited, yet unlabeled transactions are abundant. By combining the knowledge from known instances of fraud with

the intrinsic patterns present in the broader dataset, semi-supervised graph learning techniques have demonstrated remarkable accuracy and adaptability. Two prominent applications that have substantially benefited from semi-supervised graph learning techniques are:

- **Credit Card Fraud Detection:** In this domain, the rapid influx of transaction data often results in a scenario where only a fraction of the data is labeled, typically the confirmed fraud cases. Semi-supervised methods can bridge this gap, effectively using the limited labeled data to guide the model's learning from the vast swathes of unlabeled transactions, identifying potentially suspicious activities that might have otherwise been overlooked.
- **Insurance Fraud Detection:** Insurance fraud encompasses a wide range of illicit activities, from exaggerated claims to false information on policy applications. Like in other domains, labeled instances of insurance fraud are scarce compared to the volume of overall claims and applications. Semi-supervised approaches adeptly analyze patterns within this largely unlabeled data, helping to identify fraudulent claims or applications by learning from the smaller number of known fraud cases. This approach is particularly effective in adapting to new methods of fraud that constantly emerge in the insurance sector.
- **Anti-Money Laundering (AML):** Money laundering schemes often manifest as intricate patterns that subtly deviate from normative behaviors. While there might be known instances or patterns of money laundering, the ever-evolving tactics used by perpetrators demand a more flexible detection mechanism. Here, semi-supervised graph learning techniques prove invaluable, extrapolating from known laundering patterns to detect newer, subtler schemes in the broader, mostly unlabeled dataset.

The versatility and adaptability of semi-supervised graph learning techniques make them a cornerstone in the modern arsenal against financial fraud, catering to diverse applications and constantly evolving threats.

2.3.2 Credit Card Fraud Detection

Here we introduce three main existing works (**GTAN** [22], **FMLN** [31], and **Semi-GNN** [32]) that fall in this scope. Note that FMLN is also federated-learning-based method. GTAN has achieves

the best performance in both semi-supervised and supervised fraud detection tasks according to experimental results reported [22]. Semi-GNN [32] is the first work, compared with the other two, to use graph-attentive networks for financial fraud detection.

2.3.2.1 Semi-GNN [32].

Current financial fraud detection methods like neural networks [45] and SVM [46] mainly use machine learning to identify fraud patterns, focusing on users' statistical features from profiles, behaviors, and transactions. Yet, these methods often overlook user interactions, which are crucial in financial settings. Traditional machine learning approaches, using models like logistic regression or neural networks, fail to consider these interactions and the rich data they offer, such as social connections and merchant transactions. Additionally, most fraud detection data is unlabeled, a challenge not fully addressed by existing graph-based methods, which also lack interpretability. Semi-GNN addresses these gaps by leveraging both labeled and unlabeled data for fraud detection, bridging supervised and unsupervised learning. It confronts the challenge of integrating sparse labeled data with abundant unlabeled data to enhance detection performance using novel techniques.

- **Low-level view-specific user embedding:**

$$h_k^u = \sigma\left(\sum_{i \in N_v^u} a_{ui}^{(k)} W^{(k)} x_i\right)$$

  where N_v^u is the set of neighbors of user u in the v-th view-specific graph, $a_{ui}^{(k)}$ is the attention coefficient between user u and neighbor i in the k-th layer, $W^{(k)}$ is the weight matrix in the k-th layer, and x_i is the input feature of neighbor i. The function σ is a non-linear activation function.

- **High-level view-specific user embedding:**

$$h_L^u = \sigma\left(\sum_{k=1}^{L} \phi_k h_k^u\right)$$

  , where L is the number of layers, h_k^u is the low-level embedding of user u in the k-th layer, and ϕ_k is the layer-specific weight. The function σ is a non-linear activation function.

2.3.2.2 FMLN [31].

The publicly available dataset for fraudulent credit card detection is imbalanced and limited. Approximately 2% of all credit card transactions are associated with fraudulent activities. Owing to concerns related to data security and privacy, individual banks are typically unwilling to share their transaction datasets with one another. Besides, traditional techniques [23, 47] neglect the exploration of the relationship among training samples. They only focus on the comparison with samples from different classes.

To tackle these problems, the paper proposes a novel meta-learning-based model FMLN for the task of detecting fraudulent credit card activities. This model leverages a federated learning strategy. This approach enables various banks to collectively train a shared model while maintaining the privacy of their imbalanced training data, which is stored within their individual private databases. The proposed approach demonstrates notably improved performance in comparison to state-of-the-art methods when evaluated on four publicly available datasets. The paper also introduces a novel meta-learning-based classifier to effectively learn the discriminative feature extraction on unseen class samples. This strategy delivers remarkably superior performance beyond other meta-learning-based methods.

- **Feature Extraction Model:** The paper uses a K-Tuplet Network utilizing ResNet-34 as the feature extraction tool. The output of this model is considered to be network features. The feature extraction model, denoted as f_{network}, produces feature maps to represent the feature extraction function.

- **Relation Model:** The relation model, denoted as J_{relation}, depicts the similarity between the support set $D_{c_{\text{support}}}$ and the query set $D_{c_{\text{query}}}$. The relation model takes the network features and applies a sigmoid function and consider the output as similarity scores.

- **Federated Meta-Learning Framework:** The paper applies federated learning to facilitate the collaborative sharing of datasets among different banks, enabling the construction of an efficient fraud detection model without compromising the privacy of each bank's customers. The objective loss function is defined as:

$$\min l(x_i; y_i; w), \quad \text{where} \quad l(x_i; y_i; w) = \frac{1}{n_c} \sum_{i \in D_i} L_{c_{\text{relation}}}(x_{i_c}; y_{i_c}; w) \tag{2.1}$$

Here, $n_{\text{whole}} = \sum_{c=1}^{C} n_c$ represents all the data samples involved in the whole process. $L_{c_{\text{relation}}}$ is the loss function for the relation model.

- **K-Tuplet Network:** An enhanced metric learning approach, referred to as the deep K-tuplet network, has been developed. This network extends the capabilities of the triplet network by enabling simultaneous comparisons with K negative samples within each mini-batch.
- **Federated Learning:** The paper uses a horizontal federated learning framework, which is used in scenarios where datasets have similar feature space but differ in samples. The server initializes all parameters of the fraud detection model. At each communication round t, a random fraction seed of banks is selected. These banks establish direct communication with the server, retrieve the latest global model parameters, and compute the average gradient of the loss function f_c using their respective private datasets at the current fraud detection model parameters, employing a fixed learning rate η.

2.3.2.3 GTAN [22].

In this work, the authors note that in real-world applications, only a small portion of transactions are labeled as fraudulent or non-fraudulent. This lack of labeled data makes it difficult to train effective machine-learning models for fraud detection.

To address this challenge, the authors propose a semi-supervised learning approach that leverages both labeled and unlabeled data. Their method, called Attribute-Driven Graph Representation Learning, uses the attributes of transactions to construct a graph and then applies graph representation learning to capture the complex relationships between transactions. By employing this approach, the model can learn from both labeled and unlabeled data, thereby enhancing its capacity to identify fraudulent transactions.

The model is designed to handle the challenge of limited labeled data in credit card fraud detection by leveraging labeled and unlabeled data in a semi-supervised learning approach. The technical contribution can be concluded as follows:

- **Attribute Embedding and Feature Learning:** The authors introduce a novel approach for transforming the numerical attributes of each record into a tensor format, symbolized as $X_{num} \in \mathbb{R}^{N \times d}$. Here, N is the total count of transactions, and d represents the dimensions of the features. For categorical attributes like card, transaction, and merchant category, they adopt distinct attribute embedding layers to formulate these attributes into a separate matrix $X_{cat} \in \mathbb{R}^{N \times ds}$. These embeddings are then merged to form a comprehensive categorical representation of each transaction, computed by add-pooling as $x^{(u)}cat = \sum ix^{(u)}cat, i$, where i encompasses various categories such as card, transaction, and merchant. The notation $x^{(u)}cat \in \mathbb{R}^{1 \times d}$ signifies the aggregated category embedding vector for each transaction record indexed by u.

- **Gated Temporal Attention Networks:** In this section, the authors utilize a sequence of transaction embeddings, denoted as $X = \{x_{t0}, x_{t1}, ...x_{tn}\}$, to capture the temporal dynamics of each transaction. This is achieved by integrating both categorical and numerical attributes as inputs to the Gated Temporal Attention Network (GTAN). The formulation is given by $x_{ti} = x_{num}^{(ti)} + x_{cat}^{(ti)}$. Initially, at the first TGAT layer, $H_0 = X$ is set as the base input embedding matrix. They apply a multi-head attention mechanism to evaluate the significance of each neighboring transaction and accordingly update the embeddings, enhancing the model's ability to discern relevant temporal patterns.

- **Risk Embedding and Propagation:** Here, the authors suggest embedding the partially observable risk attributes (or labels) into a unified feature space, aligning them with other node features. This process involves creating risk embedding vectors for nodes with labels and assigning zero vectors for those without. The combined node features, now inclusive of risk embeddings, are represented as $x_{ti} = x_{num}^{(ti)} + x_{cat}^{(ti)} + \hat{y}^{(t1)}W_r$, with W_r denoting the tunable parameters for risk embedding. This method allows for a more nuanced incorporation of risk factors into the overall feature set, potentially enhancing the model's predictive accuracy.

- **Fraud Risk Prediction Techniques:** Upon aggregating the transaction embeddings, the authors utilize a sophisticated two-layer Multi-Layer Perceptron (MLP) to estimate fraud risk. This process is articulated as $\hat{y} = \sigma(\text{PReLU}(HW_0 + b_0)W_1 + b_1)$, where $\hat{y} \in \mathbb{R}^{N \times 1}$ encapsulates the predicted risk levels for all transactions. The parameters W and b are adjustable within the MLP, allowing the model to fine-tune its predictions based on the learned

embeddings. This advanced prediction methodology is expected to yield more accurate and reliable fraud risk assessments.

2.3.3 Insurance Fraud Detection

2.3.3.1 InfDetect [33]

The *InfDetect* paper introduces a large-scale, graph-based fraud detection system for e-commerce insurance, leveraging advanced graph learning techniques. This approach employs both supervised and unsupervised graph learning algorithms, including DeepWalk (unsupervised), Graph Neural Networks (supervised and semi-supervised node classification), and DistRep (supervised edge classification), to analyze complex fraud patterns in e-commerce transactions.

Various graph types, such as device-sharing, transaction, and friendship graphs, are analyzed. The system preprocesses these graphs to optimize classification performance by removing isolated accounts. Additionally, the data processing phase involves feature collection and processing, employing techniques like data scaling, categorical feature encoding, and denoising Autoencoder (DAE) for unsupervised feature transformation.

The system's primary objective is to identify fraudsters in the claim stage by classifying accounts or orders as fraudulent. The methodology is particularly effective in detecting fraudster gangs using graph learning techniques. The paper includes both qualitative and quantitative evaluations, demonstrating the system's enhanced fraud detection capabilities over traditional rule-based methods. This novel approach significantly contributes to the field of fraud detection in e-commerce, showcasing the power of graph learning in uncovering and analyzing fraudulent activities.

2.3.4 Anti-Money Laundering

This subsection delves into two sophisticated anti-money laundering (AML) methods: GNN-AML and GraphSense. GNN-AML employs scalable graph convolutional neural networks to analyze complex financial networks, tackling challenges in large, noisy datasets. It emphasizes graph compression and neural network scalability. GraphSense, on the other hand, analyzes Bitcoin transactions related to ransomware using graph-based methodologies, focusing on transaction and

cluster analysis in the Bitcoin blockchain. Both methods highlight innovative approaches to combating financial crimes through advanced data analysis techniques.

2.3.4.1 GNN-AML [34]

As to anti-money laundering (AML), there is a needle-in-a-haystack problem of entity classification and hidden pattern discovery in extensive, constantly changing, high-dimensional, time-series transaction datasets with high noise-to-signal ratios, combinatorial complexity, and non-linearity. Datasets frequently exhibit fragmentation, inaccuracy, incompleteness, and inconsistencies, both within individual organizations and across them. Automating the synthesis of information from diverse data streams proves to be a formidable challenge, often necessitating the involvement of human analysts who may have limited resources at their disposal.

Besides, The challenge of AML involves dealing with vast, high-dimensional graph data that map billions of relationships (edges) among millions of entities (nodes). In the realm of AML transaction monitoring, a node entity can represent either an individual account or a collection of linked accounts, whose connections are already established or inferred through clustering to function as a unified entity. The known attributes encompass explicitly defined data, constituting information that has been specifically collected through standard Know Your Customer (KYC) procedures or multimodal data extracted from public or partner information sources. This category also includes observable transactions and any accompanying flags or Suspicious Activity Reports (SARs) that have been filed. These challenges highlight the complexity and scale of the problem, as well as the societal impact of failing to effectively address it.

The technical contribution of this paper is centered around the use of scalable graph convolutional neural networks for forensic analysis of financial data in anti-money laundering (AML). The GNN-AML paper discusses some key concepts and results:

- **Graph Compression with Ligra+:** The researchers applied a graph compression tool called Ligra+ and conducted experiments on both a simulated AML graph and several deep learning benchmark graphs. Ligra+ serves as a compression system aimed at reducing space usage, while maintaining competitive or improved performance compared to running algorithms on uncompressed graphs using multi-core machines. They managed to obtain a

compression ratio of up to 2x, although compressing larger graphs presents more significant challenges. This development creates the potential for memory-efficient training and inference in deep neural networks by harnessing compression and reordering techniques.

- **Scalable Graph Convolutional Neural Networks:** The paper offers an initial exploration of scalable graph convolutional neural networks for the forensic analysis of vast, dense, and ever-changing financial data. They present preliminary experimental findings based on a sizable synthetic graph generated by their proprietary data simulator, known as AMLSim, featuring 1 million nodes and 9 million edges.

- **High Efficiency:** The paper explores prospects for achieving high efficiency in both computation and memory usage, and it also provides findings from a straightforward experiment involving graph compression. The outcomes of their study substantiate their initial hypothesis that employing graph deep learning for anti-money laundering (AML) holds significant potential in combating illicit financial practices.

2.3.4.2 GraphSense [35]

The *GraphSense* paper introduces a novel methodology for analyzing Bitcoin transactions related to ransomware using advanced graph learning techniques. This approach extends beyond traditional clustering heuristics and is applied to 35 different ransomware families. Key to this methodology are two different types of network representations over the entire Bitcoin blockchain: the *Address Graph* and the *Cluster Graph*. The *Address Graph* represents each Bitcoin address as a vertex and each transaction as a directed edge, facilitating the computation of summary statistics, such as transaction numbers and value flows. The *Cluster Graph* partitions addresses into maximal subsets or clusters, likely controlled by the same real-world actor, using the multiple-input clustering heuristic. This method significantly contributes to dataset expansion, identifying numerous Bitcoin addresses associated with ransomware attacks and tracing their outgoing relationships to key addresses. Furthermore, it assesses the minimum direct financial impact of each ransomware family by analyzing monetary flows to these key addresses. Implemented on the open-source *GraphSense* platform, this approach enables comprehensive transaction extraction and analysis, offering new insights into the ransomware economy and enhancing the understanding of its financial implications through graph analysis.

2.4 Supervised Approaches

2.4.1 Overview

Supervised graph learning techniques for financial fraud detection employ labeled data, which comprises instances of both genuine and fraudulent financial activities, to train models. These labeled datasets serve as foundational knowledge, allowing graph learning models to learn intricate patterns and relationships indicative of fraud or other financial risks. The strength of supervised graph learning techniques lies in their ability to achieve high precision in detecting and categorizing financial activities based on past labeled examples.

The following applications represent notable areas where supervised graph learning techniques have been extensively applied.

- **Credit/Loan Risk Assessment:** Supervised graph learning techniques are used to evaluate the potential risk associated with lending money to individuals or institutions. By analyzing past borrowing behavior and related financial activities, these models can predict the likelihood of timely loan repayment, aiding financial institutions in making informed lending decisions.
- **Loan Default Analysis:** In this domain, supervised models analyze transactional data and financial histories to predict the probability of a borrower defaulting on a loan. Such insights can guide lenders in devising risk mitigation strategies and shaping their lending policies.
- **Insurance Fraud Detection:** Insurance claims come with diverse data points, from personal histories to event details. Supervised graph learning techniques parse through this data to identify patterns that might indicate fraudulent insurance claims, ensuring that insurers can validate claims with higher accuracy.
- **Credit Card Fraud Detection:** Given the vast amounts of daily credit card transactions, detecting fraudulent activities requires models that can rapidly and accurately identify suspicious patterns. Supervised graph learning techniques, trained on historical instances of fraud, excel in pinpointing such activities amidst a sea of legitimate transactions.

- **Anti-Money Laundering (AML):** While semi-supervised methods are crucial for AML due to evolving money laundering tactics, supervised graph learning techniques play a vital role when there's a substantial amount of labeled data on known laundering techniques. These models can detect and flag transactions that align with known illicit money movement strategies, acting as a first line of defense.

The application of supervised graph learning techniques spans a broad spectrum of financial domains, offering tools that are both precise and responsive to the intricacies of each application.

2.4.2 Credit/Loan Risk Assessment

In this subsection, we delve into three cutting-edge methodologies. Firstly, Tem-GNN [36] is notably adept at extracting credit risk insights from dynamic graphs. The HGAR [37] approach employs a higher-order graph attention mechanism to discern the potential of loan defaults. Lastly, DGANN [38] introduces an end-to-end dynamic graph neural network for predicting risks associated with networked loans.

2.4.2.1 Tem-GNN [36]

The authors highlight three challenges in credit risk prediction from temporal graphs: time interval irregularity, integration of structural and temporal information, and static and short-term temporal factors. In response, the Tem-GNN model integrates a pathway for static factors, short-term graph encoders, and a time-series model. Key features include:

- **Static Feature Learning Model:** Using an MLP, the model extracts stable user features and projects them into a high-level space.
- **Short-term Graph Encoder:** Focuses on recent neighbor information through a graph convolution module.
- **Temporal Model:** Incorporates an interval-decayed attention mechanism defined as:

$$\alpha'(t) = \frac{\exp(s'(h_t))}{\sum_{i\in\{0,1,\ldots,T\}} \exp(s'(h_{ti}))} \tag{2.2}$$

where $s'(h_t)$ is the output of a single-layer feed-forward neural network with a tanh activation function. To handle interval irregularity:

$$g(\delta_t) = \frac{1}{\log(e + \delta_t)} \tag{2.3}$$

where δ_t is the time interval between the event t to now.

- **User Embedding and Risk Score Calculation:** Combines temporal attention to define user embedding and uses an MLP for risk score prediction.

2.4.2.2 HGAR [37]

The authors of this work note that companies are allowed to guarantee each other to enhance loan security, but when more and more companies are involved, they form complex guarantee networks. These complex structures can be a double-edged sword for the national economy, especially during economic downturns where many companies may fail to repay loans, causing a domino effect of defaults.

To address this challenge, the paper proposes a high-order graph attention representation method (HGAR) for the default risk assessment of networked guarantee loans. The main contributions of HGAR can be summarized as follows:

- **Binary Roles of Nodes:** Nodes in the guarantee network have binary roles, guarantor or guarantee.
- **Graph Attentional Layer:** Captures node importance in a network.
- **High-Order Adjacent Approximation:** Preserves binary node roles and high-order adjacency.
- **Loan Default Prediction Layer:** Uses both first-order and high-order attentional representation for default prediction.

2.4.2.3 DGANN [38]

Loans from commercial banks form complex directed-network structures, necessitating an adaptive strategy to efficiently identify and address any systematic crises. Therefore, the authors propose an end-to-end dynamic graph-based attention neural network (DGANN) to predict risk guarantees by learning on the interconnected loans in a network. Their approach can be split into three parts:

- **Graph Convolution Network (GCN) with Structure Attention:** Uses temporal guarantee networks for high-order graph representation.
- **Graph Recurrent Network (GRN) with Temporal Attention:** Extracts edge attributes and learns temporal patterns. The model updates the hidden state as:

$$
\begin{aligned}
r_t &= \sigma(W_r * [h_{t-1}, e_t]) \\
z_t &= \sigma(W_z * [h_{t-1}, e_t]) \\
\hat{e}_h &= \tanh(W_e * [r_t \odot h_{t-1}, e_t]) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{e}_h
\end{aligned}
$$

  where r_t and z_t denote the reset and update gates of the tth object respectively, $\hat{e}_h$ represents the representation of the candidate hidden layer, and W are the weights dynamically updated during the model training phase.

- **Prediction Layer:** Estimates risk probability with a global view of guarantee networks.

2.4.3 Loan Default Analysis

In this subsection, we delve into two methods for loan default prediction. Formally, the loan default prediction task falls under node anomalies and edge anomalies in the networked-loans. The ST-GNN [39] targets Small and Medium Enterprises (SMEs), addressing data deficiency issues often seen with online financial institutions. By adeptly extracting genuine supply chain relationships from the SME graph, this approach enhances loan default prediction accuracy. Conversely,

AMG-DP [40] emphasizes the intricate properties intrinsic to loan default prediction: Communicability, Complementation, and Induction. By leveraging relation-specific layers and attention mechanisms, AMG-DP ensures precise predictions, even for new users with sparse data.

2.4.3.1 ST-GNN [39]

This research addresses data scarcity in financial risk analysis for Small and Medium Enterprises (SMEs), particularly in online financial institutions like Ant Financial and WeBank, where acquiring credit-related SME data is challenging.

Key concerns include extracting authentic supply chain relationships from the SME graph and using these relationships in financial risk analysis. The solution involves two components: a Spatial-Temporal aware Graph Neural Network (ST-GNN) and a loan default prediction module:

1. **Spatial-Temporal aware Graph Neural Network (ST-GNN)**: Designed to capture both the local graph structure and temporal dynamics of the SME graph, ST-GNN is trained semi-supervisedly for link prediction, focusing on identifying genuine supply chain links. It assigns confidence scores to each edge, refining the SME graph into a supply-chain graph by removing low-confidence edges.
2. **Loan Default Prediction Module**: Utilizing the refined supply-chain graph, this module aggregates data from neighboring SMEs into a target SME, enabling more accurate financial risk assessments. The method incorporates ST-GNN for supervised node classification to predict future loan repayment failures of SMEs.

In summary, ST-GNN efficiently extracts supply chain relationships and predicts loan defaults for SMEs, offering a novel approach in the realm of financial risk analysis for SMEs.

2.4.3.2 AMG-DP [40]

As to loan default prediction task in financial scenarios, there are three key intrinsic properties of this problem:

- **Communicability:** The patterns of loan default behaviors are discerned through the examination of the local network structure.
- **Complementation:** Describing the profile of a financial defaulter is intricate and challenging to capture comprehensively with just one source of information, particularly for new users who have limited prior behavioral data.
- **Induction:** In practical financial scenarios, new users emerge on a daily basis, necessitating models to retain the capacity to make predictions for this category of users who possess only limited profile information.

Previous solutions have been ineffective in harnessing multiplex relationships in financial settings, thus neglecting these crucial inherent features of loan default detection. The authors propose a novel Attributed Multiplex Graph-based Loan Default Prediction approach (AMG-DP) to address these issues. The authors also acknowledge that the factors leading a user to fail in meeting required repayments are intricate, and expressing such multiplex information is challenging with feature-based approaches. The AMG-DP model is designed with relation-specific receptive layers that utilize an adaptive breadth function to integrate vital information obtained from the local structure in each facet of the AMG. Additionally, the model incorporates multiple propagation layers to investigate higher-order connectivity information. It leverages a relation-specific attention mechanism to highlight pertinent information throughout the end-to-end training process.

The adaptive fusion function is implemented with a relation-specific attention mechanism, which is formulated as follows:

$$\alpha_{u,r} = \frac{\exp(\gamma_r \tanh(W_\gamma h_u + b_\gamma))}{\sum_{r' \in R} \exp(\gamma_{r'} \tanh(W_\gamma h_u + b_\gamma))} \tag{2.4}$$

where $\{W_\gamma, b_\gamma, \gamma_r\}$ form the parameter set of the adaptive fusion function. The function $\tanh(\cdot)$ is used as the activation function.

The relation-specific receptive layer is formulated as follows:

$$\alpha_{u,i} = \frac{\exp(v_r^T \tanh(W_r^{(l)}[h_u^{(l-1)}||h_i^{(l-1)}||e_{u,i}^{(l-1)}]))}{\sum_{(u,j)\in E_r} \exp(v_r^T \tanh(W_r^{(l)}[h_u^{(l-1)}||h_j^{(l-1)}||e_{u,j}^{(l-1)}]))} \tag{2.5}$$

The final user representation is calculated as follows:

$$z_u = \sum_{r \in R} \alpha_{u,r} \cdot h_{u,r} \tag{2.6}$$

The model is evaluated on a large-scale real-world dataset and has shown effectiveness compared with state-of-the-art models. The model also has the inductive ability to predict new users by aggregating their neighbors with multiplex relations.

2.4.4 Insurance Fraud Detection

2.4.4.1 INS-GNN [41]

This paper focuses on the detection and prevention of fraudulent insurance claims, particularly those made by organized groups of fraudsters. This issue is significant in the context of Alibaba's return-freight insurance, which faces numerous potentially fraudulent claims daily. Deliberate misuse of the insurance policy can result in substantial financial losses. The paper outlines three key challenges in fraud detection:

- **Concept drift:** This term describes the situation in which new forms of fraud emerge over time and become progressively less predictable. This primarily happens because fraud detection systems use changing features that don't stay constant.
- **Label uncertainty:** In the past, the rule-based fraud detection system assigned a risk tag to each account, but it's uncertain if the accounts labeled as 'no observable risk' are genuinely risk-free or not.
- **Excessive human effort:** Traditional insurance fraud detection settings require significant human effort for labeling tasks and evaluation tasks. The authors aim to focus on automated risk control that requires negligible human effort, except for a periodical evaluation conducted by insurance professionals.

Importantly, to uncover organized fraudsters and capture complex relations among colluding fraudsters, the paper constructs three types of graphs: the device-sharing graph, the transaction graph, and the friendship graph.

- **Device-sharing graph:** This graph reveals the relations between accounts that share a device. It consists of two types of vertices: device vertices (representing User Machine IDs, UMIDs) and account vertices. Edges exist between device vertices and account vertices, indicating the log-in activities in the history. The device-sharing graph captures the device-sharing relationships among fraudsters and regular customers.
- **Transaction graph:** This graph represents the fund exchange relations between accounts. It consists of account vertices and edges that indicate the existence of established transactions between accounts. The transaction graph provides information about financial interactions between fraudsters and regular customers.
- **Friendship graph:** This graph is built based on the friendship relationships at Alipay, a product of Ant Financial with social networking features. It captures the social connections between accounts. The friendship graph can reveal social networks and connections among fraudsters and regular customers.

The paper compares and analyzes these graphs to determine which is most suitable for fraud detection. Specifically, they focus on the device-sharing graph, which exhibits contrasting patterns between colluding fraudsters and regular customers.

2.4.5 Credit Card Fraud Detection

Researchers have presented techniques to enhance detection rates for credit card fraud detection. Among these are the PC-GNN [42], which seeks to address the class imbalance through a supervised GNN-based approach; the MAHINDER [43] model, a unique tool that capitalizes on Multi-view Attributed Heterogeneous Information Networks to detect defaulters; and HACUD [44], a dedicated method that utilizes a hierarchical attention mechanism for cash-out user detection in an Attributed Heterogeneous Information Network.

2.4.5.1 PC-GNN [42].

The primary challenge tackled by the paper is the issue of imbalanced classes in graph-based fraud detection. This problem arises due to the significant disparity between the number of instances of the majority class (non-fraudulent entities) and the minority class (fraudulent entities). This imbalance often leads to poor performance of detection algorithms, particularly critical for the minority class, which is more important, i.e., the fraudsters.

The paper illustrates three key challenges in creating graph neural networks for fraud detection that grapple with class imbalance:

- **Redundant Link Information:** Fraudsters often employ deceptive tactics, like camouflage, by generating misleading information to make their identification more challenging. For example, spammers may utilize legitimate accounts to post spammy reviews, creating numerous connections between the spam reviews and genuine users, effectively concealing their true intentions within the pool of legitimate users.
- **Lack of Necessary Link Information:** Ones committing fraud usually avoid trading with each other to escape being caught together. For instance, in financial situations, individuals engaging in fraud would steer clear of conducting transactions with each other to evade joint detection. Therefore, there might not exist a connection between the two nodes, which can negatively impact the performance of methods based on Graph Neural Networks (GNNs).
- **Dilution of Minority Class Features:** The difficulty arises during the process of gathering messages in Graph Neural Networks (GNNs), which can result in the dilution of features associated with the minority class. In imbalanced settings, it's common for most of the neighbors connected to a central node to belong to the majority class. Consequently, the characteristics of fraudulent neighbors can be easily neglected, and predictions can be heavily influenced by the majority benign ones.

To address these challenges, the paper proposes a graph learning based imbalanced learning approach for graph-based fraud detection. A label-balanced sampler is designed to select nodes and edges for training. The probability assigned for each node is inversely proportional to its label frequency. This ensures that nodes from the minority class are more likely to be chosen. To

address application challenges, they suggest using a neighborhood sampler to select neighbors based on a distance function with adjustable parameters. To focus on the fraud-labeled node, unnecessary links can be removed by selecting neighbors that are distant from the target based on distance measurements. While essential links contribute to fraud prediction can be established by selecting nodes similar to the fraud class and considering them as neighbors. Specifically, the PC-GNN framework can be decoupled into three steps: Pick, Choose, and Aggregate, which can be described as follows:

- **Pick:** The central nodes are chosen using a balanced sampler to create a well-balanced sub-graph for training in mini-batches. This step helps to ensure that the minority class is sufficiently represented in the training process.
- **Choose:** With a distance function that can be adjusted using parameters, the neighborhood of the minority class is sampled more than required, and that of the majority class is sampled less. This step helps to ensure that the model is exposed to a balanced representation of the classes during training.
- **Aggregate:** Messages from chosen neighbors and various relationships is combined to obtain the ultimate representations of the target. This step helps to ensure that the model captures the relevant information from the graph structure for the prediction task.

2.4.5.2 MAHINDER [43]

The problem of detecting default users on online payment service platforms has at least one of these two challenges: 1) They consider different aspects of user behaviors to create personal profiles, acknowledging the inherent nature of financial default. It's challenging to capture these inherent aspects of default users, and there's an urgent need for a more accurate approach to modeling user profiles; 2) Certain illegal economic activities, like cash-out operations, are closely linked to financial fraud. Criminals engaging in illicit activities might intentionally create intricate behaviors, such as moving money between numerous users and attempting to lengthen the money transfer path to evade regulations. This adversarial nature of financial default makes it challenging to detect.

It's important to mention that the current task involves identifying users at risk of default using multi-view data from diverse information networks. This particular approach hasn't been explored previously.

Therefore, the authors proposed a novel approach called MAHINDER (Multi-view Attributed Heterogeneous Information Network based financial DEfault useR detection) to address these challenges. MAHINDER uses multiple user relationships to construct a multi-view attributed heterogeneous information network (MAHIN) for better user profile modeling. It also takes into account the local structures of assigned meta-paths to extract detailed behavioral patterns at a finer level. The main contribution of MAHINDER can be summarized as follows:

- **Problem Statement:** The issue of identifying financial defaulters is structured as a binary classification problem within a Multi-view Attributed Heterogeneous Information Network (MAHIN). Given an MAHIN $G = \{V, E, X_V, X_E\}$ consisting of m specific views, the purpose is to detect defaulters from the target user set $U \subseteq V$. Each user $u \in U$ is assigned a label $y_u \in \{0, 1\}$ to indicate whether they are a defaulter or not.
- **The MAHINDER Model:** The MAHINDER model is proposed to address the challenges of detecting default users. The model uses multiple user relationships to construct an MAHIN for better user profile modeling. The approach also takes into account the nearby patterns of assigned meta-paths to uncover detailed behavioral trends. The model uses meta-paths from different perspectives to comprehensively model user profiles. It also includes a path encoder based on meta-paths to grasp the local structural patterns present in both nodes and links. Attention mechanisms are used at various levels, including nodes, links, and meta-paths, to automatically determine the significance of different elements. The MAHINDER model includes attention mechanisms to meticulously model the selected paths. There's an additional attention layer applied to different meta-paths to filter out irrelevant perspectives.

2.4.5.3 HACUD [44]

The authors address the challenge of detecting cash-out users in credit payment services, a significant fraud problem in financial services. Cash-out fraud involves users seeking cash gains through illegal or insincere means. The authors identify that conventional detection methods rely mainly

on the statistical features of users and often overlook the interaction relations between them. They propose modeling the problem as a classification task in an Attributed Heterogeneous Information Network (AHIN) and introduce the HACUD model, which uses meta-path-based neighbors and a hierarchical attention mechanism.

The HACUD model is detailed as follows:

- **Feature Transformation:** The original user features are transformed into latent representations. The transformation function is:

$$f' = \sigma(Wf + b) \tag{2.7}$$

  where W is the weight matrix, b is the bias vector, σ is the activation function (ReLU is used), and f is the original feature vector.

- **Neighbor Feature Fusion:** The latent representations of a user and their neighbors, derived from each meta-path, are combined.. The fusion function $g(\cdot, \cdot)$ can be concatenation, addition, or element-wise product.

- **Hierarchical Attention Mechanism:** Different users may have different preferences over features based on meta-paths and attribute information. The attention weight of meta-path ρ for user u is:

$$\beta_{u,\rho} = \frac{\exp(z_\rho^T f''_{uC})}{\sum \exp(z_\rho^T f''_{uC})} \tag{2.8}$$

  where z_ρ is the importance of meta-path ρ and f''_{uC} is the representation of neighbors for user u based on meta-path ρ.

- **Model Learning:** The model aims to minimize the loss function:

$$L(\Theta) = \sum (y_u \log(p_u) + (1 - y_u) \log(1 - p_u)) + \lambda ||\Theta||_2^2 \tag{2.9}$$

  where y_u and p_u are the ground truth and predicted cash-out probability of user u, respectively, Θ is the model's parameter set, and λ is the regularizer parameter. The model is trained using SGD or its variants.

2.4.6 Anti-Money Laundering

2.4.6.1 GAGNN [21]

The main challenge addressed in the work is the detection of money laundering activities in large-scale transaction networks. Traditional methods for detecting money laundering often fail to capture the complex and evolving patterns of these activities. Moreover, they often suffer from high false-positive rates and are unable to effectively leverage the rich relational information in transaction networks.

To address these challenges, the authors propose a novel group-aware deep graph learning framework, GAGNN, which is designed to capture group-level money laundering patterns. The framework is capable of learning the complex and evolving patterns of money laundering activities, reducing false positives, and effectively leveraging the rich relational information in transaction networks.

The technical contribution of this paper can be summarized into three main parts:

- **Group-aware Graph Neural Network (G2N2):** The authors propose a novel graph neural network model, G2N2, which is designed to capture the group structure in financial transactions. The model is defined by the following equation:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{CONCAT}\left(h_v^{(l)}, \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} h_u^{(l)}, \frac{1}{|\mathcal{G}(v)|}\sum_{g\in\mathcal{G}(v)} h_g^{(l)}\right)\right) \tag{2.10}$$

  where $h_v^{(l+1)}$ is the hidden state of node v at layer $l+1$, σ is the activation function, $W^{(l)}$ is the weight matrix at layer l, CONCAT is the concatenation operation, $\mathcal{N}(v)$ is the set of neighbors of node v, $\mathcal{G}(v)$ is the set of groups that node v belongs to, $h_u^{(l)}$ is the hidden state of node u at layer l, and $h_g^{(l)}$ is the hidden state of group g at layer l.

- **Group Embedding:** The authors introduce a method to learn the embedding of groups. The group embedding is computed as the average of the embeddings of the nodes in the group. The group embedding is updated during the training process. The group embedding

is defined by the following equation:

$$h_g^{(l+1)} = \frac{1}{|\mathcal{V}(g)|} \sum_{v \in \mathcal{V}(g)} h_v^{(l+1)} \tag{2.11}$$

where $h_g^{(l+1)}$ is the hidden state of group g at layer $l+1$, $\mathcal{V}(g)$ is the set of nodes in group g, and $h_v^{(l+1)}$ is the hidden state of node v at layer $l+1$.

- **Group-aware Loss Function:** The authors propose a group-aware loss function to train the G2N2 model. The loss function encourages the model to assign similar labels to the nodes in the same group. The group-aware loss function is defined by the following equation:

$$\mathcal{L} = \mathcal{L}_{\text{node}} + \lambda \mathcal{L}_{\text{group}} \tag{2.12}$$

where $\mathcal{L}_{\text{node}}$ is the node classification loss, $\mathcal{L}_{\text{group}}$ is the group consistency loss, and λ is a hyperparameter to balance the two terms.

2.5 Discussions and Future Works

2.5.1 Financial Fraud Gang Replaces Traditional Isolated Fraud Event

Traditionally, financial fraud was often committed by individuals or small groups acting in isolation. However, with the advancement of technology and the increasing sophistication of fraudsters, we are now seeing the emergence of large-scale, organized financial fraud gangs. These gangs often have complex structures and employ advanced techniques, making their activities harder to detect and prevent. A similar topic has been noticed and discussed, but still in need of further research on detection algorithms [48].

Future Work: To tackle this challenge, future research could focus on developing more sophisticated detection algorithms that are capable of identifying complex patterns and relationships indicative of organized fraud. This could involve the use of advanced machine learning techniques, including deep learning and graph neural networks, which are capable of modeling complex structures and relationships. Additionally, cooperation between financial institutions and regulatory bodies could be enhanced to share information and intelligence about suspected fraud gangs.

2.5.2 The Sensitiveness and Complexity of Graph Data in Financial Fraud Detection

Graph data, representing entities (e.g., individuals, accounts) as nodes and relationships (e.g., transactions, connections) as edges, is increasingly being used in financial fraud detection. However, this type of data is often sensitive (due to privacy concerns) and complex (due to the large number of nodes and edges and the potential for complex relationships). This situation has been bothering traditional fraud detection according to Sangers et al. [49]. Privacy and security are highly expected in graph data when detecting financial fraud.

Future Work: Future work could focus on developing methods for effectively and securely handling sensitive graph data. This could involve the use of privacy-preserving techniques, such as differential privacy or federated learning, which allow for the analysis of sensitive data while minimizing the risk of privacy breaches. Additionally, new methods for handling the complexity of graph data, such as scalable graph processing algorithms or graph simplification techniques, can be developed.

2.5.3 Interpretability and Robustness of Graph Neural Network-based Financial Fraud Detection Models

Graph Learning strategies such as Graph Neural Networks (GNNs) have shown great promise in financial fraud detection due to their ability to model complex relationships in graph data. However, these models often lack interpretability, meaning it can be difficult to understand why they have made a particular prediction. Additionally, they may not be robust to changes in the data or to adversarial attacks. Much effort has been made in other related topics by using attention mechanisms and designing feature extraction methods [50, 51]. At the same time, financial fraud detection calls for interpretability more due to the complexity of data.

Future Work: To address the interpretability challenge, future work can focus on developing methods for explaining the predictions of graph learning models. This can involve the use of techniques such as LIME (Local Interpretable Model-Agnostic Explanations) or SHAP (SHapley Additive exPlanations), which provide insights into the contribution of each feature to a model's prediction. To improve the robustness of graph learning models, we can investigate methods for

making these models more resistant to changes in the data or to adversarial attacks. This can be achieved by involving techniques such as adversarial training or robust optimization. Besides, real-world graph data are usually dynamic and time-evolving. Dynamic GNNs [52, 53] can help model temporal dependencies in financial networks

2.6 Conclusion

In this paper, we have provided a comprehensive overview of the current landscape of financial fraud detection, with a particular focus on the use of graph learning techniques. We have traced the evolution of financial fraud from isolated events to complex, organized activities and discussed how this has necessitated a shift in detection methods. Then, we delved into the use of graph learning methods for financial fraud detection. We discussed how those methods, by modeling entities as nodes and relationships as edges, are able to capture the complex patterns and relationships inherent in financial fraud activities. We highlighted the advantages of graph learning methods over traditional methods, including their ability to handle heterogeneous data and their capacity for learning high-level features. However, we also acknowledged that the use of graph learning methods in financial fraud detection is not without its challenges. In the final part of our survey, we discussed these challenges in detail, including the sensitivity and complexity of graph data and the issues of interpretability and robustness in Graph-learning-based models. We highlighted the need for methods that can handle sensitive graph data securely and efficiently, as well as techniques that can improve the interpretability and robustness of graph learning methods.

In conclusion, while graph learning methods hold great promise for financial fraud detection, there are still many open research problems. Future research can focus on addressing the challenges we have identified, as well as exploring new methods and techniques for leveraging the power of graph learning methods. We hope that this survey will serve as a valuable resource for researchers and practitioners in this field and inspire further work in the fight against financial fraud.

Chapter 3

Semi-Supervised Credit Card Fraud Detection

3.1 Chapter Overview

We introduced a novel solution for semi-supervised credit card fraud detection in this chapter, which is published in [22]. Section 3.2 introduces the background of current progress in credit card fraud detection. Section 3.3 gives the literature review on graph-based semi-supervised learning and fraud detection. Section 3.4 introduces our proposed GTAN. Section 3.5 reports the performance comparison between baselines and our GTAN. Section 3.6 concludes the experimental results and discusses the contribution of this chapter.

3.2 Background of Credit Card Fraud Detection

The great losses caused by financial fraud have attracted continuous attention from academia, industry, and regulatory agencies. For instance, as reported in [1, 2], financial fraud detection plays a critical role to support the sustainable economic growth. However, fraudulent behaviors against online payments, such as illegal card swiping, have caused property losses to online payment users [3].

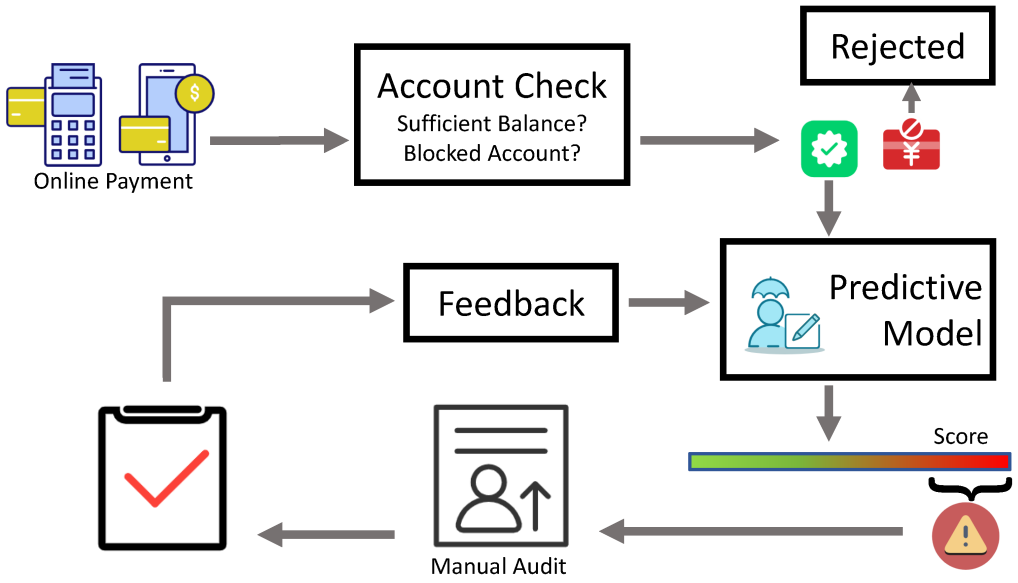


FIGURE 3.1: The illustration of typical credit card fraud detection process. The detection system of card issuer assesses each transaction with an online predictive model once it has passed account checking.

An important line of research in financial fraud detection is credit card fraud detection, where credit card fraud is a general term for the unauthorized use of funds in a transaction, typically by means of a credit or debit card [54]. Figure 3.1 shows a typical fraud detection framework deployed in the commercial system [24]. A direct way to detect fraud is to match each transaction according to specific rules such as card blacklists and budget checking. However, criminals will also obtain the knowledge of vulnerabilities from the response of the pre-designed rule system, thus invalidating the original system. To solve the invalidation problem, the predictive model is designed to automatically detect fraud patterns and produces a fraud risk score. Domain experts then can thereby focus on the high-risk transactions.

State-of-the-art. In the literature, many existing predictive models have been extensively studied to deal with fraud transactions (e.g., [23, 55]), which can be classified into two categories: (1) *Rule-based methods* directly generate sophisticated rules by domain experts to identify the suspicious transactions. For instance, authors in [10] proposed an association rule method for mining frequent fraud rules; (2) *Machine learning-based methods* learn static models by exploring large amounts of historical data. For example, authors in [13] extracted features based on neural networks and built

supervised classifiers for detecting fraudulent transactions. Recently, graph machine learning-based methods have been proposed [51] where the transactions are modeled as a graph, and the advanced graph embedding techniques are deployed.

Motivation. The state-of-the-art fraud detection techniques [56–58] can well capture the temporal or graph-based patterns of the transactions and significantly advance the performance of credit card fraud detection. However, they have at least one of the following three main limitations: (1) ignoring unlabeled data containing rich fraud pattern information; (2) ignoring categorical attributes, which are ubiquitous in the real production environment; and (3) requiring too much time on feature engineering, especially for categorical features.

These motivate us to design a semi-supervised graph neural network for credit card fraud detection. In particular, to capture the relationships among the credit card transactions associated with temporal information, we leverage a temporal transaction graph to model the time-relevant patterns. Besides, labeling the transactions is time-consuming and cost expensive. Only a tiny proportion (much less than 10%) of transactions are labeled in billions of real-life transactions, which contains many fraud patterns that have not been detected. Therefore, it is crucial to exploit the natural features from unlabeled data. In this chapter, we design a Gated Temporal Attention Network (GTAN) for the temporal transaction graph, which can extract temporal fraud patterns and exploit both labeled and unlabeled data. In addition, categorical attributes are ubiquitous and useful in real applications. Therefore, it is necessary to leverage useful information through an attribute-driven model. In this chapter, we introduce an attribute learning layer for preprocessing the transaction attributes and add risk embedding as new categorical attributes, which can better model fraud patterns (e.g., attribute embedding learning and risk propagation).

Contributions of our work are summarized as follows:

- We model credit card behaviors as a temporal transaction graph and formulate a credit card fraud detection problem as a semi-supervised node classification task.
- We present a novel attribute-driven temporal graph neural network for credit card fraud detection. Specifically, we propose a gated temporal attention network to extract temporal and attribute information. And we pass attributes and risk information on the temporal transaction graph to exploit both labeled and unlabeled data.

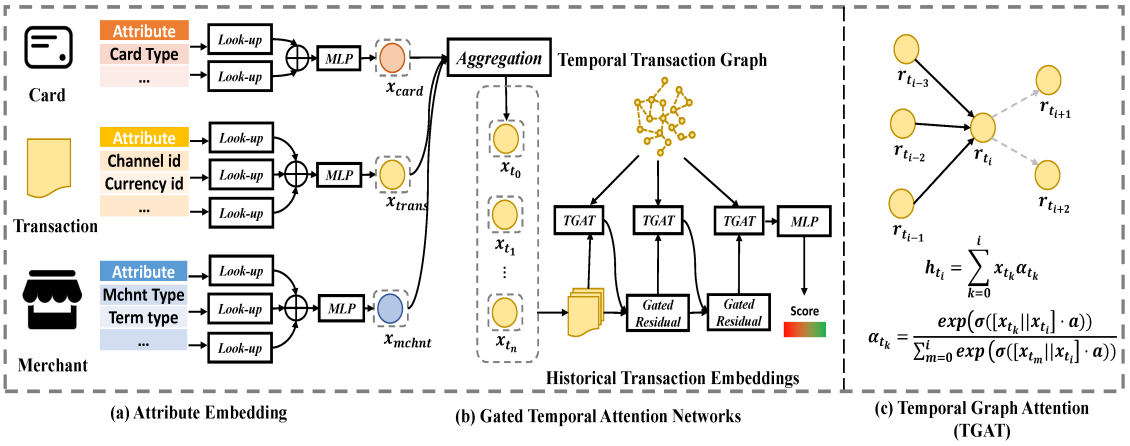


FIGURE 3.2: The illustration of the proposed model architecture and temporal graph attention mechanism.

- Extensive experiments on three datasets show the superiority of our proposed GTAN on fraud detection. Semi-supervised experiment results show that, when leveraging rich information from the unlabeled data and a bit of labeled data, our proposed method detects more fraud transactions than baselines.

3.3 Related Works

3.3.1 Credit Card Fraud Detection

Several machine learning techniques have been proposed in the literature to address the credit card fraud detection problem. For instance, in [59], Bayesian Belief Networks (BBN) and Artificial Neural Networks (ANN) were applied on a real dataset obtained from Europay International. In [60] decision trees and support vector machines (SVMs) are applied on a real-world national bank dataset. The authors in [23] showed that using convolution to extract patterns can achieve higher accuracy than non-convolution neural networks. Recently, graph-based fraud detection techniques were proposed. For instance, CARE-GNN [56] was proposed to tackle fraud detection on relational graphs. PC-GNN [58] was proposed for imbalanced supervised learning on graphs. [13] also proposed a generative adversarial network to improve the classification performance. In [24, 61], authors proposed joint feature learning based on spatial and temporal patterns. However, they modeled the fraud patterns by using only one transaction/cardholder, thereby not being

able to exploit the unlabeled data in real-life credit card transactions. The approach we present in this chapter is radically different, as we employ a semi-supervised architecture, where the fraud patterns on both unlabeled and labeled data are jointly learned within an attribute-driven graph neural network framework.

3.3.2 Graph-Based Semi-supervised Learning

Many recent works have shown the benefit of using unlabeled node attributes in graph neural networks for a wide range of prediction tasks [62, 63], such as text classification [64], molecule property prediction [65] or language understanding [66]. For instance, graph convolutional networks (GCN) were employed on partially labeled citation networks for node property prediction [67]. GraphSAGE [68] was proposed to generate low-dimensional embeddings for previously unseen data. Graph attentive network model and random walks [51] were deployed on social graphs to link the unlabeled and labeled data and pass messages among them. However, they still face at least one of the following two limitations: (1) cannot scale up to real-world graphs over millions of nodes (e.g., vanilla graph attention networks [69] have a space complexity $O(N^2)$, where N denotes the number of nodes, which is unaffordable for tasks with millions of nodes); (2) cannot propagate and learn the categorical attribute embeddings, especially for risk embeddings. Differently, our approach addresses the fraud detection problem via a message-passing model using categorical attributes, including risk information from node neighbors. Our work exploits attribute-driven model and semi-supervised graph neural networks to find more fraud patterns, which significantly improve the accuracy of credit card fraud detection.

3.4 Proposed Method

In this section, we first introduce the framework of our proposed GTAN. Then, we present the process of feature engineering, the gated temporal attention networks, risk embedding and propagation, and the fraud detection classifier. Lastly, we introduce the optimization strategy.

3.4.1 Model Architecture

The general model architecture of our proposed method is illustrated in Figure 3.2. Raw attributes of transaction records are first learned by the attribute embedding look-up and feature learning layer, which includes feature aggregation with a multi-layer perception (MLP). In our implementation, the attributes of the card include the card type, cardholder type, card limit, remaining limit, etc. The transaction attributes include the channel ID, currency ID, transaction amount, etc. The merchant attributes contain merchant type, terminal type, merchant location, sector, charge ratio, etc. Then, we devise a gated temporal attention network to aggregate and learn the importance of historical transaction embeddings. Afterward, we leverage a two-layer MLP to learn the fraud probability from these representations. The whole model can be optimized in an end-to-end mechanism jointly with the existing stochastic gradient descent algorithm.

3.4.2 Attribute Embedding and Feature Learning

Given transaction records $\mathbf{r} = (r_1, r_2, \cdots, r_N)$, each record r_i contains card attributes f_c^i, transaction attributes f_r^i, and merchant attributes f_m^i as $r_i = f_c^i, f_r^i, f_m^i$. In preprocessing, different from [61], we do not filter out any cards or merchants with few authorized transaction records. As the number of cards and merchants which have never been checked manually is much larger than checked, we adopt full transaction records of users to maintain all potential frauds. Afterward, we construct the numerical attribute representation of each record into tensor format $\mathbf{X}_{num} \in \mathbb{R}^{N \times d}$, where N denotes the number of transactions, and d denotes the dimensions of features. Besides, we extract the card, transaction, and merchant category attributes $\mathbf{X}_{cat} \in \mathbb{R}^{N \times d}$ separately through attribute embedding layers, which can be formulated as follows:

$$\begin{aligned} e_{attr} =& \text{onehot}(f_{attr}) \odot \mathbf{E}_{attr}, \\ x_{cat,i} =& \text{MLP}_i(\sum_{\forall j \in table_i} e_j), i \in \{card, trans, mchnt\}, \end{aligned} \tag{3.1}$$

where $j \in table_i$ denotes the column j in our input table data i, $e_{attr} \in \mathbb{R}^{1 \times d}$ denotes the embedding of attribute $attr$, onehot$(\cdot)$ denotes the one-hot encoding, f_{attr} denotes the single attribute of one transaction, and $\mathbf{E}_{attr} \in \mathbb{R}^{m \times d}$ denotes the embedding matrix of attribute $attr$, where m denotes the maximum number of attribute $attr$. After obtaining the embedding vector of each

attribute in the card, transaction, and merchant tables, we aggregate these embeddings to obtain each transaction's categorical embedding through add-pooling with $x_{cat}^{(u)} = \sum_i x_{cat,i}^{(u)}, i \in \{card, trans, mchnt\}$, where $x_{cat}^{(u)} \in \mathbb{R}^{1\times d}$ denotes the category embedding vector of the u-th transaction record. To address the heterogeneity of categorical attributes, our proposed feature learning layer can model all categorical attributes and project them to a unified spatial dimension, which benefits our attribute-driven graph learning model.

3.4.3 Gated Temporal Attention Networks

To learn the temporal fraud patterns, we generate the temporal transaction graph [70, 71] and aggregate messages on this graph to update the embedding of each transaction. Particularly, the directed temporal edges are generated with the previous transactions as the source and the current ones as the target, as illustrated in Figure 3.2(c). Then we aggregate messages through Temporal Graph Attention (TGAT) mechanism [72]. The number of generated temporal edges per node is a hyper-parameter, which will be studied in the experiment section.

Temporal Graph Attention. After the feature engineering and attribute embedding, we leverage a series of transaction embeddings $\mathbf{X} = \{x_{t_0}, x_{t_1}, ...x_{t_n}\}$ to learn the temporal embedding of each transaction record. First, we combine categorical attributes and numerical attributes as the input of GTAN network with $x_{t_i} = x_{num}^{(t_i)} + x_{cat}^{(t_i)}$. At the first TGAT layer, we set $\mathbf{H}_0 = \mathbf{X}$ as the input embedding matrix. Afterward, we leverage multi-head attention to separately calculate the importance of each neighbor and update embeddings, which can be formulated as follows:

$$\mathbf{H} = \text{Concat}(\text{Head}_1, ..., \text{Head}_{h_{att}})\mathbf{W}_o, \tag{3.2}$$

where h_{att} denotes the number of heads, $\mathbf{W}_o \in \mathbb{R}^{d\times d}$ denotes learnable parameters, $\mathbf{H}$ denotes the aggregated embeddings with $\mathbf{H} = \{h_{t_0}, h_{t_1}, ..., h_{t_n}\}$, and each attention head is formulated as follows:

$$\begin{aligned} \text{Head} &= \sum_{x_i\in\mathcal{X}} \sigma\Big(\sum_{x_t\in\mathcal{N}(x_i)} \alpha_{x_t,x_i} x_t\Big), \\ \alpha_{x_t,x_i} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[x_t||x_i]))}{\sum_{x_j\in\mathcal{N}(x_t)} \exp(\text{LeakyReLU}(\mathbf{a}^T[x_t||x_j]))}, \end{aligned} \tag{3.3}$$

where $\mathcal{N}(x_i)$ denotes the temporal neighbors of the i-th transaction, α_{x_t,x_i} denotes the importance of temporal edge (x_t, x_i) in each attention head, and $\mathbf{a} \in \mathbb{R}^{2d}$ denotes the weight vector of each head. In practice, to avoid extra space consumption in extreme cases (such as high-frequency transactions in a short period), we use a neighbor sampling and truncation strategy to control the number of neighbor nodes $|\mathcal{N}(x_t)|$ (i.e., the number of associated temporal edges per node) through which the temporal graph attention layer propagates messages. Besides, to avoid borrowing future information, the neighbor transactions sampled for each transaction must be the past transactions from the same cardholder so that we can model the temporal fraud pattern through message passing on the temporal transaction graph.

Attribute-driven Gated Residual. To further improve the effectiveness and interpretability of our method, after obtaining aggregated embeddings, we leverage the embeddings and raw attributes to infer the importances of the aggregated embeddings and raw attributes after each layer of TGAT, which can be formulated as follows:

$$\begin{aligned} \text{gate}_{t_i} =& \sigma([x_{cat,t_i}||x_{num,t_i}||h_{t_i}]\beta_{t_i}), \\ z_{t_i} =& \text{gate}_{t_i} \cdot h_{t_i} + (1 - \text{gate}_{t_i}) \cdot x_{t_i}, \end{aligned} \tag{3.4}$$

where $\text{gate}_{t_i} \in [0, 1]$ denotes the gate variable of the t_i-th transaction, σ denotes the sigmoid activation function, $\beta_{t_i} \in \mathbb{R}^{3d\times 1}$ denotes the gate vector, and z_{t_i} denotes the output vector of each TGAT layer, which is fed into the next layer as input. According to our framework, if we stack a new TGAT layer with the attribute-driven gated residual mechanism, we use the output of the k-th gating mechanism as the input of the $k + 1$-th TGAT.

3.4.4 Risk Embedding and Propagation

Inspired by unifying label propagation with feature propagation [73], we propose to take the manually annotated label as one of the categorical attributes of the transaction, and get the embedding of this categorical attribute, which we call *risk embedding*. Specifically, we take the manually annotated label as the risk feature of each transaction, where the category of unlabeled data is 'unlabeled', and the category of the rest of the data is 'fraud' or 'legitimate'. Then, we add this feature to the transaction data as one of our input categorical attributes. Due to concerns about

label leakage, this attribute has not been used in previous fraud detection solutions. We will discuss the techniques for avoiding label leakage later. Then, we propose to embed the partially observed risk attributes (i.e., labels) into the same space as the other node features, which consist of the risk embedding vectors for labeled nodes and zero embedding vectors for the unlabeled ones. Then, we add the node features and risk embeddings together as input node features with $x_{t_i} = x_{num}^{(t_i)} + x_{cat}^{(t_i)} + \tilde{y}^{(t_1)}\mathbf{W}_r$, where $\mathbf{W}_r$ denotes the learnable parameters of risk embedding. [73] have proved that by mapping partially-labeled $\hat{\mathbf{Y}}$ and node features $\mathbf{X}$ into the same space and adding them up, we can use one graph neural network to achieve both attribute propagation and label propagation. Therefore, our fraud detection model can jointly model the temporal fraud patterns and risk propagation by adding the transaction label as one of the transaction categorical attributes.

3.4.5 Fraud Risk Prediction

After obtaining the aggregated embeddings of transactions, we leverage a two-layer MLP to predict the fraud risk, which is formulated as follows:

$$\hat{\mathbf{y}} = \sigma(\text{PReLU}(\mathbf{H}\mathbf{W}_0 + \mathbf{b}_0)\mathbf{W}_1 + \mathbf{b}_1), \tag{3.5}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{N\times 1}$ denotes the risk prediction results of all transactions, and $\mathbf{W}$ and $\mathbf{b}$ denote the learnable parameters of MLP. Afterward, we calculate the objective function $\mathcal{L}$ via binary cross-entropy, which is formulated as follows:

$$\begin{aligned}\mathcal{L} = -\frac{1}{N}\sum_{i=0}^{N}[&\mathbf{y}_i \cdot \log p(\hat{\mathbf{y}}_i|\mathbf{X}, \mathbf{A}) + \\ &(1-\mathbf{y}_i)\cdot\log(1-p(\hat{\mathbf{y}}_i|\mathbf{X}, \mathbf{A}))],\end{aligned} \tag{3.6}$$

where $\mathbf{y}$ denotes the ground-truth label of transactions. The proposed GTAN can be optimized through the standard SGD-based algorithms.

3.4.6 Masking to Avoid Label Leakage

Previous works only took risk information as optimization objectives to supervise their fraud detection model training. Unlike previous credit card fraud detection solutions, we semi-supervise our model by propagating transaction attributes and risk embeddings among labeled and unlabeled transactions. Using an unmasked objective for our fraud detection model will result in label leakage in the training process. In this case, our model will directly take observed labels and neglect the complicated hidden fraud patterns, which cannot be generalized in predicting future fraud transactions. Therefore, we propose to learn from the risk information of each transaction's neighbor transactions instead of learning from the label of itself. Specifically, a masked fraud detection training strategy is leveraged. During each training step, we randomly sample a batch of nodes, namely center nodes, along with the neighbor nodes corresponding to each center node. Then, we convert the partially observed labels $\hat{\mathbf{Y}}$ into $\tilde{\mathbf{Y}}$ by masking all the center nodes' risk embeddings to zero embeddings and keeping the others unchanged. Then, our objective function is to predict $\hat{\mathbf{Y}}$ with given $\mathbf{X}$, $\tilde{\mathbf{Y}}$ and $\mathbf{A}$:

$$\begin{aligned}\mathcal{L} = -\frac{1}{|V|}\sum_{i=0}^{|V|}[\mathbf{y}_i \cdot \log p(\hat{\mathbf{y}}_i|\mathbf{X}, \tilde{\mathbf{Y}}, \mathbf{A}) + \\ (1-\mathbf{y}_i) \cdot \log(1 - p(\hat{\mathbf{y}}_i|\mathbf{X}, \tilde{\mathbf{Y}}, \mathbf{A}))],\end{aligned} \tag{3.7}$$

where $|V|$ represents the number of center nodes with masked labels. In this way, we can train our model without the self-loop leakage of risk information; and during inference, we employ all observed labels $\hat{\mathbf{Y}}$ as input categorical attributes to predict the risk of the transactions out of the training set. The optimization objective of our model can be intuitively summarized as modeling the fraud patterns by the attribute information of neighboring transaction nodes and the attribute information.

3.5 Experiments

In this section, we first describe the datasets used in the experiments, then compare our fraud detection performance with other state-of-the-art baselines on two supervised graph-based fraud

Dataset	YelpChi	Amazon	FFSD
#Node	45,954	11,948	1,820,840
#Edge	7,739,912	8,808,728	31,619,440
#Fraud	6,677	821	33,858
#Legitimate	39,277	11,127	141,861
#Unlabeled	-	-	1,645,121

TABLE 3.1: Statistics of the three fraud detection datasets.

detection datasets and one semi-supervised dataset. Then, we perform ablation studies by evaluating two variants of the proposed GTAN, which demonstrates the effectiveness of our proposed method and attribute-driven mechanism.

3.5.1 Experiment Settings

3.5.1.1 Datasets

To the best of our knowledge, we did not find any public semi-supervised credit card fraud detection dataset. Therefore, we collect the partially labeled records from our collaborated partners, namely Finacial Fraud Semi-supervised Dataset (**FFSD**). The ground truth labels are obtained on cases reported by consumers and confirmed by financial domain experts. If a transaction is reported by a cardholder or identified by financial experts as fraudulent, we label it as 1; otherwise, it is labeled as 0. Besides, we also experimented on two public supervised fraud detection datasets. The **YelpChi** graph dataset [74] contains a selection of hotel and restaurant reviews on Yelp. Nodes in the graph of the YelpChi dataset are reviews with 32-dimensional features, and the edges are the relationships among reviews. The **Amazon** graph dataset [75] includes product reviews of musical instruments. The nodes in the graph are users with 25-dimensional features, and the edges are the relationships among reviews. Some basic statistics of the three datasets are shown in Table 3.1.

3.5.1.2 Compared Methods.

The following methods are compared to highlight the effectiveness of the proposed GTAN.

Dataset	YelpChi			Amazon			FFSD		
	AUC	F1	AP	AUC	F1	AP	AUC	F1	AP
GEM	0.5270	0.1060	0.1807	0.5261	0.0941	0.1159	0.5383	0.1490	0.1889
Player2Vec	0.7003	0.4121	0.2473	0.6185	0.2451	0.1291	0.5278	0.2147	0.2041
FdGars	0.7332	0.4420	0.2709	0.6556	0.2713	0.1438	0.6965	0.4089	0.2449
Semi-GNN	0.5161	0.1023	0.1811	0.7063	0.5492	0.2254	0.5473	0.4485	0.2758
GraphSAGE	0.5364	0.4508	0.1712	0.7502	0.5795	0.2624	0.6527	0.5370	0.3844
GraphConsis	0.7060	0.6041	0.3331	0.8782	0.7819	0.7336	0.6579	0.5466	0.3876
CARE-GNN	0.7934	0.6493	0.4268	0.9115	0.8531	0.8219	0.6623	0.5771	0.4060
PC-GNN	0.8174	0.6682	0.4810	0.9581	0.9153	0.8549	0.6795	0.6077	0.4487
GTAN	**0.9241***	**0.7988***	**0.7513***	**0.9630***	**0.9213***	**0.8838***	**0.7616***	**0.6764***	**0.5767***

TABLE 3.2: Fraud detection performance on three datasets, compared with popular benchmark methods. We report the results of the area under the roc curve (AUC), macro average of F1 score (F1-macro), and averaged precision (AP).

- *GEM.* Heterogeneous GNN-based model proposed in [76]. We set the learning rate to 0.1 and the number of hops of neighbors to 5.
- *FdGars.* Fraudster detection via the graph convolutional networks proposed in [77]. We set the learning rate to 0.01 and the hidden dimension to 256.
- *Player2Vec.* Attributed heterogeneous information network proposed in [78]. We set the same parameters as the FdGars model.
- *Semi-GNN.* A semi-supervised graph attentive network for financial fraud detection proposed in [51]. We set the learning rate to 0.001.
- *GraphSAGE.* The inductive graph learning model proposed in [68]. We set the embedding dimension to 128.
- *GraphConsis.* The GNN-based model tackling the inconsistency problem, proposed in [57]. We used the default parameters suggested by the original paper.
- *CARE-GNN.* The GNN-based model tackling fraud detection on a relational graph [56]. We used default parameters from the original paper.
- *PC-GNN.* A GNN-based model remedying the class imbalance problem, proposed in [58]. We used the default parameters from the original paper.

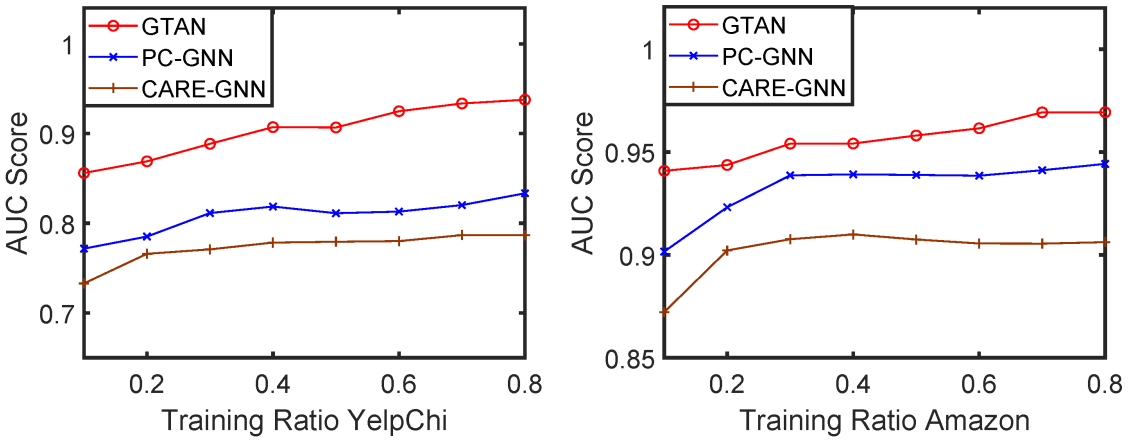


FIGURE 3.3: The result of semi-supervised experiments with different ratios of labeled training data.

- **GTAN.** The proposed gated temporal attention network model.[1] We also evaluate two variants of our model, GTAN-A and GTAN-R, in which the temporal graph attention component and risk embedding component are not considered, respectively. We set the batch size to 256, the learning rate to 0.0003, the input dropout ratio to 0.2, the number of heads to 4, the hidden dimension d to 256, and train the model with the Adam optimizer for 100 epochs with early stopping.

3.5.1.3 Evaluation Metrics

We evaluate the experimental results on credit card fraud detection and opinion fraud datasets by the area under the ROC curve (AUC), macro average of F1 score (F1-macro), and averaged precision (AP), which are calculated as follows:

We count the number of True Positive N_{TP} (i.e. correct identification of positive labels), False Positive N_{FP} (i.e., incorrect identification of positive labels), and False Negatives N_{FN} (i.e., incorrect identification of negative labels). Then, F1 score and AP as formulated with $F1_{macro} = \frac{1}{l}\sum_{i=1}^{l}\frac{2\times P_i \times R_i}{P_i+R_i}$ and $AP = \sum_{i=1}^{l}(R_i - R_{i-1})P_i$, where $P_i = N_{TP}/(N_{TP}+N_{FP})$ and $R_i = N_{TP}/(N_{TP}+N_{FN})$. We also report the AUC[2] in our experiments.

[1]The sources of our proposed method GTAN will be available at `https://github.com/finint/antifraud`.

[2]`https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html`

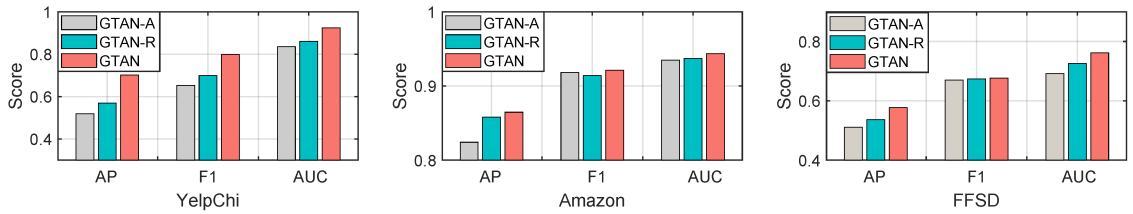


FIGURE 3.4: The ablation study results on three datasets. Gray bars represent the GTAN-A variant, blue bars represent the GTAN-R variant, and red bars represent the GTAN model.

3.5.2 Fraud Detection Performance

In YelpChi and Amazon datasets, we set the ratio of training to testing as 2:3. In the FFSD dataset, transactions of the first 7 months are used as training data, and then we detect fraud transactions in the following 3 months (August, September, and October of 2021). We repeat the experiments ten times for each method and show the average performance in Table 3.2. * denotes that the improvements are statistically significant for $p < 0.01$ according to the paired t-test.

The first five rows of Table 3.2 report the results of some classic graph-based methods, including GEM, Player2Vec, FdGars, Semi-GNN, and GraphSASE. It is clear that the result of GEM is not satisfactory, showing the limitation of a shallow model in addressing the complex fraud patterns. Player2Vec and FdGars improve the performance, partially due to the enlarged model capacity. Semi-GNN and GraphSAGE are close to each other and better than the first three methods. The result demonstrates the effectiveness of deep graph learning-based models in detecting fraud transactions. PC-GNN achieves more competitive results by employing the transaction graphs in the learning process, which is considerably better than the above baselines. The last row of Table 3.2 shows that our methods GTAN significantly outperforms all baselines with at least 10%, 0.5%, and 6% AUC improvements in three datasets, respectively. Besides, our proposed GTAN also outperforms other baselines with at least 27%, 2.9%, and 12.8% AP improvements in three datasets, respectively, which strongly proves the effectiveness of employing a semi-supervised temporal graph attention network for fraud detection.

3.5.3 Semi-supervised Experiment

To compare the capability of semi-supervised learning, we further set the ratio of the training set to the whole dataset to different values. For brevity of the diagrams, we select the two most competitive baselines (i.e., PC-GNN and CARE-GNN) for the following semi-supervised experiments. We vary the percentages of nodes used for training from 10% to 80% with an incremental of 10% for eight sets of experiments, with the remaining nodes as the test set in each set of experiments. We perform experiments on the YelpChi and Amazon datasets since they are fully annotated, which allows us to vary the ratio of labeled data in a wide range. The experimental results are shown in Figure 3.3.

On the YelpChi dataset, we can observe that GTAN always has the best performance under different training ratios. At the same time, in scenarios with little labeled data (10% training ratio), GTAN still performs well. As the number of labeled data increases, there is a steady improvement in the performance of GTAN. On the Amazon dataset, we can also observe that GTAN always has the best performance under different training ratios. Compared with the YelpChi dataset, the GTAN model on the Amazon dataset is less sensitive when changing the training ratio with no more than 2% variations in the AUC. This fully demonstrates that GTAN can achieve good performance even when there is a small portion of labeled data available (i.e., as low as 10%). Therefore, we conclude that the GTAN model is robust to training ratio changes and consistently outperforms PC-GNN and CARE-GNN, which shows the superiority of GTAN model in semi-supervised learning.

3.5.4 Ablation Study

The proposed model contains some key components, and we verify their effectiveness by ablating each component, respectively. Specifically, we evaluate two variants, namely GTAN-A and GTAN-R. In the GTAN-A model, we remove the TGAT component, and the central node aggregates messages directly collected from neighboring nodes. In this case, we obtain the transaction embeddings from all the neighbor embeddings with the same weight instead of adaptively adjusting the weights of neighbor nodes. In the GTAN-R model, we remove the risk embedding component and only use the original node attributes $\mathbf{X}$.

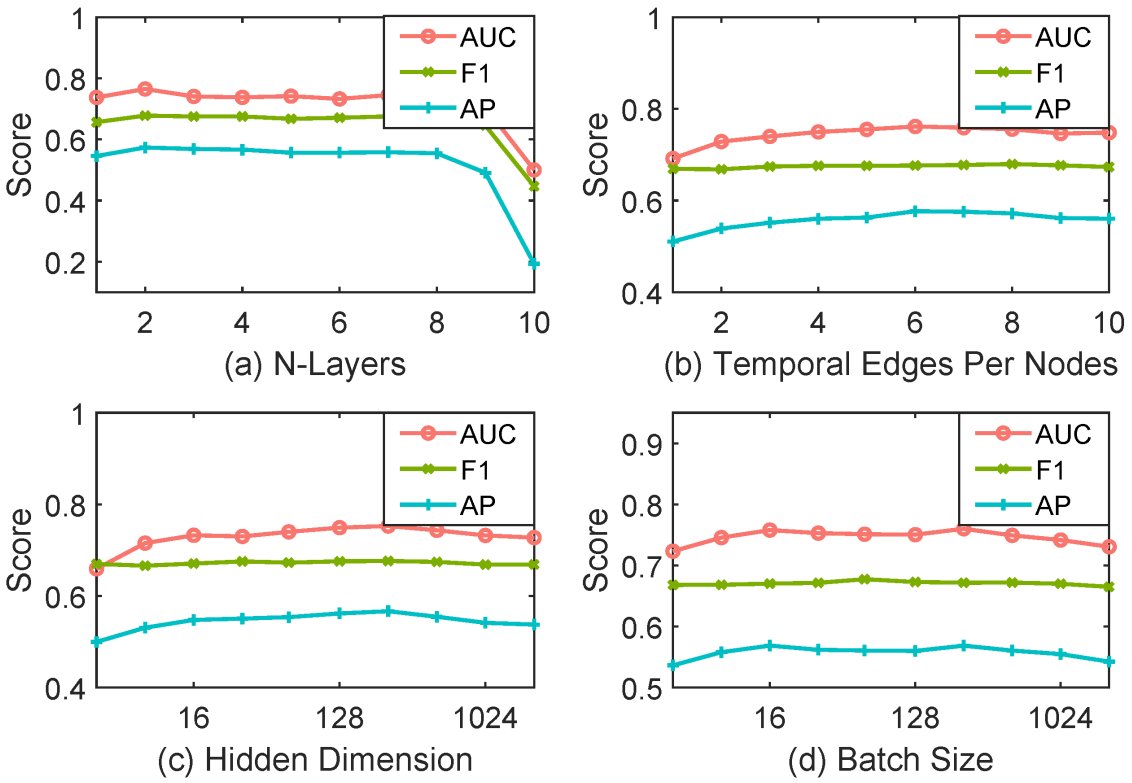


FIGURE 3.5: Parameter sensitivity analysis with respect to (a) the number of GNN layers; (b) the number of temporal edges per node; (c) hidden dimension; and (d) the batch size.

The grey bars in Figure 3.4 show the removal of the attention mechanism has the greatest impact on the accuracy metrics of the model. This proves that the reweighting of temporal transaction neighbors from the temporal graph attention mechanism is effective in our graph-based method. Besides, the green bars in Figure 3.4 gets the second-highest scores compared with GTAN, which proves that the risk embedding is effective in modeling credit card fraud patterns from transaction risk propagation. In summary, removing either component deteriorates the performance of GTAN, which proves that the temporal graph attention mechanism and risk embedding are both effective in graph-based fraud detection.

3.5.5 Parameter Sensitivity

In this section, we study the model parameter sensitivity by varying the depth of temporal graph attention layers, the number of temporal edges per node, the hidden dimension, and the batch size. The experimental results in the fraud detection dataset are reported in Figure 3.5.

We vary the depth of temporal graph attention layers from 1 to 10. As shown in Figure 3.5(a), our model's performance remains stable with up to 8 GNN layers. With deeper layers, our model tends to aggregate the temporal information from larger neighborhoods. Our model performs the best with two GNN layers when the AUC and AP reach the peak; therefore, we set the default depth to 2. The performance is degraded if we keep on increasing the depth of TGAT layers. The reason might be that deeper GNNs result in over-smoothing [79] in transaction embeddings. Figure 3.5(b) shows that when we increase the number of temporal edges per node from 1 to 10, our proposed model could consider neighbors in a wider range. Besides, our model requires at least 2 neighbors to learn graph-based transaction embeddings, and it reaches peak performance when the number of edges is 6. Beyond that, an excessive increase in the number of edges in the graph is not beneficial to the model's accuracy. Figure 3.5(c) shows that when we increase the hidden dimensions from 4 to 2048, our model maintains stable model performance and reaches a relative performance peak at 256. Figure 3.5(d) shows that our model performs best when the batch size is set as 64 or 256. Considering the training efficiency of the model, we set the batch size as 256. Generally, for values ranging from 16 to 512, the model is not sensitive to the hidden dimension and batch size, with less than 3% variations in the AUC.

3.6 Conclusion

In this chapter, we studied an important real-world problem of credit card fraud detection. Considering that the labeling of fraud transactions is time-consuming and cost-expensive, we proposed an effective semi-supervised credit card fraud detection method by modeling data with temporal transaction graphs and developing attribute-driven gated temporal attention networks. Considering the ubiquitous categorical attributes and human-annotated labels, we proposed an attribute representation and risk propagation mechanism to model the fraud patterns accurately. The comprehensive experiments demonstrated the superiority of our proposed methods in three fraud detection datasets compared with other baselines. Semi-supervised experiments demonstrate the excellent fraud detection performance of our model with only a tiny proportion of manually annotated data. Our approach has been deployed in a transaction fraud analysis system. In the future, we will explore to study the temporal fraud patterns and risk-propagation fraud patterns in an effective and efficient way.

Chapter 4

Heterogeneous Graph Neural Network for Fraud Detection

4.1 Chapter Overview

In this chapter, we proposed a heterogeneous graph-based solution for fraud detection. This chapter has been published on APWeb 2024. Section 4.2 gives the background and current progress of heterogeneous graph-based fraud detection. Section 4.3 introduces the preliminaries of graph-based fraud detection. Section 4.4 shows our proposed Relation-aware Heterogeneous Graph Neural Network (RHGNN). Section 4.5 reports the experimental results. Section 4.6 concludes this chapter.

4.2 Background

Detecting fraudulent behavior is used in finance and social media to reduce economic loss. For example, credit card fraud is a major issue in the financial sector, causing significant global economic losses each year [80, 81]. As digital transactions grow, so does the complexity and volume of fraud, making effective detection systems necessary to protect consumer trust and financial integrity [82]. The need for robust fraud detection mechanisms is driven by the evolving nature of

fraud techniques, which exploit the growth of e-commerce and digital banking [83]. Detecting fraud has thus become crucial in data mining.

Fraud detection systems are categorized based on their methodology. Early systems used rule-based methods, where typical fraudulent behaviors were hardcoded [84]. As fraud tactics evolved, machine learning techniques were incorporated, allowing automatic detection based on historical data [85]. Recently, deep learning methods have shown significant improvements in detecting complex fraud patterns [86]. These systems analyze large amounts of data to identify hidden patterns and anomalies indicating fraud. The use of Graph Neural Networks (GNNs) has introduced a new paradigm in fraud detection, leveraging relationships and connectivity patterns among nodes to enhance detection accuracy [22]. This shift towards relational data analysis emphasizes understanding not just individual nodes but also their connections. Traditional rule-based fraud detection systems define explicit conditions that trigger alerts when met. These systems, derived from known fraud patterns and expert knowledge, are straightforward but rigid, lacking the flexibility to adapt to new fraud tactics without manual updates [87]. Machine learning-based methods offer a dynamic approach by learning from historical data, automatically adapting and improving detection algorithms over time [54]. Techniques such as decision trees, neural networks, and support vector machines have been widely applied, enabling the detection of complex fraudulent behaviors not explicitly defined in rule sets [81, 86]. This adaptability makes machine learning methods effective against evolving fraud strategies, continuously refining their models with new data.

Recent advancements in fraud detection have integrated graph-based methods, utilizing relationships between nodes to detect fraud more effectively. These methods consider not only individual node features but also their relationships within a network, capturing complex patterns typically missed by traditional methods. GNNs, in particular, have shown a remarkable ability to detect fraudulent activities by leveraging connected data in real-world networks [42, 88]. GNNs apply deep learning principles to graph data, allowing information propagation across nodes and enabling the system to identify suspicious link patterns indicative of fraud [89, 90]. This relational approach is advantageous for detecting organized fraud rings, where fraudulent transactions are interconnected. Innovations such as attention mechanisms and node embeddings have further enhanced the efficacy of these methods, providing nuanced understandings of node relationships within graphs [91].

Despite their success, existing graph-based fraud detection methods face several limitations. Many suffer from low efficiency, particularly when scaling to large datasets common in financial systems [89]. The computational complexity of processing large graphs, especially with multiple iterations, can lead to prohibitive processing times. Real-world graphs are usually heterogeneous, with various types of nodes and edges representing different entities [90]. For example, according to Figure 4.1, there are three types of nodes in financial fraud detection tasks. Most GNNs assume a homogeneous graph structure, oversimplifying real-life networks. This can result in a loss of critical information and less effective fraud detection.

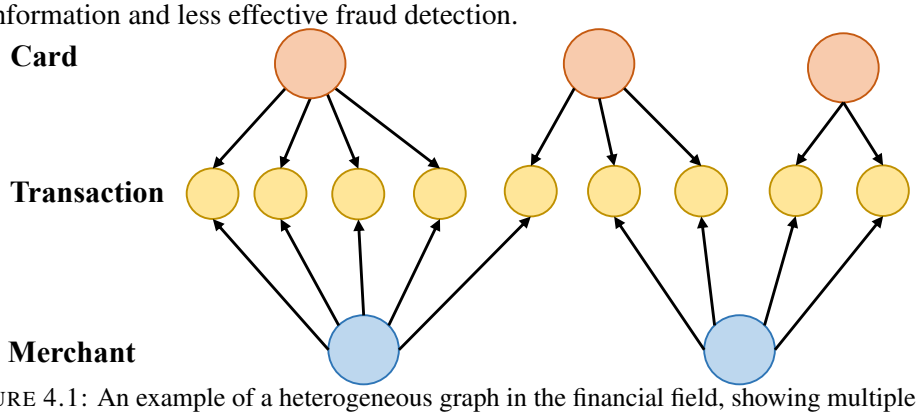


FIGURE 4.1: An example of a heterogeneous graph in the financial field, showing multiple types of nodes/edges in real-world graphs.

In response to these limitations, this chapter introduces a novel Relation-Aware Heterogeneous Graph Neural Network (RHGNN) model to address the shortcomings of existing graph-based fraud detection methods. Our model handles the heterogeneity of real-world graphs by recognizing and processing different types of nodes and edges as distinct but interconnected components. By leveraging computation graph preprocessing and a hybrid propagation scheme, our approach enhances the efficiency of information processing across the graph, reducing computational overhead and facilitating scalability. The integration of stochastic projection reduction manages feature dimensionality, maintaining efficiency even when scaling to large datasets. This dual focus on addressing heterogeneity and improving computational efficiency sets our method apart, offering a more robust and scalable solution for detecting fraudulent activities in expansive real-world networks. Finally, extensive experiments on benchmark datasets, including Amazon, Yelpchi, and T-Finance, demonstrate our method's superior performance over existing graph-based techniques in accuracy and efficiency in detecting fraudulent transactions.

The contributions of this chapter are manifold and are itemized as follows:

- I) We delineated the complexities of fraud patterns in real-world networks, highlighting the need for models that can handle the heterogeneity of graph data effectively.
- II) We introduced a Relation-Aware Heterogeneous Graph Neural Network model. This model incorporates the computation graph pre-process and hybrid propagation scheme to enhance feature propagation efficiency and includes stochastic projection reduction to manage feature dimensionality without performance loss.
- III) Our method is rigorously tested on several benchmark datasets, including Amazon, Yelpchi, and T-Finance, where it demonstrates superior performance in fraud detection accuracy compared to existing state-of-the-art fraud detection techniques.

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$	A heterogeneous graph
$\mathcal{V}$	Set of nodes
$\mathcal{E}$	Set of edges
$\mathcal{A}$	Set of node types
$\mathcal{R}$	Set of edge types
$v \in \mathcal{V}$	A node in the graph
$e \in \mathcal{E}$	An edge in the graph
$A \in \mathcal{A}$	A node type
$R \in \mathcal{R}$	An edge type
$\mathbb{A}_{R_i}$	Adjacency matrix for edge type R_i
$P = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$	A meta-path
y_t	Label of transaction node t

TABLE 4.1: Notation Summary.

4.3 Preliminaries

In this section, we define the notations and terminologies used throughout this chapter. We focus on the problem of fraud detection within the context of the heterogeneous graphs. We also discuss the related works in this section.

4.3.1 Problem Definition

Unlike traditional graph-based methods that treat input data as homogeneous graphs to predict potential fraudulent activities, our model represents the real-world graph as a heterogeneous graph,

automatically generated from real-world transactions or databases. In these heterogeneous graphs, nodes denote individual entities (e.g., users/transactions/merchants) and edges represent their relationships, encoded in the form of adjacency matrices $\mathbb{A}_{R_i}$ for each edge type R_i. Therefore, in this part, we formally defined the fraud detection problem in the context of heterogeneous graphs and the meta-path. All notations are summarized in the Table 4.1.

Heterogeneous Graph: A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $\mathcal{A}$ is the set of node types, and $\mathcal{R}$ is the set of edge types. Each node $v \in \mathcal{V}$ belongs to one particular node type $A \in \mathcal{A}$, and each edge $e \in \mathcal{E}$ belongs to one particular edge type $R \in \mathcal{R}$. Each edge type R_i is associated with an adjacency matrix $\mathbb{A}_{R_i}$, which defines the connectivity between node types. For example, the financial heterogeneous graph has three types of nodes, i.e., card, merchant, and transaction.

Meta-path: A meta-path P is a sequence of node types and edge types in the form $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{l-1}} A_l$, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$. A meta-path captures a specific type of relationship between nodes in a heterogeneous graph. For example, in the financial heterogeneous graph, a meta-path can be composed by $card \xrightarrow{R_1} transaction \xrightarrow{R_2} merchant$.

Graph-based Fraud Detection: Given a heterogeneous graph $\mathcal{G}$ representing real-world relationships, where nodes represent entities such as users, merchants, and transactions, and edges represent relationships such as transaction occurrences and user-transaction associations, the goal is to predict whether a transaction node $t \in \mathcal{V}$ is fraudulent. This can be formulated as a binary classification problem where the model assigns a label $y_t \in \{0, 1\}$ to each transaction node t, with $y_t = 1$ indicating a fraudulent transaction and $y_t = 0$ indicating a legitimate one. The goal is to develop an end-to-end model that can accurately and efficiently identify fraudulent nodes by leveraging the heterogeneous nature of the graph and the complex relationships captured by meta-paths.

4.3.2 Related Works

4.3.2.1 Non-Graph Based Fraud Detection

Non-graph based fraud detection has been extensively studied using various machine learning and statistical techniques. Traditional methods include supervised learning approaches such as logistic regression [92], decision trees [93], and support vector machines [94], which rely on hand-crafted features extracted from input data [95]. Recently, deep learning techniques have been employed to automatically learn complex features from raw fraud detection data, improving the accuracy of fraud detection systems [86]. However, these deep-learning approaches typically focus on individual entities and do not fully exploit the relationships between entities (such as users and transactions). Our work addresses this limitation by utilizing advanced heterogeneous graph neural networks (HGNNs) to capture "cross-user" information, providing a more comprehensive view of fraudulent activities.

4.3.2.2 Graph-Based Fraud Detection

Graph-based methods have gained popularity in fraud detection due to their ability to capture relationships between entities. These methods represent input data as graphs, where nodes represent entities (e.g., users, transactions) and edges represent relationships (e.g., transaction links). Techniques such as graph convolutional networks (GCNs) and graph attention networks (GATs) have been applied to fraud detection, showing improved performance over traditional methods [32]. CARE-GNN [88] was proposed to select better neighbors for graph-based fraud detection tasks. PC-GNN [42] was designed to solve the node label imbalance problem in fraud detection. BW-GNN [96] was proposed to address the "right-shift" issue in the graph anomaly detection. GTAN [22] was designed for semi-supervised fraud detection in graphs. However, these approaches often assume a homogeneous graph structure and may not fully exploit the heterogeneity inherent in real-world financial transaction networks. Our work addresses this limitation by employing scalable heterogeneous graph neural networks (HGNNs) that can effectively handle the diverse types of nodes and edges present in financial graphs. By leveraging the heterogeneity of these graphs, our model captures more complex and realistic interactions between different types of entities. Importantly, this enhanced modeling capability is achieved without significantly

increasing computational costs, making our approach both practical and efficient for large-scale credit card fraud detection.

4.3.2.3 Heterogeneous Graph Neural Networks

Heterogeneous graph neural networks (HGNNs) extend the concept of GNNs to heterogeneous graphs, which contain multiple types of nodes and edges. HGNNs have been used to model complex interactions in various domains, including recommendation systems, social networks, and fraud detection [97, 98]. By incorporating meta-paths and relation-specific transformations, HGNNs can capture the rich semantics of heterogeneous graphs, leading to improved performance in fraud detection tasks [99]. Our work advances the fraud detection task by employing scalable HGNNs specifically designed for modeling real-world fraudulent activities. Unlike previous GNN-based approaches, we propose an effective solution tailored to the heterogeneity of real-life graphs. Our model leverages the diverse types of nodes and edges to capture more realistic and intricate interactions between different entities. This results in a robust and efficient fraud detection system that outperforms traditional methods and other GNN-based solutions, providing significant improvements in both detection accuracy and computational efficiency.

4.4 Proposed Method

In this section, we introduce our proposed method for fraud detection using a Relation-Aware Heterogeneous Graph Neural Network (RHGNN) model. This section includes the model architecture, computation graph pre-process, heterogeneous propagation, and fraud detection and optimization.

4.4.1 Model Architecture

Different from other graph-based fraud detection techniques, we consider real-world data as heterogeneous graphs in the fraud detection task. For example, in the application of credit card fraud detection, the transaction node is the *"target"* node, and card and merchant nodes are the other

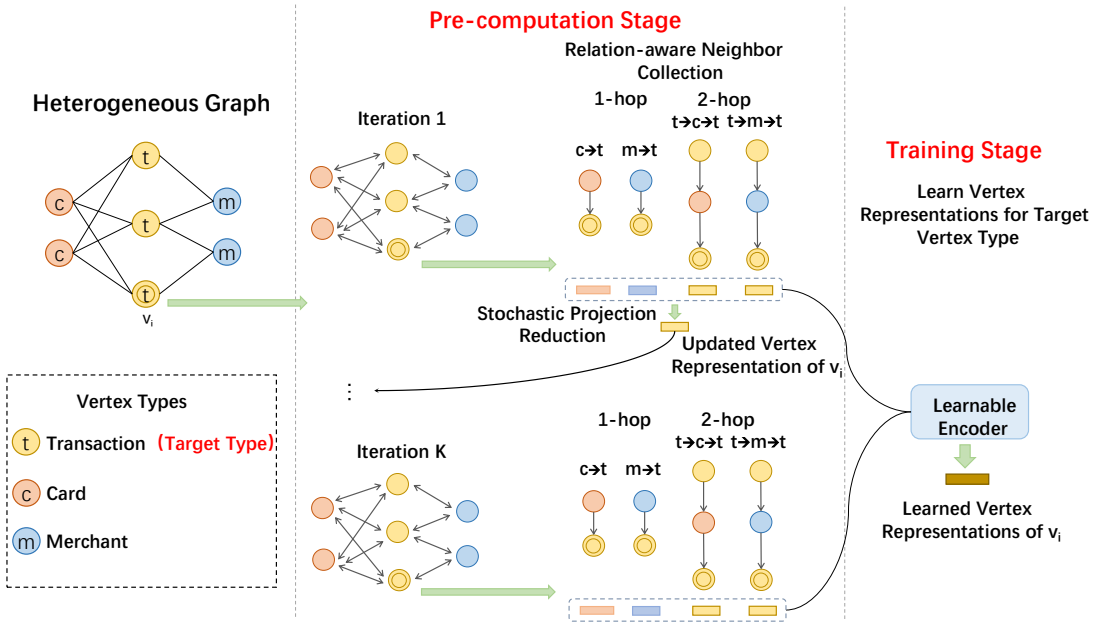


FIGURE 4.2: Overall Framework of the proposed Relation-Aware Heterogeneous Graph Neural Network model (RHGNN). (1) Pre-computation Stage: To leverage the efficiency of hybrid propagation, the main framework of RHGNN consists of alternating propagation iterations, where we introduce a Stochastic Projection Reduction component to manage dimensionality reduction for updated vertex representations. To exploit the granularity of relation-specific information for low information loss, we introduce a hybrid propagation scheme, which alternates focus between 1-hop and 2-hop neighbors, collecting neighbor information based on different relational contexts separately. (2) Training Stage: After K iterations, the collected neighbor information (of target vertices) is used as the input for a heterogeneous graph encoder to generate the vertex representations of target vertices, enhancing the model's ability to detect fraudulent activities accurately and efficiently.

types of nodes. Thanks to the success of Random Projection Heterogeneous Graph Neural Network (RpHGNN) [100], our model architecture leverages the strengths of heterogeneous graph neural networks (HGNNs) to capture the complex and varied interactions present in fraud detection input data. The architecture integrates both efficient relation-aware pre-computation and heterogeneous graph neural networks, enabling precise and scalable fraud detection. The core components of our model include:

- **Relation-Aware Node Mapping:** This component involves a propagation scheme that utilizes different local relations to gather potential fraudulent information. For each node type, we identify relevant edge types and apply message passing to collect and aggregate neighbor information, which aids in constructing an enriched feature representation for each node.

- **Stochastic Node Reducing:** To manage the high dimensionality of aggregated neighbor information, this method employs a stochastic dimensionality reduction technique. By applying stochastic node reducing, it effectively reduces the feature space dimensionality while preserving essential information, which is crucial for maintaining computational efficiency and model performance.

- **Intra-Relation Node Convolution:** Within the same type of relation, this convolution operation processes features through specified filters. It aggregates localized feature information from direct neighbors under the same relation type, enhancing the node's feature representation with context-specific information that is critical for identifying patterns indicative of fraud.

- **Heterogeneous Node Information Propagation:** This crucial step propagates information across various node types and relations within the heterogeneous graph. It employs meta-paths to guide the aggregation process, integrating diverse and complex interactions across the graph. This comprehensive aggregation allows the model to capture and leverage the nuanced dynamics characteristic of real-world data structures.

4.4.2 Computation Graph Pre-Process

Before training the model, we preprocess the computation graph to optimize the efficiency and effectiveness of the heterogeneous graph neural network operations for fraud detection.

Relation-Aware Node Mapping. In the relation-aware node mapping phase, we collect neighbor information using a propagation scheme that incorporates various local relations to gather potential fraudulent information. For a node type $A_j \in \mathcal{A}$, we identify a set of edge types ending with it. Each edge type $R_m = (A_i, *, A_j)$ is treated as a local relation $A_i \xrightarrow{R_m} A_j$, and our model performs message passing along these edges to collect neighbor information:

$$(H_{R_m})^{(k)} = (D_{R_m})^{-1} \mathbb{A}_{R_m} (H_{A_i})^{(k-1)} \tag{4.1}$$

Here, $(H_{A_i})^{(k-1)}$ denotes the vertex representation matrix from the $(k-1)$-th iteration, or the raw feature matrix if $k = 1$. $\mathbb{A}_{R_m}$ is the adjacency matrix for edge type R_m, and D_{R_m} is the degree matrix for vertices of type A_j.

Stochastic Node Reduction. To manage high-dimensional neighbor information, we employ a stochastic node reduction technique, maintaining efficiency while reducing dimensionality. For each node type A_j and its collected neighbor information $\{(H_{R_m})^{(k)} | R_m = (*, *, A_j)\}$, we update the vertex representation of A_j as follows:

$$(H_{A_j})^{(k)} = \sum_{R_m=(*,*,A_j)} \text{Norm}((H_{R_m})^{(k)}(W_{R_m})^{(k)}) \tag{4.2}$$

where $W_{R_m}^{(k)}$ is the stochastic projection weight matrix, ensuring that the dimensionality of the updated vertex representations remains constant. This approach avoids the efficiency issues of exponentially growing dimensionality seen in naive concatenation methods.

By leveraging the relation-aware node mapping and stochastic node reducing methods, we ensure that the collected neighbor information reflects the fine granularity of different relations, optimizing both the efficiency and the effectiveness of the graph neural network operations.

4.4.3 Heterogeneous Propagation

The heterogeneous propagation stage involves two key processes: intra-relation node convolution and heterogeneous node information propagation. Especially, we use 1-hop and 2-hop hybrid propagation scheme to ensure scalabiilty without performance loss.

Hybrid Propagation Scheme. In the traditional propagation scheme, each iteration collects neighbor information via 1-hop relations, requiring K iterations to capture relations within K hops. A simple way to reduce iterations is to use a 2-hop propagation scheme, extending local relations to 2 hops. This reduces the required iterations to $\frac{K}{2}$, lowering the frequency of untrainable vertex representation updates. However, this approach can be inefficient due to the potentially overwhelming number of 2-hop relations in certain graphs.

To address this efficiency issue, we design a hybrid 2-hop and 1-hop propagation scheme. This method selects twice as many 2-hop relations as local ones, including both 1-hop and 2-hop relations.

Intra-Relation Node Convolution. After confirming propagation scheme, in this step, we perform convolution operations within the same type of relation. For a given relation type $R \in \mathcal{R}$, we define the intra-relation convolution as:

$$\mathbf{H}_R^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}_R(v)} \frac{1}{c_{vu}} \mathbf{H}_u^{(l)} \mathbf{W}_R\right), \tag{4.3}$$

where $\mathbf{H}_u^{(l)}$ is the feature vector of node u at layer l, $\mathbf{W}_R \in \mathbb{R}^{d \times d}$ is the weight matrix for relation R, c_{vu} is a normalization constant, and σ is an activation function. This intra-relation convolution extracts localized features specific to each type of relation.

Heterogeneous Node Information Propagation. This process propagates information across different types of nodes and edges, ensuring that the model captures the complex interactions that span various entities in the financial graph. Let $\mathbf{H}^{(l+1)}$ be the aggregated feature matrix at layer $l+1$:

$$\mathbf{H}^{(l+1)} = \sigma\left(\sum_{R \in \mathcal{R}} \sum_{u \in \mathcal{N}_R(v)} \frac{1}{c_{vu}} \mathbf{H}_u^{(l)} \mathbf{W}_R\right), \tag{4.4}$$

where the information is aggregated from multiple relations and nodes, guided by meta-paths that define the sequence of relations to be followed.

4.4.4 Fraud Detection and Optimization

The final stage of our method focuses on detecting fraudulent transactions and optimizing the model's performance.

After propagating information through the heterogeneous graph, we apply a classification layer to identify potentially fraudulent transactions. The classification layer uses the enriched node representations generated by the previous steps to make accurate predictions. The final node representation $\mathbf{H}^*$ is passed through a softmax function to obtain the probability of each node being

fraudulent:

$$\hat{y}_v = \text{softmax}(\mathbf{H}_v^* \mathbf{W}_c), \tag{4.5}$$

where $\mathbf{W}_c$ is the weight matrix for the classification layer. To optimize the model, we employ techniques such as stochastic projection reduction to manage feature dimensionality, ensuring that the model remains efficient even with large-scale data. Additionally, we utilize loss functions tailored for imbalanced datasets, such as weighted cross-entropy, to improve the detection of rare fraudulent events:

$$\mathcal{L} = -\sum_{v \in \mathcal{V}} w_v y_v \log(\hat{y}_v), \tag{4.6}$$

where w_v is the weight assigned to node v based on class imbalance.

Our experimental results demonstrate that the proposed method not only improves fraud detection accuracy but also significantly reduces training time, making it suitable for real-time fraud detection systems.

4.5 Experiments

4.5.1 Experiments Settings

4.5.1.1 Datasets

To evaluate the fraud detection performance on financial networks, we conducted experiments on the T-Finance [96] dataset. The T-Finance dataset aims to find the anomaly accounts in transaction networks. The nodes are unique anonymized accounts with 10-dimension features related to registration days, logging activities and interaction frequency. The edges in the graph represent two accounts that have transaction records. Human experts annotate nodes as anomalies if they fall into categories like fraud, money laundering and online gambling. In the T-finance dataset, there are 39357 nodes, 21222543 edges and 10-dimension features.

Besides, we also conducted experiments on two public fraud detection datasets, the Yelp and Amazon dataset. The Yelp dataset consists of filtered spam and reviews recommending hotels and restaurants, while the Amazon dataset comprises reviews of musical instrument products. We

extracted 45954 nodes, 3846979 edges, and 32 manual features from [101] as the original graph and node features of the Yelp dataset. We also extracted 11944 nodes, 4398392 edges, and 25 manual features from [102] as Amazon's original graph and node features.

4.5.1.2 Compared Methods

We compare our proposed techniques with the baselines: GCN [103], GEM [104], GAT [105], FdGars [106], Play2Vec [107], GeniePath [108], Graphconsis [109], SEMIGNN [110]. These methods are popular for graph-based node classification tasks or fraud detection.

4.5.1.3 Parameter Setting and Evaluation Metrics

In the experiment, each baseline is conducted separately and adopts the optimal parameters as they were originally proposed. For our model, we set the model embedding size to 256, the hidden size to 256, the learning rate to 0.003, and using Adam as the optimizer. Our model's hyperparameter setting is a combination of several hyperparameters, which include the input dropout rate, the hidden dropout rate, and squash k which is a hyperparameter that regulates the degree of feature dimensionality reduction achieved through stochastic node reducing techniques. We set the model input drop rate to 0.01, the drop rate to 0.01, and squash k to 6. In terms of the running environment, we train the models on a Linux machine with 2 Intel(R) Xeon(R) Silver 4208 CPU @ 2.10GHz, 128GB RAM, and an RTX 6000 with 48GB of GPU memory. We evaluate the experimental results on financial fraud detection and opinion fraud datasets by the area under the ROC curve (AUC), F1 score(F1-macro), and average precision (AP). For all three metrics, the higher score indicates the higher performance of the methods.

4.5.2 Fraud Detection Experiment

In the fraud detection experiment, we evaluate the performance of our proposed model against the baselines on the T-Finance, Amazon, and YelpChi datasets. The results are summarized in Table 4.2. Our model outperforms all baselines in terms of AUC, F1-macro, and AP, demonstrating its effectiveness in detecting fraudulent behaviors.

TABLE 4.2: Fraud Detection Experimental Results.

Model	YelpChi			Amazon			T-Finance		
	AUC	F1	AP	AUC	F1	AP	AUC	F1	AP
GCN	0.8207	0.7847	0.7575	0.8401	0.8057	0.7986	0.8303	0.7944	0.8077
GEM	0.8267	0.7897	0.7635	0.8461	0.8109	0.8046	0.8363	0.8003	0.8142
GAT	0.8372	0.7968	0.7705	0.8576	0.8175	0.8113	0.8474	0.8071	0.8210
FdGars	0.8515	0.8093	0.7832	0.8714	0.8309	0.8241	0.8610	0.8202	0.8348
Play2Vec	0.8603	0.8183	0.7934	0.8807	0.8396	0.8332	0.8703	0.8287	0.8429
GeniePath	0.8671	0.8250	0.8009	0.8872	0.8467	0.8404	0.8773	0.8351	0.8498
Graphconsis	0.8732	0.8301	0.8055	0.8938	0.8529	0.8473	0.8834	0.8440	0.8551
SEMIGNN	0.8806	0.8364	0.8118	0.9207	0.8591	0.8543	0.9109	0.8463	0.8612
RHGNN	**0.9431**	**0.8458**	**0.8156**	**0.9734**	**0.8788**	**0.8561**	**0.9573**	**0.8935**	**0.8641**

In the fraud detection experiment, we observed that our proposed model, RHGNN, consistently outperformed the baseline models across all three datasets: YelpChi, Amazon, and T-Finance. These results indicate the robustness and superior performance of RHGNN in detecting fraudulent activities across different types of networks. The significant improvement in metrics across all datasets demonstrates the effectiveness of our model's design and its applicability to various fraud detection scenarios.

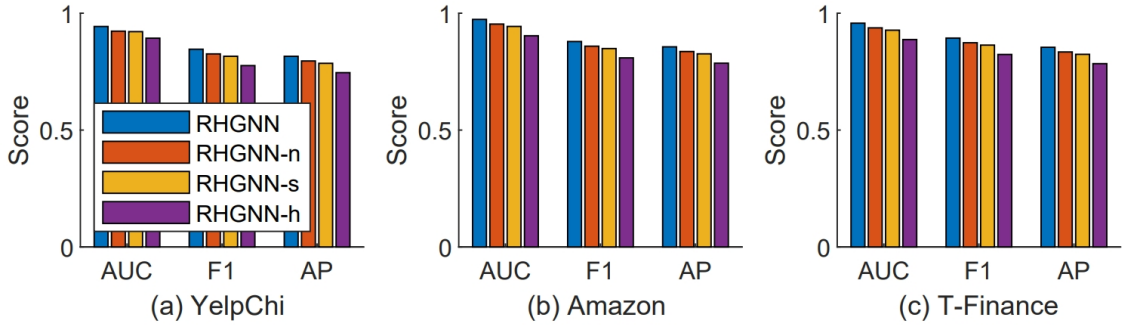


FIGURE 4.3: Ablation study results on three datasets.

4.5.3 Ablation Study

We conduct an ablation study to understand the contributions of different components of our model. The results are shown in Figure 4.3. Removing any of the components (hybrid neighbor scheme, stochastic node reduction, or heterogeneous propagation) results in a noticeable drop in performance, indicating the importance of each component. In this figure, RHGNN is our proposed method, RHGNN-n is the model with non-hybrid neighbor scheme, RHGNN-s denotes the

model without stochastic node reduction, and RHGNN-h denotes the model without heterogeneous propagation (treat the edges as the same type). Performance metrics analyzed are the AUC, F1 Score, and Average Precision (AP).

As shown in Figure 4.3(a), the base RHGNN model consistently outperforms other variants across all metrics on the YelpChi dataset. The RHGNN model achieves notably higher AUC, F1, and AP scores, indicating its comprehensive capability in handling the complexity and heterogeinity specific to YelpChi data. Both RHGNN-n and RHGNN-s variants show a slight reduction in performance, suggesting that the non-hybrid scheme and no stochastic node reduction slightly hampers effectiveness. The RHGNN-h shows diminished effectiveness, particularly in F1 and AP scores, underscoring the importance of the utilization of heterogeneous information. We can obtain similar results in the experimental results of the other two datasets.

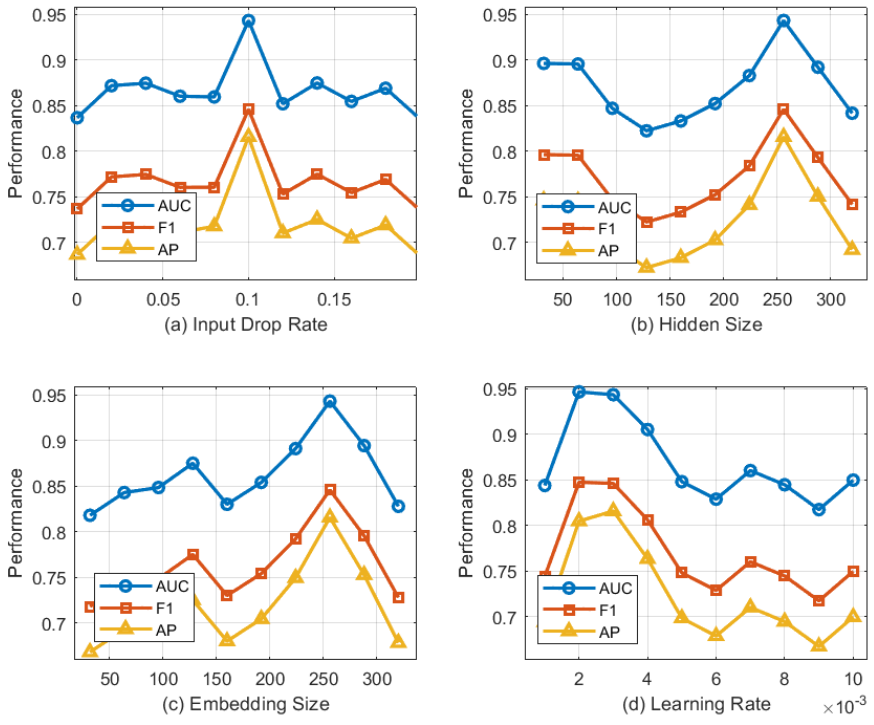


FIGURE 4.4: Parameter sensitivity analysis for the RHGNN model.

4.5.4 Parameter Sensitivity

To examine the sensitivity of our model to its hyperparameters, we vary key parameters such as embedding size, learning rate, and dropout rates. Figure 4.4 illustrates the impact of these parameters on the model's performance.

Specifically, Figure 4.4(a) shows the impact of varying the input drop rate on model performance. As the drop rate increases from 0 to 0.15, significant fluctuations in performance metrics are observed. The AUC metric remains relatively stable, maintaining high performance across different drop rates. However, both the F1 score and AP decline sharply as the drop rate exceeds 0.1, indicating a loss of critical information that adversely affects precision and recall balance. Figure 4.4(b) analyzes sensitivity to hidden size. The model's performance increases as the hidden size grows from 50 to around 150. Beyond this point, metrics peak and then slightly decline, particularly the F1 score and AP, suggesting an optimal range around 150-200. This is likely due to the model's ability to capture more complex patterns with more hidden units without yet overfitting. The trends in embedding size, shown in Figure 4.4(c), reveal a clear peak in performance at an embedding size of 150. Both the AUC and F1 scores improve significantly as the embedding size increases from 50 to 150, after which performance plateaus or slightly declines. This behavior suggests that up to a certain point, increased dimensionality provides sufficient representational space, beyond which additional dimensions do not contribute to better generalization. Lastly, Figure 4.4(d) evaluates the effect of learning rate adjustments. A lower learning rate of around 0.002 to 0.004 appears optimal, as indicated by higher values in all performance metrics. The AP and F1 scores decrease substantially with higher learning rates, suggesting that too large a step may lead to instability in training or skipping over optimal solutions. We choose our best parameter settings according to these results.

4.6 Conclusion

In this study, we proposed a novel Relation-Aware Heterogeneous Graph Neural Network (RHGNN) for fraud detection in financial and social media data. Our approach leverages the intricate relationships between different types of entities to enhance the detection capabilities beyond traditional machine learning methods, which typically rely on the features of individual nodes alone. Through extensive experiments, we demonstrated that RHGNN significantly outperforms existing state-of-the-art methods across various metrics, including AUC, F1 score, and average precision. The results confirm the efficacy of incorporating relation-awareness in graph neural networks, providing a more nuanced understanding of fraudulent patterns. This advancement opens new avenues for developing even more sophisticated models that can handle the complexity and diversity

of real-world data. Future work will focus on further enhancing the scalability of RHGNN and exploring its applicability to other domains beyond fraud detection. We also aim to investigate the integration of additional data sources and the impact of temporal dynamics on the model's performance.

Chapter 5

Black Box Attack for Credit Card Fraud Detection

5.1 Chapter Overview

In this chapter, we show how to implement a black box attack on credit card fraud detection. This chapter has been submitted on Information Sciences journal. In section 5.2, we give the background of credit card fraud detection. And related works on fraud detection are introduced in Section 5.3. The following sections give the proposed solutions and show the performance comparisons. The Section 5.7 concludes this chapter.

5.2 Background

Credit card fraud detection (CCFD) has become a critical area of research due to the increasing prevalence of online transactions and the associated financial risks. While machine learning models have significantly improved the accuracy of fraud detection, their robustness against adversarial attacks remains a pressing concern. Among various attack strategies, black-box attacks have gained significant attention because they align closely with real-world scenarios, where attackers typically lack detailed knowledge of the target system. Unlike white-box and gray-box attacks, which require varying degrees of access to the target model's internal parameters or architecture,

black-box attacks [111–113], especially transfer-based black-box attacks [112, 114], operate without privileged information. This makes them more generalizable and practical for evaluating fraud detection systems under adversarial settings, ensuring their reliability and robustness.

With the rapid development of machine learning, many highly generalizable classification models, as well as anomaly detection and fraud detection models, have been proposed, which have demonstrated strong performance in CCFD tasks. These models can be categorized into differentiable models, non-differentiable models, and unsupervised learning approaches. Differentiable models, such as Logistic Regression (LR) [115] and Deep Neural Networks (DNN) [116], rely on gradient-based optimization techniques to learn complex relationships between features and labels, making them effective for tasks requiring high accuracy. Non-differentiable models include Decision Tree (DT) [117], Linear Support Vector Machines (Linear-SVM) [118], Radial Basis Function Support Vector Machines (RBF-SVM) [118], and Extreme Gradient Boosting (XGBoost) [119]. These models are often favored for their interpretability or their ability to handle non-linear decision boundaries. Lastly, unsupervised learning approaches, such as Isolation Forest (IF) [120] and Denoising Autoencoders (DAE) [121], are designed to detect anomalies by identifying patterns that deviate from the norm, making them particularly useful for identifying novel fraud patterns. These algorithms, either relying on their strong generalizability or leveraging their exceptional fraud and anomaly detection capabilities, demonstrate outstanding performance in CCFD tasks.

In recent years, significant efforts have been devoted to exploring the security of machine learning models, particularly through adversarial attacks [122–124]. These studies primarily focus on generating adversarial examples to exploit vulnerabilities in machine learning systems. Transfer-based attacks [112, 114, 125], introduced by [114], leverage the transferability of adversarial examples between substitute and target models. This approach involves training a substitute model to approximate the target model's behavior and crafting adversarial examples against the substitute model, which are then used to attack the target model. While effective, these existing methods often require a large number of queries to the target model to train the substitute model, making it impractical in real-world scenarios. Score-based attacks [126, 127] estimate the gradient of the target model's loss function using confidence scores for input samples. This enables the construction of adversarial examples without requiring access to the model's internal parameters. However, these methods rely on the availability of precise confidence scores, which may not always be accessible in real-world applications. Decision-based attacks [128, 129] rely solely on

the final classification decision of the target model to iteratively refine adversarial perturbations. Although this approach eliminates the need for confidence scores, it still requires a substantial number of queries to achieve effective attacks. There are many other outstanding works related to adversarial learning. Despite these advancements, most existing studies focus on domains such as image [124, 130], point clouds [131, 132], language [133, 134] and speech recognition [135, 136], leaving the security of CCFD models under adversarial attacks largely unexplored. This gap underscores the need for tailored attack strategies that address the unique challenges and constraints of CCFD systems.

The current state-of-the-art attack model for CCFD is the Semi-supervised Transfer Black-box Attack (STBA) [14]. This model combines semi-supervised learning techniques with transfer-based black-box attacks to evaluate the security of machine learning-based CCFD systems. The STBA framework operates by first training a substitute model using a small number of labeled samples and a large number of unlabeled samples, which are selectively labeled through semi-supervised learning methods. The substitute model is then used to generate adversarial examples. Using these samples, a robust fraud classification model can be trained, which produces an adversarial perturbation vector. This vector enables black-box attacks on unknown target models. Compared to earlier works, STBA overcomes several critical limitations. It significantly reduces the reliance on target model queries, enhancing practicality. Moreover, the optimization model employed by STBA is specifically designed for the CCFD domain, incorporating industry-specific constraints to enable the generation of more realistic adversarial examples. However, STBA still has limitations. Its design remains relatively straightforward and static, with limited performance and adaptability to diverse scenarios, which restricts its effect and generalizability.

Motivated by the above method, we propose our model for attacking CCFD, named Black-box Adaptive Fraud Detection Attack model (BADA). BADA first leverages the Ensemble-guided Adversarial Generation module, enabling the initialization of adversarial vectors with enhanced generalizability. This design leverages the complementary strengths of diverse models, improving the transferability of adversarial vectors and improving the adaptability and effectiveness of the attack across diverse scenarios. Next, drawing partial inspiration from the static space memory technology [137], we propose the Memory-based Adversarial Defense module to build a robust surrogate model that simulates the target model under attack. To improve attack effectiveness

without increasing the query burden on the target model, the module maintains a static knowledge space. This static memory helps stabilize the learning process of the Adversarial Defense Surrogate model, enhancing its robustness and reliability under adversarial settings. By closely approximating the behavior of the robust target model, the surrogate model provides an optimized learning environment for the subsequent attack module, thereby enhancing its attack capabilities. Finally, BADA incorporates a surrogate-based Adaptive Attack module, which leverages the surrogate model to perform effective and generalizable attacks against CCFD models. This module refines the initial adversarial perturbation vectors through an iterative optimization process, aiming to maximize the degradation of the target models' performance. Through the integration of these components, BADA achieves superior performance and generalizability compared to state-of-the-art black-box attack models for CCFD while requiring only a minimal number of queries to the target model. By integrating these components, BADA achieves superior performance and generalizability compared to state-of-the-art black-box attack models for CCFD while requiring only a minimal number of queries to the target model. The contribution of this work is summarized as follows:

- To the best of our knowledge, BADA is the first attack model to introduce adaptive adversarial optimization and static space memory technology into CCFD under transfer-based black-box settings. By leveraging these innovations, BADA provides a tailored solution for evaluating the robustness of CCFD systems.
- BADA introduces several novel modules, including an Ensemble-guided Adversarial Generation module that initializes generalizable adversarial perturbation vectors, a Memory-based Adversarial Defense module designed to establish a robust surrogate model, and a surrogate-based Adaptive Attack module that refines perturbations through optimization. These innovations collectively enhance the effectiveness and generalizability of adversarial attacks on CCFD models.
- Extensive experiments validate the superior performance of BADA across multiple datasets and target models. The results demonstrate BADA's ability to achieve stronger attack performance compared to state-of-the-art baselines, demonstrating its practical applicability and effectiveness across diverse real-world scenarios.

5.3 Related Work

Credit card fraud detection models. In recent years, significant progress has been made in credit card fraud detection (CCFD) models, which play a critical role in financial risk management and fraud prevention [138, 139]. CCFD is typically formulated as a binary classification problem, divided into supervised and unsupervised learning approaches. Supervised models, such as Logistic Regression (LR) [115], Deep Neural Networks (DNN) [116], Decision Trees (DT) [117], Linear Support Vector Machines (Linear-SVM) [118], Radial Basis Function Support Vector Machines (RBF-SVM) [118], and Extreme Gradient Boosting (XGBoost) [119], excel in classification accuracy [14]. For instance, DNNs are particularly effective in capturing complex feature relationships, while XGBoost demonstrates strong performance in handling imbalanced datasets. These models learn mappings between transaction features and labels, enabling the rapid identification of known fraud patterns. Unsupervised models, including Isolation Forest (IF) [120] and Denoising Autoencoders (DAE) [121], do not rely on labeled data and can detect novel fraud patterns. For instance, IF identifies anomalies by detecting outliers, while DAE distinguishes normal and fraudulent transactions through reconstruction errors. These models are particularly effective in scenarios with limited labels or unknown fraud patterns. However, their performance can be sensitive to parameter selection and may degrade when applied to high-dimensional datasets. Additionally, some studies utilize balance theory and matrix decomposition to predict fraud transactions as edge signs in graphs [140–142]. Graph neural networks (GNNs) and graph embedding techniques have also been applied to fraud detection, effectively modeling complex relationships in graph structures [143–148]. These methods excel in capturing intricate dependencies between entities but often face challenges related to computational complexity and scalability in large-scale datasets. Existing research provides a solid theoretical and practical foundation for constructing CCFD models, paving the way for designing effective attack models. However, the robustness of these models under adversarial settings remains underexplored, highlighting the need for tailored attack strategies to evaluate and enhance their security.

Black-box adversarial attacks. Significant advancements have been made in the study of adversarial attacks on machine learning models, particularly in black-box scenarios, which are highly relevant to real-world applications such as CCFD. Existing methods are broadly categorized into

transfer-based, score-based, and decision-based approaches, each with distinct strengths and limitations. Transfer-based attacks exploit the transferability of adversarial examples between substitute and target models [112, 114, 125]. This approach is advantageous as it does not require direct access to the target model's parameters, making it suitable for black-box scenarios. However, it often requires numerous queries to train accurate substitute models, which can limit its practicality in real-world applications. Score-based methods estimate gradients using confidence scores, bypassing the need for direct access to model parameters [126, 127]. While this approach is effective in scenarios where confidence scores are accessible, its reliance on precise confidence scores also limits its practicality, where such information is often unavailable. Decision-based attacks, such as boundary attacks, rely solely on classification decisions [128, 129]. This eliminates the need for confidence scores, making it particularly suitable for scenarios where only classification decisions are accessible, such as fully black-box settings. However, it typically demands extensive queries to iteratively adjust inputs and locate decision boundaries, which can be computationally expensive. Despite their progress, all three approaches face challenges in reducing query costs, which remains a critical barrier to their practical application in CCFD tasks. In the context of CCFD, the state-of-the-art attack method is the Semi-supervised Transfer Black-box Attack (STBA) [14]. This method combines semi-supervised learning techniques with transfer-based black-box attack algorithms, significantly reducing the number of queries required to train substitute models. By leveraging a small number of labeled samples and a large number of unlabeled samples, STBA enhances the similarity between substitute and target models, thereby improving attack effectiveness. However, its relatively static design limits its adaptability to diverse scenarios, leaving room for improvement in attack performance and generalizability. To address the limitations of existing approaches, we propose BADA, a novel and more effective black-box adversarial attack model for CCFD. It incorporates well-designed modules, including an Ensemble-guided Adversarial Generation module, a Memory-based Adversarial Defense module, and an Adaptive Attack module. Compared to STBA, BADA introduces dynamic and adaptive mechanisms, enhancing its generalizability and effectiveness across diverse target models. These innovations provide valuable insights for evaluating and designing robust CCFD systems.

5.4 Preliminaries

In this chapter, we focus on launching a transfer black-box adversarial attack against credit card fraud detection (CCFD) models. To facilitate understanding, we summarize the notations used throughout this chapter in 5.1. To ground our study, we first define the structure of a credit card transaction dataset, which serves as the basis for training and evaluating fraud detection models. The dataset is formally described as follows:

Definition 5.1 (Credit Card Transaction Dataset)**.** A credit card transaction dataset is represented as $\mathcal{D} = (\boldsymbol{H}, \boldsymbol{Y})$, where $\boldsymbol{H} \in \mathbb{R}^{N \times d}$ denotes the matrix of N transaction records, each with a d-dimensional feature vector, and $\boldsymbol{Y} \in \{0, 1\}^N$ is the label set, where 0 indicates a non-fraudulent transaction and 1 indicates a fraudulent transaction.

The credit card transaction dataset serves as the foundation for training and evaluating fraud detection models. However, the robustness of these models can be challenged by adversarial attacks, especially in scenarios where attackers aim to manipulate the input data to deceive the model. Among various attack strategies, black-box attacks are particularly common and warrant significant attention for defense, as they require minimal information about the target model. To formalize this concept, we define the notion of black-box attacks as follows:

Definition 5.2 (Black-Box Attack)**.** A black-box attack on a classification model is an adversarial strategy where the attacker has limited or no knowledge of the model's internal structure, parameters, or dataset. The goal of the attack is to perturb the input features in a way that causes the model to make incorrect predictions.

The attack models proposed in this work must demonstrate transferability, ensuring their effectiveness against unseen CCFD models, i.e., models that were not encountered during the training phase. This property is critical as it reflects the practical challenges of real-world adversarial scenarios. To further clarify the scope of our study, we now formalize the research problem in the context of semi-supervised transfer black-box adversarial attacks. The problem is described as follows:

Problem Statement. The objective of this work is to develop a transferable black-box attack model that targets a pre-trained classifier $f(\boldsymbol{x}) : \mathbb{R}^d \rightarrow \{0, 1\}$, which predicts whether input

Notation	Description
$\mathcal{D} = (\boldsymbol{H}, \boldsymbol{Y})$	The credit card transaction dataset, where $\boldsymbol{H} \in \mathbb{R}^{N \times d}$ is the feature matrix and $\boldsymbol{Y} \in \{0,1\}^N$ is the label set.
$\mathcal{D}_t$, $\mathcal{D}_a$	The dataset used by the target model for training ($\mathcal{D}_t$), and the dataset available to the attacker ($\mathcal{D}_a$) with the majority of labels masked.
$\mathcal{D}_{a,l} = (\boldsymbol{H}_{a,l}, \boldsymbol{Y}_{a,l}), \mathcal{D}_{a,u} = (\boldsymbol{H}_{a,u})$	Subsets split from $\mathcal{D}_a$, where $\mathcal{D}_{a,l}$ is the labeled subset with feature matrix $\boldsymbol{H}_{a,l}$ and corresponding labels $\boldsymbol{Y}_{a,l}$, and $\mathcal{D}_{a,u}$ is the unlabeled subset with feature matrix $\boldsymbol{H}_{a,u}$.
$\mathcal{D}_{sur}, \mathcal{D}_{atk}$	Datasets split from $\mathcal{D}a, l$, where $\mathcal{D}sur$ is used for training the surrogate model and $\mathcal{D}_{atk}$ is used for evaluating the attack effectiveness.
$\boldsymbol{w}, \boldsymbol{w}_i, \boldsymbol{w}_\mathrm{a}$	The adversarial perturbation vector ($\boldsymbol{w}$), its initialized version ($\boldsymbol{w}_i$), and the optimized perturbation vector used to attack the target model ($\boldsymbol{w}_\mathrm{a}$).
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$	Weight matrices used in the surrogate model for feature transformation and prediction.

TABLE 5.1: Notation Table

feature vectors represent non-fraudulent or fraudulent transactions. The goal of the attack is to reduce the fraud detection accuracy of the target model by causing it to misclassify fraudulent transactions as non-fraudulent. Specifically, the model leverages an credit card transaction dataset with limited labeled samples to optimize an adversarial perturbation vector $\boldsymbol{w} \in \mathbb{R}^{d'}$, where $d' \ll d$, to modify only a subset of input features while maintaining the transferability of the attack across unseen models.

5.5 Proposed Model

In this section, we present our proposed model BADA. It is structured into four key components: the preprocessing stage, the Ensemble-guided Adversarial Generation module, the Memory-based Adversarial Defense module, and the Adaptive Attack module. 5.1 provides an overview of the framework. First, the preprocessing stage prepares the dataset and target model for subsequent attack simulations by normalizing and balancing the data. Then, the Ensemble-guided Adversarial Generation module initializes a coarse-grained perturbation vector $\boldsymbol{w}_i$ using ensemble learning techniques, providing a robust starting point for adversarial attacks. Next, the Memory-based Adversarial Defense module constructs a robust surrogate model to simulate the target model's

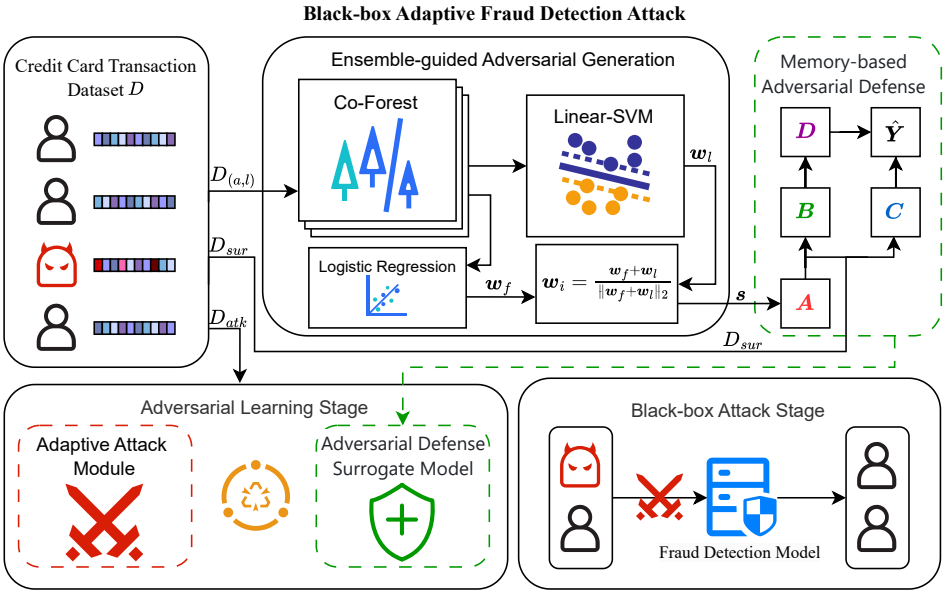


FIGURE 5.1: Framework of BADA

behavior, enabling effective adversarial learning. Finally, the Adaptive Attack module refines $\boldsymbol{w}_a$ through iterative optimization, maximizing the attack's effectiveness while maintaining stealthiness. These interconnected components collectively form a robust framework that enables adaptable and stealthy attacks.

5.5.1 Preprocessing stage

To ensure robust and effective attack performance, we begin by preprocessing the dataset to prepare the data for subsequent attack simulations. Given a dataset $\mathcal{D} = (\boldsymbol{H}, \boldsymbol{Y})$, the feature matrix $\boldsymbol{H}$ is normalized to map each feature into a range $[a, b]$:

$$\bar{\boldsymbol{H}}_{i,j} = a + \frac{(\boldsymbol{H}_{i,j} - \min(\boldsymbol{H}_j))}{\max(\boldsymbol{H}_j) - \min(\boldsymbol{H}_j)} \times (b - a), \tag{5.1}$$

where $\boldsymbol{H}_{i,j}$ is the value of the feature in the i-th sample, and $\min(\boldsymbol{H}_j)$, $\max(\boldsymbol{H}_j)$ denote the minimum and maximum values of the feature across all samples, respectively. This normalization mitigates scale variability among features, ensuring that all dimensions contribute comparably to model predictions. For consistency, we normalize all features to the range $[0, 1]$, yielding the feature matrix $\bar{\boldsymbol{H}}$ utilized in subsequent steps. The preprocessed dataset is partitioned into two subsets: $\mathcal{D}_t$, used for training the target model, and $\mathcal{D}_a$, reserved for black-box attack simulation.

This partitioning strategy reflects real-world scenarios where attackers typically have access to only a limited portion of the data, with most labels in $\mathcal{D}_a$ masked to simulate the limited knowledge of attackers regarding the true class labels. This setup ensures that the attack model is evaluated under realistic conditions, highlighting its robustness and adaptability. The parameters of the target models are fixed after training, thus emulating a black-box attack scenario.

5.5.2 Ensemble-guided Adversarial Generation

After obtaining $\mathcal{D}_a$, the Ensemble-guided Adversarial Generation module initializes the perturbation vector, which serves as the foundation for crafting the final adversarial perturbation vector. This initialization leverages the weight vector of a fraud detection classifier, guided by the geometric properties of decision boundaries [14]. In black-box attacks, decision boundaries play a critical role, as they define the regions of feature space corresponding to different classes. For a well-performing classifier, its decision boundary vector effectively separates different classes, providing a robust basis for generating transferable adversarial perturbations. Because most available transaction records in the attack process lack label information, BADA employs semi-supervised learning using the Co-Forest method [149] on $\mathcal{D}_a$. This approach begins by initializing m base classifiers, denoted as $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, each trained on a distinct subset of the labeled data. Specifically, the labeled dataset $\mathcal{D}_{a,l} = \{(\boldsymbol{H}_{a,j}, \boldsymbol{Y}_{a,j}) \mid (\boldsymbol{H}_{a,j}, \boldsymbol{Y}_{a,j}) \in \mathcal{D}_a, \boldsymbol{Y}_{a,j} \neq \varnothing\}$ is randomly partitioned into m subsets. For clarity, we denote the feature vectors and labels in the labeled dataset $\mathcal{D}_{a,l}$ as $\boldsymbol{H}_{a,l}$ and $\boldsymbol{Y}_{a,l}$, respectively. Each classifier $\mathcal{C}_i$ is trained on a different subset of $\mathcal{D}_{a,l}$. In this work, decision trees are chosen as the base classifiers due to their robustness in handling high-dimensional data and imbalanced class distributions. By training each base classifier on a unique subset of the labeled data, the Co-Forest establishes a foundation for collaborative learning in subsequent procedures.

Next, the module enters an iterative process to progressively label the unlabeled data $\mathcal{D}_{a,u}$ with high confidence, where $\mathcal{D}_{a,u} = \{(\boldsymbol{H}_{a,j}, \boldsymbol{Y}_{a,j}) \mid (\boldsymbol{H}_{a,j}, \boldsymbol{Y}_{a,j}) \in \mathcal{D}_a, \boldsymbol{Y}_{a,j} = \varnothing\}$. At the start of each iteration, a random subset $\mathcal{D}^i_{a,u} \subset \mathcal{D}_{a,u}$ is sampled and allocated to each base classifier $\mathcal{C}_i$. This subset serves as the candidate pool for pseudo-labeling, enabling the ensemble to iteratively refine its predictions and improve the quality of the labeled dataset. For every sample $\boldsymbol{H}^i_{a,u} \in \mathcal{D}^i_{a,u}$, the module aggregates the predictions from all other classifiers $\{\mathcal{C}_j \mid j \neq i\}$. The average confidence

score for class $c \in \{0, 1\}$ is computed as:

$$P_c(\boldsymbol{H}^i_{a,u}) = \frac{1}{|\boldsymbol{H}_{a,u}| - 1} \sum_{j \neq i} \mathcal{C}_j(\boldsymbol{H}^i_{a,u}), \tag{5.2}$$

where $\mathcal{C}_j(\boldsymbol{H}^i_{a,u})$ represents the probability predicted by the j-th classifier $\mathcal{C}_j$ that $\boldsymbol{H}^i_{a,u}$ belongs to class c. Samples with confidence scores exceeding a predefined threshold θ are selected for pseudo-labeling. The pseudo-labeled samples are then added to the labeled dataset of the corresponding classifier as follows:

$$\mathcal{D}'_{a,u} = \{(\boldsymbol{H}^i_{a,u}, c) \mid P_c(\boldsymbol{H}^i_{a,u}) \geq \theta\}. \tag{5.3}$$

Each base classifier is then retrained using the labeled set, which includes both the original labeled samples and the newly pseudo-labeled ones. This process is repeated iteratively for a fixed number of iterations or until convergence. Upon the completion of the iterative pseudo-labeling process, the module consolidates the labeled data to train a final classifier, which integrates the collective knowledge of these base classifiers. The training dataset for this classifier is defined as the union of the original labeled samples $\mathcal{D}_{a,l}$ with the pseudo-labeled samples $\mathcal{D}'_{a,u}$:

$$\mathcal{D}'_a = \mathcal{D}_{a,l} \cup \mathcal{D}'_{a,u}. \tag{5.4}$$

To mitigate the impact of uncertainty in black-box attacks, we employ two classifiers, Logistic Regression and Linear-SVM, which are combined to form an ensemble-based final classifier in Co-Forest. They are selected for their ability to generate linear decision boundaries, which facilitate adversarial perturbations. Once optimized, we extract the decision boundary vectors $\boldsymbol{w}_f$ and $\boldsymbol{w}_l$ from the coefficients of Logistic Regression and Linear-SVM, respectively. By integrating these vectors, the initialization vector $\boldsymbol{w}_i$ is computed as follows:

$$\boldsymbol{w}_i = \frac{\boldsymbol{w}_f + \boldsymbol{w}_l}{\|\boldsymbol{w}_f + \boldsymbol{w}_l\|_2}. \tag{5.5}$$

where $\| \cdot \|_2$ denotes the Euclidean norm. This boundary vector leverages the insights from both classifiers, ensuring robustness and generalizability. It serves as the foundation for the attack module, enabling effective boundary-oriented perturbations that align closely with the target model's decision boundaries.

5.5.3 Memory-based Adversarial Defense

The initialized perturbation vector $\boldsymbol{w}_i$ obtained from the preprocessing stage provides a preliminary attack direction. To refine the attack at a finer granularity, we construct a surrogate model to simulate the behavior of target models. A standard multilayer perceptron (MLP) can serve as a surrogate model, but its simplicity limits its effectiveness as it struggles to capture complex decision boundaries. To enhance the surrogate model's robustness and predictive accuracy, the Memory-based Adversarial Defense module is proposed. It integrates $\boldsymbol{w}_i$ into its architecture to enhance adversarial adaptability and employs a structured mapping mechanism to approximate the target model's behavior more effectively.

Given the labeled attacker-accessible dataset $\mathcal{D}_{a,l}$, we first divide it into two disjoint subsets, $\mathcal{D}_{sur} = (\boldsymbol{H}_{sur}, \boldsymbol{Y}_{sur})$ and $\mathcal{D}_{atk} = (\boldsymbol{H}_{atk}, \boldsymbol{Y}_{atk})$. Then, we train the Adversarial Defense Surrogate model by leveraging the dataset $\mathcal{D}_{sur}$. The structure of the Memory-based Adversarial Defense module comprises a static adversarial space $\boldsymbol{s}$ and four linear transformations, represented by weight matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$, where:

$$\boldsymbol{s} = \boldsymbol{w}_i, \quad \boldsymbol{A} \in \mathbb{R}^{d' \times d}, \quad \boldsymbol{B} \in \mathbb{R}^{d \times \frac{d}{2}}, \quad \boldsymbol{C} \in \mathbb{R}^{d \times 2}, \quad \boldsymbol{D} \in \mathbb{R}^{\frac{d}{2} \times 2}. \tag{5.6}$$

The forward pass of the module takes an input feature vector $\boldsymbol{x} \in \mathbb{R}^d$ and processes it as follows. First, a global influence vector $\boldsymbol{s}'$ is computed from the static adversarial space $\boldsymbol{s}$ using a weight matrix $\boldsymbol{A}$:

$$\boldsymbol{s}' = \boldsymbol{s}\boldsymbol{A}. \tag{5.7}$$

This transformation projects the adversarial influence into a latent space, providing a reference for subsequent perturbation attacks and enhancing the surrogate model's knowledge of adversarial behaviors. Next, it is applied to the entire feature matrix $\boldsymbol{H}_{sur}$. Each row of $\boldsymbol{H}_{sur}$ is adjusted by subtracting a scaled version of $\boldsymbol{s}'$ to create a perturbed feature matrix $\boldsymbol{H}_p \in \mathbb{R}^{|\boldsymbol{H}_{sur}| \times d}$ for the surrogate model:

$$\boldsymbol{H}_p = \boldsymbol{H}_{sur} - \beta \mathbf{1} \boldsymbol{s}', \tag{5.8}$$

where $\beta > 0$ is a scaling factor, and $\mathbf{1} \in \mathbb{R}^{|\boldsymbol{H}_{sur}| \times 1}$ is a column vector of ones ensuring proper broadcasting across all rows. By incorporating the vector $\boldsymbol{s}'$ into its architecture, the module

effectively simulates the attack scenario, enabling the surrogate model to achieve enhanced adversarial resilience and more accurately approximate the target model's decision-making, laying the groundwork for subsequent perturbation vector optimization. Then, a non-linear mapping is applied through a weight matrix $\boldsymbol{B}$:

$$\boldsymbol{X}_b = \sigma(\boldsymbol{H}_p \boldsymbol{B}), \tag{5.9}$$

where $\sigma(\cdot)$ is a non-linear activation function. This process captures the complex feature interactions influenced by perturbations. Simultaneously, to retain the surrogate model's ability to directly capture unperturbed features, the original input $\boldsymbol{H}_{sur}$ undergoes a linear transformation using the weight matrix $\boldsymbol{C}$:

$$\boldsymbol{X}_c = \boldsymbol{H}_{sur} \boldsymbol{C}. \tag{5.10}$$

This process provides access to original information for the surrogate model during training while reducing the impact of attack scenario simulations on its performance. Finally, by integrating $\boldsymbol{X}_b$ and $\boldsymbol{X}_c$ with the weight matrix $\boldsymbol{D}$, the surrogate model computes the fraud detection result as:

$$\boldsymbol{Y'} = \text{Softmax}(\boldsymbol{X}_b \boldsymbol{D} + \boldsymbol{X}_c), \tag{5.11}$$

where $\text{Softmax}(\boldsymbol{z}) = \frac{e^{\boldsymbol{z}}}{\sum_{i=1}^{|\boldsymbol{z}|} e^{\boldsymbol{z}_i}}$ is the softmax function, and $\boldsymbol{z} = \boldsymbol{X}_b \boldsymbol{D} + \boldsymbol{X}_c$. This function normalizes the output into a probability distribution over the two classes (fraudulent and non-fraudulent). Then, we take the last column of $\boldsymbol{Y'} \in \mathbb{R}^{|\boldsymbol{H}_{sur}| \times 2}$ as the fraud probability predicted by the final model, denoting it as $\hat{\boldsymbol{Y}}$. Because the surrogate model solves binary classification problems, it is trained using cross-entropy loss as follows:

$$\mathcal{L}_{sur} = -\frac{1}{|\boldsymbol{H}_{sur}|} \sum_{i=1}^{|\boldsymbol{H}_{sur}|} \left[\boldsymbol{Y}_{sur}^{(i)} \log(\hat{\boldsymbol{Y}}^{(i)}) + (1 - \boldsymbol{Y}_{sur}^{(i)}) \log(1 - \hat{\boldsymbol{Y}}^{(i)}) \right] \tag{5.12}$$

By fully leveraging the limited information as a foundation, this module introduces adversarial perturbation simulation, providing a more powerful and flexible surrogate model to guide subsequent attack optimization. For clarity, the trained Adversarial Defense Surrogate model is denoted as $\text{ADS}(\cdot; \theta)$, where θ denotes the fixed parameters of the model following optimization.

5.5.4 Adversarial Learning Stage

Relying on the surrogate model's approximation of target models' behavior, we proceed to optimize $\boldsymbol{w}_i$ to enhance the attack's effectiveness. This is achieved using second-order derivative-based optimization to improve the convergence rate and attack performance. Since our focus is solely on guiding the target model to incorrectly predict fraudulent behavior, we optimize $\boldsymbol{w}_i$ exclusively using the fraudulent behavior data in $\mathcal{D}_{atk}$. We denote this subset of the dataset as $\mathcal{D}'_{atk} = \{(\boldsymbol{H}^{(i)}_{atk}, \boldsymbol{Y}^{(i)}_{atk}) \in \mathcal{D}_{atk} \mid \boldsymbol{Y}^{(i)}_{atk} = 1\}$, where $\boldsymbol{H}^{(i)}_{atk}$ and $\boldsymbol{Y}^{(i)}_{atk}$ represent the i-th vector and element of $\boldsymbol{H}_{atk}$ and $\boldsymbol{Y}_{atk}$, respectively. Similarly, for clarity in subsequent explanations, we define $\boldsymbol{H}'_{atk}$ and $\boldsymbol{Y}'_{atk}$ to represent the feature vectors and labels in $\mathcal{D}'_{atk}$, respectively. The optimized perturbation vector $\boldsymbol{w} \in \mathbb{R}^{d'}$ is initialized by $\boldsymbol{w}_i$. To craft adversarial examples, the matrix $\boldsymbol{H}'_{atk}$ is modified by subtracting the corresponding elements of $\boldsymbol{w}$ from the perturbable dimensions:

$$\boldsymbol{H}_{mod} = \boldsymbol{H}'_{atk} - (\boldsymbol{M} \odot \mathrm{PL}(\boldsymbol{w})), \tag{5.13}$$

where $\odot$ denotes the Hadamard product and $\boldsymbol{H}_{mod}$ denotes the perturbed feature matrix for attack model. $\mathrm{PL}(\cdot)$ is a mapping operation that takes the perturbation vector $\boldsymbol{w}$ as input and outputs a d-dimensional vector, where the values of $\boldsymbol{w}$ are aligned with the d' perturbable dimensions in d, and all other dimensions are set to 0. $\boldsymbol{M} \in \mathbb{R}^{|\boldsymbol{Y}_{atk}| \times d}$ is a binary mask matrix where d' columns corresponding to the perturbable dimensions are set to 1, and all other columns are set to 0. Next, we utilize the surrogate model $\mathrm{ADS}(\cdot;\theta)$ to predict the fraud probabilities:

$$\hat{\boldsymbol{Y}}_{mod} = \mathrm{ADS}(\boldsymbol{H}_{mod}; \theta), \tag{5.14}$$

where each element $\hat{\boldsymbol{Y}}^{(i)}_{mod} \in \hat{\boldsymbol{Y}}_{mod}$ represents the likelihood that the i-th sample is classified as fraudulent by the surrogate model. In continuation, to optimize $\boldsymbol{w}$, we utilize the cross-entropy loss function to guide the optimization of the perturbation vector, leading to a decrease in the effectiveness of the surrogate model:

$$\mathcal{L}_{atk}(\boldsymbol{w}) = -\frac{1}{|\boldsymbol{H}_{mod}|} \sum_{i=1}^{|\boldsymbol{H}_{mod}|} \log\left(1 - \hat{\boldsymbol{Y}}^{(i)}_{mod} + \epsilon\right), \tag{5.15}$$

where ϵ is a small positive constant ensuring numerical stability. This objective imposes a penalty on high fraud detection probabilities, thereby guiding the optimization process to refine the perturbation vector for better concealing fraudulent activities. At this stage, the L-BFGS optimization algorithm is employed to iteratively update $\boldsymbol{w}$ by minimizing the objective function $\mathcal{L}_{atk}(\boldsymbol{w})$:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \mathcal{H}_t^{-1} \nabla_{\boldsymbol{w}^{(t)}} \mathcal{L}_{atk}(\boldsymbol{w}^{(t)}), \tag{5.16}$$

where the gradient $\nabla_{\boldsymbol{w}^{(t)}} \mathcal{L}_{atk}(\boldsymbol{w}^{(t)})$ is computed with respect to the t-th iterative perturbation vector $\boldsymbol{w}^{(t)}$, initialized as $\boldsymbol{w}^{(0)} = \boldsymbol{w}$. The matrix $\mathcal{H}_t^{-1}$ provides a low-rank approximation of the inverse Hessian matrix at iteration t, as employed in the L-BFGS algorithm to balance computational efficiency and the use of second-order information. At each step, L-BFGS updates $\mathcal{H}_t^{-1}$ using a recursive rule that incorporates curvature information derived from successive differences in weights ($\boldsymbol{p}_t$) and gradients ($\boldsymbol{q}_t$):

$$\mathcal{H}_t^{-1} = \mathcal{H}_{t-1}^{-1} + \frac{(\boldsymbol{p}_t^\top \boldsymbol{q}_t + \boldsymbol{q}_t^\top \mathcal{H}_{t-1}^{-1} \boldsymbol{q}_t)(\boldsymbol{p}_t \boldsymbol{p}_t^\top)}{(\boldsymbol{p}_t^\top \boldsymbol{q}_t)^2} - \frac{(\mathcal{H}_{t-1}^{-1} \boldsymbol{q}_t \boldsymbol{p}_t^\top + \boldsymbol{p}_t \boldsymbol{q}_t^\top \mathcal{H}_{t-1}^{-1})}{\boldsymbol{p}_t^\top \boldsymbol{q}_t}, \tag{5.17}$$

where $\boldsymbol{p}_t = \boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}$ and $\boldsymbol{q}_t = \nabla_{\boldsymbol{w}^{(t+1)}} \mathcal{L}_{atk}(\boldsymbol{w}^{(t+1)}) - \nabla_{\boldsymbol{w}^{(t)}} \mathcal{L}_{atk}(\boldsymbol{w}^{(t)})$. This recursive approach balances computational efficiency, memory usage, and the incorporation of second-order curvature information. We denote the final optimized perturbation vector as $\boldsymbol{w}_a$. It minimizes the performance of target models while ensuring that the perturbations remain subtle.

As a final step, we can achieve a transfer black-box attack on different target fraud detection models. To ensure subtle and controlled perturbations, we first normalize $\boldsymbol{w}_a$ to derive a scaled vector $\bar{\boldsymbol{w}}_a$. During the attack, when a feature matrix $\boldsymbol{H}$ is about to be input into the target model, we perturb the perturbable dimensions as follows:

$$\boldsymbol{H} = \boldsymbol{H} - \delta \cdot (\boldsymbol{M} \odot \mathrm{PL}(\bar{\boldsymbol{w}}_a)), \tag{5.18}$$

where δ is a parameter controlling the strength of the attack. By leveraging the carefully designed and imperceptible perturbation vector $\boldsymbol{w}_a$, the target models' ability to detect fraudulent transactions is significantly diminished, which is accomplished without relying on access to nearly all of the target models' internal information.

TABLE 5.2: Dataset statistics

	# Records	# Features	# Fraudulent	# Normal loans	Imbalance rate
LC	490,477	66	102,684	387,793	3.776566943
IEEE	590,540	148	20,663	569,877	27.5795867
IEEE-80K	80,000	148	2,125	77,875	36.64705882

5.6 Experiments

In this section, we conduct extensive experiments to validate the effectiveness and robustness of the proposed attack model. Specifically, we first present the experimental setup, including the details of the datasets, baseline methods, and evaluation metrics. Subsequently, we perform comprehensive evaluations, including comparisons with state-of-the-art baselines, ablation studies, and parameter sensitivity analyses, to demonstrate the superiority of our approach.

5.6.1 Experimental Setup

Datasets. To ensure a fair comparison with baselines [14], we employ identical datasets and preprocessing procedures. The detailed descriptions of these datasets are as follows:

- **LendingClub (LC) dataset** [1]**.** The LC dataset, collected from the largest U.S. peer-to-peer lending platform between 2015Q1 and 2016Q3, contains 490,477 transactions, of which 102,684 are identified as fraudulent. Our preprocessing pipeline removes features with more than 60% missing values and applies multiple imputation (mode for categorical/discrete variables and mean for continuous variables) to handle the remaining missing data.
- **IEEE-CIS Vesta Credit (IEEE) dataset** [2]**.** The IEEE dataset originates from the 2019 IEEE Computational Intelligence Society competition. Compared to LC, it exhibits a significantly higher class imbalance, with 590,451 transactions and only 20,663 identified as fraudulent. The dataset includes fundamental transaction-related attributes, such as transaction amount, product category, transaction location, and transaction frequency, along with engineered features developed by Vesta Corporation, including credit ranking, frequency-related attributes, and network-based features. To evaluate our model's performance on

[1] https://github.com/INFORMSJoC/2021.0076/tree/main/data
[2] https://www.kaggle.com/competitions/ieee-fraud-detection

a smaller dataset with imbalanced labels, we construct an additional benchmark dataset (IEEE-80K) by sampling 80,000 transactions from the original IEEE dataset.

The statistics of these datasets are summarized in 5.2. For attack simulations, we configure $|\boldsymbol{H}_{sur}| : |\boldsymbol{H}_{atk}| = 9 : 1$ and assume the first 10 feature dimensions in the ranking are perturbable. Additionally, we apply random under-sampling to these datasets, using a sampling ratio of 0.3. After training the target models, we limit the number of frauds to 100 to simulate real situations. These curated datasets enable the standardized evaluation of CCFD model robustness against black-box attacks.

Baselines. We consider four baseline methods in our experiments: RBF-classic, Linear-classic, STBA-RBF, and STBA-Linear. The first two, RBF-classic and Linear-classic, are based on the traditional transfer-based black-box attack approach introduced by [150]. These methods train a substitute model using a small set of labeled data and generate adversarial examples as the transferability of adversarial samples. The key difference between them lies in the type of substitute model: RBF-classic employs an SVM with a radial basis function (RBF) kernel, while Linear-classic uses a Linear-SVM, representing nonlinear and linear models, respectively. The latter two methods, STBA-RBF and STBA-Linear, are state-of-the-art baselines [14]. STBA addresses the challenge of limited labeled data by incorporating semi-supervised learning techniques to augment the labeled dataset with a large pool of unlabeled samples. This improves the quality of the substitute model without requiring extensive querying of the target model. Similar to the classic baselines, STBA-RBF and STBA-Linear differ in their choice of substitute model: RBF-SVM and Linear-SVM, respectively.

Evaluation metric. To ensure consistency with prior research, we adopt the Security Evaluation Index (SEI) to measure model robustness. The SEI quantifies the resilience of a CCFD model against adversarial attacks by computing the normalized area under the security evaluation curve. Let $\mathrm{Acc}(\rho)$ denote the classification accuracy of a target model on an adversarial dataset D_{adv} under an attack strength ρ. The SEI is formally defined as:

$$\mathrm{SEI} = \frac{\int_0^{\rho_{\max}} \mathrm{Acc}(\rho)\, d\rho}{\rho_{\max}}, \tag{5.19}$$

where $\rho_{\max}$ represents the maximum attack strength considered in the evaluation. Since the exact functional form of $\mathrm{Acc}(\rho)$ is unknown, we approximate it numerically using discrete observations:

$$\mathrm{SEI} \approx \frac{1}{2\rho_{\max}} \sum_{i=1}^{t-1} (\rho_{i+1} - \rho_i) \cdot (\mathrm{Acc}(\rho_{i+1}) + \mathrm{Acc}(\rho_i)), \tag{5.20}$$

where t denotes the number of attack experiments, and ρ_i represents the attack strength at the i-th experiment, satisfying $\rho_1 < \rho_2 < \cdots < \rho_t = \rho_{\max}$. A higher SEI indicates greater robustness, while a lower SEI suggests a stronger attack effect, making it a reliable metric for evaluating adversarial attack performance.

Parameter settings. The default parameter configurations for our model are as follows: $\alpha = 0.3$, $|\boldsymbol{H}_t| : |\boldsymbol{H}_a| = 7 : 3$, $m = 6$ base classifiers, $d' = 10$ attack dimensions, $\beta = 0.1$, and set the fraud rate in $\mathcal{D}_a$ to 0.05. To ensure statistical reliability, each experiment is repeated 10 times with different random seeds, evaluating 20 discrete attack strengths ($\delta \in [0.1, 2.0]$ with uniform intervals) to comprehensively validate model robustness. For the surrogate model, we adopt the Adam optimizer with a learning rate of 0.05. On the LC dataset, the model is trained for 500 epochs with a maximum of 5 optimization iterations, while on the IEEE dataset, it is trained for 50 epochs with a maximum of 10 optimization iterations. For the IEEE-80K dataset, we reduce the learning rate to 0.01, train the model for 250 epochs, and limit the maximum optimization iterations to 10 to ensure model performance.

Experimental environment. Consistent with prior research, we evaluated the efficacy of the attack models against eight distinct target models. These include Logistic Regression (LR) and Deep Neural Networks (DNN) as differentiable models, Decision Tree (DT), Linear Support Vector Machines (Linear-SVM), Radial Basis Function Support Vector Machines (RBF-SVM), and Extreme Gradient Boosting (XGBoost) as non-differentiable models, as well as Isolation Forest (IF) and Denoising Autoencoders (DAE) representing unsupervised learning approaches. This selection ensures a comprehensive assessment of various classification paradigms, covering both traditional and anomaly detection techniques. All experiments use the Scikit-learn implementation of LinearSVC, which leverages the liblinear library optimized for large-scale data. By default, it employs the One-vs-Rest (OvR) strategy for multi-class classification while maintaining equivalence to standard Linear-SVM in binary cases. The hardware configuration includes an NVIDIA A2 GPU with 16GB VRAM, an Intel Xeon E-2488 CPU, and 128GB system memory.

TABLE 5.3: SEIs of target models under attacks on dataset LC

	RBF-classic	Linear-classic	STBA-RBF	STBA-Linear	BADA	Promote
LR	0.09075	0.00850	0.23510	0.00833	0.00750	9.91%
DT	0.13500	0.09875	0.24100	0.08680	0.05875	32.32%
XGBoost	0.09975	0.00675	0.22005	0.00640	0.00600	6.25%
Linear-SVM	0.08350	0.00475	0.24133	0.00455	0.00325	28.57%
RBF-SVM	0.09050	0.00625	0.18620	0.00630	0.00625	0.00%
DNN	0.10975	0.10025	0.20010	0.10235	0.07983	20.37%
IF	0.74600	0.50375	0.64043	0.47610	0.45025	5.43%
DAE	0.32193	0.19315	0.37825	0.18923	0.18913	0.05%

5.6.2 Effectiveness of Attack Algorithms

To evaluate the effectiveness of the proposed attack model, in this section, we compare its performance against state-of-the-art baselines across multiple fraud detection tasks. The results, as summarized in 5.3 5.4 5.5, indicate that BADA consistently outperforms the baselines across different datasets and target models. It should be noted that the "Promote" column in these tables represents the improvement in attack effectiveness of BADA compared to the second-best model. Quantitatively, BADA achieves an improvement of up to 63.98% in SEI metric compared to the best-performing baseline, highlighting its superior capability in perturbing fraud detection systems.

Analysis of experimental results on LC. 5.3 presents the attack performance of BADA and baseline methods against different target models on the LC dataset. The results show that target models achieve the lowest SEI under the perturbation of BADA, demonstrating BADA's superior attack capability. SEI quantifies the robustness of a target model under attack, with lower values indicating stronger attack performance. The "Promote" column highlights the percentage improvement of BADA over the second-best baseline, further validating its effectiveness. Compared to the best baseline method, BADA reduces SEI by 32.32% on DT and achieves an average reduction of 12.86% across all models in this dataset. Among differentiable models, BADA is particularly effective against LR and DNN, with reductions of 9.91% and 20.37%, respectively, compared to the second-best baseline. This highlights the advantage of the strategy of BADA. For non-differentiable models, BADA achieves the largest SEI reduction of 32.32% on DT, demonstrating its ability to exploit broader and more refined decision boundaries. Linear-SVM, characterized by its rigid margin-based decision boundary, is also highly vulnerable to BADA, which significantly

outperforms all baseline methods on this model. The attack performance improvement of BADA on RBF-SVM on the LC dataset is limited compared to the second-best attack method. This is primarily because RBF-SVM, when applied to datasets with relatively balanced labels, tends to construct highly localized and flexible decision boundaries using its radial basis function kernel. While this characteristic allows RBF-SVM to effectively separate classes in balanced datasets, it also makes the model highly sensitive to adversarial perturbations, as small input changes can lead to significant shifts in its decision boundaries. Consequently, BADA demonstrates limited additional improvement on LC, as the baseline methods have already leveraged the inherent weaknesses of RBF-SVM in adversarial scenarios. Similarly, under the attack of BADA, XGBoost achieves an SEI reduction of only 6.25% compared to the second-best attack method on the LC dataset. This is because, although XGBoost leverages multiple classifiers, all of its base classifiers share the same structure, making them susceptible to simultaneous attacks. As a result, existing attack methods are already capable of achieving relatively effective results against XGBoost, leaving limited room for BADA to demonstrate significant improvement. For unsupervised models, BADA achieves SEI reductions of 5.43% and 0.05% for IF and DAE. These results indicate that label-based attack strategies are generally less effective on unsupervised algorithms. Nonetheless, BADA's stronger generalization capabilities enable it to outperform other baselines, demonstrating its adaptability even in challenging scenarios. Overall, BADA achieves notable improvements over other baselines across all target models, demonstrating its superior attack performance. It validate the effectiveness of BADA's design, which combines adversarial learning-based optimization and robust initialization strategies to achieve state-of-the-art performance in black-box attack scenarios.

Analysis of experimental results on LEEE and IEEE-80K. 5.4 5.5 summarizes the attack performance of BADA and baseline methods across IEEE and IEEE-80K. The overall trend aligns with the results observed on the LC dataset. Due to IEEE and IEEE-80K having similar label distributions, the experimental results also tend to be consistent. BADA still outperforms all baseline methods on both datasets, achieving the best attack performance. On average, it achieves a 27.8% SEI reduction compared to the second-best model, demonstrating its effectiveness across datasets with different distributions. For differentiable target models, it can be observed that under BADA's attack, the robustness of LR and DNN decreases by 61.3% and 49.61%, respectively, compared to the existing best attack method. This further validates that BADA's optimization

TABLE 5.4: SEIs of target models under attacks on dataset IEEE

	RBF-classic	Linear-classic	STBA-RBF	STBA-Linear	BADA	Promote
LR	0.35975	0.30125	0.39865	0.38195	0.12463	58.63%
DT	0.60100	0.65750	0.64458	0.70903	0.56480	6.02%
XGBoost	0.57225	0.54575	0.57320	0.55100	0.48765	10.65%
Linear-SVM	0.38400	0.22200	0.42225	0.34890	0.12538	43.52%
RBF-SVM	0.31825	0.29200	0.33748	0.32765	0.12885	55.87%
DNN	0.50550	0.47250	0.55770	0.57198	0.21305	54.91%
IF	0.52750	0.57600	0.53548	0.56645	0.56420	-6.96%
DAE	0.40453	0.41138	0.39660	0.41430	0.21303	46.29%

TABLE 5.5: SEIs of target models under attacks on dataset IEEE-80K

	RBF-classic	Linear-classic	STBA-RBF	STBA-Linear	BADA	Promote
LR	0.42000	0.43775	0.46980	0.39178	0.14113	63.98%
DT	0.56550	0.58150	0.60158	0.61055	0.55923	1.11%
XGBoost	0.53175	0.57875	0.58055	0.58215	0.51453	3.24%
Linear-SVM	0.41925	0.42750	0.48915	0.25665	0.24485	4.60%
RBF-SVM	0.27700	0.29450	0.30088	0.29050	0.16858	39.14%
DNN	0.45950	0.46550	0.53590	0.36418	0.20283	44.31%
IF	0.56075	0.53950	0.52645	0.55400	0.63415	-20.46%
DAE	0.44038	0.46833	0.44400	0.45405	0.26455	39.93%

framework, which incorporates both decision boundary refinement and adversarial learning, can exploit weaknesses in differentiable classifiers more effectively than existing attacks. For non-differentiable target models, the most significant improvement over baseline methods is observed on RBF-SVM and Linear-SVM, with an average SEI reduction of 47.51% and 24.06%, respectively. This confirms that SVM-based models are particularly vulnerable to BADA when subjected to a more adaptive optimization process on imbalanced datasets, as their rigid decision boundaries are more easily exploited by adversarial perturbations. For unsupervised models, BADA achieves an average SEI reduction of 43.11% on DAE. This indicates that DAE has a lower ability to defend against BADA's attack strategies, particularly on datasets with lower fraud ratios. BADA shows a certain degree of performance degradation in attacks on IF on these two datasets. This is because IF heavily relies on the sparsity of fraud nodes for detection. Due to the limited dimensions of perturbable features, the perturbation vectors generated by BADA, which are highly effective against other target models, tend to focus on non-critical features for IF. This inadvertently enhances IF's ability to isolate sparse anomalies, reducing the attack's overall effectiveness.

5.6.3 Ablation Study

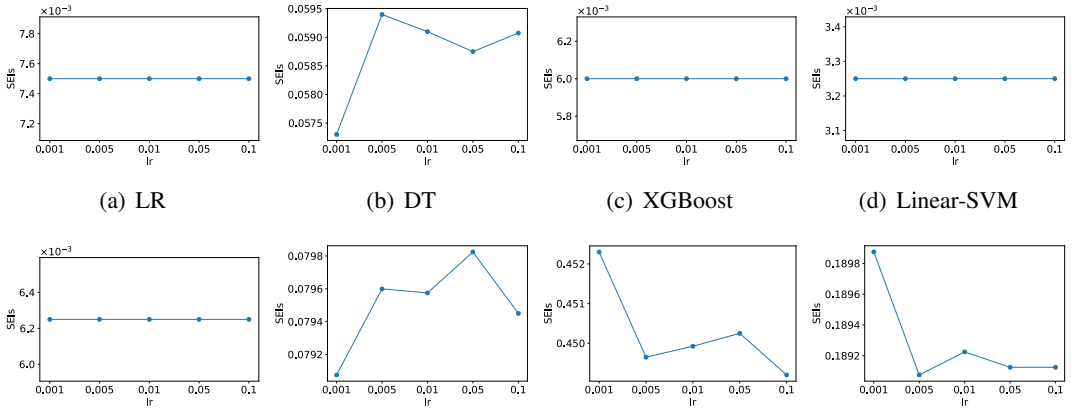
In this section, we conduct an ablation study to evaluate the contribution of each component in BADA. 5.6 presents the results of this study on the LC dataset, highlighting the impact of different components on attack performance. As shown in the table, the full BADA model achieves the lowest SEI for most of these target models, demonstrating the critical role of each component in enhancing attack effectiveness. The results of STBA w/ W reflect the attack performance when using $\boldsymbol{w}_i$ as the perturbation vector. While the performance degrades across all target models compared to the full BADA, it still surpasses the state-of-the-art baseline STBA-Linear. The variant BADA-W further investigates the impact of $\boldsymbol{w}_i$, representing a version of BADA where $\boldsymbol{w}_i$ is initialized using only a single model. A significant gap in attack effectiveness between BADA and this variant is observed, underscoring the importance of leveraging multiple classifiers to construct a more effective attack model. These findings suggest that BADA relies on the proposed initialization strategy for $\boldsymbol{w}_i$ to achieve optimal performance. The performance of the variant BADA-S evaluates the impact of replacing the Memory-based Adversarial Defense module with a standard MLP. The results reveal a significant drop in performance, with SEI increasing across all target models compared to the complete BADA. This highlights the critical role of the Memory-based Adversarial Defense module in constructing a robust surrogate model and enhancing attack effectiveness. The variant BADA-WS disables both the Memory-based Adversarial Defense module and the proposed initialization strategy for the perturbation vector. This configuration results in worse performance compared to BADA-W and BADA-S, approaching the baseline STBA-Linear. These findings reinforce the conclusion that both the Memory-based Adversarial Defense module and the initialization strategy are essential for achieving superior attack performance. Overall, these results demonstrate that each component of BADA plays a critical role in achieving superior attack performance. The best results are obtained when all components are combined, highlighting the synergy between each component in BADA.

5.6.4 Parameter Sensitivity Analysis

To further analyze the performance of our model and its sensitivity to key hyperparameters, we conducted a series of parameter sensitivity experiments. Specifically, we investigate the impact of

TABLE 5.6: Ablation study on LC

	STBA-Linear	STBA w/ W	BADA-W	BADA-S	BADA-WS	BADA
LR	0.00833	0.00828	0.00763	0.00783	0.00823	0.00750
DT	0.08680	0.08508	0.07590	0.07128	0.06255	0.05875
XGBoost	0.00640	0.00613	0.00630	0.00635	0.00670	0.00600
Linear-SVM	0.00455	0.00415	0.00340	0.00370	0.00415	0.00325
RBF-SVM	0.00630	0.00625	0.00625	0.00648	0.00688	0.00625
DNN	0.10235	0.09740	0.08673	0.08398	0.08213	0.07983
IF	0.47610	0.46823	0.48915	0.52700	0.56695	0.45025
DAE	0.18923	0.18990	0.18750	0.19065	0.19635	0.18913



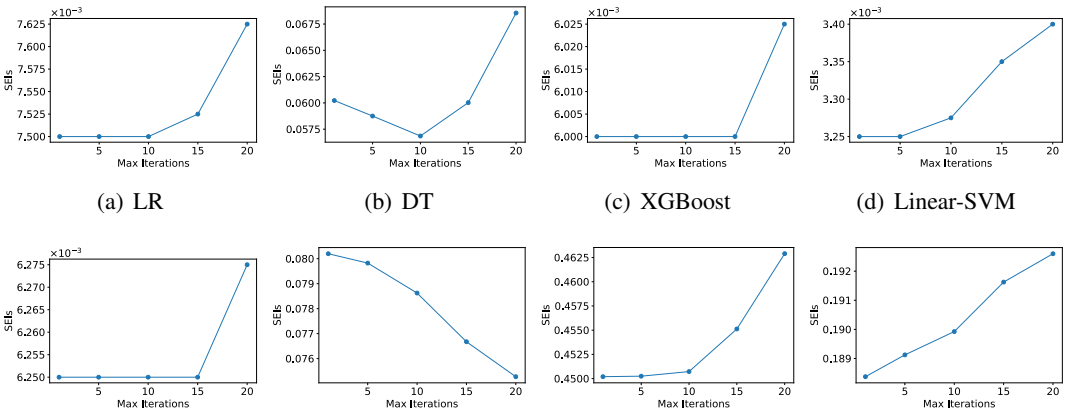
(a) LR (b) DT (c) XGBoost (d) Linear-SVM

(e) RBF-SVM (f) DNN (g) IF (h) DAE

FIGURE 5.2: SEIs results for Learning Rate

the learning rate and the maximum optimization iterations of the surrogate model. These hyperparameters are critical for balancing training stability, efficiency, and attack effectiveness.

Learning rate sensitivity analysis. The learning rate is a critical hyperparameter that affects both the stability and efficiency of model training. A lower learning rate ensures stable convergence but may prolong training and yield suboptimal results, while a higher learning rate accelerates learning but risks instability. Understanding its impact on the surrogate model is essential for improving its simulation of target models in fraud detection and optimizing the attack performance of BADA. From the experimental results in 5.2, we observe notable variations in attack performance under different learning rates. For differentiable models, distinct patterns emerge. The attack performance against LR remains relatively stable, indicating that the surrogate model's learning rate has minimal impact on this simple optimization-based model. In contrast, DNN exhibits fluctuations,



(a) LR (b) DT (c) XGBoost (d) Linear-SVM

(e) RBF-SVM (f) DNN (g) IF (h) DAE

FIGURE 5.3: SEIs results for Max Iterations

suggesting that an inappropriate learning rate can degrade the surrogate model's ability to approximate deep networks, thereby reducing attack effectiveness. For non-differentiable models, DT shows clear sensitivity to the learning rate, with attack performance worsening at higher values. This indicates that reducing the optimization amplitude of the surrogate model can improve attack results for this target model. In contrast, Linear-SVM, RBF-SVM, and XGBoost exhibit minimal variation, suggesting that attacks against these models are largely unaffected by the surrogate model's learning rate. For unsupervised models, including IF and DAE, a significant drop in SEIs at lower learning rates highlights the importance of a properly tuned surrogate model for enhancing attack efficacy. Overall, these results emphasize that while some target models are robust to variations in the surrogate model's learning rate, attacks against DNN and DT require careful tuning of this hyperparameter. Based on these findings, we set the learning rate of BADA's surrogate model to 0.05 for the LC dataset to ensure strong performance across all target models.

Maximum iterations sensitivity analysis. The maximum number of iterations is a key hyperparameter that controls the second-order derivative-based optimization process for refining $\boldsymbol{w}$. Setting this parameter appropriately is crucial for ensuring effective attack performance. A small iteration count may result in insufficient optimization, while an excessively large count may cause over-adjustment, potentially degrading attack effectiveness. From the experimental results in 5.3, we observe a general trend where increasing the number of iterations leads to worse attack performance across most target models. This suggests that excessive iterations may introduce optimization instability or overfit the surrogate model to behaviors that do not generalize well for

adversarial attacks. However, for DNN, increasing iterations results in a steady improvement in attack effectiveness. This indicates that extended updates allow the surrogate model to better approximate the complex decision boundaries of such target models, leading to more effective adversarial strategies. Given these observations, selecting a relatively small number of iterations ensures consistently strong attack performance across all target models. Based on these findings, we set the maximum number of iterations to 5 for the LC dataset, balancing optimization efficiency and attack effectiveness.

5.7 Conclusion

This chapter presents BADA, a black-box fraud detection attack model designed to evaluate and expose vulnerabilities in fraud detection systems. By generating adaptive perturbations, BADA effectively challenges the robustness of these systems while demonstrating strong adaptability across various fraud detection environments. Through extensive experiments on three real-world datasets and eight fraud detection models, we validate the effectiveness of BADA, which achieves significant improvements over existing baselines. These results emphasize the urgent need to address adversarial threats in the financial domain and underscore the importance of developing more resilient fraud detection mechanisms. The insights provided by BADA not only deepen our understanding of adversarial attacks but also offer valuable guidance for future research aimed at enhancing the robustness and reliability of fraud detection systems.

Chapter 6

EPILOGUE

In this thesis, we explored and addressed the core challenges of financial fraud detection by leveraging advanced deep graph learning techniques, focusing on data imbalance, heterogeneous information utilization, and model robustness in real-world financial scenarios.

We began with Chapter 1, which established the research background, clarified the significance of combating evolving financial fraud, and outlined the thesis structure and key objectives. Chapter 2 systematically synthesized existing research on fraud detection—covering traditional machine learning, basic graph-based methods, and early deep graph learning approaches—while identifying critical research gaps: the underutilization of semi-supervised learning for imbalanced fraud data, the failure to fully exploit heterogeneous financial entities and relations, and the lack of attention to model vulnerability against adversarial attacks.

Building on these gaps, Chapter 3 proposed a semi-supervised deep graph learning framework for credit card fraud detection, which leverages limited labeled data and large-scale unlabeled transactions via graph contrastive learning to mitigate data imbalance. Chapter 4 developed a specialized heterogeneous graph neural network, constructing a multi-entity (users, accounts, merchants) and multi-relation (transactions, bindings) financial graph with a tailored aggregation mechanism to capture complex interactive fraud patterns. Chapter 5 focused on black box attacks against credit card fraud detection models, analyzing system vulnerabilities and proposing a scenario-adapted attack method to guide the design of more robust detection systems. A central focus of this thesis was the alignment of technical innovations with practical financial needs. Our methods addressed

industry pain points: the semi-supervised approach solves the common data scarcity of fraud cases, the heterogeneous GNN unlocks value from fragmented financial data, and the black box attack analysis fills the gap in model robustness evaluation for fraud detection.

The works presented in this thesis, reflected in a series of academic publications, have demonstrated superior performance compared to state-of-the-art methods in terms of detection accuracy, adaptability to complex financial scenarios, and robustness against adversarial interference. Looking ahead, the field of deep graph learning for financial fraud detection holds substantial potential for further exploration. Future research directions include integrating multimodal financial data (e.g., text from user profiles, behavioral logs), enhancing model interpretability to meet regulatory compliance requirements, developing real-time adaptive detection systems for dynamic fraud tactics, and exploring cross-scenario transferability of fraud detection models. The current work lays a solid foundation for these endeavors, pushing the boundaries of practical application of deep graph learning in financial risk management.

In conclusion, this thesis makes significant contributions to the intersection of deep graph learning and financial fraud detection, advancing the state of the art in addressing complex, real-world fraud challenges and providing actionable insights for both academic research and industrial practice.

Bibliography

[1] Latifa AlFalahi and Haitham Nobanee. Conceptual building of sustainable economic growth and corporate bankruptcy. *Available at SSRN 3472409*, 2019.

[2] Domicián Máté, Rabeea Sadaf, Judit Oláh, József Popp, and Edit Szűcs. The effects of accountability, governance capital, and legal origin on reported frauds. *Technological and Economic Development of Economy*, 25(6):1213–1231, 2019.

[3] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian K. Tharakunnel, and J. Christopher Westland. Data mining for credit card fraud: A comparative study. *Decis. Support Syst.*, 50: 602–613, 2011.

[4] Chloe Tergiman and Marie Claire Villeval. The way people lie in markets: Detectable vs. deniable lies. *Management Science*, 69(6):3340–3357, 2023.

[5] Yidi Liu, Xin Li, and Zhiqiang Zheng. Smart natural disaster relief: Assisting victims with artificial intelligence in lending. *Information Systems Research*, 2023.

[6] Chen Jin, Luyi Yang, and Kartik Hosanagar. To brush or not to brush: Product rankings, consumer search, and fake orders. *Information Systems Research*, 34(2):532–552, 2023.

[7] Fatemeh Delkhosh, Ram D Gopal, Raymond A Patterson, and Niam Yaraghi. Impact of bot involvement in an incentivized blockchain-based online social media platform. *Journal of Management Information Systems*, 40(3):778–806, 2023.

[8] Hengshu Zhu, Hui Xiong, Yong Ge, and Enhong Chen. Discovery of ranking fraud for mobile apps. *IEEE Transactions on knowledge and data engineering*, 27(1):74–87, 2014.

[9] Yiangos Papanastasiou, S Alex Yang, and Angela Huyue Zhang. Improving dispute resolution in two-sided platforms: The case of review blackmail. *Management Science*, 2023.

[10] KR Seeja and Masoumeh Zareapoor. Fraudminer: A novel credit card fraud detection model based on frequent itemset mining. *The Scientific World Journal*, 2014, 2014.

[11] EMSW Balagolla, WPC Fernando, RMNS Rathnayake, MJMRP Wijesekera, AN Senarathne, and KY Abeywardhana. Credit card fraud prevention using blockchain. In *2021 6th international conference for Convergence in Technology (I2CT)*, pages 1–8. IEEE, 2021.

[12] Markus Weinmann, Joseph Valacich, Christoph Schneider, Jeffrey L Jenkins, and Martin Thomas Hibbeln. The path of the righteous: Using trace data to understand fraud decisions in real time. *MIS Quarterly*, 2021.

[13] Ugo Fiore, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 2017.

[14] Jin Xiao, Yuhang Tian, Yanlin Jia, Xiaoyi Jiang, Lean Yu, and Shouyang Wang. Black-box attack-based security evaluation framework for credit card fraud detection models. *INFORMS Journal on Computing*, 2023.

[15] Jennifer J Xu, Dongyu Chen, Michael Chau, Liting Li, and Haichao Zheng. Peer-to-peer loan fraud detection: Constructing features from transaction data. *MIS quarterly*, 46(3), 2022.

[16] Fan Zhou, Guanyu Wang, Kunpeng Zhang, Siyuan Liu, and Ting Zhong. Semi-supervised anomaly detection via neural process. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[17] Yi Yang, Kunpeng Zhang, and Yangyang Fan. Analyzing firm reports for volatility prediction: A knowledge-driven text-embedding approach. *INFORMS Journal on Computing*, 34 (1):522–540, 2022.

[18] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, pages 21076–21089. PMLR, 2022.

[19] Keli Xiao, Qi Liu, Chuanren Liu, and Hui Xiong. Price shock detection with an influence-based model of social attention. *ACM Transactions on Management Information Systems (TMIS)*, 9(1):1–21, 2017.

[20] Jinwen Sun, Keli Xiao, Chuanren Liu, Wenjun Zhou, and Hui Xiong. Exploiting intra-day patterns for market shock prediction: A machine learning approach. *Expert Systems with Applications*, 127:272–281, 2019.

[21] Dawei Cheng, Yujia Ye, Sheng Xiang, Zhenwei Ma, Ying Zhang, and Changjun Jiang. Anti-money laundering by group-aware deep graph learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–13, 2023.

[22] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *AAAI*, 2023.

[23] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, pages 483–490. Springer, 2016.

[24] D. Cheng, X. Wang, Y. Zhang, and L. Zhang. Graph neural network for fraud detection via spatial-temporal attention. *IEEE TKDE*, pages 1–1, 2020.

[25] Rongrong Jing, Hu Tian, Gang Zhou, Xingwei Zhang, Xiaolong Zheng, and Daniel Dajun Zeng. A gnn-based few-shot learning model on the credit card fraud detection. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pages 320–323. IEEE, 2021.

[26] Yangming Zhou, Jiaqi Li, Jin-Kao Hao, and Fred Glover. Detecting critical nodes in sparse graphs via "reduce-solve-combine" memetic search. *INFORMS Journal on Computing*, 2023.

[27] Dawei Cheng, Zhibin Niu, and Yiyi Zhang. Contagious chain risk rating for networked-guarantee loans. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[28] Enxia Li, Mengshi Chen, Sheng Xiang, and Ling Chen. Graph-learning-empowered financial fraud detection: Progress and future directions. *Intelligent Computing*.

[29] Xiangfeng Li, Shenghua Liu, Zifeng Li, Xiaotian Han, Chuan Shi, Bryan Hooi, He Huang, and Xueqi Cheng. Flowscope: Spotting money laundering based on graphs. In *AAAI Conference on Artificial Intelligence*, 2020.

[30] Tianyi Chen and Charalampos Tsourakakis. Antibenford subgraphs: Unsupervised anomaly detection in financial networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2762–2770, 2022.

[31] Wenbo Zheng, Lang Yan, Chao Gou, and Fei-Yue Wang. Federated meta-learning for fraudulent credit card detection. In *International Joint Conference on Artificial Intelligence*, 2020.

[32] Daixin Wang, Yuan Qi, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, and Shuang Yang. A semi-supervised graph attentive network for financial fraud detection. *2019 IEEE International Conference on Data Mining (ICDM)*, pages 598–607, 2019.

[33] Cen Chen, Chen Liang, Jianbin Lin, Li Wang, Ziqi Liu, Xinxing Yang, Jun Zhou, Shuang Yang, and Yuan Qi. Infdetect: a large scale graph-based fraud detection system for e-commerce insurance. *2019 IEEE International Conference on Big Data (Big Data)*, pages 1765–1773, 2019.

[34] Mark Weber, Jie Chen, Toyotaro Suzumura, Aldo Pareja, Tengfei Ma, Hiroki Kanezashi, Tim Kaler, Charles E. Leiserson, and Tao B. Schardl. Scalable graph learning for anti-money laundering: A first look. In *NeurIPS*, Worksop on Challenges and Opportunities for AI in Financial Services, 2018.

[35] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoît Dupont. Ransomware payments in the bitcoin ecosystem. *ArXiv*, abs/1804.04080, 2018.

[36] Daixin Wang, Zhiqiang Zhang, Jun Zhou, Peng Cui, Jingli Fang, Quanhui Jia, Yanming Fang, and Yuan Qi. Temporal-aware graph neural network for credit risk prediction. In *SDM*, 2021.

[37] Dawei Cheng, Yi Tu, Zhenwei Ma, Zhibin Niu, and Liqing Zhang. Risk assessment for networked-guarantee loans using high-order graph attention representation. In *IJCAI*, pages 5822–5828. AAAI Press, 2019.

[38] Dawei Cheng, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Risk guarantee prediction in networked-loans. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

[39] Shuo Yang, Zhiqiang Zhang, Jun Zhou, Yang Wang, Wang Sun, Xingyu Zhong, Yanming Fang, Quan Yu, and Yuan Qi. Financial risk analysis for smes with graph-based supply chain mining. In *International Joint Conference on Artificial Intelligence*, 2020.

[40] Binbin Hu, Zhiqiang Zhang, Jun Zhou, Jingli Fang, Quanhui Jia, Yanming Fang, Quan Yu, and Yuan Qi. Loan default analysis with multiplex graph learning. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[41] Chen Liang, Ziqi Liu, Bin Liu, Jun Zhou, Xiaolong Li, Shuang Yang, and Yuan Qi. Uncovering insurance fraud conspiracy with network learning. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[42] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. *Proceedings of the Web Conference 2021*, 2021.

[43] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *Proceedings of The Web Conference 2020*, pages 785–795, 2020.

[44] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):946–953, 7 2019.

[45] Efstathios Kirkos, Charalambos Spathis, and Yannis Manolopoulos. Data mining techniques for the detection of fraudulent financial statements. *Expert Syst. Appl.*, 32:995–1003, 2007.

[46] Pediredla Ravisankar, Vadlamani Ravi, G. Raghava Rao, and Indranil Bose. Detection of financial statement fraud and feature selection using data mining techniques. *Decis. Support Syst.*, 50:491–500, 2011.

[47] Youngjoon Ki and Ji Won Yoon. Pd-fds: Purchase density based online credit card fraud detection system. In Archana Anandakrishnan, Senthil Kumar, Alexander Statnikov, Tanveer Faruquie, and Di Xu, editors, *Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance*, volume 71 of *Proceedings of Machine Learning Research*, pages 76–84. PMLR, 8 2018.

[48] Jiahui Wang, Yi Guo, Xinxiu Wen, Zhihong Wang, Zhen Li, and Minwei Tang. Improving graph-based label propagation algorithm with group partition for fraud detection. *Applied Intelligence*, 50:3291–3300, 2020.

[49] Alex Sangers, Maran van Heesch, Thomas Attema, Thijs Veugen, Mark Wiggerman, Jan Veldsink, Oscar Bloemen, and Daniël Worm. Secure multiparty pagerank algorithm for collaborative fraud detection. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, pages 605–623. Springer, 2019.

[50] Fatima Khalid, Ali Javed, Hafsa Ilyas, Aun Irtaza, et al. Dfgnn: An interpretable and generalized graph neural network for deepfakes detection. *Expert Systems with Applications*, 222:119843, 2023.

[51] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 598–607. IEEE, 2019.

[52] Yuanyuan Xu, Wenjie Zhang, Ying Zhang, Maria Orlowska, and Xuemin Lin. Timesgn: Scalable and effective temporal graph neural network. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 3297–3310. IEEE, 2024.

[53] Yuanyuan Xu, Wenjie Zhang, Xiwei Xu, Binghao Li, and Ying Zhang. Scalable and effective temporal graph representation learning with hyperbolic geometry. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[54] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3): 602–613, 2011.

[55] Raghavendra Patidar, Lokesh Sharma, et al. Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, 1(32-38), 2011.

[56] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 315–324, 2020.

[57] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[58] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, pages 3168–3177, 2021.

[59] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pages 261–270, 2002.

[60] Yusuf G Şahin and Ekrem Duman. Detecting credit card fraud by decision trees and support vector machines. 2011.

[61] Dawei Cheng, Sheng Xiang, Chencheng Shang, Yiyi Zhang, Fangzhou Yang, and Liqing Zhang. Spatio-temporal attention-based neural network for credit card fraud detection. In *AAAI*, pages 362–369, 2020.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[63] Zixing Song, Xiangli Yang, Zenglin Xu, and Irwin King. Graph-based semi-supervised learning: A comprehensive review. *IEEE transactions on neural networks and learning systems*, PP, 2022.

[64] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018.

[65] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. *arXiv preprint arXiv:1801.09927*, 2018.

[66] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[67] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[68] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[69] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2018.

[70] Sheng Xiang, Dong Wen, Dawei Cheng, Ying Zhang, Lu Qin, Zhengping Qian, and Xuemin Lin. General graph generators: experiments, analyses, and improvements. *The VLDB Journal*, 2021.

[71] Sheng Xiang, Dawei Cheng, Jianfu Zhang, Zhenwei Ma, Xiaoyang Wang, and Ying Zhang. Efficient learning-based community-preserving graph generation. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1982–1994, 2022.

[72] Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. Temporal and heterogeneous graph neural network for financial time series prediction. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022.

[73] Yunsheng Shi, Zhengjie Huang, Wenjin Wang, Hui Zhong, Shikun Feng, and Yu Sun. Masked label prediction: Unified massage passing model for semi-supervised classification. In *IJCAI*, 2021.

[74] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[75] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908, 2013.

[76] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2077–2085, 2018.

[77] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 310–316, 2019.

[78] Yiming Zhang, Yujie Fan, Yanfang Ye, Liang Zhao, and Chuan Shi. Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 549–558, 2019.

[79] Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *ArXiv*, abs/1909.12223, 2020.

[80] Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical science*, pages 235–249, 2002.

[81] Rahul Goyal and Amit Kumar Manjhvar. Review on credit card fraud detection using data mining classification techniques & machine learning algorithms. *Data Science & Analytics eJournal*, 2020.

[82] Jarrod West and Maumita Bhattacharya. Intelligent financial fraud detection: A comprehensive review. *Comput. Secur.*, 57:47–66, 2016.

[83] Samaneh Sorournejad, Zahra Zojaji, Reza Ebrahimi Atani, and Amir Hassan Monadjemi. A survey of credit card fraud detection techniques: Data and technique oriented perspective. *ArXiv*, abs/1611.06439, 2016.

[84] Christopher Whitrow, David J. Hand, Piotr Juszczak, David John Weston, and Niall M. Adams. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18:30–55, 2009.

[85] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)*, pages 1–9. IEEE, 2017.

[86] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, 2016.

[87] Clifton Phua, Vincent Cheng-Siong Lee, Kate Smith-Miles, and Ross W. Gayler. A comprehensive survey of data mining-based fraud detection research. *ArXiv*, abs/1009.6119, 2010.

[88] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[89] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001.

[90] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[91] Petar Velivckovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[92] Fayaz Itoo, Meenakshi, and Satwinder Singh. Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, 13:1503 – 1511, 2020.

[93] Petr Hájek, Mohammad Zoynul Abedin, and Uthayasankar Sivarajah. Fraud detection in mobile payment systems using an xgboost-based framework. *Information Systems Frontiers*, pages 1 – 19, 2022.

[94] Naoufal Rtayli and Nourddine Enneya. Enhanced credit card fraud detection based on svm-recursive feature elimination and hyper-parameters optimization. *J. Inf. Secur. Appl.*, 55: 102596, 2020.

[95] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Neural Information Processing Systems*, 2017.

[96] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, pages 21076–21089. PMLR, 2022.

[97] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4821–4830, 2019.

[98] Jinghan Shi, Houye Ji, Chuan Shi, Xiao Wang, Zhiqiang Zhang, and Jun Zhou. Heterogeneous graph neural network for recommendation. *ArXiv*, abs/2009.00799, 2020.

[99] Rui Bing, Guan Yuan, Mu Zhu, Fanrong Meng, Huifang Ma, and Shaojie Qiao. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artificial Intelligence Review*, 56(8):8003–8042, 2023.

[100] Jun Hu, Bryan Hooi, and Bingsheng He. Efficient heterogeneous graph learning via random projection. *ArXiv*, abs/2310.14481, 2023.

[101] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[102] Julian J. McAuley and Jure Leskovec. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. *CoRR*, abs/1303.4402, 2013.

[103] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[104] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. *CoRR*, abs/2104.06643, 2021.

[105] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[106] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion proceedings of the 2019 World Wide Web conference*, pages 310–316, 2019.

[107] Zheng Wang, Cheng Long, Gao Cong, and Ce Ju. Effective and efficient sports play retrieval with deep representation learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 499–509, 2019.

[108] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, and Le Song. Geniepath: Graph neural networks with adaptive receptive paths. *CoRR*, abs/1802.00910, 2018.

[109] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. *CoRR*, abs/2005.00625, 2020.

[110] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. *CoRR*, abs/2003.01171, 2020.

[111] Wenxuan Wang, Bangjie Yin, Taiping Yao, Li Zhang, Yanwei Fu, Shouhong Ding, Jilin Li, Feiyue Huang, and Xiangyang Xue. Delving into data: Effectively substitute training for black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4761–4770, 2021.

[112] Renyang Liu, Wei Zhou, Tianwei Zhang, Kangjie Chen, Jun Zhao, and Kwok-Yan Lam. Boosting black-box attack to deep neural networks with conditional diffusion models. *IEEE Transactions on Information Forensics and Security*, 2024.

[113] Mingxing Duan, Kailun Jiao, Siyang Yu, Zhibang Yang, Bin Xiao, and Kenli Li. Mc-net: realistic sample generation for black-box attacks. *IEEE Transactions on Information Forensics and Security*, 19:3008–3022, 2024.

[114] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[115] Mary Frances Zeager, Aksheetha Sridhar, Nathan Fogal, Stephen Adams, Donald E Brown, and Peter A Beling. Adversarial learning in credit card fraud detection. In *2017 Systems and Information Engineering Design Symposium (SIEDS)*, pages 112–116. IEEE, 2017.

[116] Jin Xiao, Yanlin Jia, Xiaoyi Jiang, and Shouyang Wang. Circular complex-valued gmdh-type neural network for real-valued classification problems. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12):5285–5299, 2020.

[117] Namsik Chang and Olivia R Liu Sheng. Decision-tree-based knowledge discovery: single- vs. multi-decision-tree induction. *INFORMS Journal on Computing*, 20(1):46–54, 2008.

[118] J Paul Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations research*, 59(2):467–479, 2011.

[119] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[120] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[121] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[122] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International conference on machine learning*, pages 2484–2493. PMLR, 2019.

[123] Junhao Zheng, Chenhao Lin, Jiahao Sun, Zhengyu Zhao, Qian Li, and Chao Shen. Physical 3d adversarial attacks against monocular depth estimation in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24452–24461, 2024.

[124] Xuanming Cui, Alejandro Aparcedo, Young Kyun Jang, and Ser-Nam Lim. On the robustness of large multimodal models against image adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24625–24634, 2024.

[125] Yanbo Chen and Weiwei Liu. A theory of transfer-based black-box attacks: Explanation and implications. *Advances in Neural Information Processing Systems*, 36:13887–13907, 2023.

[126] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *CoRR*, abs/1612.06299, 2016.

[127] Sizhe Chen, Zhehao Huang, Qinghua Tao, Yingwen Wu, Cihang Xie, and Xiaolin Huang. Adversarial attack on attackers: Post-process to mitigate black-box score-based query attacks. *Advances in Neural Information Processing Systems*, 35:14929–14943, 2022.

[128] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[129] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7714–7722, 2019.

[130] Ruikui Wang, Yuanfang Guo, and Yunhong Wang. Ags: Affordable and generalizable substitute training for transferable adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5553–5562, 2024.

[131] Keke Tang, Zhensu Wang, Weilong Peng, Lujie Huang, Le Wang, Peican Zhu, Wenping Wang, and Zhihong Tian. Symattack: Symmetry-aware imperceptible adversarial attacks on 3d point clouds. In *ACM Multimedia 2024*, 2024.

[132] Jianping Zhang, Wenwei Gu, Yizhan Huang, Zhihan Jiang, Weibin Wu, and Michael R. Lyu. Curvature-invariant adversarial attacks for 3d point clouds. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 7142–7150. AAAI Press, 2024. doi: 10.1609/AAAI.V38I7.28542.

[133] Vinu Sankar Sadasivan, Shoumik Saha, Gaurang Sriramanan, Priyatham Kattakinda, Atoosa Chegini, and Soheil Feizi. Fast adversarial attacks on language models in one gpu minute. *arXiv preprint arXiv:2402.15570*, 2024.

[134] Elias Abad Rocamora, Yongtao Wu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Revisiting character-level adversarial attacks for language models. In *Forty-first International Conference on Machine Learning*, 2024.

[135] Zheng Fang, Tao Wang, Lingchen Zhao, Shenyi Zhang, Bowen Li, Yunjie Ge, Qi Li, Chao Shen, and Qian Wang. Zero-query adversarial attack on black-box automatic speech recognition systems. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors,

Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024, pages 630–644. ACM, 2024. doi: 10.1145/3658644.3670309.

[136] Guoming Zhang, Xiaohui Ma, Huiting Zhang, Zhijie Xiang, Xiaoyu Ji, Yanni Yang, Xiuzhen Cheng, and Pengfei Hu. Laseradv: Laser adversarial attacks on speech recognition systems. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association, 2024.

[137] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[138] Qingfeng Zeng, Li Lin, Rui Jiang, Weiyu Huang, and Dijia Lin. Nnensleg: A novel approach for e-commerce payment fraud detection using ensemble learning and neural networks. *Information Processing & Management*, 62(1):103916, 2025.

[139] Banghua Wu, Xuebin Lv, Abdullah Alghamdi, Hamad Abosaq, and Mesfer Alrizq. Advancement of management information system for discovering fraud in master card based intelligent supervised machine learning and deep learning during sars-cov2. *Information Processing & Management*, 60(2):103231, 2023.

[140] Tyler Derr, Cassidy Johnson, Yi Chang, and Jiliang Tang. Balance in signed bipartite networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1221–1230, 2019.

[141] Junjie Huang, Huawei Shen, Qi Cao, Shuchang Tao, and Xueqi Cheng. Signed bipartite graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 740–749, 2021.

[142] Charu C Aggarwal, Yao Li, S Yu Philip, and Yuchen Zhao. On edge classification in networks with structure and content. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 187–190. IEEE, 2017.

[143] Yadan Luo, Zi Huang, Hongxu Chen, Yang Yang, Hongzhi Yin, and Mahsa Baktashmotlagh. Interpretable signed link prediction with signed infomax hyperbolic graph. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[144] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. Bipartite graph embedding via mutual information maximization. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 635–643, 2021.

[145] Jianke Yu, Hanchen Wang, Xiaoyang Wang, Zhao Li, Lu Qin, Wenjie Zhang, Jian Liao, and Ying Zhang. Group-based fraud detection network on e-commerce platforms. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5463–5475, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599836.

[146] Jianke Yu, Hanchen Wang, Xiaoyang Wang, Zhao Li, Lu Qin, Wenjie Zhang, Jian Liao, Ying Zhang, and Bailin Yang. Temporal insights for group-based fraud detection on e-commerce platforms. *IEEE Transactions on Knowledge and Data Engineering*, 37(2):951–965, 2025. doi: 10.1109/TKDE.2024.3485127.

[147] Zheng Zhang, Jun Wan, Mingyang Zhou, Zhihui Lai, Claudio J Tessone, Guoliang Chen, and Hao Liao. Temporal burstiness and collaborative camouflage aware fraud detection. *Information Processing & Management*, 60(2):103170, 2023.

[148] Zhengyang Liu, Hang Yu, and Xiangfeng Luo. A noise-resistant model for graph-based fraud detection. *Information Processing & Management*, 62(5):104198, 2025.

[149] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1088–1098, 2007.

[150] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine learning and knowledge discovery in databases: European conference, ECML pKDD 2013, prague, czech Republic, September 23-27, 2013, proceedings, part III 13*, pages 387–402. Springer, 2013.