Connecting the Dots: Graph Neural Networks for Auditing Accounting Journal Entries

Qing Huang

Rutgers University
Marshall University

Marco Schreyer

International Computer Science Institute, University of California Berkeley
University of St. Gallen

Nilson Romero Michiles Junior

Federal Data Processing Service (Serpro), Brazil
University of Brasilia

Miklos A. Vasarhelyi

Rutgers University

Acknowledgements:

This article is based on Chapter 4 of Qing Huang's Ph.D. dissertation (Huang 2024).
The research of Marco Schreyer is supported by an IFI fellowship (No. 57515245) of the German Academic Exchange Service (DAAD), and a Mobi.Doc fellowship (No. 1031606) of the University of St. Gallen, Switzerland.

The opinions expressed are those of the authors and do not represent the opinions or positions of the organizations.

Connecting the Dots: Graph Neural Networks for Auditing Accounting Journal Entries

Abstract
Based on the double-entry bookkeeping mechanism, each transaction is recorded in at least two ledger accounts, with either debit or another credit. Journal entry data, in the context of accounting, contains a rich network of information that can be effectively translated into a graph. This study explores how to use graph neural networks to learn graph representations from journal entry data and to systematically understand the intricate patterns and connections inherent in journal entries at the transaction level. The real-world application results demonstrate that the unsupervised graph neural network framework of journal entry data offers a promising methodology for detecting fraud and error in auditing work.

1. Introduction

The formalization of double-entry bookkeeping can be traced back to the late 15th century in Italy by Luca Pacioli (Sangster 2007). At that time, the accounting data is recorded manually by bookkeepers. With the application of computers and Enterprise Resource Process (ERP) systems, there comes a massive increase in accounting data volume and a tremendous shift in how the organization records these data (Davenport 1998). No matter how the accounting data volume and recording methods are changed, auditors are mandated to audit the journal entries and consider fraud and unusual transactions in a financial statement audit, as required by the International Auditing and Assurance Standards Board (IAASB 2009), the Public Company Accounting Oversight Board (PCAOB 2020) and the American Institute of Certified Public Accountants (AICPA 2023). Journal entries (JEs) are viewed as the DNA of accounting information (Gray and Debreceny 2014); thus, the test of journal entries is of great importance to auditors.

Graph theory has been successfully applied in various domains, such as information systems, knowledge graphs, ecosystems, sociology, and biological networks (Xia et al. 2021). However, in accounting and auditing, research has primarily utilized graph theory for social network analysis (Jans, Alles, and Vasarhelyi 2014) or for visualizing journal entry data (Guo, Yu, and Wilkin 2022). There is a notable gap in research focused on applying graph learning techniques to journal entries. To address this gap, this paper explores the application of graph representation learning to audit journal entry data by constructing a graph neural network framework.

This study first constructs two matrices based on journal entry data: one is the general ledger account matrix, and another is the transaction variables matrix. Then, deep learning algorithms are employed to learn these variables and graph representations. This algorithm extracts the essence of the graph and learns the patterns present within the journal entries. Abnormal transactions will be detected based on the learned representations. Such transactions, significantly different from the learned mainstream patterns, could indicate irregularities such as errors, fraud, or other non-standard financial activity. By applying deep learning and graph representation, this research aims to advance financial auditing by offering a more proactive and intelligent approach to identifying abnormal transactions.

This paper contributes to the accounting and auditing practice in the following ways. First, this research proposes a data transformation technique that constructs two essential matrices from journal entry data: a general ledger account matrix and a transaction variables matrix. This transformation enables effective graph representation learning. Second, this study applies deep learning algorithms to learn the variables and their graph representations. This approach effectively navigates the complexities of the graph, identifying intricate patterns within the journal entries. Last but not least, this paper proposes a method to detect abnormal accounting journal entries that deviate from the learned posting activities of an organization.

The remainder of this paper is structured as follows: Section 2 discusses related literature in graph analysis, representation learning, and graph representation learning in real-world accounting data. Section 3 provides the background information about graph theory and the graph representation of journal entry data. Section 4 explains the proposed methodology. Section 5 discusses the data used in this study and the main results. Section 6 concludes the paper, summarizes the findings, and discusses the limitations and future study.

2. Literature Review

In recent years, techniques based on (deep) machine learning have gradually been applied to various audit tasks (Cho, Vasarhelyi, Sun, and Zhang 2020; Sun 2019). Simultaneously, graph representation learning has triggered significant academic research (Fey and Lenssen 2019; Hamilton, Ying, and Leskovec 2017; Hamilton 2020). Within the scope of this work, the study focuses the literature review on (i) graph analysis, (ii) representation learning, and (iii) graph representation learning of real-world accounting data.

2.1 Graph Analysis of Accounting Data

Graph theory has been effectively utilized in various fields, including information systems, knowledge graphs, ecosystems, sociology, and biological networks (Xia et al. 2021). Deep learning on graphs has also been recognized as an influential tool for graph analysis (Xia et al. 2021; Zhang, Cui, and Zhu 2022). Several studies have demonstrated the potential of graph theory for understanding relationships in the accounting and auditing domain. For example, Arya, Fellingham, Glover, Schroeder, and Strang (2000) first represent the double entry system in directed graphs; and then apply this representation to auditing work and financial statement analysis (Arya, Fellingham, Mittendorf, and Schroeder 2004). Menon and Williams (2004) and Lennox (2005) find that social ties between companies and accounting firms could negatively affect audit quality. Hwang and Kim (2009) document that directors of publicly traded firms are more independent than they appear. They also found that CEOs often select directors within their social networks, and these connections considerably influence the directors' effectiveness in oversight and monitoring roles.

McGlohon, Bay, Anderle, Steier, and Faloutsos (2009) propose using domain knowledge and link knowledge to detect misstated accounts in general ledger data, but their approach was limited to the node or account level. Certain ledger accounts are predictably high-risk, as they are more susceptible to manipulation or misstatement due to inherent complexities and subjective estimates. Examples include accounts receivable, inventory, revenue, and capitalized expenses. The real challenge lies in uncovering suspicious transactions or identifying fraud at the edge or transaction level. Node-level anomalies provide minimal added value to auditors since they already know which accounts are more suspicious. Jans et al. (2014) use graph analysis techniques to detect segregation of duties violations in auditing business processes, offering another potential application of graph theory in auditing practice. Recently, Guo et al. (2022) suggest visualizing journal entry data at the account level to identify fraud. However, this analysis is constrained to aggregated transaction data, which may overlook critical information.

2.2 Representation Learning of Accounting Data

The exploration of representation learning and deep learning techniques in auditing accounting data has been a focus of recent research, yielding novel insights, especially in anomaly detection and the interpretation JEs. Schreyer, Sattarov, Borth, Dengel, and Reimer (2017), Lenderink (2019), and Župan, Letinić, and Budimir (2018) propose to use deep Autoencoders (AENs) and Variational Autoencoders (VAEs) to detect anomalies within large-scale accounting data. Their works demonstrate the utility of reconstruction error as an effective anomaly detection measure through the analyses of anonymized datasets from real-world Enterprise Resource Planning (ERP) systems. The capability of learning accounting data representations to enhance audit procedures has also been highlighted in recent studies. Schultz and Tropmann-Frick (2020), Zupan, Letinic, and Budimir (2020) and Nonnenmacher, Kruse, Schumann, and Gómez (2021) detail the application of AENs, VAEs and Long Short-Term Memory (LSTM) architectures for detecting irregularities in accounting JEs. These contributions highlight the potential of representation learning technologies to improve audit processes.

Subsequent investigations have expanded the scope of anomaly detection. Schreyer, Sattarov, Schulze, Reimer, and Borth (2019) explore the use of deep Adversarial Autoencoder (AAE) networks to derive semantically meaningful representations of accounting JEs, using reconstruction error and the divergence from a specified prior distribution as indicators of anomalies. The introduction of the Vector Quantised-Variational Autoencoder (VQ-VAE) by Schreyer, Sattarov, Gierbl, Reimer, and Borth (2020) further advanced this field by facilitating the learning of quantized, low-dimensional representations of JEs. Hemati, Schreyer, and Borth (2021) propose proposed a technique for continuous unsupervised anomaly detection using AENs, addressing catastrophic model forgetting and reducing false positives and negatives. A recent study by Yadav, Zhang, and Jin (2023) presents an anomaly detection algorithm using a modified Transformer architecture for sub-ledger accounting JEs, demonstrating superior performance and adaptability compared to traditional methods.

2.3 Graph Representation Learning of Accounting Data

Very few studies have focused on graph representation learning in accounting data. Recently, Boersma, Sourabh, Hoogduin, and Kandhai (2018) propose a method to aggregate and visualize the nodes and edges of financial statements networks, finding that the visualization could help auditors evaluate a company's organizational complexity and serve as an indicator of risk. Nguyen et al. (2023) propose a method leveraging graph motifs to detect fraudulent accounting transactions at the graph level within a graph database. Boersma, Wolsink, Sourabh, Hoogduin, and Kandhai (2023) utilize network representations of companies from their transaction datasets, measuring their similarities to classify them further and assist in audit work. Although these studies utilize graph representation learning of financial data, their focus remains on graph structure information at a higher level and fails to identify transaction-level anomalies.

In conclusion, the reviewed literature emphasizes the potential of applying graph theory and deep learning to accounting data. This study integrates previous findings to facilitate graph anomaly detection in large-scale journal entry data at the transaction level. To the best of our knowledge, this work represents the first step toward graph representation learning of real-world accounting data at the transaction level.

3. Background

Recent advancements in graph theory and its application to financial data have introduced novel methodologies for auditing processes. This section provides an overview of graph theory fundamentals, highlights the importance of financial data in graph representations, and demonstrates how journal entries can be conceptualized as graphs to facilitate graph representation learning.

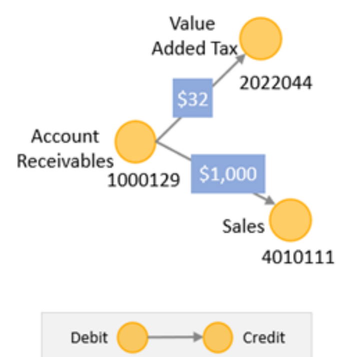
3.1 Graph Theory Fundamentals

A graph consists of a set of vertices (or nodes), a set of edges, and a relationship that pairs each edge with two vertices (West 2001). Nodes represent entities, and edges represent the connections between these entities. Graph data, historically overlooked, is prevalent in financial information. In double-entry bookkeeping, each transaction is recorded in at least two general ledger accounts, involving one debit and one credit. These connections can be effectively represented as graphs. Nodes represent general ledger accounts, and edges represent debit and credit connections of specific transactions, establishing a web of interconnected financial activity (see Figure 1). This perspective allows for the representation of large volumes of data in a manageable form, facilitating efficient analysis and pattern recognition. Depending on the need to differentiate the direction from one node to another, graphs can be classified as directed or undirected.

Figure 1 Graph Representation of Journal Entry

Journal Entry Data

Company	Entry ID	Line Item	Currency	Amount	D/C	Ledger
AAA	100011	1	USD	1,032.00	D	1000129
AAA	100011	2	USD	1,000.00	C	4010111
AAA	100011	3	USD	32	C	2022044
...	...	...	...	...	...	…



3.2 Journal Entries as Directed Graphs

The accounting journal entries provide a chronological record of a company's financial transactions. Representing these entries as directed graphs facilitates the application of graph representation learning to audit journal entry data.

3.2.1 Structure of Journal Entries

Recording journal entries enables organizations to document the effect of business transactions on various accounts. According to the Collins English Dictionary, a journal entry denotes an entry made directly into the general journal of an organization to accurately and adequately record every business transaction through a process known as “journalizing” (Schreyer 2023). A typical journal entry includes various categorical and numerical attributes required to record a transaction accurately, such as date, general ledger accounts (name and/or number), amount, and debit or credit designation (Schreyer 2023).

3.2.2 Graph Representation of Journal Entries

Transforming journal entries into graphs provides a novel perspective on accounting data. This method enables the application of graph representation learning techniques to audit complex journal entry data by conceptualizing accounts as nodes and journal entries as edges:

- Nodes: Represent accounts within the financial ledger.
- Edges: Represent the transactions between accounts, mapping the operational dynamics of financial processes.

This representation facilitates the visualization of transaction flows across accounts, enabling auditors to identify unusual patterns that could signify errors or fraudulent activities more effectively than traditional methods. Distinguishing between directed and undirected graphs offers varying perspectives on transaction flows. In the accounting domain, the concepts of debit and credit play an important role, as they signify different types of financial movements – increases or decreases in certain accounts. Consequently, this study uses directed graphs to capture this information. Specifically, this study differentiates the directions of connections based on the flow from debit to credit.

- Directed Graph: Indicates transaction flow from debited to credited accounts, reflecting transaction directionality.
- Undirected Graph: Depicts bidirectional relationships without specifying transaction flow.

3.2.3 Example of Graph-Based Journal Entry

Consider journal entry #100011, where Account Receivables (AR) is debited $1,032.00, Sales is credited $1,000.00, and Value Added Tax (VAT) is credited $32.00 (See Figure 1). This entry can be represented as a directed graph with AR as the originating node and Sales and VAT as terminal nodes, reflecting the precise nature of financial transactions.

Table 1 Journal entry #100011 in tabular format

Entry	Line Item	Account	Account Description	D/C	Debit (USD)	Credit (USD)
100011	1	1010	Account Receivables (AR)	D	1,032.00	-
100011	2	4000	Sales	C	-	1,000.00
100011	3	2050	Value Added Tax (VAT)	C	-	32.00

Transitioning from this tabular format (Table 1) to a directed graph captures the directional flow of resources, laying the foundation for auditing through graph representation learning techniques as proposed in the subsequent sections.

4. Methodology

This section outlines the methodology employed for graph representation learning in financial auditing, particularly focusing on journal entry data. The methodology leverages the Variational Graph Autoencoder (VGAE) framework to encode graph data into a latent space, facilitating various downstream audit tasks. The subsequent sections detail the attributes of journal entries, their representation as directed graphs, the components and functioning of the VGAE, and the application of learned representations for conformance checking and anomaly detection.

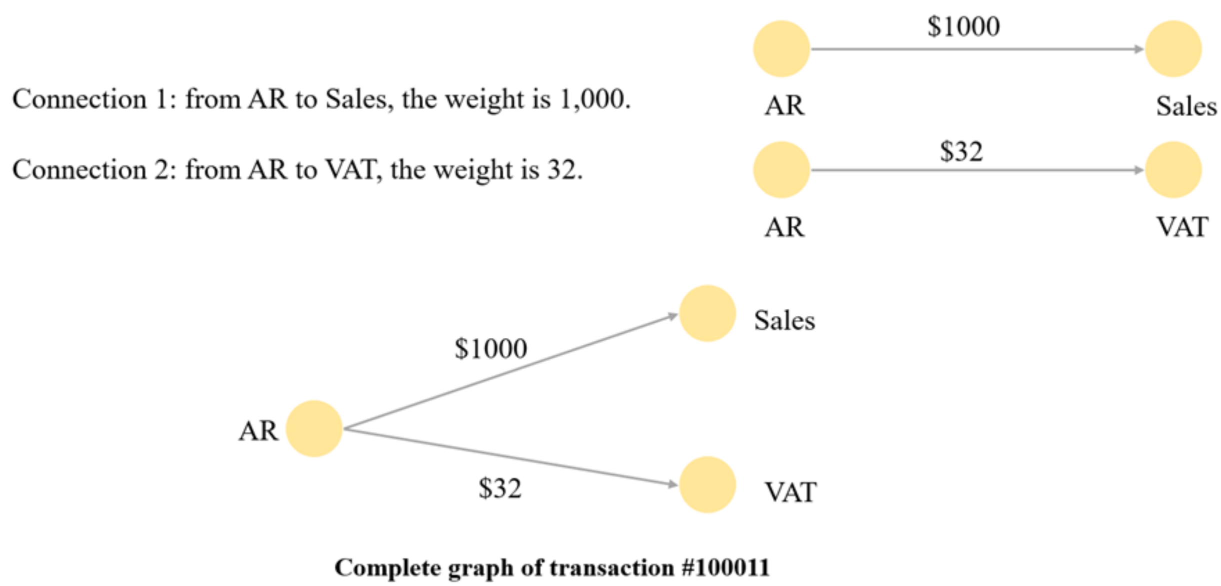
4.1 Overview of Journal Entry Attributes

In general, JEs can be characterized by two main classes of attributes: categorical and numerical. Formally, let A represents the number of categorical attributes, denoted as $x_i^{cat} = \{x_i^1, x_i^2, \ldots, x_i^a, \ldots, x_i^A\}$, where $a = 1, 2, \ldots, A$. These attributes include details such as posting type, general ledger, and date. And let B represents the number of numerical attributes, denoted as $x_i^{num} = \{x_i^1, x_i^2, \ldots, x_i^b, \ldots, x_i^B\}$, where $b = 1, 2, \ldots, B$. These attributes include details such as tax and transaction amounts. A single JE is thus represented by a union of categorical and numerical attributes, formally defined as $x_i^j = x_i^a + x_i^b$, where $j = 1, \ldots, M$ with $M = A + B$ denoting the total number of JE attributes. A dataset of JE, $\mathcal{D}$, represents a collection of entries as $\mathcal{D} = \{x_1, x_2, \ldots, x_i, \ldots, x_N\}$, where N is the number of entries.

4.2 Graph Representation of Journal Entries

Following the detailed tabular representation, a directed graph captures the essence of journal entry #100011 (See figure 2). In this visual representation, the Account Receivable (AR) account is depicted as the originating node, with directed edges flowing towards the Sales and Value Added Tax (VAT) accounts, signifying the debit to credit movement. Each node and edge are labeled with account names and transaction amounts, providing a clear perspective of the transaction's structure and flow.

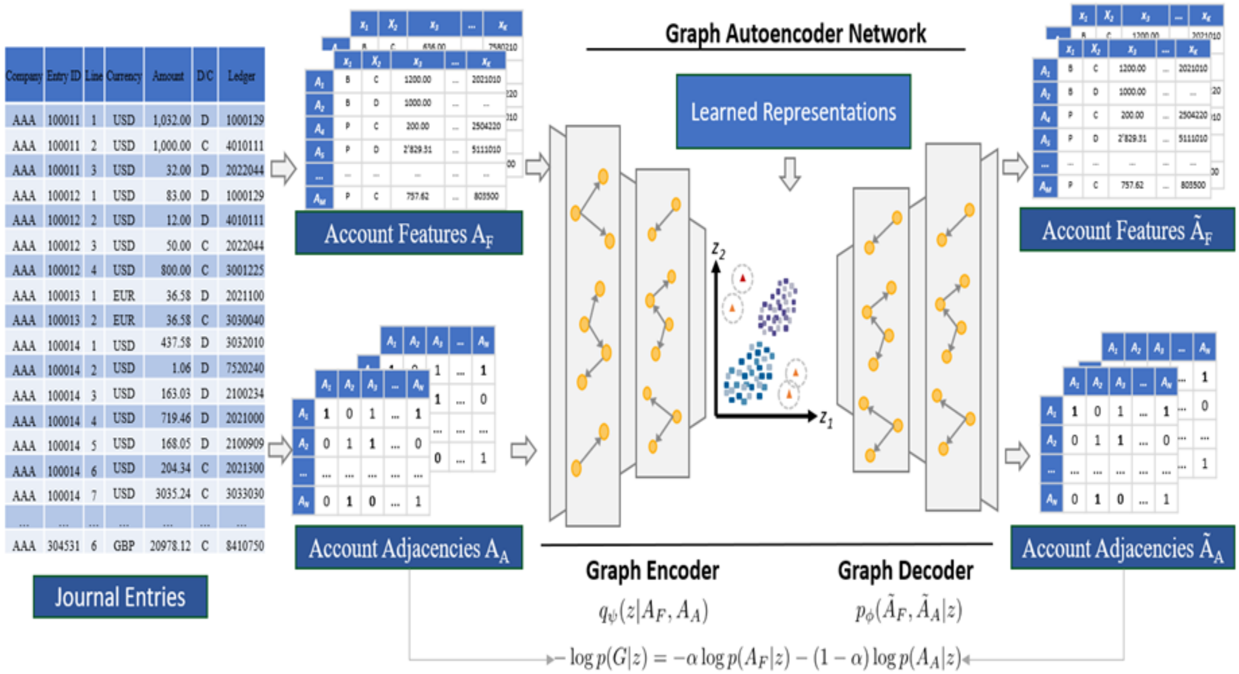
Figure 2 Directed Graph Representation Example



4.3 Graph Autoencoder Neural Networks

The Variational Graph Autoencoder (VGAE), originally proposed by Kipf and Welling (2016), enables the encoding of both graph topology and node attributes into a latent space. VGAEs address the non-Euclidean nature of graph data. By integrating Graph convolutional networks (GCNs), VGAEs could capture graph structures, which is critical for identifying patterns and ensuring model resilience against data sparsity. This technique is applicable for graph reconstruction, clustering, and, notably, anomaly detection. This study proposes a graph autoencoder network (GAEN) (see Figure 3) based on the VGAE framework. This process is facilitated through several key components:

Figure 3 Graph Neural Network Auditing Framework



- **Adjacency Matrix:** The adjacency matrix $A \in \{0,1\}^{N \times N}$ is a binary or weighted matrix representing the edges between nodes in the graph, where N denotes the number of nodes. In the VGAE framework, A serves as both an input to the encoder and the target for reconstruction by the decoder, guiding the learning process towards capturing the graph's structural essence.
- **Feature Matrix:** Accompanying A, the feature matrix $X \in \mathbb{R}^{N \times D}$ contains D-dimensional attributes for each node, extending beyond structural properties to provide additional information. In financial auditing, X may include details such as transaction amounts, account types, and other relevant information, offering a comprehensive view of each node's characteristics. Integrating X enriches the model's ability to recognize patterns by combining structural and feature information.

- **Graph Convolutional Network Layers:** Central to the VGAE architecture are Graph Convolutional Network (GCN) layers, which apply convolutional operations to graph-structured data. GCNs facilitate feature learning as formally defined by:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right),$$

where $H^{(l)}$ represents the activation matrix at the l-th layer, $\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections I_N, $\tilde{D}$ is the degree matrix of $\tilde{A}$, $W^{(l)}$ denotes the layer-specific trainable weight matrix, and σ signifies a non-linear activation function such as the ReLU. The GCN layers aggregate and transform node features with local neighborhood graph structures.

- **Encoder Network:** The encoder maps the input graph into a latent space representation by transforming the feature matrix, X, and the adjacency matrix, A, of the graph through a series of GCN layers. The encoding process approximates the posterior distribution of the latent variables as Gaussian distributions with means and variances parameterized by the output of the GCN layers:

$$q(Z|X, A) = \prod_{i=1}^{N} \mathcal{N}(z_i|\mu_i, \mathrm{diag}(\sigma_i^2)),$$

where the mean $\mu = GCN_\mu(X, A)$, and log variance, $\sigma^2 = GCN_\sigma(X, A)$, are obtained from a series of GCN layers. The transformation enables the encoder to efficiently capture both the salient features of the graph's structure and node attributes, embedding it into a probabilistic latent space Z.

- **Decoder Network:** The decoder reconstructs the adjacency matrix A from the latent space representation Z, using a pairwise inner product followed by a sigmoid activation function:

$$\hat{A} = \sigma(ZZ^T),$$

where $\sigma(\cdot)$ denotes the sigmoid function, producing probabilities that estimate the existence of edge between nodes. This operation allows the model to predict the graph structure, emphasizing the decoder's ability to infer the graph's connectivity pattern from the compact latent representations.

The whole process is governed by the loss function shown below. By leveraging these components, the proposed framework could provide a robust methodology for analyzing complex graph-structured data in journal entry data, enabling the detection of anomalies and enhancing the overall audit process.

$$log p\ (\mathrm{D}\ |z) = -a log p\ (A\ |z) - (1 - a)\ log p\ (\mathrm{X}\ |z)$$

where α is the weight of the Adjacency matrix.

4.4 Graph Representation Learning

The framework excels in graph representation learning by encoding graph data into a dense, latent space, thereby unveiling complex, nonlinear relationships that are not immediately apparent in the raw data. The encoding process, facilitated by GCN layers, transforms the graph into a latent representation Z , which is then utilized by the decoder to reconstruct the adjacency matrix A, thus preserving the graph's structural nuances.

The efficacy of VGAEs in pattern recognition can be attributed to two primary factors:

Latent Pattern Recognition: VGAEs identify latent patterns, including community structures and topological motifs that are obscured in the raw graph data. The model's encoder condenses the graph into a comprehensive latent representation that conveys topological and nodal feature patterns.

Probabilistic Embedding: VGAEs introduce a probabilistic dimension to the embeddings, adopting a variational inference approach to model the distribution of latent variables. This approach is particularly beneficial for addressing the uncertainty and noise inherent in real-world datasets, thereby enhancing the model's capabilities in pattern recognition.

In financial auditing, the ability of VGAEs to capture and analyze complex transactional relationships is particularly valuable. The learned representations can be utilized for different downstream audit tasks. In this paper, we focus on the downstream audit tasks of (a) journal entry anomaly detection and (b) journal entry conformance checking.

Anomaly Detection: The learned graph representations can also be employed for anomaly detection by analyzing deviations from the main clusters in the latent space Z . By identifying journal entries that significantly deviate from the established clusters, auditors can catch transactions that exhibit unusual patterns or structures. This deviation analysis facilitates the detection of potential anomalies, such as errors or fraudulent activities, that do not conform to normative transaction patterns.

Conformance Checking: The learned graph representations can be utilized for journal entry conformance checking by analyzing the main clusters of representations in the latent space

Z. By examining these clusters, auditors can identify the typical patterns and structures of legitimate transactions. This analysis allows for the verification of whether journal entries conform to the established organizational processes, ensuring consistency and adherence to expected accounting standards.

5. Experimental Setup and Results

This section mainly discusses the experimental setup and results. To evaluate the quantitative performance of the proposed methodology, this study injects anomalies into the data to simulate the unusual transactions. Inspired by Breuniq et al. (2000), two types of outliers are embedded: global and local accounting outliers. In this study, global outliers are constructed using general ledger accounts that have never been used in normal journal entries, while local outliers refer to journal entries that display unusual general ledger account combinations in this study.

Then, unsupervised outlier detection algorithms are applied to examine outliers, since the unsupervised approach does not require labels and is more practical in real-world applications (Fan, Xiao, Zhao, and Wang 2018). In this process, the input data is the latent representation of the original data, and the output is the outlier scores generated by outlier detection algorithms.

A conformance check of dataset A is also discussed to demonstrate how the proposed methodology could help auditors interpret the results.

5.1 Experimental Setup

The proposed methodology is tested on two journal entry datasets: one case study dataset from EY Academic Resource Center (EYARC) referred to as Dataset A, and a real-world dataset from a partner company referred to as Dataset B.

Below are three datasets that are used in this study.

- Dataset A: This case study dataset contains 37,869 journal entry line items, corresponding to 18,934 journal entries and 74 journal ledger accounts.
- Dataset B: This real-world dataset contains 1,348,829,408 journal entry line items, corresponding to 128,045 journal entries and 441 general ledger accounts.

Following the categorical attributes are encoded into dummy ("one-hot") variables, and the numerical attributes are normalized. Subsequently, a feature matrix is constructed by aggregating the processed attributes at the transaction level. Table 2 illustrates an example of a feature matrix, where each row represents a transaction, and each column represents a feature.

Table 2 Feature Matrix Example

Entry	Amount (Normalized)	Document Type_a	Document Type_b	…	Profit Center_x
810019	0.5	1	0	…	0
810020	0.3	0	1	…	1
…	…	…	…	…	…

Next, for each journal entry, an adjacency matrix is constructed to enable graph representation learning. The adjacency matrix represents the relationships between nodes in a specific journal entry, where each node corresponds to a general ledger account. The following steps outline the computation of the adjacency matrix using the general ledger information:

1. Identify General Ledger Accounts: For each journal entry, identify the general ledger accounts involved. Each of these accounts will correspond to a node in the graph, representing the various elements of the financial journal entry.

2. Initialize the Adjacency Matrix: Create an N^N adjacency matrix A , where N is the number of unique general ledger accounts in the journal entry. Initialize all elements of A to 0. The diagonal elements are always set to 0 to indicate no self-loops within the accounts[1].

3. Insert Edges Based on Journal Entries: For each pair of accounts (nodes) involved in a journal entry, if account_i (e.g., Accounts Receivable) is debited and account_j (e.g., Sales) is credited, set $A[_{i,j}] = 1$. This indicates a directed edge from node_i to node_j. Continue this process for all account pairs to complete the adjacency matrix.

For illustration, previous journal entry #100011 is used as an example. This journal entry involves three general ledger accounts: AR, Sales, and VAT, resulting in a 3 x 3 adjacency matrix. In this journal entry, AR is debited, and Sales and VAT are credited. Therefore, there is one connection from AR to Sales and another connection from AR to VAT. As a result, elements $A\,[_{1,2}]$ and $A\,[_{1,3}]$ in this matrix are set to 1. Since no other connections exist in this journal entry, all remaining elements in this matrix are set to 0.

---

[1] The diagonal can be set either to 1 or 0 in graph learning. In this study, the accounts are assumed not self-related, so the diagonal is set to 0.

Table 3 Adjacency Matrix Example

	AR	Sales	VAT
AR	0	1	1
Sales	0	0	0
VAT	0	0	0

In this example, AR is the originating node, with directed edges flowing towards Sales and VAT, indicating the flow and direction of the journal entry between these accounts. This adjacency matrix captures the essence of the journal entry's structure and flow, setting the foundation for graph-based analysis in financial auditing.

5.1.2 Graph Representation Learning

Our architectural setup follows the VGAE framework, consisting of an encoder and a decoder as shown in Figure 3. The encoder uses ReLU activation functions, except in the last layer where no non-linearity is applied. The encoder comprises multiple Graph Convolutional Network (GCN) layers to capture both graph topology and node attributes. The decoder reconstructs the adjacency matrix from the latent space representation. Table 4 provides the architectural details of both the encoder and decoder networks.

Table 4 Experimental Parameters[2]

Net	Dataset	Layer 1	Layer 2	Layer 3	Layer 4	…	Final Layer

[2] We set Z-dim (latent space dimension) to 2 for visualization purposes in the GCN layers. The maximum training epochs is set to 500, with a mini-batch size of 64 journal entries, and the

Encoder	A	256	128	64	32	…	2
Decoder	A	2	32	64	128	…	256
Encoder	B	108	64	32	16		2
Decoder	B	2	4	8		…	108

Upon completion of the model training, the learned encoder is utilized to obtain the JEs graph representation for downstream audit tasks such as journal entry conformance checking and anomaly detection.

5.2 Experimental Results

This study evaluates the performance of the proposed methodology in both quantitative and qualitative ways.

5.2.1 Quantitative Results

In the Quantitative way, this study uses the Average Precision (AP) to measure the performance of the proposed methodology (See Pedregosa (2011) for more details about AP) on dataset A. Twenty outliers are embedded into the case study data, ten are global outliers, and ten are local outliers. Table 5 represents the quantitative results for Dataset A. The performance of the proposed methodology is compared with that of the baseline, conventional outlier detection algorithms, and autoencoder. As shown in Table 5, the random setup serves as a baseline that randomly guesses anomalies, with an AP of 0.0011 overall, and 0.0005 for both global and local anomalies. Both the autoencoder and outlier detection algorithms – namely, the Local Outlier

---

model also apply early stopping once the loss converges. We use the Adam optimizer with $\beta_1 = 0.9$, h $\beta_2 = 0.999$ and a learning rate schedule following a cosine decay.

Factor (LOF) and Histogram-based Outlier Detection (HBOS) used in this study – exhibit better performance than the baseline. The GAEN alone performs better than the baseline but not as well as outlier detection algorithms or Autoencoder. However, when combined with outlier detection algorithms, GAEN demonstrates improved performance. Notably, GAEN combined with LOF significantly outperforms the other methods, particularly in the detection of global anomalies.

Table 5 Quantitative Results

Learning Setup		**AP $_{all}$**	**AP $_{global}$**	**AP $_{local}$**
Random	Random	0.0011	0.0005	0.0005
Autoencoder	ReError	0.0041	0.0058	0.0001
LOF	LOF	0.0044	0.0025	0.0022
HBOS	HBOS	0.0023	0.0019	0.0008
GAEN	ReError	0.0022	0.0022	0.0008
GAEN	LOF	**0.0635**	0.1060	**0.0084**
GAEN	HBOS	0.0055	**0.0097**	0.0005

5.2.2 Qualitative Results

As part of the tests, a qualitative assessment was conducted using a database from a partner company for the fiscal year 2022 (Dataset B). In collaboration with expert accounting auditors, model records were analyzed to identify the model accuracy in presenting data with possible doubtful or unusual entries from a financial accounting perspective. Additionally, there was an examination of whether they were interested in receiving alerts for these records as timely alerts or red flags during the fiscal year execution.

It is noteworthy that the dataset provided by the partner was used, ranked by Anomaly Score (GAEN - Outlier Algorithm) and Reconstruction Error (GAEN), without the injection of anomalous data. Due to processing limitations, accounting entries with a maximum of 1000 sequential entries were included, meaning nodes with a maximum of 1000 degrees. Table 6 below presents the analyses conducted by the auditors:

Table 6 Qualitative Results for Dataset B

Period	Ranked Score	# of Journal Entry Selected to Investigate	Auditor Qualitative Feedback			Sample Situations filled by Auditors (not Exhaustive)
			Is there a potential error?	Is it rare/unusual?	Would you like to receive alters like this?	
Q1*	GAEN + IForest	10	Yes (10%)	Yes (80%)	Yes (10%)	-Duplicate records
	GAEN + ReError	10	Yes (40%)	Yes (50%)	Yes (50%)	-Self-postings -Records with 0 value
Q1 & Q2*	GAEN + IForest	10	Yes (70%)	Yes (100%)	Yes (90%)	-Self-postings
	GAEN + ReError	10	Yes (100%)	Yes (100%)	Yes (100%)	-Unexpected reversal -Unexpected Transfer between ledger accounts
Fiscal Year*	GAEN + IForest	860	Yes (98.26%)	Yes (98.84%)	Yes (98.26%)	-Unusual manual entries -Unexplained reclassification entries
	GAEN + ReError	860	Yes (99.53%)	Yes (100%)	Yes (99.53%)	-Reversal without explanations -Unexpected Transfer between ledger accounts -Self-postings

* Q1 has 179,692 journal entry line items.
* Q1 and Q2 have 411,987 journal entry line items.
* The whole fiscal year has 1,064,147 journal entry line items.

As shown in Table 6, in auditing Quarter 1 data, auditors selected the top ten risky transactions ranked by the combination of GAEN with Isolation Forest (IForest) and GAEN reconstruction error, respectively. Upon investigation, for the top ten of GAEN + IForest, auditors considered one of them a potential error, eight truly rare or unusual, and only one transaction for which auditors would like to receive similar alerts in the future. For the top ten selected by GAEN reconstruction error, auditors identified four as potential errors, five as truly rare or unusual, and would like to receive similar alerts for five of them in the future. Auditors categorized these errors as self-postings, duplicate records, and records with a 0 value.

In the audit of Q1 and Q2 data, auditors followed the same selection process. For the ten transactions selected by GAEN + IForest, auditors found seven to be potential errors, all to be rare and unusual, and would like to receive alerts for all of them in the future. The ten transactions selected by GAEN reconstruction error were more noteworthy. Auditors considered all of them potential errors and unusual or rare, and would like to receive alerts for all of them in the future. These errors or anomalies were categorized as unexpected reversals, unexpected transfers between ledger accounts, and self-postings.

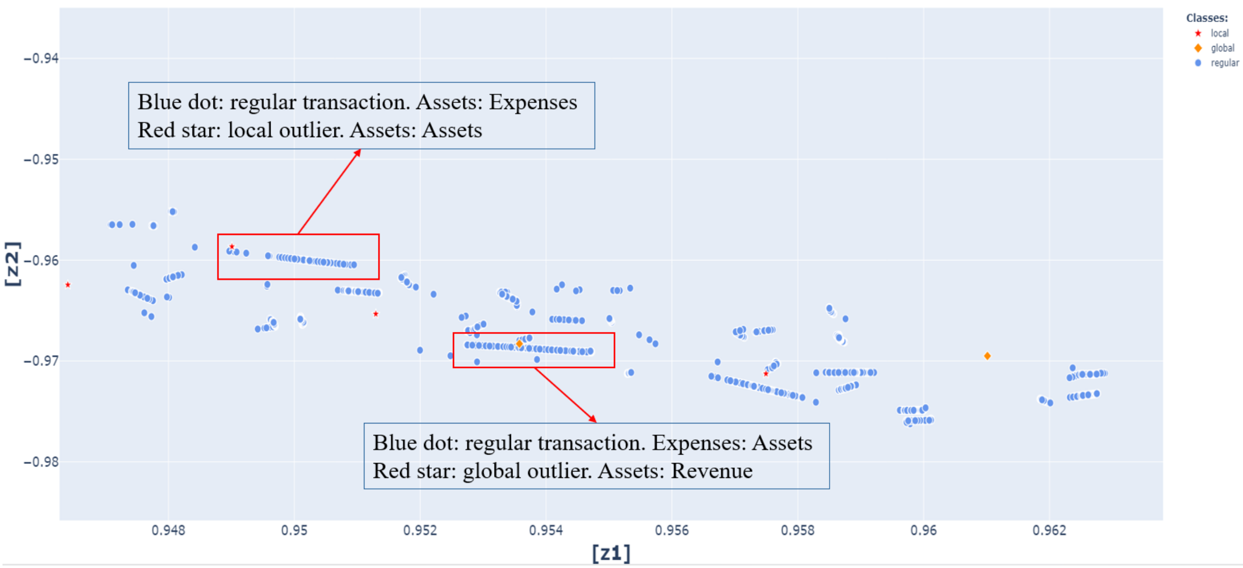
In the audit of the entire fiscal year, auditors select 860 transactions under these two selection methods. For transactions selected by GAEN + IForest, 98.26% are potential errors, 98.84% are rare or unusual, and auditors would like to receive similar alerts for 98.26% of them in the future. For transactions selected by GAEN Reconstruction Error, auditors think all of them are rare or unusual, 99.53% are potential errors, and auditors would like to receive alerts for 99.53% of the select transactions in the future. Auditors categorize these errors or anomalies as unusual manual entries, unexplained reclassification entries, reversals without explanations, unexpected transfers between ledger accounts, and self-postings.

Both GAEN + IForest and GAEN + Reconstruction Error have good performance in catching suspicious transactions, and the GAEN + Reconstruction Error works slightly better than the other one, especially in identifying potential errors. The qualitative results also demonstrate that the proposed methodology can not only detect suspicious transactions related to general ledger accounts combination and transfer, but also identify other types of risky transactions, such as unexpected reversal, duplicate records, and records with zero values.

5.3 Conformance Check

Figure 4 below shows partial latent space visualization of learned dataset A. The JEs data has been processed through a graph encoder to generate the learned representations in a two-dimensional space labeled as axes z1 and z2. The blue dots represent regular data points, while the orange diamonds and red stars represent global and local anomalies, respectively. Each cluster in the figure stands for a specific type of transaction. We randomly zoom into two clusters to check the transactions in detail. For the left highlighted cluster, it can be found that all blue dots in this area represent transactions that debit assets and credit expenses. While there is one exception that debits assets and credits assets, and this transaction is also labeled as a local outlier by the proposed methodology. Another highlighted cluster on the right exhibits another type of transactions that debit expenses and credit assets. One transaction debit asset and credit revenue is labeled as a global outlier in this cluster. By randomly examining transactions from these visualizable clusters or categories, auditors could enhance the conformance check work effectively.

Figure 4 Latent space representation of Dataset A



6. Conclusion

This study highlights the use of the Variational Graph Autoencoder (VGAE) for learning latent representations of journal entries. The VGAE model is adept at identifying subtle patterns and anomalies within the data, which are crucial for detecting fraudulent activities or errors in financial statements. By integrating graph representation learning with financial auditing practices, the proposed methodology offers a new approach to analyzing journal entries. The proposed framework enables the identification of accounting patterns and anomalies, contributing to more effective auditing.

Unlike traditional methods that view data in isolation, graph neural networks leverage the inherent graph structure of accounting entries, making them particularly suited for the domain where transactions are interconnected. While deep learning models are often criticized for their “black box” nature, the graph-based approach provides a more interpretable framework, as the relationships and transactions can be visualized and understood directly from the graph structure.

GNNs, through their structured representation learning, provide more accurate detection of anomalies than traditional methods, which might not effectively handle the high-dimensional and interconnected nature of journal entry data. Applying GNNs, specifically the VGAE, to audit accounting journal entries has demonstrated significant advancements in detecting anomalies and predicting financial outcomes.

This work contributes to the accounting and auditing practice from three perspectives. First, it proposes a shift in how data is processed and analyzed in accounting. As these technologies continue to evolve, their integration into mainstream financial practices is expected to grow, further transforming the financial analysis and auditing landscape. Second, the utilization of GNNs reduces the time and effort required by human auditors, improves the auditor's ability to handle large volumes of data, potentially reducing audit cycle times. Finally, the proposed methodology can model and analyze the complex relationships between different accounts and transactions as graphs, allowing auditors to detect anomalies and potential fraud more effectively. This contextual understanding helps identify unusual patterns that could indicate fraudulent activities or errors that are not apparent through non-graph auditing techniques.

Despite the promising results, the study has several limitations that need to be addressed. First, this study primarily utilizes two datasets, which may not comprehensively represent the diversity and complexity of journal entries encountered in various industries and organizations. Second, implementing GNNs, particularly in processing large-scale datasets, may involve significant computational resources and time. The complexity increases with the network's size and the graph structures' sophistication. This computational demand can limit the practical applicability of the proposed methods in real-time auditing environments. Moreover, there is a risk of overfitting, especially given the complex nature of GNNs and the depth of layers used in the model. Overfitting can lead the model to perform well on training data but poorly on unseen data, thus affecting its generalization capability. Ensuring the model generalizes well to new, unseen datasets is crucial for its application in diverse auditing environments. Finally, the effectiveness of the GNN model relies on the quality of the input data and the preprocessing steps applied. Inaccuracies in data preprocessing can significantly skew the model's outputs. Further research could explore the integration of GNNs with audit procedures to create a comprehensive framework for using GNNs in financial auditing.

In sum, applying GNNs in auditing journal entries represents a significant advancement in the field. By structuring data into graph formats, GNNs enable a more nuanced understanding of the relationships and interactions between accounts. This is crucial for detecting irregularities such as fraud or errors. The ability of GNNs to process and learn from large-scale, complex datasets meets a critical need in financial auditing. GNNs have the potential to enhance the accuracy, efficiency, and reliability of audits, effectively managing the increasing complexity of financial statements.

References

American Institute of Certified Public Accountants (AICPA). 2023. Consideration of Fraud in a Financial Statement Audit. *AU-C 240*. https://us.aicpa.org/content/dam/aicpa/research/standards/auditattest/downloadabledocuments/au-c-00240.pdf.

Arya, A., J. C. Fellingham, J. C. Glover, D. A. Schroeder, and G. Strang. 2000. Inferring Transactions from Financial Statements. *Contemporary Accounting Research* 17 (3).

Arya, A., J. C. Fellingham, B. Mittendorf, and D. A. Schroeder. 2004. Reconciling Financial Information at Varied Levels of Aggregation*. *Contemporary Accounting Research* 21 (2).

Boersma, M., S. Sourabh, L. Hoogduin, and D. Kandhai. 2018. Financial statement networks: An application of network theory in audit. *Journal of Network Theory in Finance* 4 (4).

Boersma, M., J. Wolsink, S. Sourabh, L. A. Hoogduin, and D. Kandhai. 2023. Measure cross-sectoral structural similarities from financial networks. *Scientific Reports* 13 (1).

Breuniq, M. M., H. P. Kriegel, R. T. Ng, and J. Sander. 2000. LOF: Identifying density-based local outliers. *SIGMOD Record (ACM Special Interest Group on Management of Data)* 29 (2).

Cho, S., M. A. Vasarhelyi, T. Sun, and C. Zhang. 2020. Learning from machine learning in accounting and assurance. *Journal of Emerging Technologies in Accounting*.

Davenport, T. H. 1998. Putting the enterprise into the enterprise system. *Harvard business review*, *76*(4), 121-131.

Fan, C., F. Xiao, Y. Zhao, and J. Wang. 2018. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied Energy* 211.

Fey, M., and J. E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric.

Gray, G. L., and R. S. Debreceny. 2014. A taxonomy to guide research on the application of data mining to fraud detection in financial statement audits. *International Journal of Accounting Information Systems* 15 (4).

Guo, K. H., X. Yu, and C. Wilkin. 2022. A Picture is Worth a Thousand Journal Entries: Accounting Graph Topology for Auditing and Fraud Detection. *Journal of Information Systems* 36 (2).

Hamilton, W. L. 2020. *Graph Representation Learning*. *Graph Representation Learning*.

Hamilton, W. L., R. Ying, and J. Leskovec. 2017. Representation Learning on Graphs: Methods and Applications.

Hemati, H., M. Schreyer, and D. Borth. 2021. Continual Learning for Unsupervised Anomaly Detection in Continuous Auditing of Financial Accounting Data.

Huang, Q. 2024. Three Essays on Anomaly Detection in Accounting: Channel Stuffing, Journal Entry Testing, and Graph Learning. Ph.D. dissertation, Rutgers University.

Hwang, B. H., and S. Kim. 2009. It pays to have friends. *Journal of Financial Economics* 93 (1).

International Auditing and Assurance Standards Board (IAASB). 2009. The Auditor's Responsibilities Relating to Fraud in an Audit of Financial Statements. *International*

Standard on Auditing (ISA) 240. https://www.ifac.org/_flysystem/azure-private/publications/files/A012%202013%20IAASB%20Handbook%20ISA%20240.pdf.
Jans, M., M. G. Alles, and M. A. Vasarhelyi. 2014. A field study on the use of process mining of event logs as an analytical procedure in auditing. *Accounting Review* 89 (5).
Kipf, T. N., and M. Welling. 2016. Variational Graph Auto-Encoders.
Lenderink, R. J. 2019. Unsupervised Outlier Detection in Financial Statement Audits. Master's Thesis, University of Twente.
Lennox, C. 2005. Audit quality and executive officers' affiliations with CPA firms. *Journal of Accounting and Economics* 39 (2).
McGlohon, M., S. Bay, M. G. Anderle, D. M. Steier, and C. Faloutsos. 2009. SNARE: A link analytic system for graph labeling and risk detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
Menon, K., and D. D. Williams. 2004. Former audit partners and abnormal accruals. *Accounting Review*.
Nguyen, H. T., P. J. Liang, and L. Akoglu. 2023. Detecting Anomalous Graphs in Labeled Multi-Graph Databases. *ACM Transactions on Knowledge Discovery from Data* 17 (2).
Nonnenmacher, J., F. Kruse, G. Schumann, and J. M. Gómez. 2021. Using autoencoders for data-driven analysis in internal auditing. In *Proceedings of the Annual Hawaii International Conference on System Sciences. Vol. 2020-January.*
Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12.
Public Company Accounting Oversight Board (PCAOB). 2020. Consideration of Fraud in a Financial Statement Audit. *Auditing Standards (AS 2401)*. https://pcaobus.org/oversight/standards/auditing-standards/details/AS2401.
Sangster, A. 2007. The printing of Pacioli's Summa in 1494: How many copies were printed?. *Accounting Historians Journal*, 34(1), 125-145.
Schreyer, M. 2023. Representation Learning in Financial Auditing Using Deep Neural Networks. Ph.D. Dissertation, St. Gallen University.
Schreyer, M., T. Sattarov, D. Borth, A. Dengel, and B. Reimer. 2017. Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks.
Schreyer, M., T. Sattarov, A. Gierbl, B. Reimer, and D. Borth. 2020. Learning sampling in financial statement audits using vector quantised variational autoencoder neural networks. In *ICAIF 2020 - 1st ACM International Conference on AI in Finance*.
Schreyer, M., T. Sattarov, C. Schulze, B. Reimer, and D. Borth. 2019. Detection of Accounting Anomalies in the Latent Space using Adversarial Autoencoder Neural Networks.
Schultz, M., and M. Tropmann-Frick. 2020. Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits. In *Proceedings of the Annual Hawaii International Conference on System Sciences*. Vol. 2020-January.
Sun, T. S. 2019. Applying deep learning to audit procedures: An illustrative framework. *Accounting Horizons* 33 (3).

West, D. B. 2001. *Introduction to graph theory*. Vol. Vol. 2. Upper Saddle River: Prentice hall.

Xia, F., K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu. 2021. Graph Learning: A Survey. *IEEE Transactions on Artificial Intelligence*.

Yadav, D., S. Zhang, and T. Jin. 2023. Transformer based Anomaly Detection on Multivariate Time Series Subledger Data. *In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 06â•fi10, 2023, Long Beach, CA, USA.* 1.

Zhang, Z., P. Cui, and W. Zhu. 2022. Deep Learning on Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 34 (1).

Župan, M., S. Letinić, and V. Budimir. 2018. Journal Entries with Deep Learning Model. *International Journal of Advance Computational Engineering and Networking (IJACEN)* 6 (10).

Zupan, M., S. Letinic, and V. Budimir. 2020. Accounting Journal Reconstruction with Variational Autoencoders and Long Short-term Memory Architecture. In *CEUR Workshop Proceedings*. Vol. 2646.