GraphFC: Customs Fraud Detection with Label Scarcity

Karandeep Singh*
Institute for Basic Science
Daejeon, Korea
ksingh@ibs.re.kr

Yu-Che Tsai*
National Taiwan University
Taipei, Taiwan
roytsai27@gmail.com

Cheng-Te Li
National Cheng Kung University
Tainan, Taiwan
chengte@ncku.edu.tw

Meeyoug Cha
Data Science Group, IBS & School of Computing, KAIST
Daejeon, Korea
meeyoungcha@kaist.ac.kr

Shou-De Lin
National Taiwan University
Taipei, Taiwan
sdlin@csie.ntu.edu.tw

ABSTRACT

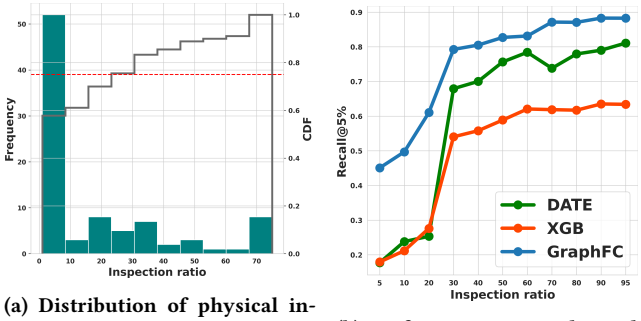
Custom officials across the world encounter huge volumes of transactions. Associated with customs transactions is the customs fraud - the intentional manipulation of goods declarations to avoid the taxes and duties. Due to limited manpower, the custom offices can only manually inspect a small number of declarations, necessitating automation of customs fraud detection by machine learning (ML) techniques. Limited availability of manually inspected ground truth data makes it essential for the ML approach to be able to generalize well on unseen data. However, current customs fraud detection models are not well suited and designed for this setting. In this work, we propose GraphFC (***Graph*** neural networks for ***C***ustoms ***F***raud), a model-agnostic, domain-specific, graph neural network based customs fraud detection model that is designed to work in a real-word setting with limited ground truth data. Extensive experimentation using real customs data from two different countries demonstrate that GraphFC generalizes well over unseen data and outperforms various baselines and other models with a large margin.

KEYWORDS

Graph Neural Network, Frauds detection, Multi-task learning, Customs Fraud Detection

1 INTRODUCTION

Customs is an authority in a country responsible for collecting tariffs and controlling the flow of goods into and out of country. According to the World Trade Organization, the world merchandise trade volume for exports and imports in 2020 alone was about 36.5 trillion dollars[1]. With globalization and increased connectivity, the trade volumes continue to grow further. International trade and commerce via customs encounter malicious transaction declarations that involve intentional manipulation of trade invoices to avoid ad valorem taxes and duties [8, 18, 26]. Administrators inspect trading goods and invoices to secure revenue generation from trades and develop automated systems to detect fraudulent transactions. This is an important undertaking as with customs transactions running into millions, even a slight increase in the detection accuracy will result in collection of hundreds of thousands of dollars of additional revenue. However, limited manpower and the colossal scale of trade volumes make the manual inspection of all transactions practically impossible. In fact, as per the World Bank reports (Figure 1a), administrations end up inspecting only a small portion, typically 5% - 10% or lower, of the total transactions [5, 14]. Recently, the World Customs Organization[2] (WCO) and its partner countries have made a progress in developing machine learning models, which can help officials identify and investigate any suspicious transactions e.g. [19, 22, 33]. However, the existing works on customs, including previous state-of-the-art (SoTA) DATE [19], fall short in adequately addressing the real-world scenario with low inspection rates and generalizing well over unseen data.



(a) Distribution of physical inspection rate among 90 countries across the world.

(b) Performance w.r.t. physical inspection ratio (%).

Figure 1: Physical inspection rate across the world and performance of different models on B-Data.

Customs fraud detection under low inspection rate: Manual inspection in customs refers to the physical examination of goods, packages, or shipments by customs officials to verify trade declaration and identify potential violations. Due to high volume of international trade, and limited human resources, only a small fraction of declarations actually undergo manual inspection. As shown in Figure 1b with two popular models DATE and XGBoost, the scarcity of labeled data (i.e., inspection rate lower than 30%) severely impacts the performance of existing customs fraud detection machine learning models. On the contrary, our proposed

*Both authors contributed equally to this research.
[1]Source: World Integrated Trade Solution, https://wits.worldbank.org/CountryProfile/en/WLD

[2]http://www.wcoomd.org/

CIKM '23, October 21–25, 2023, Birmingham, UK
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

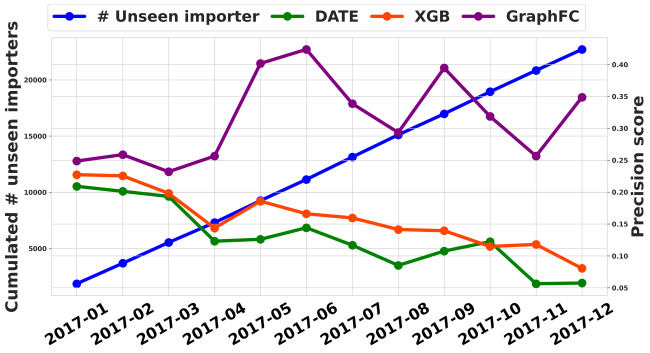


Figure 2: Comparison of model performance *w.r.t* # unseen importers on B-Data in 2017.

GraphFC is less effected by the label scarcity as it leverage the rich information in unlabeled transactions that improves the model robustness.

Performance degradation on unseen data. We adopt a temporal approach to realistically assess the impact of unseen data on machine learning models. By splitting the data into a training set (2014-2016) and a testing set (2017), we examine the models' performance on unseen data. Results in Figure 2 reveal a steady increase in unseen importers over time, posing a challenge for the models as they encounter unfamiliar patterns. This leads to a decline in performance, highlighting the struggle to adapt to evolving and unseen importer profiles. To address this, our proposed GraphFC leverages the inductive capability of GNNs, exhibiting invariance to unseen data (Figure 2). This underscores its effectiveness in mitigating the impact of unseen features and handling evolving importers.

To tackle the aforementioned challenges, we propose to model the tabular customs data as a connected graph of transactions, and then address fraud detection problem from the graph perspective. Representing customs data as a graph structure offers significant advantages, including enhanced connectivity among customs transactions and the ability to leverage unlabeled data to learn more comprehensive and representative embeddings (refer Figure 3). Second, integrating data from the modeled graph enables utilization of unlabeled data. Previous models rely heavily on labeled data and miss out on useful information that may exist in the unlabled transactions. Model's exposure to a wider and richer set of examples enhances its ability to generalize better over unseen data.

Extensive experimentation demonstrates that the proposed model substantially mitigate the performance drop with label scarcity and offers major improvements over strong baselines like XGBoost and SoTA DATE [19]. We summarize our contributions as follows:

- We develop ***Graph*** neural networks for ***C***ustoms ***F***raud (GraphFC) and showcase its strength in multi-year real customs data from two countries by demonstrating substantial improvement over the previous state-of-the-art and various other baselines.
- GraphFC adopts self-supervised and semi-supervised learning techniques to extract rich information from unlabeled data, which substantially alleviate the performance degradation due to label scarcity.
- Results from two datasets demonstrate that GraphFC achieves average improvements of 38% in precision, 39% in recall, and 22% in revenue compared to the state-of-the-art baseline.

2 RELATED WORK

Most of the customs administrations are using legacy rule-based systems [23] which are hard to maintain and are heavily dependent on expert knowledge [21, 27]. In general, advances in data science have led to the development of various fraud-detection models. A popular choice is tree-based models [1–3, 41]. Due to the non-availability of customs data in the public domain, the published literature on customs fraud is understandably scarce. However, there are some known efforts, like using support vector machine-based learner (Belgium) [35]; unsupervised spectral clustering (Columbia) [11]; ensemble of tree-based approaches, support vector machine and neural networks (Indonesia) [7]. Similarly, Netherlands' customs fraud detection model is built based on the Bayesian network and neural networks [34]. Other countries like Brazil have developed their proprietary customs fraud detection systems [13].

GNNs are increasingly being used in fraud detection. For instance [44] addressed the homogeneity and heterogeneity of issue of networks for fraudulent invitation detection, anomaly detection on dynamic graphs with sudden bouts of increased and decreased activities is proposed in [10], [38] proposed a semi-supervised approach and an attention-based approach for fraud detection in Alipay. In [29], authors designed a dynamic heterogeneous graph from the users' registrations to prevent the suspicious massive account registrations, [12] designs a GNN based fraud detection model to detect the camouflaged fraudsters, [39] designs a graph convolutional network for fraudster detection in online app review system, [28] undertakes the task of fraud detection in credit cards by learning temporal and location-based graph features. One of the recent works in this area was by Huang et al. [17], who proposed an AO-GNN for fraud detection. They addressed the label-imbalance problem in GNNs by maximizing the AUC metric, which is unbiased with label distribution. Liu et al. [24] proposed PC-GNN for imbalanced supervised learning on graphs. They devised a label-balanced sampler to construct sub-graphs for mini-batch training. While these approaches made advances in the GNN methodologies and applications, their application in customs domain is completely unexplored.

3 PROBLEM SETTING

Undervaluation is the most common type of customs fraud. Importers or exporters declare the value of their trade goods lower than the actual value, mainly to avoid ad valorem customs duties and taxes [30]. For this work, we limit the definition of an illicit customs transaction to that of undervaluation. In this light, we formulate the problem of customs selection as follows:

Problem: *Given a transaction o_i with its importer ID imp_i and HS-code c_i of the goods, the goal is to predict both the fraud score $\hat{y}_i^{cls}$ and the raised revenue $\hat{y}_i^{rev}$ obtainable by inspecting transaction o_i.*

By selecting top-n transactions the highest predicted values $\hat{y}_i^{cls}$ and $\hat{y}_i^{rev}$, customs administration could identify the most suspicious transactions to be inspected. Meanwhile, since we focus on mitigating the low inspection ratio problem in this work, we

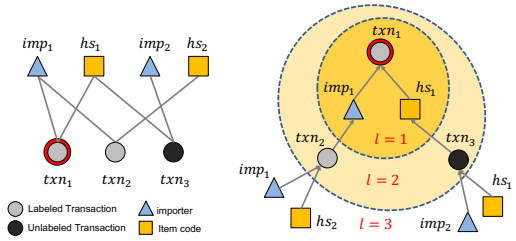


Figure 3: Transaction graph construction in GraphFC

assume we have the labeled set $L = \{(x_i, y_i^{cls}, y_i^{rev}), \forall i = 1, \ldots, n\}$ and unlabeled set $U = \{(x_i), \forall i = n+1, \ldots, m\}$, where $|L| \ll |U|$.

4 METHODOLOGY

The proposed model, GraphFC, leverages the strengths of Gradient Boosting Decision Trees (GBDTs) and GNNs, as depicted in Fig. 4. Initially, the customs data is fed into an XGBoost (XGB) model, popular for its effectiveness in tabular data [19, 25]. We extract cross-features from the fitted XGBoost model and utilize them for initializing the embeddings of transaction nodes. A self-supervised pretraining approach is then employed on to start learning the node representations within the transaction graph. Finally, GraphFC is fine-tuned on the labeled dataset using a dual-task optimization scheme, aiming to identify fraudulent transactions that maximize revenue collection.

4.1 Transactions Graph from Tabular Data

A crucial aspect of our work involves transforming customs data into a graph. The conventional approach of connecting transaction nodes based on similar features, like importer ID, leads to an extremely dense graph with astronomically high space complexity. Such a graph becomes impractical for real-world usage due to both memory constraints and the over-smoothing problem in GNNs. To address this challenge and enable practical implementation, we introduce virtual nodes $\mathcal{V}_C$ that only connect to transactions sharing the same feature value. This step significantly reduces the complexity from $O(n^2)$ to $O(n)$ (see Fig. 3), where the number of edges for a single node is proportional to the number of transactions multiplied by the number of virtual nodes.

As per the domain experts and prior research, we choose importer ID and HS-code as categorical features to establish links between transactions. $\mathcal{V}_C$ consists of importer ID and HS-code nodes with unique values as node IDs. The transaction graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{V}_\mathcal{T} \cup \mathcal{V}_C$ represents individual transactions and virtual nodes, and $\mathcal{E}$ represents connections based on shared feature values. The feature matrix $\mathbf{D} \in \mathbb{R}^{h \times m}$ consists of $\mathbf{h} = |\mathcal{V}|$ nodes with associated $\mathbf{m}$ features. During the testing phase, only historical transactions are considered for inferring the transaction embeddings. The features of nodes in $\mathcal{V}_\mathcal{T}$ are obtained from cross features in the GBDT step, while nodes in $\mathcal{V}_C$ are initialized with zero vectors.

4.2 Cross Features from GBDT

GBDT is an ensemble technique where multiple weak learner models are combined to provide a powerful overall learner. The decision path in the fitted trees could be seen as a new set of features, with each path representing a new cross feature made from the original features. Inspired by some recent studies such as [16, 40], we utilize state-of-the-art XGBoost model [9] and extract cross features from the fitted model. Let us assume W represents the total number of leaves in the ensemble. An input vector $\mathbf{x}$ ends up in one of the leaf nodes according to the decision rules of each fitted decision tree. We represent each activated leaf node in the t-th decision tree as a one-hot encoding vector $\mathcal{F}_t$. We concatenate these together and produce a multi-hot vector $\mathbf{p} \in \mathbb{R}^W$, where 1 indicates the activated leaves and 0 the non-activated ones.

4.3 Transaction Interaction Learning with Message Passing

4.3.1 Local subgraph extraction via neighbor sampling. To mitigate the problems associated with transductive training of GNNs, the neighborhood sampling technique [15] is applied to extract a local subgraph from a target node for inductive mini-batch training. Specifically, it samples a tree rooted at each node by recursively expanding the root node's neighborhood by K steps with a fixed sample size of $\{T_1, T_2, ..., T_K\}$, where T_i is the number of nodes sampled in i-th step. Note that we alternately sample T_i nodes from either $\mathcal{V}_C$ or $\mathcal{V}_\mathcal{T}$ at each step to form the subgraph. Afterwards, the desired message passing operation could be applied to the subgraph.

4.3.2 Transaction interaction learning with graph neural network. Graph neural networks (GNNs) offer a powerful framework for learning better embeddings by considering high-order interactions among transactions. In GNNs, the interaction between nodes can be viewed as an embedding propagation process, often referred to as message passing. By stacking multiple message passing operations, the node representations gradually accumulate information from a local subgraph, as depicted in Figure 3 and discussed in Section 4.1. The message aggregation stage combines all the neighboring messages obtained from the previous message construction stage, consolidating them into a single representation that serves as the final node embedding. This mechanism allows GNNs to capture and incorporate the high-order dependencies and interactions among transactions, leading to improved transaction embeddings that capture the underlying patterns and nuances of the customs data. Specifically, we adopt the Graph Attention Network [36] as the backbone of GNN layer where the embedding for node m could be represented as:

$$s_m^{(k+1)} = ReLU\left(\sum_{s_n \in N(m)} \alpha_{mn}^{(k)} \mathbf{W}_1^{(k)} s_n^{(k)}\right), \tag{1}$$

where $\alpha_{mn}^{(k)}$ denotes the attention weight from node m to node m, $N(m)$ returns the set of neighbors for node m , and $\mathbf{W}_1^{(k)}$ is a learnable weight matrix at the k-th layer. Note that we have $s_m^{(0)} = \mathbf{p}_m$ be the cross feature of the node m. The attention weights α_{mn} can be derived by:

$$\alpha_{mj}^{(k)} = \frac{\exp\left(\sigma\left(\mathbf{r}_k^\top [\mathbf{W}_2^{(k)} s_j^{(k)} \parallel \mathbf{W}_2^{(k)} s_m^{(k)}]\right)\right)}{\sum_{n \in N(m)} \exp\left(\sigma\left(\mathbf{r}^\top [\mathbf{W}_2^{(k)} s_n^{(k)} \parallel \mathbf{W}_2^{(k)} s_m^{(k)}]\right)\right)}, \tag{2}$$

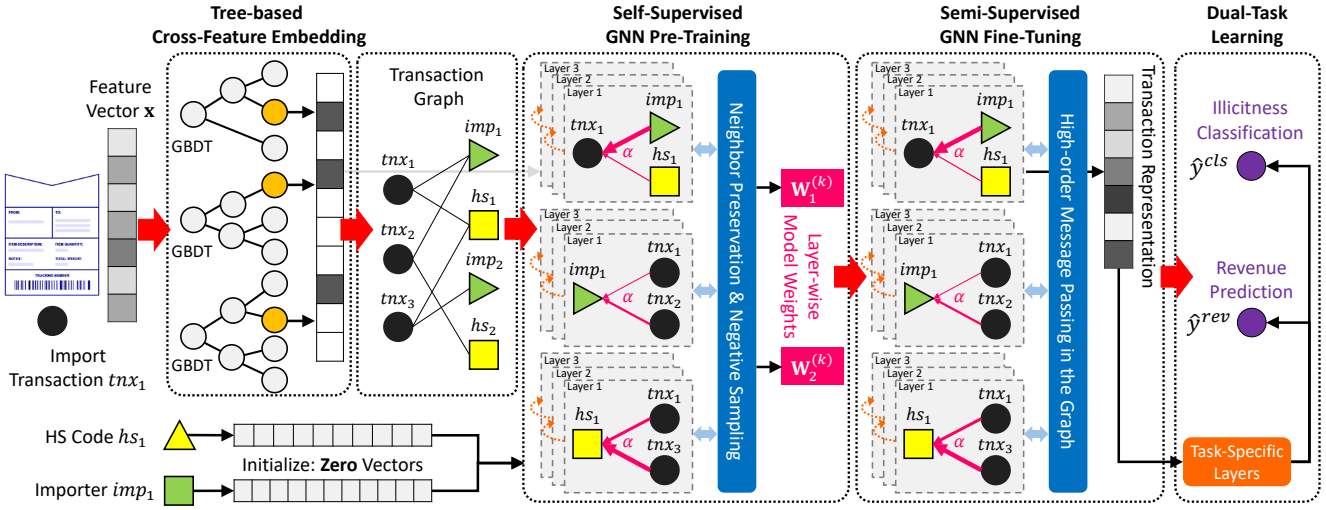


Figure 4: Model architecture of GraphFC. Cross features extracted from GBDT step act as node features in the transaction graph. In the pre-training stage, GraphFC transaction embeddings. Afterwards, the model is fine-tuned with labeled data with dual-task learning framework to predict the illicitness and the additional revenue.

where $\mathbf{r}_k$ is the learnable vector that projects the embedding into a scalar, $\|$ denotes the concatenation operation, and $\mathbf{W}_2^{(k)}$ is a learnable weight matrix.

4.3.3 Order of Message Passing. In the message passing operation, one of the key components of GraphFC is to make use of the virtual node $\mathcal{V}_C$ to aggregate features from its neighboring transactions, with $\mathcal{V}_C$ being connected to both labeled and unlabeled transactions. As the node embeddings of virtual node $\mathcal{V}_C$ is initialized with zero vectors, it is critical to decide the order of message passing to avoid collecting information from an empty vector. We address this issue by updating embeddings of $\mathcal{V}_C$ and $\mathcal{V}_\mathcal{T}$ in the following manner: In step 1, the embeddings for $\mathcal{V}_C$ are updated by collecting information from $\mathcal{V}_\mathcal{T}$ via E.q 1. It is followed by step 2 that updates embeddings of $\mathcal{V}_\mathcal{T}$ as per new representations of V_C. Step 1 and step 2 performed alternately to obtain the final embeddings.

4.4 Self-Supervised Pretraining

We employ a self-supervised GNN pretraining approach in our GraphFC framework. This allows us to learn transaction embeddings from labeled and unlabeled data, preserving the transaction graph structure and enhancing generalization [6, 20, 37]. The self-supervised objective function, similar to [15], guides the learning process. This flexible approach leverages entire data without requiring ground-truth labels. Additionally, it enhances generalization by aligning the distribution of unlabeled data with that of labeled data, thereby reducing the extrapolation phenomenon.

4.5 Prediction and Optimization

The following dual-task learning objective is used to achieve two goals: 1. provide the probability of a transaction being illicit 2. predict the additional revenue (i.e., taxes) after inspecting suspicious transactions. Hence, transaction embedding is used (i.e., $s_m^{(k)}$) for both the tasks of binary illicit classification and maximizing revenue prediction [19]. Given the transaction feature $s_m^{(k)}$, task-specific layer is defined:

$$\begin{aligned} \hat{y}^{cls}(s_m^{(k)}) &= \phi\left(\mathbf{r}_1^\top s_m^{(k)} + \mathbf{b}_1\right), \\ \hat{y}^{rev}(s_m^{(k)}) &= \mathbf{r}_2^\top s_m^{(k)} + \mathbf{b}_2, \end{aligned} \tag{3}$$

where $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^d$ denotes the hidden vectors of task-specific layers that project $s_m^{(k)}$ into the prediction tasks of binary illicitness and raised revenue, respectively. ϕ is the sigmoid function. $\hat{y}^{cls}(s_m^{(k)})$ is the predicted probability of a transaction being illicit, and $\hat{y}^{rev}(s_m^{(k)})$ is the predicted raised revenue value of a transaction. The final objective function $\mathcal{L}_{\text{GraphFC}}$ is given by:

$$\mathcal{L}_{\text{GraphFC}} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{rev} + \lambda\|\Theta\|^2, \tag{4}$$

where Θ denotes all learnable model parameters, $\mathcal{L}_{cls}$ is the cross-entropy loss for binary illicitness classification, $\mathcal{L}_{rev}$ is the mean-square loss for raised revenue prediction. The hyperparameter α is used to balance the contributions. Finally, we use mini-batch gradient descent to optimize the objective function $\mathcal{L}_{\text{GraphFC}}$, along with the *Ranger* [42] optimizer.

5 EVALUATION

5.1 Experimental Setup

We employed multi-year transaction-level import data from two partner countries of WCO. Data contains only the import transactions and those reported with undervaluation. Due to data confidentiality policy, we abbreviate these countries as A and B. Subject to availability, the two datasets have a different number of records. A-data has about 2.3 million transactions will illicit rate at 8.16%. These number stand at about 1.4 million and 4.14% for B-data. Further, as data belongs to different customs administrations, there are minor differences in features, but the most relevant features are the similar across the datasets. Among others, these include the HS-code, importer IDs, the notional value of the transaction, date, taxes paid, the quantity of the goods. We also add a binary feature

representing historic fraudulent behavior or total taxes paid per unit value.

Data Split To mimic the actual setting, we split the train, valid, and test data temporally. Since we have multi-year data, we treat the data from the most recent year as the test, and the older data is split as train and validation sets.

Performance under Low Inspection Ratio: We assume a prior inspection rate of 5% for the available datasets.[3] This implies that only 5% of the ground truth data is available and the remaining 95% data is unlabeled.

Evaluation Criteria: We imitate the limited manual inspection setting by evaluating model performance on the top top-n% (we demonstrate results for 1%, and 5%) of the suspicious transactions suggested by the model. Thus, e.g., precision at 1% is equivalent to classification precision when top 1% of transactions suggested by GraphFC are *manually inspected* and the ground-truth for those transactions is obtained. We report precision (Pre.), recall (Rec.), and revenue (ratio of total revenue that may be collected if all fraudulent transaction are manually inspected) (Rev.) as evaluation the metrics for the model.

Deployment: In April 2023, the author delivered an invited lecture at the Korea Customs Week 2023 [32] held in Seoul, South Korea, which was attended by 81 customs authorities, international organizations including UNESCAP, ICC, APEC, AFCFTA, UNODC, WCO, GEA, IDB OECD, and WIPO. Various customs authorities, including South Korea, Israel, India, France, Mauritius, and Mozambique acknowledged the significance of the work and expressed interest in accessing the model and materials. GraphFC can consume the digital declarations as they are made and officers can inspect the top-n suggestions by the model.

5.2 Performance Comparison

GraphFC performance is compared with 4 baseline methods:

- **XGBoost** [9]: XGBoost is a tree-based model which is widely used for modeling tabular data.
- **Tabnet** [4]: Tabnet is a self-supervised learning-based framework especially designed for tabular data.
- **VIME** [43]: VIME adopts self-and semi-supervised learning techniques for tabular data modeling.
- **DATE** [19]: This fraud detection method which utilizes tree-aware embeddings and attention network.
- **GraphFC$_{\text{RGCN}}$**: A GraphFC variant using RGCN [31] aggregator designed for relational and heterogeneous graphs.

We compare detection results under 5% inspection rate across various baselines in Table 1. GraphFC shows a consistent and remarkable improvement over the baselines and DATE, which verifies the effectiveness of the proposed model. It is worth emphasizing that the significant improvement of GraphFC over the self- and semi-supervised VIME and Tabnet models further validates the superiority of the proposed model. We also present the model performance by varying the inspection rates in {1%, 2%, 5%, 10%, 20%}(Fig. 5). GraphFC and its variants consistently outperform all the baselines. Our results firmly establish that GraphFC is robust against different degrees of label scarcity, and generalize well in different countries with respective data patterns. Additionally, model's performance with different GNN backbone highlights the flexible nature of the approach and indicates that any desired aggregation strategy can be used as per the requirements.

[3] More experiments with additional dataset and different inspection scenarios, including full supervised setting, can be found on project's Github page.

Table 1: Model performance for a prior inspection rate of 5%.

Model	n=1%			n=5%		
	Pre.	Rec.	Rev.	Pre.	Rec.	Rev.
A-Data						
XGB	0.671	0.070	0.120	0.445	0.234	0.359
Tabnet	0.715	0.05	0.081	0.452	0.231	0.334
VIME	0.725	0.076	0.116	0.471	0.249	0.362
DATE	0.803	0.085	0.158	0.472	0.249	0.38
GraphFC$_{\text{RGCN}}$	0.819	0.086	0.146	0.525	0.277	0.424
GraphFC	**0.869**	**0.092**	**0.17**	**0.535**	**0.282**	**0.445**
B-Data						
XGB	0.151	0.061	0.109	0.045	0.092	0.184
Tabnet	0.149	0.06	0.089	0.046	0.085	0.171
VIME	0.064	0.024	0.052	0.043	0.087	0.208
DATE	0.152	0.061	0.105	0.057	0.115	0.210
GraphFC$_{\text{RGCN}}$	0.259	0.104	0.127	0.179	0.362	**0.334**
GraphFC	**0.402**	**0.162**	**0.172**	**0.200**	**0.406**	0.306

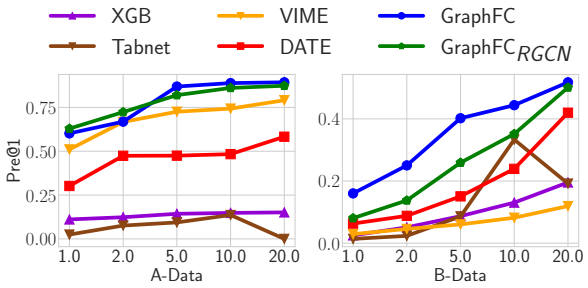


Figure 5: Performance comparison by varying inspection rate. The x-axis denotes the different inspection rate.

Table 2: Performance comparison on different training graphs under 5% inspection rate.

Model	n=1%			n=5%		
	Pre.	Rec.	Rev.	Pre.	Rec.	Rev.
B-Data						
Q($\mathcal{G}_L$) + F($\mathcal{G}_L$)	0.245	0.099	0.134	0.147	0.298	0.259
Q($\mathcal{G}_U$ + F($\mathcal{G}_L$)	0.278	0.112	0.141	0.186	0.377	0.304
Q($\mathcal{G}$) + F($\mathcal{G}_L$)	0.286	0.115	0.141	0.18	0.363	0.285
Q($\mathcal{G}_L$) + F($\mathcal{G}$)	0.292	0.118	0.149	0.163	0.329	0.269
Q($\mathcal{G}_U$) + F($\mathcal{G}$)	0.402	0.162	0.155	0.173	0.349	0.286
Q($\mathcal{G}$) + F($\mathcal{G}$)	**0.402**	**0.162**	**0.172**	**0.2**	**0.406**	**0.305**

5.3 Role of Unlabeled Data

GraphFC improves its representation ability by utilizing the unlabeled data in the pretraining and fine-tuning stage. The key to making use of unlabeled data lies in the construction of transaction graph as presented in Fig. 3. The transaction graph comprises of V_T and V_C, where V_T includes both labeled and unlabeled transactions. To verify the effectiveness of unlabeled data in V_T, two variants of

Table 3: Performance comparison on different subgroups of importers and HS codes in terms of various frequencies. The evaluation results with $n = 1\%$ on B-data are reported.

	GraphFC			DATE			
Subgroup	Pre.	Rec.	Rev.	Pre.	Rec.	Rev.	Illicit rate
$\text{Imp}_{[0]}$	89.85%	32.74%	25.58%	7.65%	2.79%	3.54%	2.74%
$\text{Imp}_{(0,10]}$	42.96%	16.93%	10.43%	20.32%	8.01%	6.77%	2.53%
$\text{Imp}_{(10,50]}$	28.11%	17.17%	25.13%	14.43%	8.80%	18.57%	2.15%
$\text{Imp}_{(50,100]}$	15.66%	10.48%	3.24%	12.05%	8.06%	11.19%	1.64%
$\text{Imp}_{(100,\infty)}$	12.20%	5.68%	23.45%	19.51%	9.09%	25.04%	1.50%
$\text{HS}_{[0]}$	48.51%	11.21%	12.22%	22.77%	5.26%	12.76%	4.36%
$\text{HS}_{(0,20]}$	33.56%	8.78%	9.27%	16.70%	4.37%	5.65%	3.82%
$\text{HS}_{(20,150]}$	34.01%	11.85%	18.12%	16.28%	5.67%	13.44%	2.87%
$\text{HS}_{(150,1000]}$	53.71%	27.78%	27.45%	9.57%	4.93%	4.06%	1.93%
$\text{HS}_{(1000,\infty)}$	83.89%	69.33%	77.79%	1.97%	1.62%	1.04%	1.21%

$\mathcal{G}$ are made with the following rule: **(a)** $\mathcal{G}_L$: keeps only the labeled transaction and its corresponding edges with V_C. **(b)** $\mathcal{G}_U$: keeps only the unlabeled transaction and its corresponding edges with V_C. We then compare the results using different graphs in the pretraining(Q) and fine-tuning (F) stage with 6 combinations and list the performance with a semi-supervised setting of B-Data in Table 2. Each row represents the graphs used in pretraining and fine-tuning, for example, $Q(\mathcal{G}_L) + F(\mathcal{G}_U)$ means pretraining with graph $\mathcal{G}_L$ and fine-tune on graph $\mathcal{G}_U$. Notably, only using labeled data (variant $Q(\mathcal{G}_L) + F(\mathcal{G}_L)$) yields the worst results. Incorporating unlabeled data in either pretraining or fine-tuning significantly improves performance. Fine-tuning with unlabeled data generally outperforms using only labeled data, highlighting the benefit of considering both types of information. Comparing $Q(\mathcal{G}_L)$, $Q(\mathcal{G}_U)$, and $Q(\mathcal{G})$, $Q(\mathcal{G}_L)$ is notably inferior, while $Q(\mathcal{G}_U)$ and $Q(\mathcal{G})$ yield similar performance. This further emphasizes the effectiveness of adding unlabeled data in pretraining, as it increases training instances and enhances embedding space connectivity for better predictions.

5.4 Performance on Test Subgropus

Within the realm of customs declarations, certain importer IDs and HS codes exhibit higher frequencies compared to others. To assess the influence of occurrence frequency on performance, we have partitioned the test set into several subgroups, as outlined below:

- **Importer**: We categorized importers into five subgroups as per their occurrence frequency in the training set. For instance, the subgroup denoted as $\text{Imp}_{[0]}$ consists of new importers who did not appear in the training set, while $\text{Imp}_{(0,10]}$ encompasses importers who made appearances between 1 and 10 times.
- **HS**: We divide HS codes into five subgroups in a similar way.

In the context of customs fraud detection, encountering unseen features like new HS-codes and importer IDs is inevitable due to the large volume of transactions. The results in Table 3 compare GraphFC and DATE on prediction performance and illicit ratio for various subgroups. Notably, active traders ($\text{Imp}_{(100,\infty)}$) show lower fraud rates than inactive traders ($\text{Imp}_{(0,10]}$ and $\text{Imp}_{[0]}$), suggesting that importers may potentially utilize new IDs to evade detection. Similar trends are observed for HS-codes, with unpopular items having a higher likelihood of being as illicit. Therefore, detecting fraudulent transactions involving new importers and items is of paramount importance, and it constitutes the primary focus of this work. The results demonstrate that GraphFC outperforms the DATE model, particularly on unseen data ($\text{Imp}_{[0]}$ and $\text{HS}_{[0]}$), highlighting the inductive capability of GNNs and their ability to achieve superior performance on previously unseen data. Additionally, GraphFC exhibits remarkable recall and precision rates for commonly-occurring HS codes. This can be attributed to the transaction interaction learning, which aids in capturing the distribution of transactions sharing the same item code and ultimately enhances the detection performance.

Table 4: Component Analysis

Model	n=1%			n=5%		
	Pre.	Rec.	Rev.	Pre.	Rec.	Rev.
A-Data						
$\text{GraphFC}_{\text{semi}}$	0.829	0.087	0.143	**0.548**	**0.289**	0.434
$\text{GraphFC}_{\text{joint}}$	0.807	0.061	0.135	0.491	0.249	0.401
$\text{GraphFC}_{\text{only}}$	0.764	0.08	0.115	0.49	0.258	0.382
$\text{GraphFC}_{\text{sparse}}$	0.712	0.075	0.123	0.473	0.25	0.376
GraphFC	**0.869**	**0.092**	**0.17**	0.535	0.282	**0.445**
B-Data						
$\text{GraphFC}_{\text{semi}}$	0.281	0.113	0.162	0.197	0.398	**0.367**
$\text{GraphFC}_{\text{joint}}$	0.218	0.076	0.092	0.105	0.203	0.212
$\text{GraphFC}_{\text{only}}$	0.068	0.027	0.069	0.034	0.069	0.157
$\text{GraphFC}_{\text{sparse}}$	0.067	0.07	0.12	0.044	0.234	0.359
GraphFC	**0.401**	**0.162**	**0.171**	**0.2**	**0.405**	0.304

5.5 Component Analysis

We evaluate the following settings for component analysis:

- **GraphFC**: Use full model.
- $\textbf{GraphFC}_{\textbf{semi}}$: Skip the unsupervised pre-train step.
- $\textbf{GraphFC}_{\textbf{joint}}$: Remove pretraining and jointly optimizing with self-supervised objective and Eq. 4.
- $\textbf{GraphFC}_{\textbf{only}}$: Remove the GBDT step of the model and utilize the data from transactions directly.
- $\textbf{GraphFC}_{\textbf{sparse}}$: Remove one of the categorical nodes utilized for building the graph structure and build a *sparser* graph.

Results of different variants are demonstrate in Table 4. It can be noticed that in general removing any component in GraphFC leads to degradation of performance with a few excpetions. In general, neither $\text{GraphFC}_{\text{semi}}$ nor $\text{GraphFC}_{\text{joint}}$ improves GraphFC which verifies the necessity our pretraining step. Specifically, the difference between GraphFC and $\text{GraphFC}_{\text{semi}}$, $\text{GraphFC}_{\text{joint}}$ are significant when n is small (e.g. $n = 1\%$) which is important since customs usually maintain a low inpsection rate lower than 5%.

6 DISCUSSION AND CONCLUSION

In this work, we propose GraphFC, a GNN based customs fraud detection model that is designed for a real-world setting with limited availability of ground truth labels. GraphFC optimizes the identification of illicit transactions while maximizing the additional revenue collected from these transactions. Extensive analysis on two real customs datasets exhibits substantially improved performance over various baselines. The proposed model offers a solution for customs aiming to implement automated fraud detection, even in cases where the administration has limited labeled data.

REFERENCES

[1] Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications* 68 (2016).

[2] Aderemi O Adewumi and Andronicus A Akinyelu. 2017. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management* 8, 2 (2017).

[3] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md. Rafiqul Islam. 2016. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems* 55 (2016), 278–288.

[4] Sercan O Arik and Tomas Pfister. 2019. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442* (2019).

[5] Jean-François Arvis, Lauri Ojala, Christina Wiederer, Ben Shepherd, Anasuya Raj, Karlygash Dairabayeva, and Tuomas Kiiski. 2018. Connecting to compete 2018. (2018).

[6] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. arXiv:1707.03815 [stat.ML]

[7] Canrakerta, Achmad Nizar Hidayanto, and Yova Ruldeviyani. 2020. Application of business intelligence for customs declaration: A case study in Indonesia. *Journal of Physics: Conference Series* 1444 (2020), 012028.

[8] Andrea Cerioli, Lucio Barabesi, Andrea Cerasa, Mario Menegatti, and Domenico Perrotta. 2019. Newcomb–Benford law and the detection of frauds in international trade. *Proceedings of the National Academy of Sciences* 116, 1 (2019), 106–115.

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *KDD*. 785–794.

[10] Zhe Chen and Aixin Sun. 2020. Anomaly Detection on Dynamic Bipartite Graph with Burstiness. In *2020 IEEE International Conference on Data Mining (ICDM)*. 966–971.

[11] Daniel de Roux, Boris Perez, Andrés Moreno, Maria del Pilar Villamil, and César Figueroa. 2018. Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. In *KDD*. 215–222.

[12] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing Graph Neural Network-Based Fraud Detectors against Camouflaged Fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.

[13] Jorge Jambeiro Filho and Jacques Wainer. 2008. HPB: A model for handling BN nodes with high cardinality parents. *JMLR* 9 (2008), 2141–2170.

[14] Anne-Marie Geourjon, Bertrand Laporte, Ousmane Coundoul, Massene Gadiaga, T Cantens, R Ireland, and G Raballand. 2013. Inspecting Less to Inspect Better. The Use of Data Mining for Risk Management by Customs Administrations'. *Reform by Numbers. Measurement Applied to Customs and Tax Administrations in Developing Countries, Washington DC: World Bank* (2013).

[15] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).

[16] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *ADKDD*. 1–9.

[17] Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2022. Auc-oriented graph neural network for fraud detection. In *Proceedings of the ACM Web Conference 2022*. 1311–1321.

[18] Michael Keen. 2003. *Changing Customs: Challenges and Strategies for the Reform of Customs Administration*. International Monetary Fund, USA.

[19] Sundong Kim, Yu-Che Tsai, Karandeep Singh, Yeonsoo Choi, Etim Ibok, Cheng-Te Li, and Meeyoung Cha. 2020. DATE: Dual Attentive Tree-aware Embedding for Customs Fraud Detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[20] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. arXiv:1611.07308 [stat.ML]

[21] Maria Krivko. 2010. A hybrid model for plastic card fraud detection systems. *Expert Systems with Applications* 37, 8 (2010), 6070–6076.

[22] Anuj Kumar and Vishnuprasad Nagadevara. 2006. Development of hybrid classification methodology for mining skewed data sets: A case study of Indian customs data. (2006).

[23] Yiğit Kültür and Mehmet Ufuk Çağlayan. 2017. Hybrid approaches for detecting credit card fraud. *Expert Systems* 34, 2 (2017), e12191.

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

[25] Mingxuan Lu, Zhichao Han, Susie Xi Rao, Zitao Zhang, Yang Zhao, Yinan Shan, Ramesh Raghunathan, Ce Zhang, and Jiawei Jiang. 2022. BRIGHT-Graph Neural Networks in Real-time Fraud Detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3342–3351.

[26] Kunio Mikuriya and Thomas Cantens. 2020. If algorithms dream of Customs, do customs officials dream of algorithms? A manifesto for data mobilization in Customs. *World Customs Journal* 14, 2 (2020), 3–22.

[27] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. 2018. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. *IEEE Transactions on Neural Networks and Learning Systems* 29, 8 (2018), 3784–3797.

[28] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. 2020. xFraud: Explainable Fraud Transaction Detection on Heterogeneous Graphs. arXiv:2011.12193 [cs.LG]

[29] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Mo Cheng, Yinan Shan, Yang Zhao, and Ce Zhang. 2020. Suspicious Massive Registration Detection via Dynamic Heterogeneous Graph Neural Networks. arXiv:2012.10831 [cs.LG]

[30] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack. 2016. *The geography of transport systems*. Routledge.

[31] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.

[32] Korea Customs Service. April 2023. *Korea Customs Week 2023*. https://koreacustomsweek.org/

[33] Hua Shao, Hong Zhao, and Gui-Ran Chang. 2002. Applying data mining to detect fraud behavior in customs declaration. In *Proceedings. International Conference on Machine Learning and Cybernetics*, Vol. 3. IEEE, 1241–1244.

[34] Ron Triepels, Hennie Daniels, and Ad Feelders. 2018. Data-driven fraud detection in international shipping. *Expert Systems with Applications* 99 (2018), 193–202.

[35] Jellis Vanhoeyveld, David Martens, and Bruno Peeters. 2019. Customs fraud detection: Assessing the value of behavioural and high-cardinality data under the imbalanced learning issue. *Pattern Analysis and Applications* (2019).

[36] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proc. of the ICLR*.

[37] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. arXiv:1809.10341 [stat.ML]

[38] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A Semi-Supervised Graph Attentive Network for Financial Fraud Detection. In *2019 IEEE International Conference on Data Mining (ICDM)*. 598–607.

[39] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. 2019. FdGars: Fraudster Detection via Graph Convolutional Networks in Online App Review System. In *Companion Proceedings of The 2019 World Wide Web Conference*. 310–316.

[40] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. TEM: Tree-enhanced embedding model for explainable recommendation. In *WWW*. 1543–1552.

[41] Jarrod West and Maumita Bhattacharya. 2016. Intelligent financial fraud detection: A comprehensive review. *Computers & Security* 57 (2016), 47–66.

[42] Less Wright. 2019. Ranger - a synergistic optimizer. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer.

[43] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. 2020. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems* 33 (2020), 11033–11043.

[44] Yong-Nan Zhu, Xiaotian Luo, Yu-Feng Li, Bin Bu, Kaibo Zhou, Wenbin Zhang, and Mingfan Lu. 2020. Heterogeneous Mini-Graph Neural Network and Its Application to Fraud Invitation Detection. In *2020 IEEE International Conference on Data Mining (ICDM)*. 891–899.