Enhancing Fraud Transaction Detection via Unlabeled Suspicious Records

Ye Wang[†*], Yunpeng Liu[†*], Ningtao Wang[£*], Peiyang Li[†], Jiahao Hu[£], Xing Fu[£], Weiqiang Wang[£], Kun Sun[¶], Qi Li[†], Ke Xu[‡]

[†]INSC, Tsinghua University, {wangye22, liuyp20,li-py23}@mails.tsinghua.edu.cn, qli01@tsinghua.edu.cn

[£]Ant Group, {ningtao.nt, hjh333867, zicai.fx, weiqiang.wwq}@antgroup.com

[¶]Department of Information Sciences and Technology, George Mason University, ksun3@gmu.edu

[‡]DCS, Tsinghua University & Zhongguancun Laboratory, xuke@tsinghua.edu.cn

***Abstract*—Deep learning-based classifiers have been widely used in the field of financial fraud transaction detection. However, training a high-performance classifier for fraud detection is challenging due to the lack of sufficient labeled fraud data. Particularly, it is difficult to detect stealthy fraud transactions that closely mimic benign user behaviors. We observe that the suspicious transactions identified by the online detection system can augment the feature space to improve the detection performance of machine learning-based models. In this paper, we propose a new framework GIANTESS to leverage suspicious transactions to augment the feature space and thus enhance the detection of stealthy fraud transactions. Our semi-supervised approach combines both labeled transactions and unlabeled suspicious transactions to train a detection model. Specifically, it first estimates pseudo labels of suspicious transactions and then combines the pseudo labels with ground truth labels to train the detection model. We conduct experiments on two real-world datasets to demonstrate the effectiveness of our proposed method on detecting stealthy fraud transactions. The experimental results show that GIANTESS successfully improves the recall by up to 6.3% at the fixed low false positive rate of 1%. We also perform a 9-week deployment test of our system in a real-world online payment platform to demonstrate the performance of GIANTESS.**

***Index Terms*—Fraud Detection, Deep Learning**

I. Introduction

An online payment system is a digital platform that facilitates the electronic transfer of money between parties over the Internet. It has become an indispensable application on the Internet, enabling users to make transactions from anywhere with an Internet connection. However, fraudsters may conduct unauthorized or deceitful activities in fraud transactions to obtain money or valuable assets through deceptive means. They often exploit weaknesses in security systems or manipulate individuals into disclosing sensitive information. It is crucial for financial institutions to implement robust security measures to detect fraudulent activities effectively.

A wide range of fraud detection techniques have been developed [1]–[8], and they can be classified into two categories, namely, rule-based methods and machine learning-based methods. Rule-based methods [1] detect fraud transactions according to fraud patterns discovered by experts, thus achieving good detection interpretability and deployability. However, they heavily rely on expert knowledge to establish and update rules and may fail to discover complex patterns. Machine learning-based methods [2]–[8] enable an automated approach to capturing complex fraud patterns from a large size of training data. Nevertheless, the effectiveness of these methods depends on the feature space of available transactions.

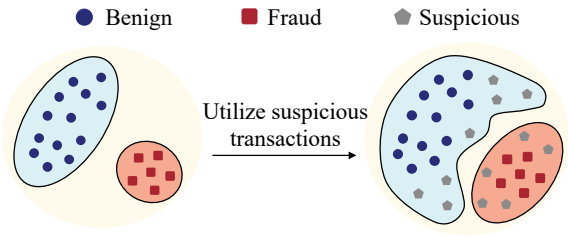


Fig. 1: The illustration of our idea. Utilizing suspicious transactions can augment the feature space and thus improve the performance of detection models.

Since fraud transactions are rarer comparing to benign transactions, it remains a challenge to distinguish the stealthy fraud transactions that have similar patterns and features to the benign transactions. For example, adversaries can compromise the accounts of normal real users [9], [10] or grow up accounts that imitate benign ones before conducting fraud transactions [11]. Such stealthy fraud transactions can easily evade existing detection systems due to their similarity to benign transactions.

In this paper, we aim to detect stealthy fraud transactions by utilizing unlabeled suspicious transaction records that are identified and interrupted by the existing detection systems. We observe that suspicious transactions include both potential fraud transactions and ambiguous benign transactions, thus providing rich and diverse information in the feature space. Therefore, suspicious transaction records are promising to effectively augment the feature space for improving the performance of detection models. Figure 1 illustrates this idea. To this end, we develop GIANTESS, a novel framework leveraging suspicious transactions to enhance the detection of stealthy fraud transactions. GIANTESS incorporates suspicious transactions into the training dataset of the detection model. The detection model is trained with suspicious transactions that are potentially fraud as a supplement to the labeled fraud

[*]Equal contribution

transactions, alleviating the risk of under-fitting caused by insufficient fraud transactions. Also, it utilizes the remaining ambiguous benign transactions to refine the decision boundary.

We face three challenges when leveraging suspicious transactions to better differentiate stealthy fraud transactions from benign ones. First, there are no accurate labels for suspicious transactions that have been interrupted. In other words, no external feedback is available since these transactions are not finished. This indicates that the exact proportion of suspicious transactions that are actual fraud remains uncertain, and this uncertainty is influenced by multiple factors, including the performance of the detection system and strategies applied by adversaries. Consequently, indiscriminately labeling suspicious transactions as all fraud or all benign will introduce massive label noise to the dataset, thereby degrading the model performance. Second, suspicious transactions exhibit a distinct distribution compared to completed transactions. This is because suspicious transactions conform to patterns defined by the existing detection system, while completed transactions do not. It complicates the process of accurately assigning labels to such transactions. Third, the volume of suspicious transactions is different in scale compared to either fraud or benign transactions. This imbalance may induce a bias in prediction.

To resolve these challenges, GIANTESS works in a semi-supervised manner in two stages to combine unlabeled suspicious transactions with labeled completed transactions. We generate pseudo labels for suspicious transactions to augment the input feature space of the model with these transactions. To overcome the first challenge, we propose a data augmentation-based training method to obtain a labeling model, which can produce scores to faithfully represent the risk of malicious transactions and serve as their pseudo soft labels. The data augmentation technique enables our model to generalize to both in-distribution and out-of-distribution samples, thereby solving the second challenge. Moreover, we obtain a detection model utilizing both labeled transactions and unlabeled suspicious transactions. To overcome the third challenge, it is trained using a specially designed hybrid loss that combines the pseudo soft labels with the ground-truth hard labels. The proposed loss function augments the hidden feature space by paying more attention to transactions that are more likely to be fraud, which alleviates the impact of class imbalance to ensure that the model fully utilizes the suspicious transactions.

We conduct an extensive evaluation of our proposed framework using two real-world datasets from a world-leading online payment platform. When fixing the false positive rate at 1%, GIANTESS successfully achieves an average improvement of the recall in the account takeover detection scenario by 6.3%, and the recall in the deception fraud scenario by 5.9%. By manually analyzing the additional account takeover fraud transactions detected by our method, we find that 45.9% of them are stealthy fraud transactions with similar patterns to benign ones. We also deploy our proposed framework in real-world production to demonstrate its effectiveness.

In summary, we make the following contributions:

- We propose a framework GIANTESS to find stealthy fraud transactions by incorporating suspicious transactions collected by online detection systems into the training of detection model.
- We design a method for generating pseudo labels, which performs data augmentation to train a labeling model that can generalize to both in-distribution and out-of-distribution samples. We also propose a hybrid loss training approach to train a detection model that combines pseudo labels with ground-truth labels and focuses on samples that are likely to be fraud.
- We conduct experiments on real-world data collected from an online payment platform, showing the effectiveness of GIANTESS to detect stealthy fraud transactions. We also perform real-world deployment test for a total of nine weeks.

II. Background and Problem Statement

In this section, we provide a brief overview of the fraud transaction detection system on online payment platforms and the identification process for suspicious transactions and then formalize our problem.

A. Background

Online payment platforms adopt fraud transaction detection systems that integrate rule-based methods and machine learning-based classifiers to capture patterns that are indicative of fraud behaviors. As illustrated in Figure 2, throughout the whole process of detection, we collect fraud, benign, and suspicious transactions. Specifically, the detection system examines each transaction submitted for execution. If the detection system identifies it as with more suspicion to be a fraud transaction, *i.e.*, is a suspicious transaction, the transaction will be interrupted. Otherwise, the transaction will proceed to be executed promptly. Once a transaction is executed, it is labeled according to external feedback with supporting evidence, *e.g.*, bank complaints and feedback from law enforcement. The platform responds to the feedback by manually examining whether the transaction is fraudulent. If no feedback is raised or the feedback is not confirmed to be fraud within a specific period from the execution of the transaction, this transaction will be automatically labeled as a benign transaction.

As discussed in the Introduction, it is non-trivial to leverage suspicious transactions to train detection models. Therefore, usually only fraud transactions and benign transactions are utilized as training samples in training supervised machine-learning models for fraud detection. Transactions identified as suspicious lack labeling because of the absence of subsequent feedback after the interruption, and are thus overlooked in the training phase. Nevertheless, these suspicious transactions manifest unique patterns that the detection system identifies, and excluding them from the model training could lead to an insufficient representation of the feature space. Particularly, the feature space of fraud samples is disproportionately impacted by the act of interrupting suspicious transactions, since it significantly diminishes the volume of fraud transactions, which in turn compromises profiling fraud samples.

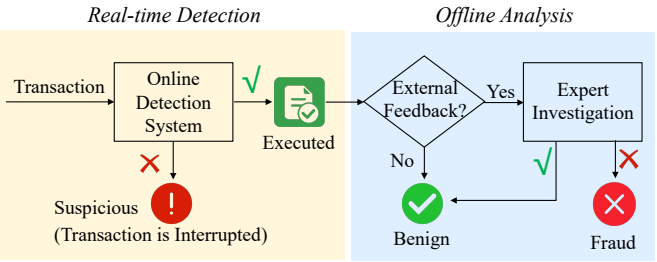


Fig. 2: The identification of benign, fraud, and suspicious transactions in an online detection system.

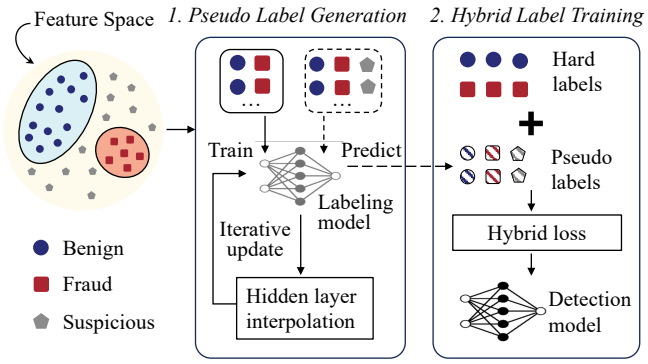


Fig. 3: The overview of our proposed framework.

B. Problem Statement

We consider that we are given both labeled transactions and unlabeled suspicious transactions on the online payment platform described in Section II-A, where each transaction is represented using numerical or categorical features with the same dimension. Labeled transactions consist of fraud transactions and benign transactions, referred to as fraud samples and benign samples, respectively. We consider labeled samples define the target distribution. Unlabeled transactions are suspicious to be fraud according to the fraud patterns defined by the detection systems. We designate these suspicious transactions as suspicious samples. Note that since unlabeled suspicious samples conform to fraud patterns while labeled samples do not, they follow a distribution different from the target distribution. We verify this claim in Section V. The underlying label of each suspicious sample should be either fraud or benign. Note that due to the nature of fraud detection systems, the quantities of fraud, benign, and unlabeled suspicious transactions exhibit significant disparities, with benign transactions predominating, followed by suspicious transactions, and fraud being the least.

We aim to develop a detection model that can distinguish between benign and fraud samples, *i.e.*, perform binary classification on the target distribution, using benign, fraud and suspicious samples for training. Formally, the dataset of labeled samples is represented by $D_L = \{(x^1, y^1), \cdots, (x^n, y^n)\}$, where x^i and $y^i \in \{0, 1\}$ denote the features and binary label of the i-th data samples. Here, $y^i = 0$ indicates a benign sample and $y^i = 1$ indicates fraud. Suspicious samples are represented by the set $D_U = \{x'^1, \ldots, x'^m\}$. We have $n \gg m$. The detection model S predicts samples that are from the same distribution as D_L. For each sample x, it outputs the probability of being fraud, denoted by $p^S(x)$.

It is worth noticing that such suspicious data is a common issue in online detection systems. Generally, online detection systems identify suspicious instances that lack high-quality labels, according to historical information. However, the proportion of suspicious instances being malicious might vary across different detection systems, and the number of labeled instances can also be significantly different according to different labeling strategies. For example, focusing on abuse account detection on online social networks, prior work leverages these suspicious accounts with a precision higher than 90%, and obtains fewer labeled accounts than these suspicious accounts [9]. In this paper, we consider a more generic setting where the proportion of suspicious instances being malicious is unknown. Moreover, since whether a transaction is fraud is determined once finished, we can determine the labels of the finished transaction in a time window, thus obtaining more labeled data than suspicious data.

III. Overview

We propose GIANTESS, a novel framework leveraging suspicious transactions to enhance the detection of stealthy fraud transactions. GIANTESS is a semi-supervised model training framework that generates accurate pseudo labels for out-of-distribution suspicious transactions and incorporates them into the detection model training to augment the input feature space. Besides, a novel hybrid loss function is proposed to effectively leverage the pseudo labels and ground truth labels simultaneously. Trained upon such an augmented input feature space with the proposed loss function, the detection model obtains a better decision boundary that can distinguish stealthy fraud transactions from benign transactions.

Figure 3 illustrates the workflow of our proposed framework. It follows the two stages below:

- **Pseudo label generation** estimates the pseudo labels of out-of-distribution suspicious samples to facilitate the training of the downstream detection model with an augmented feature space. To generate accurate pseudo labels, we design a general data augmentation-based training method to obtain a labeling model with all benign and fraud samples. The data augmentation is performed by interpolating the hidden feature space of the model under training, so that the trained labeling model possesses a smooth classification boundary that can generalize well on both out-of-distribution suspicious transactions and in-distribution transactions.
- **Hybrid label training** obtains a detection model with both labeled transactions and suspicious transactions. The model is trained using a specially designed hybrid loss that fully utilizes the labeling information of ground-truth hard labels and pseudo soft labels by paying more attention to wrongly classified transactions that are likely to be fraud. By training with the hybrid loss function, the model can more effectively distinguish stealthy fraud transactions with similar patterns to benign transactions.

IV. Design Details

In this section, we present the detailed designs of components in GIANTESS.

A. Pseudo Label Generation

To facilitate the model training with unlabeled suspicious transactions, the first step of GIANTESS is to generate pseudo labels for them with a labeling model. One trivial method for generating pseudo labels for unlabeled samples is directly training a classification model with the labeled transactions only, and then using the prediction of the model as the pseudo labels. However, this method requires that the features of unlabeled suspicious transactions follow exactly the same distribution as those of labeled transactions, which is not the case in real-world applications. A simply trained model can have poor OOD generalization performance and thus produce inaccurate pseudo labels. Considering that the volume of suspicious transactions is much larger than the labeled fraud transactions, such inaccurate pseudo labels may even downgrade the in-distribution performance of downstream models due to the overwhelming label noise. Therefore, our main goal is to generate accurate pseudo labels that can faithfully describe the risk level of both in-distribution and out-of-distribution transactions.

To achieve this goal, we propose to leverage the data augmentation technique to enhance the out-of-distribution generalization performance of the labeling model, thereby accurately predicting the risk level of the suspicious transactions as their pseudo labels. Intuitively, we interpolate the features and labels of labeled training samples, so that the model learns to predict the mixture of fraud and benign samples and finally gets a smoother decision boundary. Such data augmentation strategies are common methods to improve OOD generalization performance [12]. By augmenting the data, the hidden feature space of the model will be augmented, which enhances the quality of pseudo labels.

However, the features of transactions in the fraud detection task usually contain both numerical and categorical types. It is unreasonable to directly interpolate categorical features, since these features are discrete and directly interpolating them will break the semantics. To deal with this issue, we choose to perform interpolation on the output feature map of a selected hidden layer, following the idea of manifold mixup [13]. Such an interpolation-based augmentation method brings two benefits. First, it can naturally adapt to different fraud detection tasks since the feature maps of hidden layers are continuous no matter whether the input features are discrete or not. Second, the feature maps of hidden layers extract and amplify the risk-related patterns in the raw inputs, and thus interpolating the feature maps of benign and fraud samples is effective to improve the generalization performance on suspicious transactions and stealthy transactions whose patterns are similar to benign transactions.

The procedure for training the labeling model with the data augmentation technique is described below. First, for each batch of training samples, they are fed into the model under training and pass the forward computation of the first several hidden layers. After the computation of the selected hidden layer, the hidden feature maps of the batched samples are taken out. For each sample, we randomly choose another sample in the batch and interpolate their feature maps and labels linearly. Formally, let $h(x)$ denote the feature map of a sample with feature x after passing the selected hidden layer, the linear interpolation between a pair of samples $(x^i, y^i), (x^j, y^j) \in D_L$ is computed as follows:

$$h(x^{i,j}) = \lambda \cdot h(x^i) + (1 - \lambda) \cdot h(x^j), \tag{1}$$

$$y^{i,j} = \lambda \cdot y^i + (1 - \lambda) \cdot y^j. \tag{2}$$

The coefficient λ is sampled from a beta distribution $\mathrm{B}(\alpha, \alpha)$, where $\alpha \in (0, 1)$ is a hyper-parameter controlling the degree of interpolation. With a larger α, there is a higher probability that the sample pairs are interpolated evenly, which smooths the decision boundary at the cost of in-distribution performance since the feature map distribution is substantially changed. After interpolation, the interpolated batch of $h(x^{i,j})$ is fed back to the rest layers of the labeling model to perform the prediction, compute the loss against the interpolated labels $y^{i,j}$ and finally update parameters with backpropagation and gradient descent. Note that the interpolation calculation is differentiable, and thus all the layers in the model are trained together in an end-to-end manner.

After training the labeling model, we use the confidence score of the labeling model on suspicious samples as their pseudo labels. Instead of using the classification decision (*i.e.*, the hard label) of the labeling model, the confidences of suspicious transactions reflect the estimated risk level of these transactions. Using continuous confidence scores rather than discrete hard labels as pseudo labels can alleviate the problem of the downstream model being overconfident in classifying suspicious samples. Specifically, for each sample x, our labeling model, represented by T, generates a soft label $p^T(x)$, rather than a hard label that is either 0 or 1.

B. Hybrid Label Training

After obtaining the pseudo labels, we use the labels to train the downstream fraud detection model. To fully utilize both suspicious samples and labeled fraud and benign samples, we design a novel hybrid label training method, which combines soft pseudo labels and hard ground-truth labels and employs a novel loss function to focus on the fraud transactions that are similar to benign ones. By utilizing the unlabeled suspicious transactions, the input feature space is augmented, and the design of our hybrid loss function further augments the hidden feature space by paying more attention to the wrongly classified fraud transactions.

The design of our hybrid label loss function contains two parts, *i.e.*, the pseudo label loss and the hard label loss. The pseudo label loss calculates the discrepancy between the outputs of the detection model under training and the risk level of transactions estimated by the labeling model. To achieve this, one straightforward way is to compute the distribution distance between the detection model and the labeling model, *e.g.*, by

calculating the Kullback–Leibler divergence (KL divergence) between the two distributions. However, in real-world fraud transaction detection scenarios, the magnitude of benign transactions is far greater than that of fraud transactions. Therefore, the distributions of model outputs and labels are highly skewed toward benign, making it hard to optimize the distribution distance metric. Consequently, the detection performance on fraud transactions will be impacted.

To deal with this issue, we develop a simple yet effective loss function based on the KL divergence that can give more attention to wrongly classified transactions with high risk, yet less attention to the already correctly classified ones. The loss function is calculated as follows. First, given the sample x, we calculate the detection model's prediction $p^M(x)$ to the kth power. Then, we calculate the KL divergence between the result and the pseudo label as the pseudo label loss, denoted as $\mathcal{L}_{PL}$. The calculation is as follows:

$$\mathcal{L}_{PL} = \sum_{x \in D_U \cup D_L} \Big[p^T(x) \cdot \log \frac{p^T(x)}{p^M(x)^k} + (1 - p^T(x)) \cdot \log \frac{1 - p^T(x)}{1 - p^M(x)^k} \Big], \tag{3}$$

where $p^T(\cdot)$ denotes pseudo labels generated by the labeling model. The hyper-parameter k controls the importance of high-risk samples in the loss function. Note that both unlabeled suspicious transactions and labeled transactions are used to calculate $\mathcal{L}_{PL}$ to augment both the input and hidden feature spaces.

Now we provide the insight behind our pseudo label loss function. Mathematically, the difference between our pseudo label loss and the vanilla KL divergence $\mathcal{L}_{KL}$ (*i.e.*, $\mathcal{L}_{PL}$ when $k = 1$) is:

$$\mathcal{L}_{PL} - \mathcal{L}_{KL} = \sum_{x \in D_L \cup D_U} \Big[(1-k) p^T(x) \cdot \log p^M(x) + (1 - p^T(x)) \cdot \log \frac{1 - p^M(x)}{1 - p^M(x)^k} \Big]. \tag{4}$$

Since the model prediction is between 0 and 1, $\log p^M(x)$ is always negative. Therefore, we have the observation that when k is larger than 1, the first item of the right side of Equation (4) is greater than 0, while the second item is smaller than 0. This indicates that, when using our proposed loss instead of the vanilla KL divergence, the loss values of fraud transactions will be enlarged while the loss values of benign transactions will be reduced. Recall that the number of fraud transactions is far less than that of benign transactions. Such behavior of our loss function helps to fully utilize fraud transactions.

We also visualize the landscape of the pseudo label loss when $k = 2$ in Figure 4 as an example. It can be seen that the loss function yields a larger value when a fraud transaction is mistakenly classified as a benign one, a relatively smaller value when a benign transaction is classified as a fraud one, and the smallest value when the prediction of the detection model and the labeling model are similar. Therefore, compared to other methods for dealing with the class-imbalance problem such as assigning the fraud transactions a large sample weight,

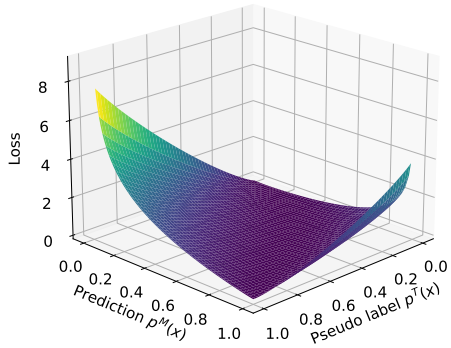


Fig. 4: The visualization of our pseudo label loss when $k = 2$.

our pseudo loss function pays more attention to the wrongly classified fraud ones, *i.e.*, ones significantly going beyond the decision boundary in the hidden feature space. By optimizing our loss function, these samples will leave the manifold of benign transactions in the hidden space and the decision boundary of the trained model will become more clear. This can enhance the detection performance on stealthy fraud transactions that are similar to benign ones. Moreover, the introduction of unlabeled suspicious transactions in the pseudo label loss function also augments the input feature space.

The proposed pseudo label loss only minimizes the differences between the predictions of the detection model and the labeling model, regardless of whether the prediction results are consistent with the ground truth. Therefore, we also calculate the cross entropy between the predictions of the detection model on labeled transactions and the corresponding ground truth hard labels. The final loss function is as follows:

$$\mathcal{L} = \epsilon \cdot \mathcal{L}_{PL} + (1 - \epsilon) \cdot \mathcal{L}_{CE}, \tag{5}$$

where ϵ is a hyper-parameter that balances the pseudo label loss and the hard label loss $\mathcal{L}_{CE}$ (*i.e.*, the cross entropy loss between $\log p^M(x)$ and the hard label). By optimizing the loss above using both suspicious and labeled transactions, we can finally obtain a detection model with augmented input and hidden feature space.

V. Evaluation

In this section, we demonstrate the effectiveness of GIANTESS with extensive experiments on real-world datasets collected from a world-leading online payment platform. The evaluations are carried out to answer the questions below:

- *Q1*. How effective is our proposed framework in detecting stealthy fraud transactions?
- *Q2*. How does each component in our framework contribute to the detection?
- *Q3*. How robust is our proposed framework under different parameter settings?

A. Experiment Settings

Datasets. We collect two datasets of different fraud detection scenarios provided by a world-leading online payment platform. The two scenarios are as below:

TABLE I: Size of datasets.

		# Fraud	# Suspicious	# Benign
Account Takeover	Train	1,098	532,650	6,316,907
	Valid	265	-	1,584,670
	Test	1,256	-	9,037,962
Deception Fraud	Train	1,109	748,910	6,316,907
	Valid	278	-	1,584,670
	Test	1,374	-	9,037,962

- *Account takeover*. In this scenario, the fraudsters take over the victims' accounts via fishing scams, social engineering, or other techniques, and then they conduct fraud transactions using the stolen accounts.
- *Deception fraud*. In this scenario, the fraudsters use various deception methods, such as telecom fraud and online cheating, to induce victims to transfer money to them.

The datasets are described in Table I. For each dataset, all transactions are split into the training set, the validation set, and the testing set. It can be seen that the datasets are extremely imbalanced, where the number of benign samples is hundreds of times of that of fraud samples. Note that the validation set and testing set do not contain suspicious transactions. This is because these transactions are detected and interrupted by the rule-based detection system and our method focuses on detecting the fraud transactions that escaped the previous detection. We collect the ground truth of transactions following the procedure described in Section II-A. Each transaction has 242 features, 30 of which are categorical and the others are numerical. These features essentially describe the security-related information as well as the historical trading behaviors of the transactors. Due to privacy and commercial confidentiality issues, the exact meanings of feature columns are not available. To validate that suspicious transactions are out-of-distribution, we train OneClassSVM outlier detectors using 20000 labeled transactions sampled from the training set of each scenario and perform detection on the suspicious transactions. Results show that 44.0% of suspicious transactions in the account takeover scenario are outliers, and 54.7% of those in the deception fraud scenario are outliers. This demonstrates our claim that suspicious samples follow a distribution different from the target distribution.

Metrics. We evaluate the detection performance using AUC, which is widely used in previous work since fraud transactions only comprise a very small proportion of all transactions. To evaluate whether fraud transactions are accurately detected without compromising legitimate users' experiences, we also report the recall at a false positive rate of x (*i.e.*, Recall@FPR=x%). Specifically, the false positive rate—the ratio between the number of benign transactions wrongly classified as fraud and the total number of benign transactions—is fixed at several low rates $x \in [0.01, 0.1, 0.5, 1, 2, 5, 10]$. We report the average scores of each metric after a 5-time repeat with the standard deviation.

Baselines. To demonstrate that our method successfully utilizes the suspicious samples to enhance the fraud detection performance, we compare our method with the methods below, which do not utilize the suspicious samples or utilize them without fully considering the special characteristics of suspicious samples:

- ***MLP***. This model is only trained using labeled samples. Therefore, it does not utilize suspicious samples.
- ***SasF***. This model is trained with the suspicious samples by treating the suspicious samples as fraud samples to alleviate the issue of insufficient fraud samples.
- ***SasB***. This model is trained with the suspicious samples by treating them as benign samples.
- ***ED***. This baseline treats suspicious samples as benign or fraud samples according to their Euclidean distances to benign and fraud samples. Specifically, for each sample, we compute its distance to the mean of features of benign and fraud samples, respectively. The sample is labeled as fraud if its distance to the mean of fraud samples is smaller, and vice versa.

Configurations and hyper-parameters. We implement all algorithms using the PyTorch framework. We choose to use an MLP model with 3 hidden layers as our base model, and the hidden dimension of each layer is 128. We interpolate the feature maps of the first hidden layer. All the models are trained with the Adam optimizer under a learning rate of 2×10^{-4} and a weight decay of 1×10^{-6}. Without further notification, we set the hyper-parameter α of the beta distribution in data augmentation to 0.2, the k of $\mathcal{L}_{PL}$ to 2, and the balance coefficient ϵ of the loss function to 0.5.

Ethical Considerations. The data we use is preprocessed as tabular data with no sensitive user information. We access all datasets that are stored on the company's devices through an internship program. To mitigate any potential disruption to the production environment, we conduct experiments in an isolated environment.

B. Effectiveness

In this section, we evaluate the effectiveness of GIANTESS by comparing its detection performance against the baselines. We implement baselines and our method and evaluate them on the two datasets. The evaluation results are shown in Table II.

From the evaluation results, we can observe that our method achieves the best performance on both two datasets. For example, in the account takeover scenarios, our method improves the AUC by 1.07% and Recall@FPR=1% by 6.27% compared to *MLP* which does not leverage the suspicious samples. In the deception fraud scenario, our method also improves the performance *MLP* by 0.46% in AUC and 5.94% in Recall@FPR=1%, which demonstrates that our method successfully utilizes the information in suspicious samples to enhance the detection of fraud transactions. Note that by treating suspicious samples as fraud samples, the performance of *SasF* is consistently worse than that of *MLP*. This validates our assumption that suspicious samples exhibit a distinct pattern from the fraud samples, and directly using them as fraud samples will harm the detection performance on in-distribution fraud samples. Meanwhile, the feature distribution

TABLE II: Fraud detection performance of GIANTESS and baselines

Dataset	Method	Recall@FPR=x%							AUC
		0.01	0.1	0.5	1	2	5	10	
Account Takeover	*MLP*	13.185 ± 0.765	27.261 ± 0.268	42.452 ± 1.431	50.048 ± 1.050	58.439 ± 0.528	69.570 ± 1.036	78.201 ± 0.889	92.431 ± 0.330
	SasF	3.025 ± 0.149	11.369 ± 0.379	20.892 ± 0.316	24.713 ± 0.155	28.328 ± 0.508	43.949 ± 2.827	68.742 ± 1.691	88.585 ± 0.628
	SasB	10.080 ± 0.641	19.411 ± 1.125	33.025 ± 0.815	41.210 ± 1.920	50.892 ± 2.031	64.793 ± 1.167	75.207 ± 1.474	91.466 ± 0.432
	ED	2.245 ± 0.248	8.726 ± 0.249	15.334 ± 0.145	18.455 ± 0.362	32.373 ± 2.485	60.748 ± 1.904	74.029 ± 1.387	90.076 ± 0.567
	GIANTESS	14.029 ± 0.572 ⇑ 6.40%	29.236 ± 0.655 ⇑ 7.24%	45.621 ± 0.687 ⇑ 7.46%	53.185 ± 0.386 ⇑ 6.27%	60.876 ± 0.813 ⇑ 4.17%	72.643 ± 0.388 ⇑ 4.42%	81.003 ± 0.670 ⇑ 3.58%	93.422 ± 0.245 ⇑ 1.07%
Deception Fraud	*MLP*	8.865 ± 1.070	28.020 ± 2.523	50.291 ± 2.524	62.445 ± 1.681	73.508 ± 0.982	85.953 ± 0.357	92.547 ± 0.151	96.997 ± 0.107
	SasF	3.261 ± 0.581	19.010 ± 1.184	42.678 ± 1.864	56.099 ± 1.217	69.418 ± 0.476	84.003 ± 0.762	91.674 ± 0.473	96.732 ± 0.099
	SasB	4.367 ± 0.186	17.278 ± 1.445	38.311 ± 2.420	50.495 ± 1.745	63.799 ± 1.444	79.854 ± 0.788	89.316 ± 0.924	95.990 ± 0.255
	ED	4.410 ± 0.351	22.009 ± 0.865	47.322 ± 2.009	59.403 ± 1.229	70.655 ± 0.759	83.464 ± 0.710	91.092 ± 0.823	96.659 ± 0.248
	GIANTESS	10.582 ± 0.610 ⇑ 19.37%	30.189 ± 1.271 ⇑ 7.74%	54.454 ± 1.001 ⇑ 8.28%	66.157 ± 1.058 ⇑ 5.94%	76.259 ± 0.962 ⇑ 3.76%	88.108 ± 0.623 ⇑ 2.51%	94.178 ± 0.632 ⇑ 1.76%	97.450 ± 0.060 ⇑ 0.46%

of suspicious samples is closer to fraud samples than benign samples, and thus treating them as benign samples introduces label noises and downgrades the performance of *SasB*. In contrast, our method designs a pseudo label generation method to yield soft labels for unlabeled suspicious samples which can estimate the risk level of suspicious samples, and further uses the hybrid label model training method to incorporate the hard label and soft label to enhance the detection performance for stealthy fraud transactions.

To gain insight into the detection performance, we ask for the help of the security experts in our cooperate online payment platform to perform detailed manual annotation of the fraud samples that our method successfully detects while the baseline *MLP* fails to detect when fixing FPR at 1%. The annotation is based on the expert experience of typical fraud events and the communication feedback between the customer service and the victims. We perform this analysis in the account takeover scenario, because it is a more difficult scenario for detection. We fix the FPR at 0.5%, which is a reasonable threshold in production to avoid excessive false alarms, and analyze the manual labels of detected false positives at this threshold. We find that, among all fraud transactions that our method can detect while *MLP* fails to, 45.9% are suspected to be conducted by the relatives of the victims or mistakenly conducted by the victims themselves. Since these transactions are conducted by people around the victims, they do not exhibit obvious anomalous characteristics such as device and location changes. Meanwhile, 29.3% of the extra samples detected are attributed to typical frauds such as information leakage, phishing scams or device lost, indicating that our method can also strengthen the detection performance using suspicious samples. In conclusion, the performance improvement of our method is largely due to the successful detection of stealthy fraud transactions which do not have obvious abnormal features. Suspicious samples also support the detection of fraud transactions whose patterns are close to existing fraud patterns.

C. Ablation Study

To validate the design of our framework, we evaluate the contribution of the components in our framework on the final detection performance. We remove different components of our framework and test the detection performance of the trained models. The compared methods are as follows:

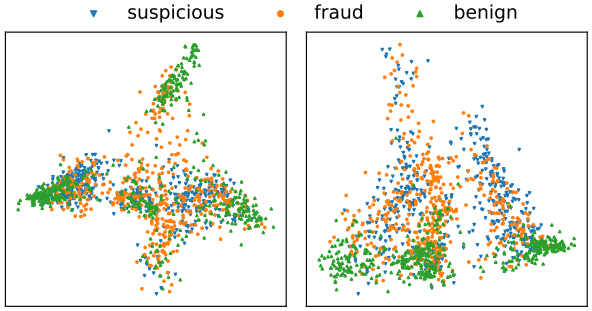
- ***MLP***. The model is trained with only labeled samples, *i.e.*, the fraud and benign transactions. All the suspicious samples are not used to train this model.
- ***DA***. The model is trained with only labeled samples under the data augmentation method of our framework, without utilizing the suspicious transaction samples.
- ***SS***. The model is trained under hybrid label training with both labeled and suspicious samples, while the data augmentation technique is not used to train the labeling model.

The evaluation results are shown in Table III. It can be seen that both the data augmentation method in the pseudo label generation component and the utilization of suspicious samples by the hybrid model training component contribute to the detection performance. By combining the two components, our method can achieve the best fraud detection performance. Note that, in the deception fraud detection scenario, the AUC of our method is slightly lower than that of *SS* by 0.015%, and similarly the AUC of *DA* is slightly lower than *MLP*, which we believe is because the data augmentation introduces additional noises by changing the feature map distribution. However, introducing our data augmentation technique significantly improves the recall at low FPR rates. For example, in the deception fraud scenario, the data augmentation component enhances Recall@FPR=0.5% for 1.86% compared to the vanilla MLP model. In real-world applications, recall at low false positive rates is a more important metric because a higher recall allows the fraud detection framework to interrupt more stealthy fraud transactions while minimizing the disruption to normal users. Therefore, the data augmentation component improves the practical value of our method.

We further dive into the hidden feature space of models to demonstrate how our method augments the feature space. We extract the hidden feature maps of transactions generated by the last hidden layers of the vanilla MLP model (*i.e.*, *MLP*) and model trained by our method. Then we visualize the hidden feature space using the PCA algorithm [14] to reduce the feature dimension. Figure 5 presents the visualization using *MLP* and GIANTESS. It can be seen that the hidden feature

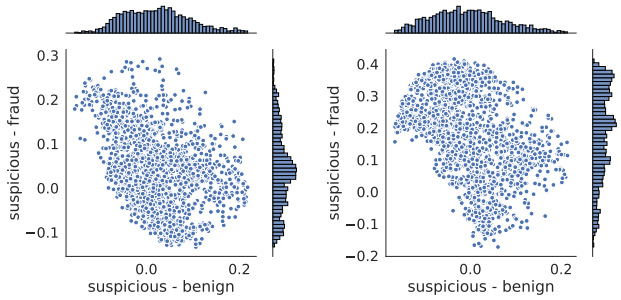
TABLE III: Performance of the proposed framework and models trained using different components of our framework.

Dataset	Method	Recall@FPR=x%							AUC
		0.01	0.1	0.5	1	2	5	10	
Account Takeover	*MLP*	13.185 ± 0.765	27.261 ± 0.268	42.452 ± 1.431	50.048 ± 1.050	58.439 ± 0.528	69.570 ± 1.036	78.201 ± 0.889	92.431 ± 0.330
	DA	14.108 ± 0.502	28.455 ± 0.502	43.455 ± 0.946	51.242 ± 1.302	59.825 ± 1.257	71.099 ± 0.964	80.223 ± 1.008	92.950 ± 0.170
	SS	13.137 ± 0.347	28.360 ± 0.472	44.522 ± 1.033	52.134 ± 0.358	60.287 ± 0.762	71.608 ± 0.767	80.016 ± 0.766	93.078 ± 0.375
	GIANTESS	14.029 ± 0.572	29.236 ± 0.655	45.621 ± 0.687	53.185 ± 0.386	60.876 ± 0.813	72.643 ± 0.388	81.003 ± 0.670	93.422 ± 0.245
Deception Fraud	*MLP*	8.865 ± 1.070	28.020 ± 2.523	50.291 ± 2.524	62.445 ± 1.681	73.508 ± 0.982	85.953 ± 0.357	92.547 ± 0.151	96.997 ± 0.107
	DA	8.661 ± 0.870	29.563 ± 0.497	52.154 ± 1.016	62.547 ± 0.265	73.566 ± 0.351	86.084 ± 0.487	92.387 ± 0.359	96.814 ± 0.127
	SS	10.044 ± 0.677	29.229 ± 1.016	53.421 ± 1.802	64.876 ± 1.306	75.604 ± 0.910	87.715 ± 0.319	93.945 ± 0.508	97.464 ± 0.037
	GIANTESS	10.582 ± 0.610	30.189 ± 1.271	54.454 ± 1.001	66.157 ± 1.058	76.259 ± 0.962	88.108 ± 0.623	94.178 ± 0.632	97.450 ± 0.060



(a) *MLP* (b) Giantess

Fig. 5: PCA visualization of the hidden feature space of models.



(a) *MLP* (b) GIANTESS

Fig. 6: Embedding similarity of models.

maps of suspicious samples and fraud samples extracted by the vanilla MLP model are heavily entangled with those of benign samples, which means that the model cannot well separate the fraud and benign transactions. In contrast, the hidden features of fraud and suspicious samples generated by our method are well separated from those of benign samples, demonstrating that our method successfully enhances the hidden feature space of the trained model. Besides, we also study the similarity of hidden feature maps of suspicious samples to those of fraud and benign samples. We sample 300 suspicious transactions and calculate their averaged cosine similarity to all fraud and benign transactions in the hidden feature space and plot the similarity in Figure 6. It is obvious that our method enlarges the similarity between the hidden features of suspicious and fraud samples, which is in line with our assumption that the suspicious samples are close to fraud transactions.

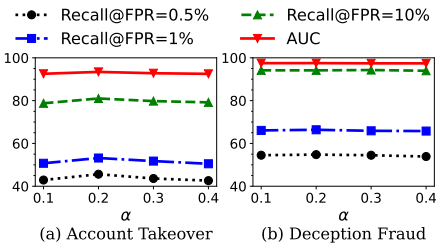
D. Parameter Sensitivity Analysis

In real-world deployment, careful parameter fine-tuning for fraud detection models is expensive due to the huge amount of data and the long training and verification cycle. Therefore, it is more preferable to have a method that is less sensitive to the hyper-parameters. We evaluate the performance of our method under different hyper-parameters, *i.e.*, α for controlling the sample interpolation coefficient, the fraud importance parameter k of the loss function, and the loss balance parameter ϵ. Specifically, we vary the value of α within the range of $[0.1, 0.2, 0.3, 0.4]$, the value of k in $[1, 2, 3]$ and the value of ϵ in $[0.3, 0.4, 0.5, 0.6, 0.7]$. When evaluating each hyper-parameter, the other hyper-parameters are fixed at their default values (*i.e.*, $\lambda = 0.2$, $k = 2$, and $\alpha = 0.7$). For each setting, we train the model under both account takeover and deception fraud settings and report the Recall@FPR=[0.5%, 1%, 10%] and the AUC score. Evaluation results are reported in Figures 7, 8 and 9, respectively.

It can be seen that the model achieves the best performance when $\alpha = 0.2$, and the metrics slightly reduce with the increase of α. The performance drop in the account takeover scenario is higher than that in the deception fraud scenario. We believe this is because the differences between account takeover fraud samples and benign samples are harder to identify, and interpolating these samples with an overly high ratio will make it even more difficult for the model to learn the identification of the fraud pattern. Nonetheless, there is an obvious improvement in recall at low false positive rates when $\alpha = 0.2$, which improves the detection ability of stealthy fraud transactions.

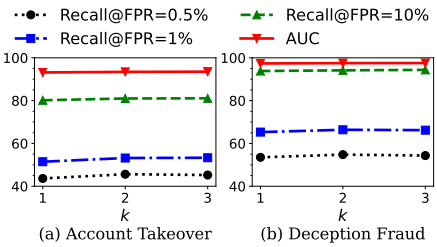
Besides, Figures 8 and 9 show that our framework is not sensitive to the choosing of fraud importance parameter k and loss balance parameter ϵ. Note that the pseudo label loss degenerates to the vanilla KL divergence loss when $k = 1$. When increasing k from 2 to 3, the AUC score in the account takeover scenario slightly increases from 93.4% to 93.5%, but the Recall@FPR=0.05% decreases from 45.6% to 45.2%. In application, the recall at low false positive rates is more important because it reflects the ability of the model to detect stealthy fraud transactions without disturbing benign users. Therefore, we still choose to set $k = 2$.

In conclusion, our method is not sensitive to the choosing of the fraud importance parameter k and the loss balance parameter ϵ, and models trained with $\alpha = 0.2$ can yield a



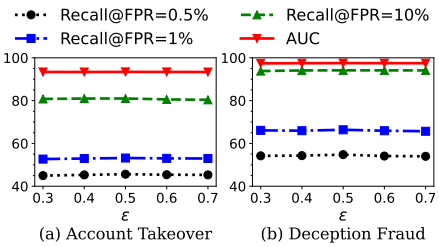
(a) Account Takeover (b) Deception Fraud

Fig. 7: Impact of the sample interpolation hyper-parameter α on the fraud detection performance.



(a) Account Takeover (b) Deception Fraud

Fig. 8: Impact of the fraud importance parameter k on the fraud detection performance.



(a) Account Takeover (b) Deception Fraud

Fig. 9: Impact of the loss balance parameter ϵ on the fraud detection performance.

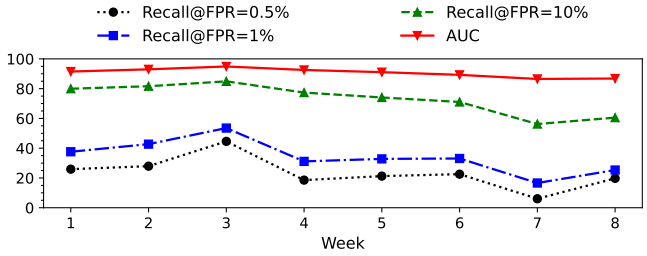


Fig. 10: Detection performance over time in the account takeover scenario.

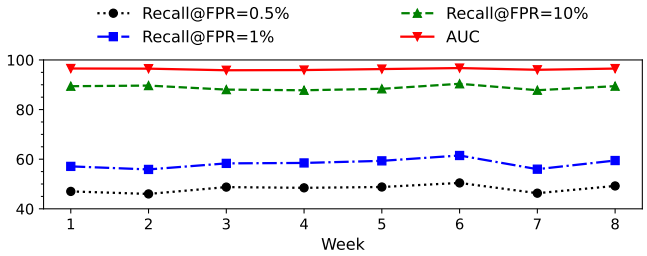


Fig. 11: Detection performance over time in the deception fraud scenario.

consistently superior performance under different scenarios.

VI. Real-World Deployment

To understand the fraud detection performance of our method in real-world applications, we deploy our framework on the platform. The real-world deployment test was conducted under the two scenarios for a total of 9 weeks, starting from June 25th, 2023. We use the first six days of data to train the fraud detection model using our framework and take the seventh day of data as the validation dataset. After training the model, we evaluate the model in the following eight weeks.

A. Performance

The model performs prediction on tens of millions of transactions each week. Finally, the transactions are labeled based on the fraud victims' complaints and manual verification by customer service. We calculate the evaluation metrics in each week and plot the metrics of the two scenarios in Figures 10 and 11, respectively.

It can be seen that models in both scenarios maintain relatively stable performance in the first four weeks. For example, in the account takeover scenario, the Recall@FPR=10% slightly reduces from 79.9% to 77.3%, while the AUC even increases from 91.5% to 92.6%. After the fourth week, the detection performance in the account takeover scenario begins to reduce gradually. However, the detection performance in the deception fraud scenario is still rather stable. For example, the Recall@FPR=10% keeps in the range from 87% to 90%, and the AUC is always around 96.5%. We believe this is because concept drift occurs over time in the fraud patterns of account takeover behaviors, while the fraud patterns in the deception fraud scenario are relatively stable. It is an interesting future work to explore how to leverage suspicious transactions to alleviate the concept drift in the account takeover scenario.

TABLE IV: Case studies of detected fraud transactions.

Type	Real-world Example
Account takeover due to device lost.	The user lost the device and then his account is used to conduct fraud transactions.
Account takeover due to fraud.	The user is induced to give the password and verification code when conducting online payment.
Account takeover by relatives.	The user's device is operated by one of his relatives to transfer money to the relative's account.

B. Case Study

To further understand the effectiveness of our method, we randomly sample some detected account takeover fraud transactions and ask for help from the security operators and customer service of our cooperating platform to investigate these cases. The feedback obtained from the users related to detected fraud transactions is analyzed. In the analysis, to protect user privacy, sensitive user information is not used. The analysis results are shown in Table IV. It can be seen that our method successfully detects different categories of account takeover fraud. Note that account takeover fraud by relatives is covert because there are no common abnormal patterns, such as device and IP changes, etc. Therefore, our method successfully detects stealthy fraud transactions.

VII. Related Work

Fraud Detection. With the popularity of e-commerce and social networks, fraud detection has been widely studied

in academia and industry. Existing methods can be divided into account and transaction levels based on the focused tasks. Account-level detection methods extract the fraudulent patterns of accounts based on users' historical behaviors [9], [15]–[19]. For example, Ianus [16] extracts synchronization and anomaly-based features for account pairs to perform detection. DEC [9] proposes a multi-stage framework to aggregate the properties and behavioral features from account graphs into deep features and perform account classification. In general, these methods focus on discovering account-level fraud patterns in specific scenarios, while we aim at designing a general method to leverage unlabeled suspicious transactions. Transaction-level detection methods focus on discovering the malicious behaviors of accounts, which is more fine-grained [3]–[8]. [3] proposes a subgraph extraction method to extract the anomalous subgraph in the transaction graph. TTAGN [4] designs a temporal aggregation graph network to utilize the temporal relationship between transactions. [5] uses the recurrent neural network to detect credit card frauds in real time. These methods focus on specific scenarios and are designed based on the special data formats in the scenarios. Compared to them, our framework aims to design a general framework to make use of suspicious transactions and is orthogonal to these methods.

Semi-supervised Learning with OOD Data. Existing literature has studied utilizing unlabeled out-of-distribution data to improve the in-distribution performance [20]–[23]. OAT [20] assumes that OOD data share the same undesirable features as the in-distribution data, and assigns labels sampled from the uniform distribution to OOD data to reduce the impact of those undesirable features. Open-sampling [21] assigns randomly sampled labels for OOD data based on class priors to alleviate the long-tailed class distribution in the multi-class classification task. COLT [22] designs a contrastive learning-based framework that leverages the OOD data to alleviate the long-tailed learning problem. These methods are based on different assumptions from our framework, *e.g.*, the OOD data belong to classes that do not exist in the in-distribution data or the task performs multi-class classification. Therefore, they cannot be applied to our scenario.

VIII. Conclusion

In this paper, we propose a novel framework, GIANTESS, to enhance the detection of stealthy fraud transactions by leveraging unlabeled suspicious records. We develop methods to combine suspicious transactions with labeled transactions to augment the feature space. We conduct experiments on two real-world datasets and real-world deployment to demonstrate its effectiveness for unveiling stealthy frauds that closely mimic benign user behaviors.

Acknowledgment

We thank all anonymous reviewers for their valuable comments. This work was supported in part by the National Science Foundation for Distinguished Young Scholars of China under No. 62425201, the Key Program of the National Natural Science Foundation of China under No. 62132011 and No. 61932016, the Science Fund for Creative Research Groups of the National Natural Science Foundation of China under No. 62221003, and the Ant Group Research Fund. Qi Li is the corresponding author of this paper.

References

[1] PayPal, "Rules vs machine learning for effective fraud prevention," 2023. [Online]. Available: https://www.paypal.com/us/brc/article/fraud-prevention-with-rules-vs-machine-learning

[2] F. Shi, Y. Cao, Y. Shang, Y. Zhou, C. Zhou, and J. Wu, "H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections," in *WWW*. ACM, 2022, pp. 1486–1494.

[3] T. Chen and C. E. Tsourakakis, "Antibenford subgraphs: Unsupervised anomaly detection in financial networks," in *KDD*. ACM, 2022.

[4] S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "TTAGN: temporal transaction aggregation graph network for ethereum phishing scams detection," in *WWW*. ACM, 2022, pp. 661–669.

[5] B. Branco, P. Abreu, A. S. Gomes, M. S. C. Almeida, J. T. Ascensão, and P. Bizarro, "Interleaved sequence rnns for fraud detection," in *KDD*. ACM, 2020, pp. 3101–3109.

[6] C. Liu, L. Sun, X. Ao, J. Feng, Q. He, and H. Yang, "Intention-aware heterogeneous graph attention networks for fraud transactions detection," in *KDD*. ACM, 2021, pp. 3280–3288.

[7] Y. Elmougy and L. Liu, "Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics," in *KDD*. ACM, 2023, pp. 3979–3990.

[8] X. Li, S. Liu, Z. Li, X. Han, C. Shi, B. Hooi, H. Huang, and X. Cheng, "Flowscope: Spotting money laundering based on graphs," in *AAAI*. AAAI Press, 2020, pp. 4731–4738.

[9] T. Xu, G. Goossen, H. K. Cevahir, S. Khodeir, Y. Jin, F. Li, S. Shan, S. Patel, D. Freeman, and P. Pearce, "Deep entity classification: Abusive account detection for online social networks," in *USENIX Security Symposium*. USENIX Association, 2021, pp. 4097–4114.

[10] P. Doerfler, K. Thomas, M. Marincenko, J. Ranieri, Y. Jiang, A. Moscicki, and D. McCoy, "Evaluating login challenges as adefense against account takeover," in *WWW*. ACM, 2019, pp. 372–382.

[11] Z. Yang, B. Wang, H. Li, D. Yuan, Z. Liu, N. Z. Gong, C. Liu, Q. Li, X. Liang, and S. Hu, "On detecting growing-up behaviors of malicious accounts in privacy-centric mobile social networks," in *ACSAC*, 2021.

[12] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *CoRR*, vol. abs/2110.11334, 2021.

[13] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *ICML*, vol. 97. PMLR, 2019, pp. 6438–6447.

[14] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 2002.

[15] H. Zheng, M. Xue, H. Lu, S. Hao, H. Zhu, X. Liang, and K. W. Ross, "Smoke screener or straight shooter: Detecting elite sybil attacks in user-review social networks," in *NDSS*. The Internet Society, 2018.

[16] D. Yuan, Y. Miao, N. Z. Gong, Z. Yang, Q. Li, D. Song, Q. Wang, and X. Liang, "Detecting fake accounts in online social networks at the time of registrations," in *CCS*. ACM, 2019, pp. 1423–1438.

[17] A. Breuer, R. Eilat, and U. Weinsberg, "Friend or faux: Graph-based early detection of fake accounts on social networks," in *WWW*. ACM / IW3C2, 2020, pp. 1287–1297.

[18] C. Wang, "The behavioral sign of account theft: Realizing online payment fraud alert," in *IJCAI*. ijcai.org, 2020, pp. 4611–4618.

[19] Z. Wang, J. Xie, T. Yu, S. Li, and J. C. Lui, "Online corrupted user detection and regret minimization," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[20] S. Lee, C. Park, H. Lee, J. Yi, J. Lee, and S. Yoon, "Removing undesirable feature contributions using out-of-distribution data," in *ICLR*, 2021.

[21] H. Wei, L. Tao, R. Xie, L. Feng, and B. An, "Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets," in *ICML*, vol. 162. PMLR, 2022, pp. 23 615–23 630.

[22] J. Bai, Z. Liu, H. Wang, J. Hao, Y. Feng, H. Chu, and H. Hu, "On the effectiveness of out-of-distribution data in self-supervised long-tail learning," in *ICLR*. OpenReview.net, 2023.

[23] A. Tadros, S. Drouyer, and R. G. von Gioi, "Out-of-distribution as A target class in semi-supervised learning," in *ICASSP*. IEEE, 2022.