FraudJudger: Fraud Detection on Digital Payment Platforms with Fewer Labels

Ruoyu Deng[1], Na Rua[1(✉)], Guangsheng Zhang[2] and Xiaohu Zhang[2]

[1] MoE Key Lab of Artificial Intelligence,
Department of CSE, Shanghai Jiao Tong University, China
{dengruoyu,naruan}@sjtu.edu.cn
[2] China Telecom Bestpay Co., Ltd.
{zhangguangsheng,zhangxiaohu}@bestpay.com.cn

Abstract. Automated fraud detection on electronic payment platforms is a tough problem. Fraud users often exploit the vulnerability of payment platforms and the carelessness of users to defraud money, steal passwords, do money laundering, etc., which causes enormous losses to digital payment platforms and users. There are many challenges for fraud detection in practice. Traditional fraud detection methods require a large-scale manually labeled dataset, which is hard to obtain in reality. Manually labeled data cost tremendous human efforts. In our work, we propose a semi-supervised learning detection model, FraudJudger, to analyze user behaviors on digital payment platforms and detect fraud users with fewer labeled data in training. FraudJudger can learn the latent representations of users from raw data with the help of Adversarial Autoencoder (AAE). Compared with other state-of-the-art fraud detection methods, FraudJudger can achieve better detection performance with only 10% labeled data. Besides, we deploy FraudJudger on a real-world financial platform, and the experiment results show that our model can well generalize to other fraud detection contexts.

Keywords: fraud detection, adversarial autoencoder, semi-supervised learning

1 Introduction

Digital payment refers to transactions that consumers pay for products or services on the Internet. With the explosive growth of electronic commerce, more and more people choose to purchase on the Internet. Different from traditional face-to-face payments, digital transactions are ensured by a third-party digital payment platform. The security of the third-party platform is the primary concern. Digital payment platforms bring huge convenience to people's daily life, but it is vulnerable to cybercrime attacks [22] [24]. Attackers have many kinds of fraud behaviors to attack digital payment platforms. For example, fraudsters may pretend to be a

staff in a digital payment platform and communicate with normal users to steal valuable information. Some fraudsters will use fake identities to transact in these platforms. An estimated 73% of enterprises report some form of suspicious activity that puts around $7.6 of every $100 transacted at risk [1]. Those frauds cause tremendous damage to companies and consumers.

Automatic detection for fraud payments is a hot topic in companies and researchers. Many researchers focus on understanding fraud users' behavior patterns. It is believed that fraud users have different habits compared with benign users. The first challenge is how to find useful features to distinguish fraud users with benign users. Sun et al. [17] use the clickstream to understand user's behavior and intentions. Some other features like transaction records [28], time patterns [8], geolocation information [6] and illicit address information [11], etc., are also proved useful in fraud detection. Fraud users have inner social connections. They always conduct fraud actions together and have relations with each other. Some researchers focus on analyzing user's social networks to find suspicious behaviors [4] [19] by graph models. They believe fraud users have some common group behaviors. The limitation of the above methods is that it is hard to find appropriate features to detect frauds manually. In traditional fraud detection methods, researchers should try many features until the powerful features are found, and these features may be partial in practice. Some information may be omitted in chosen features, and new features should be found when fraud contexts change. A proper method to learn useful features automatically is needed.

Another challenge is lacking sufficient and convincing manually labeled data in the real world. Manually labeled data are always hard to obtain in reality. It costs a vast human resource to identify fraud users manually [21]. Lacking enough labeled data to train models is a common phenomenon for many platforms. Some researchers use unsupervised learning or semi-supervised learning models to detect frauds [16]. However, for unsupervised learning, it is hard to set targets and evaluate the performance in training models. Some researchers focus on one-class detection methods which only require benign users in training [27] [9]. However, it omits information of fraud users. These works always comprise on detection performance.

In our work, we aim at overcoming these real-world challenges in fraud detection. We tackle the problem in fraud detection when insufficient labeled data are provided.

For the first challenge, we can automatically learn the best "feature" to distinguish fraud users and benign users with the help of Autoencoder [15]. Autoencoder is an unsupervised model to learn efficient data codings. It can get rid of "noise" features and only leave essential features. Origin features are encoded to latent representations by autoencoder. Makhzani et al. [14] combine autoencoder and generative adversarial network (GAN) [7], and propose a novel model called "adversarial autoencoder (AAE)". AAE can generate data's latent representations matching the aggregated posterior in an adversarial way from unlabeled data.

We propose a novel fraud detection model named FraudJudger to detect digital payment frauds automatically. FraudJudger can learn efficient features from users' operations and transaction records on digital payment platforms. In this process, FraudJudger makes full use of information in the unlabeled data. With the help of some labeled data, FraudJudger can learn how to classify users based on their latent features.

In summary, our work makes the following main contributions:

1. We propose a digital payment fraud detection model FraudJudger to overcome the shortcomings of real-world data. Our model requires fewer labeled data and can learn efficient latent features of users.
2. Our experiment is based on a real-world payment platform. The experiment result shows that our detection model achieves better detection performance with only 10% labeled data compared with other well-known supervised methods.
3. Our detection model shows strong adaptability in different contexts.

The remainder of the paper is organized as follows. In Section 2, we present related work. Our detection paradigm is provided in Section 3. Section 4 presents the details of FraudJudger. We deploy our model on a real-world payment platform, and the evaluation is in Section 5. Finally, we conclude our research in Section 6.

2 Related Work

Recently, fraud detection on digital payment platforms becomes a hot issue in the finance industry, government, and researchers. There is currently no sophisticated monitoring system to solve such problems since the digital payment platforms have suddenly emerged in recent years. Researchers often use financial fraud detection methods to deal with this problem. The types of financial fraud including credit card fraud, telecommunications fraud, insurance fraud. Many researchers regard these detection problems as a binary classification problem. Traditional detection

methods use rule-based systems [3] to detect abnormal behavior, which is eliminated by the industry environment where financial fraud is becoming more diverse and updated quickly. With the gradual maturity of machine learning and data mining technologies, some artificial intelligence models have gradually been applied to the field of fraud detection. The models most favored by researchers are Naive Bayes (NB), Support Vector Machines (SVM), Decision Tree, etc. However, these models have a common disadvantage that it is easy to overfit the training data for them. In order to overcome this problem, some models based on bagging ensemble classifier [25] and anomaly detection [2] are used in fraud detection. Besides, some researchers use an entity relationship network [18] to infer possible fraudulent activity. In recent years, more and more deep learning models are proposed. Generative adversarial network (GAN) [7] is proposed to generate adversarial samples and simulate the data distribution to improve the classification accuracy, and new deep learning methods are applied in this field. Zheng et al. [28] use a GAN based on a deep denoising autoencoder architecture to detect telecom fraud.

Many researchers focus on the imbalanced data problem. In the real world, fraud users account for only a small portion, which will lower the model's performance. Traditional solutions are oversampling minority class [5]. It does not fundamentally solve this problem. Zhang et al. [26] construct a clustering tree to consider imbalanced data distribution. Li et al. [12] propose a Positive Unlabeled Learning (PU-Learning) model that can improve the performance by utilizing positive labeled data and unlabeled data in detecting deceptive opinions.

Some researchers choose unsupervised learning and semi-supervised learning [23] due to the lack of enough labeled data in the real-world application. Unsupervised learning methods require no prior knowledge of users' labels. It can learn data distributions and have the potential to find new fraud users. Roux et al. [20] proposed a cluster detection based method to detect tax fraud without requiring historic labeled data.

In our work, we use semi-supervised learning to detect fraud users, and an unsupervised method is applied in analyzing fraud user patterns and finding potential fraud users.

3 Fraud Detection Paradigm

Our fraud detection paradigm is designed based on existing payment platforms' fraud detection workflows.

Many digital payment platforms have been devoted to fraud detection for many years. These platforms have their own fraud users blacklists, and they track and analyze fraud users on the blacklists continuously. Payment platforms have concluded many rules based on years of experience. As shown in Fig 1, platforms can use these detection rules to manually detect new fraud users and build fraud users blacklists and benign users lists. However, these labeled users only make up a small portion of all users. Most users on the platforms are unlabeled. FraudJudger is trained based on these labeled users and unlabeled users, which can make full use of every user's information. Once the detection model is trained, it can be used to classify new unknown users.

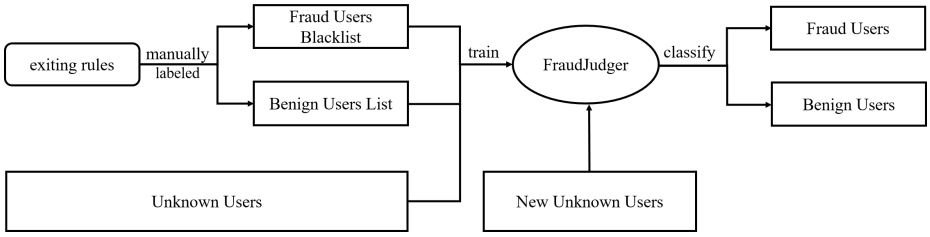


Fig. 1. Fraud detection paradigms of FraudJudger

4 FraudJudger: Fraud Detection Model

4.1 Model Overview

FraudJudger can learn the latent representations of input features and classify users. Fig 2 shows the architecture of our detection model. Each blue square box in Fig 2 corresponds to a neural network. There are four networks in FraudJudger: encoder E, decoder E' and two discriminators D_1 and D_2. The inputs of the model are user features x, and the outputs are predicted labels y and users' latent features z.

4.2 The Structure of FraudJudger

In this section, we will explain each part of FraudJudger in detail.

Encoder: First, FraudJudger learns the latent representations of origin user features x by the encoder. The dimension of origin user features x is too high to analyze directly for the following reasons:

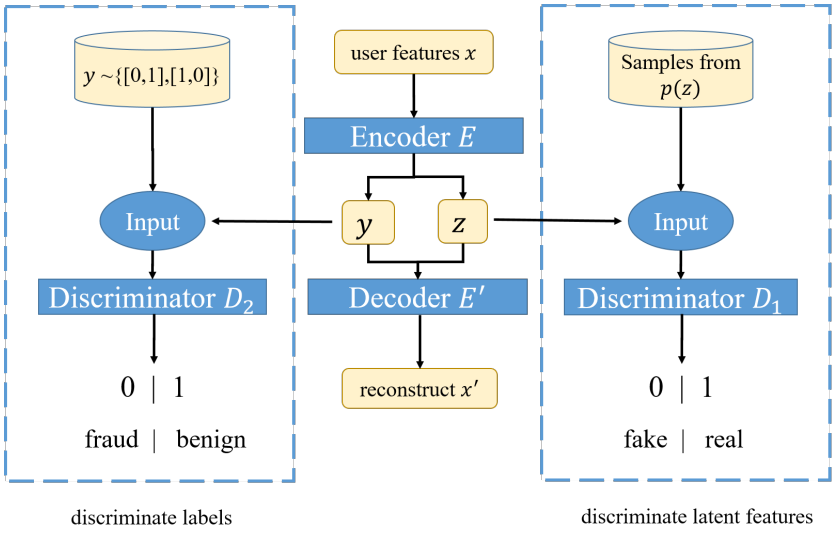


Fig. 2. The architecture of FraudJudger

1. Raw data contain irrelevant information, which is noise from our perspective. These irrelevant features will waste computation resources and affect the model's performance.
2. High dimension features will weaken the model's generalization ability. Detection model will be easily overfitted.

The encoder part reduces the dimension of features and only leave essential features. For an input merged feature x, encoder E will learn the latent representation z of x. The dimension of the latent variables z is less than the dimension of the input x, and it is determined by the output layer of the encoder's network. The encoding procedure can be regarded as dimensionality reduction. Besides, it will output an extra one-hot variable y to indicate the class of input value, which is a benign user or fraud user in our model. Our model uses y to classify an unknown user. 0 means fraud user and 1 is the benign user. The inner structure of the encoder is a multi-layer network.

$$E(x) = (y, z) \tag{1}$$

Decoder: The purpose of the decoder is learning how to reconstruct the input of the encoder from encoder's outputs. The decoder's procedure is the inverse of the encoder. Inputs of the decoder E' are outputs of the encoder E. The decoder will learn how to reconstruct inputs x from y and

z. The output of the decoder is x'. The inner structure of the decoder is also the inverse of the inner structure of the encoder.

$$E'(y, z) = x' \tag{2}$$

Discriminator: Like the discriminator of GAN, we use discriminators in our model to judge whether a variable is real or not. Since the encoder has two outputs, y and z, we need two discriminators D_1 and D_2 to discriminate them, respectively. The discriminators will judge whether a variable is in the real distribution.

4.3 Loss Fuction

Loss functions are used to measure the inconsistency between the model's outputs and expected outputs. There are four loss functions to be optimized in FraudJudger.

Encoder-Decoder Loss: The loss of the encoder and the decoder L_{e-d} is defined by mean-square loss between the input x of the encoder and output x' of the decoder. It measures the similarity between x and x'.

$$L_{e-d} = \mathbb{E}((x - x')^2) \tag{3}$$

Generator Loss: Encoding the class y and latent vectors z from x can be regarded as the generator in GAN. Let $p(y)$ be the prior distributions of y, which are the distributions of fraud users and benign users in the real world. And $p(z)$ is the prior distribution of z, which is assumed as Gaussian distribution: $z \sim \mathcal{N}(\mu, \sigma^2)$. The generator tries to generate y and z in their prior distributions to fool the discriminators. The loss function of the generator L_G is:

$$L_G = -\mathbb{E}(log(1 - D_1(z)) + log(1 - D_2(y))) \tag{4}$$

Discriminator Loss: The loss of two discriminators are defined to measure the ability in discriminating fake values.

$$\begin{aligned} L_{D_1} &= -\mathbb{E}(a_z log(D_1(z)) + (1 - a_z) log(1 - D_1(z))) \\ L_{D_2} &= -\mathbb{E}(a_y log(D_2(y)) + (1 - a_y) log(1 - D_2(y))) \\ L_D &= L_{D_1} + L_{D_2} \end{aligned} \tag{5}$$

where a_z, a_y are the true labels (fake samples or real) of inputs z and y. The total loss of the discriminator part is the sum of each discriminator.

Classifier Loss: We can teach the encoder to output the right label y with the help of a few samples with labels. And the loss function L_C is:

$$L_C = -\mathbb{E}(a'_y log(y) + (1 - a'_y) log(1 - y)) \tag{6}$$

where a'_y means the right label (fraud or benign) for a sample, and y is the output label from the encoder. When the encoder outputs a wrong label, the classifier will back-propagate the classification loss and teach the encoder how to predict labels correctly.

4.4 Training Procedure

The model learns how to optimize loss functions in the training procedure. In the training phase, the generator generates like the real label information y and latent representations z by the encoder network. Two discriminators try to judge whether the inputs are fake or real. It is a two-player min-max game. The generator tries to generate true values to fool discriminators, and discriminators are improving discrimination accuracy. Both of the generator and discriminators will improve their abilities simultaneously by optimizing loss functions L_{e-d}, L_G and L_D. Samples with labels can help to increase the classification ability of our model by optimizing the classifier loss L_C. The algorithm for training the FraudJudger model is shown in Algorithm 1.

Algorithm 1: Training FraudJudger

Input: Set of labeled users $\mathbf{U_l} = \{u_{l1}, u_{l2}, ..., u_{ln}\}$;
Set of labels of labeld users $\mathbf{a_{yl}} = \{a_{y1}, a_{y2}, ..., a_{yn}\}$;
Set of unlabeled users $\mathbf{U_n} = \{u_{n1}, u_{n2}, ..., u_{nm}\}$;
Number of epochs *ep*;
Output: Well-trained FraudJudger model;

```
Initialize parameters in FraudJudger;
for i = 1, ..., ep do
    foreach user in U_l do
        Compute latent representations y,z of the user;
        Optimize L_{e-d},L_G,L_D and L_C;
    end
    foreach user in U_n do
        Compute latent representations y,z of the user;
        Optimize L_{e-d},L_G and L_D;
    end
end
```

Algorithm 2: Classify unknown users by FraudJudger

Input: Set of unknown users $\mathbf{U} = \{u1, u2, ..., un\}$;
Well-trained FraudJudger model;
Output: The classes of users $\mathbf{Y} = \{y1, y2, ..., yn\}$;

```
foreach user in U do
    compute latent representations y,z of the user by FraudJudger;
    Y += y;
end
return Labels of users Y;
```

Once the training of our model finishes, we can use it to classify unknown users. The algorithm for classifying unknown users is shown in Algorithm 2.

5 Experiment

5.1 Platform Description

We deploy FraudJudger on a real-world payment platform. The payment platform we choose is Bestpay[3], which operates the payment and finance businesses. Bestpay is the third-largest payment platform in China, and there are more than 200 million users in Bestpay. Bestpay stores user's operation records and transaction records, and these records can be regarded as the raw features of users. These data in the platform have been anonymized before we use in case of privacy leakage. The data contains more than 29,000 user's operation behaviors and transaction behaviors in 30 days. All users in the data are manually labeled as benign or fraud. The fraud behavior in this dataset is illegal bonus-getting. We regard labels of these users as ground truth. In this data, the amount of fraud users is 4,046, which accounts for 13.78% of total users. Each user contains two kinds of data, one is operation data, and the other one is transaction data. There are 20 features in operation data and 27 features in transaction data. Some important operation features and transaction features are listed in Table 1.

As shown in Table 1, there are some common features in both operation data and transaction data. We first merge operation features and transaction features by the key feature, which is "user id". It means that features belong to the same user will be merged. Each pair of features in operation features set and transaction features set will produce

[3] https://www.bestpay.com.cn/

Table 1. Part features in operation data and transaction data

Operation Feature	Explanation	Transaction Feature	Explanation
mode	user's operation type	time	transaction time
time	operation time	device	transaction device
device	operation device	tran_amt	transaction amount
version	operation version	channel	platform type
IP	device's IP address	IP	device's IP address
MAC	device's MAC address	acc_id	account id
os	device's operation system	balance	balance after transaction
geo_code	location information	trains_type	type of transaction

new features which contain their statistic properties. After merging features, and filtering out features with a high missing rate, we get 940-dimensional merged features for each user. FraudJudger will analyze the 940-dimensional merged features to detect frauds.

5.2 Hyperparameters

The structure of the encoder, decoder, and discriminator in FraudJudger is a five-layer network, which contains three hidden layers. The number of neurons in each hidden layer is 1024, 512, 512, respectively. The dimension of the latent representation z is 128, and the training epoch is 500. Fraudjudger takes the 940 dimensions of user's features as input, and learn latent representations whose dimensions are 128. We randomly choose 20,000 users in training and another 6,000 users for evaluation.

5.3 Compared with supervised models

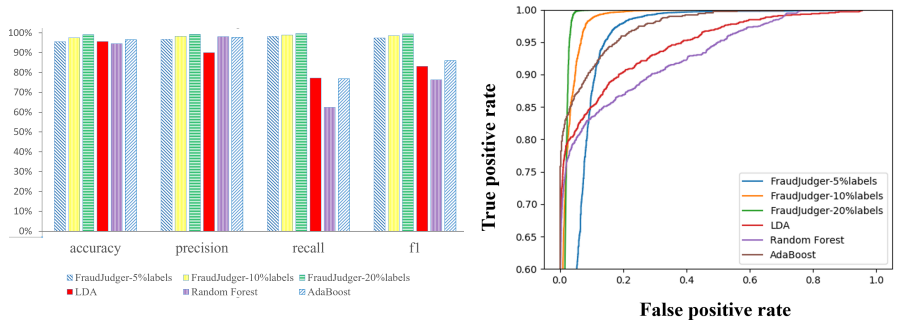
Many traditional semi-supervised algorithms sacrifice on model's performance comparing with supervised models. We compare our model's classification performance with other supervised classification models to evaluate the detection performance of FraudJudger. Three different excellent supervised machine learning models are chosen: Linear Discriminant Analysis (LDA), Random Forest, and Adaptive Boosting model (AdaBoost). All of these models' inputs are users with labels. Besides, we set three groups of FraudJudger models with 5% labels, 10% labels, and 20% labels, respectively, to evaluate FraudJudger's performance with different requirements of labeled data. The inputs to each model are the 940-dimensional merged features.

We use accuracy, precision, recall, and F1 score to measure the detecting performance of models. Precision is the fraction of true detected fraud

Table 2. AUC of FraudJudger and supervised models

Models	FraudJudger -5%labels	**FraudJudger -10%labels**	**FraudJudger -20%labels**	LDA	Random Forest	AdaBoost
AUC	0.944	**0.983**	**0.985**	0.946	0.930	0.975

users among all users classified as fraud users. Accuracy is the proportion of users who are correctly classified. Recall is intuitively the ability of the model to find all the fraud samples. F1 score is a weighted harmonic mean of precision and recall. We also use the ROC (Receiver Operating Characteristic) curve and AUC (Area Under Curve) to evaluate the result. ROC and AUC are another two measurements of the detection ability.



(a) Accuracy, Precision, Recall and F1 Score of models

(b) ROC of FraudJudger and other models

Fig. 3. Comparing FraudJudger with supervised detection models

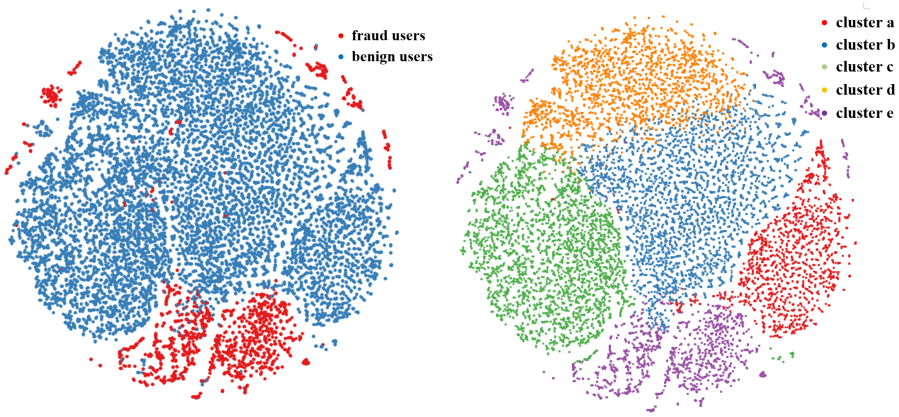
Figure 3(a) shows the accuracy, precision, recall, and F1 score of each model. FraudJudger outperforms other supervised models in recall and F1 score even with only 5% labeled data in training. It demonstrates that FraudJudger is good at detecting fraud users. And Figure 3(b) and Table 2 show the ROC and AUC results. As we can see from the results, the model's detection accuracy increases with more labeled training data. When the proportion of labeled data is larger than 10%, FraudJudger outperforms all other supervised classification models in AUC. It is reasonable because FraudJudger can automatically learn essential features and omit noisy features from raw inputs rather than using features from raw data directly like other supervised detection models. If we use fewer labels, FraudJudger still has a satisfying performance. Compared with

other supervised algorithms, FraudJudger saves more than 90% work on manually labeling data and achieves better performance.

In conclusion, FraudJudger has an excellent performance on fraud users detection even with a small ratio of labeled data. Comparing with other supervised fraud detection methods, FraudJudger has a low requirement for the amount of labeled data and can learn effective features. Our model can be applied in realistic situations.

5.4 Visualization of Latent Representation

FraudJudger uses learned latent representations to detect fraud users. In order to have an intuitively understanding of the latent representations, we use t-SNE [13] to visualize the latent representations learned from FraudJudger. T-SNE is a practical method to visualize high-dimensional data by giving each data point a location in a two-dimensional map. We visualize the latent features of users learned from FraudJudger when the ratio of labeled data is 10% in training. The dimension of learned latent representations is 100.



(a) Visualization of latent representations by t-SNE

(b) Visualization of cluster result of latent representations

Fig. 4. Visualization of latent representation

Fig 4(a) is the visualization of latent representations by t-SNE. The red points represent fraud users, and blue points represent benign users. Fraud users and benign users are well separated by latent representations

in the t-SNE map. Benign users gather together, and fraud users are isolated to benign users. It means that the latent representations learned from FraudJudger can well separate benign users and fraud users.

Furthermore, we cluster users' latent representations into five groups by K-means, and plot each group with different colors in Fig 4(b). Fig 4(b) contains five different colors, and each color represents each group of users after clustering. It is hoped that benign users and fraud users will form different groups after clustering, and the clustering result verifies it. The dividing lines between different groups are quite apparent.

Comparing Fig 4(b) with Fig 4(a), most fraud users are clustered into the same group in Fig 4(b). The fraud users in Fig 4(a) are corresponding to the purple group in Fig 4(b). Benign users with different behavior patterns are clustered into four different groups. Fraud users and benign users are well separated by cluster analysis. Since no label information is used in clustering, it verifies that the fraud users and benign users have distinct latent features learned from FraudJudger.

5.5 Evaluation on Other Contexts

In order to evaluate FraudJudger's generalization ability in other contexts, we test FraudJudger on vandals detection. Vandals are widespread on many social networks, especially on Wikipedia.

Dataset Description This evaluation is based on the UMDWikipedia dataset [10]. It contains about 33,000 Wikipedia users and 770,000 edits from Jan 2013 to July 2014. Users in the dataset are listed in the white lists or blacklists. Each user has a sequence of edit records on Wikipedia pages. The dimension of each user's feature is 200. Zheng et al. [27] choose users with the lengths of the edit sequence range from 4 to 50. After the preprocessing, the dataset contains 10528 benign users and 11495 vandals, and the dataset is available at https://github.com/PanpanZheng/OCAN/.

Comparison We compare FraudJudger with following state-of-art fraud detection methods:

(a) One-class Gaussian process (OCGP) [9] is a one-class classification model derived from the Gaussian process framework.
(b) One-class adversarial nets (OCAN) [27] builds LSTM-Autoencoder to learn the latent representation of users and uses a complementary GAN model to detect fraud users.

(c) Label Propagation (LP) [23] is a semi-supervised learning model which uses an iterative algorithm to propagate labels through the dataset along with high-density areas defined by unlabeled data.

Both of the first two methods, OVGP and OCAN, are one-class classification models, which only use positive labeled data while training. In our evaluation, we randomly choose 7,000 benign users as the training dataset to train the two models.

For group (c), we randomly choose 7,000 users, and 2.5% of them are labeled to train the LP model.

We set three other groups of experiments with different proportions of labeled samples for training FraudJudger:

(d) 2.5% labeled data. 175 labeled and 6,825 unlabeled users for training.
(e) 5.0% labeled data. 350 labeled and 6,650 unlabeled users for training.
(f) 10.0% labeled data. 700 labeled and 6,300 unlabeled users for training.

The total number of users used for training FraudJudger is also 7,000. It should be noted that in our concern, it is harder to get 7,000 reliable benign users than get 175 labeled users, which means the requirements for training data of group (c)(d)(e)(f) are more strict than group (a)(b). We randomly choose another 3,000 benign users and 3,000 vandals as the testing dataset. The measurements are precision, accuracy, recall, and F-1 score. Each model is evaluated on 10 different runs to avoid randomness. The result of each measurement is presented by the mean value and standard deviation of the 10 runs. The dimension of latent representations is 8 in FraudJudger.

Table 3. Vandal detection results (mean± std.) of models

Algorithm	Precision	Recall	F1 score	Accuracy
OCGP	0.838±0.023	0.829±0.037	0.833±0.016	0.834±0.014
OCAN	0.907±0.062	0.922±0.035	0.901±0.023	0.897±0.024
LP-2.5%	0.878±0.030	0.860±0.046	0.861±0.046	0.864±0.044
FraudJudger-2.5%	**0.975±0.011**	0.865±0.023	0.917±0.015	0.917±0.015
FraudJudger-5.0%	0.947±0.015	0.908±0.016	0.927±0.009	0.925±0.009
FraudJudger-10.0%	0.950±0.016	**0.926±0.023**	**0.938±0.011**	**0.935± 0.012**

The result is in Tabel 3. Fraudjudger achieves better performance than the other three state-of-the-art detection algorithms. Fraudjudger

has higher values in the four measurements and fewer standard deviation. It means that FraudJudger can be used in fraud detection and can have excellent performance even with a small ratio of labeled data. The model's detection accuracy and F1 score increase with more labeled training data. We find that when training with 2.5% labeled data, the precision is the highest. We argue that this is because if a model is not sensitive in classifying a user as a vandal, the model will have higher precision, but the recall will be lower. A better-trained model will have good performance both on precision and recall.

In conclusion, FraudJudger can save more work on manually labeling data and achieve better performance in vandal detection with a lower requirement for training data. It is demonstrated that FraudJudger has excellent performance in different scenarios.

6 Conclusion

In this paper, we proposed a novel fraud users detection model FraudJudger, which requires fewer labeled data in training. FraudJudger can learn latent features of users from raw data and classify users based on the learned latent features. We overcome restrictions of real-world data that it is hard to obtain enough labeled data. Our experiment is based on two different real-world contexts, and the result demonstrates that FraudJudger has a good performance in fraud detection. Compared with other well-known methods, FraudJudger has advantages in learning latent representations of fraud users and saves more than 90% manually labeling work. Our model achieves high performance on different platforms. We have seen broad prospects of deep learning in fraud detection.

7 ACKNOWLEDGMENTS

Our work is supported by National Nature Science Foundation of China (NSFC) No. 61702330; China Telecom Bestpay Co., Ltd.

References

1. Aerospike: Enabling digital payments transformation (mar 2019), https://www.aerospike.com/lp/enabling-digital-payments-transformation-ebook
2. Ahmed, M., Mahmood, A., Islam, M.R.: A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems **55**, 278–288 (01 2015)
3. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. Expert Systems with Applications **51**, 134–142 (2016)

4. Beutel, A., Xu, W., Guruswami, V., Palow, C., Faloutsos, C.: Copycatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd international conference on World Wide Web (WWW). pp. 119–130. ACM (2013)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
6. Deng, R., Ruan, N., Jin, R., Lu, Y., Jia, W., Su, C., Xu, D.: Spamtracer: Manual fake review detection for o2o commercial platforms by using geolocation features. In: International Conference on Information Security and Cryptology. pp. 384–403. Springer (2018)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the Advances in neural information processing systems (NIPS). pp. 2672–2680 (2014)
8. KC, S., Mukherjee, A.: On the temporal dynamics of opinion spamming: Case studies on yelp. In: Proceedings of the 25th International Conference on World Wide Web (WWW). pp. 369–379. ACM (2016)
9. Kemmler, M., Rodner, E., Wacker, E.S., Denzler, J.: One-class classification with gaussian processes. Pattern Recognition **46**(12), 3507–3518 (2013)
10. Kumar, S., Spezzano, F., Subrahmanian, V.: Vews: A wikipedia vandal early warning system. In: Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM (2015)
11. Lee, S., Yoon, C., Kang, H., Kim, Y., Kim, Y., Han, D., Son, S., Shin, S.: Cybercriminal minds: An investigative study of cryptocurrency abuses in the dark web. In: Network and Distributed Systems Security Symposium (NDSS) (2019)
12. Li, H., Chen, Z., Liu, B., Wei, X., Shao, J.: Spotting fake reviews via collective positive-unlabeled learning. In: Proceedings of the IEEE International Conference on Data Mining (ICDM). pp. 899–904. IEEE (2014)
13. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
14. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
15. Ng, A., et al.: Sparse autoencoder. CS294A Lecture notes **72**(2011), 1–19 (2011)
16. de Roux, D., Perez, B., Moreno, A., Villamil, M.d.P., Figueroa, C.: Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 215–222. ACM (2018)
17. Sun, J., Zhu, Q., Liu, Z., Liu, X., Lee, J., Su, Z., Shi, L., Huang, L., Xu, W.: Fraudvis: Understanding unsupervised fraud detection algorithms. In: Proceedings of the IEEE Pacific Visualization Symposium (PacificVis). pp. 170–174. IEEE (2018)
18. Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B.: Gotcha! network-based fraud detection for social security fraud. Management Science **63**(9), 3090–3110 (2016)
19. Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A.: Online human-bot interactions: Detection, estimation, and characterization. In: Proceedings of the Eleventh international AAAI conference on web and social media (ICWSM) (2017)
20. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a

local denoising criterion. Journal of machine learning research **11**(Dec), 3371–3408 (2010)
21. Viswanath, B., Bashir, M.A., Crovella, M., Guha, S., Gummadi, K.P., Krishnamurthy, B., Mislove, A.: Towards detecting anomalous user behavior in online social networks. In: Proceedings of the 23rd USENIX Security Symposium (USENIX Security). pp. 223–238 (2014)
22. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. Computers & Security **57**, 47–66 (2016)
23. Xiaojin, Z., Zoubin, G.: Learning from labeled and unlabeled data with label propagation. Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University (2002)
24. Yao, Y., Viswanath, B., Cryan, J., Zheng, H., Zhao, B.Y.: Automated crowdturfing attacks and defenses in online review systems. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). pp. 1143–1158. ACM (2017)
25. Zareapoor, M., Shamsolmoali, P., et al.: Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science **48**(2015), 679–685 (2015)
26. Zhang, Y., Liu, G., Zheng, L., Yan, C., Jiang, C.: A novel method of processing class imbalance and its application in transaction fraud detection. In: Proceedings of the IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT). pp. 152–159. IEEE (2018)
27. Zheng, P., Yuan, S., Wu, X., Li, J., Lu, A.: One-class adversarial nets for fraud detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1286–1293 (2019)
28. Zheng, Y.J., Zhou, X.H., Sheng, W.G., Xue, Y., Chen, S.Y.: Generative adversarial network based telecom fraud detection at the receiving bank. Neural Networks **102**, 78–86 (2018)