Loan Fraud Users Detection in Online Lending Leveraging Multiple Data Views

Sha Zhao[1,2*], Yongrui Huang[1*], Ling Chen[3], Chunping Wang[3], Shijian Li[1,2], Lei Chen[3], Gang Pan[1,2†]

[1] Department of Computer Science, Zhejiang University, Hangzhou 310027, China
[2] State Key Laboratory of Brain Machine Intelligence, Zhejiang University, Hangzhou 311121, China
[3] FinVolution Group (FINV), Shanghai 201203, China
{szhao, yrhuang, shijianli, gpan}@zju.edu.cn, {chenling03, wangchunping02, chenlei04}@xinye.com

Abstract

In recent years, online lending platforms have been becoming attractive for micro-financing and popular in financial industries. However, such online lending platforms face a high risk of failure due to the lack of expertise on borrowers' creditworthness. Thus, risk forecasting is important to avoid economic loss. Detecting loan fraud users in advance is at the heart of risk forecasting. *The purpose of fraud user (borrower) detection is to predict whether one user will fail to make required payments in the future.* Detecting fraud users depend on historical loan records. However, a large proportion of users lack such information, especially for new users. In this paper, we attempt to detect loan fraud users from cross domain heterogeneous data views, including user attributes, installed app lists, app installation behaviors, and app-in logs, which compensate for the lack of historical loan records. However, it is difficult to effectively fuse the multiple heterogeneous data views. Moreover, some samples miss one or even more data views, increasing the difficulty in fusion. To address the challenges, we propose a novel end-to-end deep multiview learning approach, which encodes heterogeneous data views into homogeneous ones, generates the missing views based on the learned relationship among all the views, and then fuses all the views together to a comprehensive view for identifying fraud users. Our model is evaluated on a real-world large-scale dataset consisting of 401,978 loan records of 228,117 users from January 1, 2019, to September 30, 2019, achieving the state-of-the-art performance.

Introduction

In recent years, emergence of online lending has been attractive option for micro-financing and becoming popular in financial industries. By such lending finance platforms, users (borrowers) obtain small loans without any collateral over the internet. Borrowers easily process loan procedures by specific applications, and obtain loans in a much shorter period than that from traditional financial service providers (e.g., bank). Such platforms have attracted massive number of users, and have rapidly gained market share in financial industries. However, they face a high risk of failure due to the lack of expertise on the borrowers' creditworthness. Moreover, the fact that the loans are usually uncollateralized and there are not rigorous rules for borrowers, increases the risk. Thus, risk forecasting is very important for such online lending finance platforms.

Detecting fraud users (Hu et al. 2019; Liu et al. 2018; Wang et al. 2019; Cheng et al. 2020; Jiang, Ni, and Wang 2021; Zhang et al. 2022; Wang et al. 2019; Zhan and Yin 2018; Qian et al. 2021; Wang et al. 2021; Tolstyakov and Mamedova 2021; Yang et al. 2022) is a crucial part of risk forecasting. **The purpose of fraud user detection is to predict whether one user will fail to make required payments at the stipulated repayment time.** From the viewpoint of machine learning, the problem can be transformed into a binary classification one. Traditional fraud user detection relies heavily on borrowers' historical loan records. But, a large portion of borrowers miss such information (Yang et al. 2019b), especially for the new borrowers which could result in cold-start issues. Fortunately, there are some other data sources available, which reveal clues regarding one customer's fraud probability. They compensate for the lack of historical loan records, and benefit to fraud users detection.

In this work, we attempt to detect loan fraud users from cross domain heterogeneous data views. In particular, there are different data sources related to each user, including user attributes (age, gender, income level, etc.), installed app lists (what apps installed on a smartphone), app installation behaviors (when one updates/ installs/ uninstalls apps on a smartphone), and app-in logs (interaction logs in a specific loan app). Such multiple views reflect users' fraud probability from different perspectives. For example, the fraud probability differs among users with different income levels. Installed app lists or app installation behaviors reflect users' consumption level to a degree (Zhao et al. 2019) that is related to the fraud probability. Moreover, the multiple views contain complementary information among each other. For instance, users with different attributes prefer to install different apps (Zhao et al. 2019). By exploiting the complementary characteristics among the multiple data views, multiview leaning brings about a more comprehensive description of users that is helpful for fraud user detection.

However, identifying fraud users from multiple views is nontrivial. First of all, ***for some user samples, one or even more data views are missing***. A straightforward way is to remove the samples missing views, but it usually results in

*These authors contributed equally.
†Corresponding author

notable performance degeneration. There have been many methods for generating missing views. However, there are some limitations: 1) Most of them are two-phase solutions, where they first generate missing views and then use the generated views for different downstream tasks (e.g. classification and clustering). The views generated in the first phase contain noises which could not be well optimized in the second phase, causing cumulative error and degrading the performance of the downstream tasks. Meantime, the data semantics, which are important for a specific downstream task, are obscured by such noises to a certain degree, also degrading the performance. 2) Many existing generation methods are non-friendly for multiple data views with different dimensions. Second, ***it is difficult to effectively fuse together different data views***. A naive solution considers concatenating all the views into one single view. However, the concatenation does not work, since different views are heterogeneous. Moreover, the simple concatenation ignores the relationship among different views.

To address the challenges, we propose a novel deep multiview learning approach for fraud user detection. We evaluate its effectiveness on a real world large-scale dataset provided by a financial company in China, which consists of 401,978 loan records of 228,117 users from January 1, 2019, to September 30, 2019. Our contributions are as follows:

- We propose a novel deep multiview learning approach to detect loan fraud users from multiple data views. It encodes heterogeneous views into homogeneous ones, generates missing views and fuses all the views for classifying fraud users. It is validated by a real world large-scale dataset and achieves the state-of-the-art performance.
- We develop an end-to-end trainable architecture to classify loan fraud users, where the view encoding, view generation and fusion, and classification can be jointly trained from generation loss and classification loss.
- We propose a view generation and fusion subnetwork, where missing views are generated based on the learned relationship among all the views, and all the views are fused together through an attention mechanism.

Related Work

Loan Fraud User Detection

There have been extensive studies on loan fraud users detection. At first, expertise-based manual techniques are used for detecting loan fraud behaviors, which are time consuming and difficult to make accurate decisions (Baklouti and Baccar 2013). Then, some studies have used machine learning and deep learning techniques to automatically detect loan fraud users (Wang et al. 2019)(Awotunde et al. 2021; Zhang et al. 2022; Ma et al. 2018; Jiang, Ni, and Wang 2021; Cheng et al. 2019; Attigeri, Pai MM, and Pai 2021; Xu et al. 2021). In recent years, some have used multi-view learning methods for loan fraud users detection by combing together different data sources. For example, Ge et al. (Ge et al. 2017) combined users' loan data from a large P2P lending platform and their social media profiles data from a popular social media site to predict the fraud risk. Zhong el al. (Zhong et al. 2020) designed a multi-view attributed heterogeneous information network for loan fraud detection, by aggregating user's social, funding and device views.

Multiview Learning with Missing Views

Some samples miss views, especially when multiple data views are collected from different sources. The methods for addressing multiview learning with missing views are roughly classified into two categories: filling in missing values in the origin dataset and subspace learning. For filling missing values, an alternative approach is conventional matrix completion, which provides a solution for the random missing-variable problem. But, it is not suitable for recovering the concentrated or column-wise (row-wise) missing variables (Xu, Tao, and Xu 2015). To solve this problem, Xu, Tao, and Xu (2015) restored the incomplete views used the complete views by exploiting the connections between multiple views. However, it is difficult to generate views from a simple view to a more complex one, especially when the multiple data views are heterogeneous. It also ignores the correlation between different views (Gong et al. 2021). Moreover, the method is a two-phase solution where missing views are generated in the first phase, and then the generated views are used for downstream tasks. It causes cumulative error and degrades the performance. Subspace learning projects the incomplete views into a common latent subspace for learning (Zhang et al. 2019, 2020; Arya and Saha 2021). An encoding network is designed to degrade the complete latent representation into the available views. The encoded latent representation from observations is complete and support the prediction task. However, the subspace learning method is not so adaptive to our research problem. First, it is difficult to design different project functions to map heterogeneous datasets to the same subspace. Second, the learning subspace sometimes is lack of semantics for specific tasks, such as classification or clustering.

Data Overview

Groundtruth Labels

Our dataset contains users' loan information, such as loan time, repayment time, and loan amount. Each user has 4 repayment rounds, and the interval between the adjacent ones is one month. ***We define the behavior of having more than 30 days overdue repayment time as a fraud behavior.*** The dataset indicates whether one user is a fraud one for each repayment round, including 1R30, 2R30, 3R30 and 4R30: whether one user has more than 30 days overdue the 1st-repayment time (1R30), whether she has more than 30 days overdue the 2nd-repayment time (2R30), and so on. The fraud rate of these four repayment rounds is 4.0%, 9.0%, 10.6%, and 12.1%. The fraud rate of the 1st-repayment round is the lowest, while that of the 4th is the highest. Usually, if one user has 30 days overdue the 1st-repayment, she probably has overdue the subsequent repayment rounds. ***In this work, we try to classify whether one user has overdue the 1st-, 2nd-, 3rd-, and 4th-repayment, respectively.***

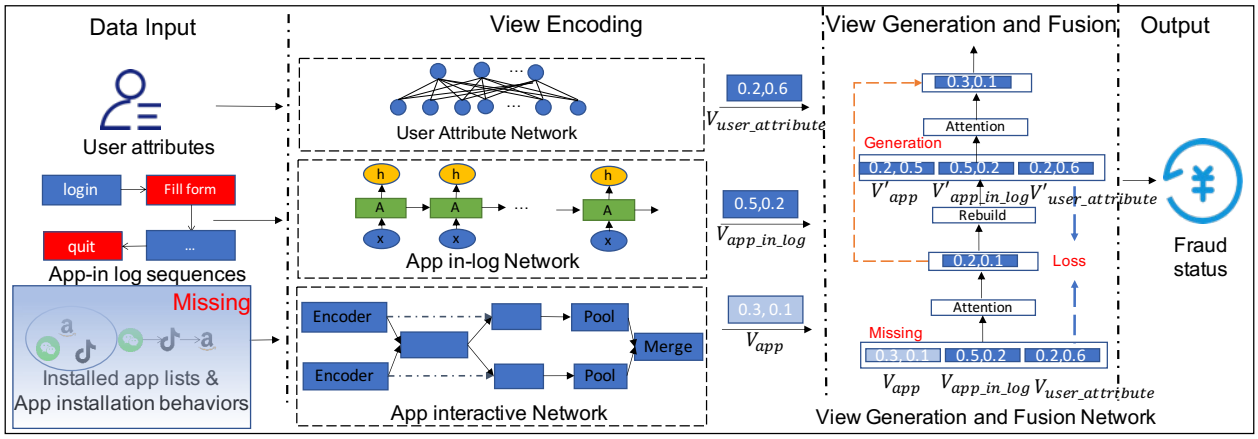


Figure 1: Overview of our proposed model.

Multiple Data Views

There are four data views in total, including user attributes, app-in logs, app installed lists and app installation behaviors. **(1)** The **user attributes** includes one user's gender, age, income, education level and marriage status. For each user in the dataset, we has her user attributes. The fraud rate varies with different user attributes. **(2) App-in logs** refer to the interaction logs with a specific loan app developed by the cooperated company, by which borrowers can obtain small loans online. App-in logs contain several app-in interaction records, each of which contains user identity, the time when the action happens, event the user clicks and the page where the action happens. For each user, we use her app-in logs before the loan time. There are 275,543,268 app-in logs in total from April, 2018 to February, 2020. **(3) Installed app lists** refer to what apps installed on a user's smartphone. Each record contain user's identity and the corresponding apps. The dataset of installed app lists contains 8,561,127 records and 127,818 apps. **(4) App installation behaviors** refer to when one user installs/ uninstalls what apps on a smartphone. Each record contains one's identity, the corresponding app, time the action happens and the action types (installation/un-installation). It contains 329,326 apps.

In our dataset, some samples miss one or more views. According to our observation, all the samples have the view of user attributes. Any one of the other three data views may be missed. For the total 402,840 samples, the users missing the view of app installed lists are the most. In our data set, there are 127,213 loan samples in total having all the views, accounting **only 30%** of all the samples.

Deep Multiview Learning Approach

Problem Definition

Given multiple data views, including user attributes, app-in logs, app installed lists, and app installation behaviors, **we predict whether one user will fail to make the required payments at the stipulated repayment time** (1R30, 2R30, 3R30, and 4R30). We transform the problem into a binary classification problem.

Model Overview

We design an end-to-end deep multiview approach where view generation and fusion, and classification are trained jointly. Fig. 1 shows the overview of our model, which consists of data input, view encoding, view generation and fusion, and output. More specifically, we first map four heterogeneous data sources into three homogeneous views with the same dimensions by developing subnetworks. The user attributes and app in-logs are encoded into $v_{user\_attribute}$ and $v_{app\_in\_log}$ by user attribute subnetwork and app-in log subnetwork. Due to the incompleteness of app installation behaviors, we propose an app interactive subnetwork to map installed app lists and app installation behaviors into one app view: v_{app}, taking advantages of the close association between them. Then, we generate the missing views and fuse the three views together by designing a view generation and fusion subnetwork. Finally, the fused vectors are used for identifying fraud users.

Encoding Heterogeneous Data Views into Homogeneous Ones

We first encode the heterogeneous data views into homogeneous ones with the same dimensionality of 256.

Encoding user attributes User attribute data consists of two parts, one of which are demographic attributes, including age, gender, income, marriage status, and education level, and the other are loan-related attributes, including loan periods and loan amount. We design a user attribute subnetwork to map all the heterogeneous user attributes into a homogeneous view, $v_{user\_attribute}$ with 256 dimensions. Here, we use a neural network and GELU (Gaussian Error Linear Units) as our activation function.

Encoding app-in logs App-in logs refer to one's interaction records in sequence with the specific loan app developed by the cooperated company. Each record is denoted as a tuple of three basic elements: one's identity u, action c, and timestamp t. In order to capture the sequential relationship between actions, we apply GRU (Gated Recurrent Unit) to learn hidden vectors at each timestamp. Then, we obtain the

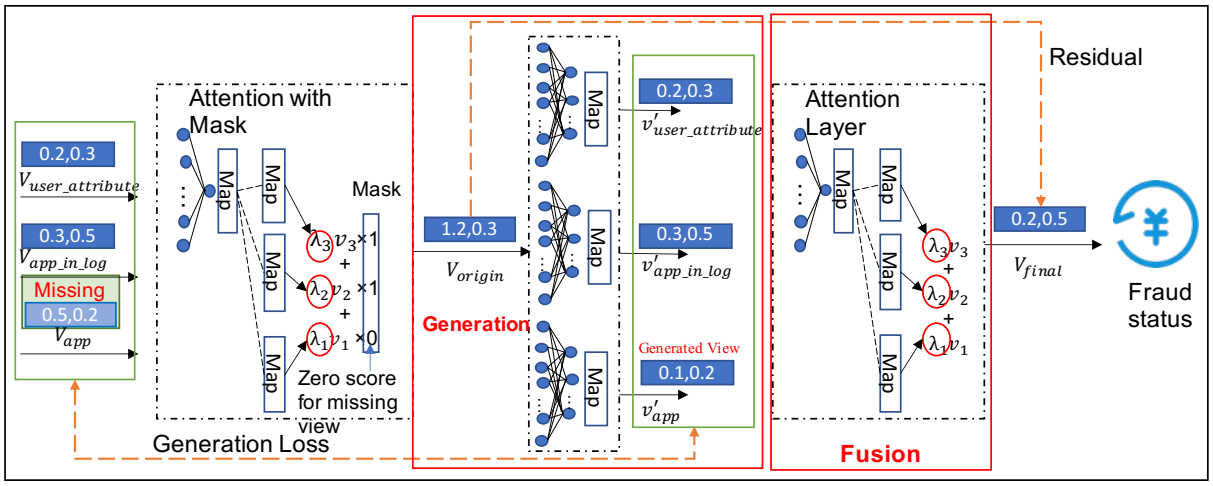


Figure 2: Illustration of view generation and fusion network.

$v_{app\_in\_log}$ with 256 dimensions by an attention layer which learns the weight of each action.

Encoding app lists and app installation behaviors In app installation behaviors, some action sequences are incomplete. For example, the groundtruth sequence is "[WeChat Installation]-[TikTok Installation]-[TikTok Uninstallation]", but the collected is "[WeChat Installation]-[TikTok Uninstallation]" and the "TikTok Installation" is missed. Fortunately, the "TikTok" can be found in app lists, which compensate the missing "TikTok Installation". The complementary information between these two datasets can mitigate the influence of the incompleteness. Therefore, *we project the app installed lists and app installation behaviors into one view: the app view of v_{app}, taking advantages of the close association between them.* Motivated by this idea and inspired by model RE2 (Yang et al. 2019a), we propose an app interactive subnetwork to combine them together to model such association to learn v_{app}.

More specifically, we first encode the app lists and app installation behaviors to dense vectors a_i and b_j by app list encoder and app installation behavior encoder, respectively. In order to take advantage of the complementary information between them, we then feed a_i and b_j to a mutual attention layer. It first extracts the importance of one view by learning the attention score by the Equation (1). Then, the other view is complemented by Equation (2), a' that is weighted sum of b and b' that is weighted sum of a.

$$e_{ij} = F(a_i)^T F(b_j) \tag{1}$$

$$\begin{aligned} a'_i &= \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} b_j \\ b'_j &= \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{kj})} a_i \end{aligned} \tag{2}$$

In order to guarantee that there is no significant difference between a_i and a', we apply a residual mechanism. Specifically, we feed a' to fusion layers, which compare local and aligned representations and then fuse them together by Equation (3) (Operation for b is similar to a). Finally, a_i^f and b_i^f are feed to a merge layer to obtain the app view v_{app} with a fixed dimension by Equation (4).

$$\begin{aligned} \bar{a}_i^1 &= G_1([a_i; a'_i]) \\ \bar{a}_i^2 &= G_2([a_i; a_i - a'_i]) \\ \bar{a}_i^3 &= G_3([a_i; a_i \circ a'_i]) \\ a_i^f &= G([\bar{a}_i^1; \bar{a}_i^2; \bar{a}_i^3]) \end{aligned} \tag{3}$$

where G_1, G_2 , G_3 and G are independent feed-forward networks and $\circ$ denotes element-wise multiplication.

$$v\_app = H([v_a; v_b; |v_a - v_b|; v_a \circ v_b]) \tag{4}$$

where H is a feed-forward neural network and v_a and v_b are our sequence's pooling results. v_{app} is app view.

Generating Missing Views and Fusing All the Views

After obtaining the three views of $v_{user\_attribute}$, $v_{app\_in\_log}$ and v_{app}, we need to fuse them together. In our dataset about 70% users miss one or even more views. In such cases, it is impossible to directly combine all the views. An easy way is to remove all user samples with any missing view. But, it will lead to a serious loss of data. Fortunately, different views are correlated with each other, and such correlation could help generate the missing views, which can further improve the performance. For example, a user attribute A is correlated to the app B in app lists, so A and B together help detect fraud users. If B is missing, we can generate B' based on its correlation with A, and use B' and A together for fraud users, of which the performance is better than only using A. Therefore, we propose a view generation and fusion subnetwork, by which we model the correlation among multiple views and generate missing views based on the correlation, and then combine all the views together for

loan fraud detection. Fig. 2 presents how our view generation and fusion network works.

In our dataset, each user has at least one view, the view of user attribute. Our view generation and fusion network aims to generate the missing views on the basis of the views one user already has. For her missing views, we add a special token $< start >$ as input. For instance illustrated in Fig. 2, one user has the views of user attribute and app-in logs, but misses the app view. As such, the $v_{user\_attribute}$, $v_{app\_in\_log}$ and the token $< start >$ representing the app view are taken as input to the subnetwork.

Then, we model the correlation among multiple views by applying an attention layer. We first aggregate the three views into v_{origin}, which embeds the importance of each input view computed by the attention mechanism in Equation (5). Here, we set the attention scores of the missing views to zeros. The v_{origin} is obtained by the weighted summation of all the views by Equation (6), which reflects the relationship among different views.

$$\lambda_i = \frac{e^{\alpha^T tanh(W v_i)}}{\sum_{i=1}^{K} e^{\alpha^T tanh(W v_i)}}, \forall i = 1, .., K \tag{5}$$

$$v_{origin} = \sum_{i=1}^{K} \lambda_i v_i \tag{6}$$

where v_i is the views, λ_i is the attention score, $\alpha^\top$ and W are trainable parameters, and K is the number of views.

Then, the v_{origin} is feed to generate three views (v'_1, v'_2 and v'_3), taking the advantage of the relationship among the three views. Specifically, the three views are generated by three different matrix mapping described in Equation (7). Here, we not only generate the missing views, but also generate the existing ones. When generating the existing views, we make them close to the initial ones as much as possible, by introducing a **generation loss function** in Equation (8). We calculate the euclidean distance between the rebuilt views and the initial ones during the training procedure for the existing views.

$$v'_i = W_{i2}^T tanh(W_{i1}^T v_{origin} + b_{i1}) + b_{i2}, \forall i = 1, 2, ..., K \tag{7}$$

where v'_i is the corresponding generated view, W_{i2}, W_{i1}, b_{i2}, b_{i1} are the trainable parameters and K is the number of views.

$$\begin{aligned} loss_{generation} &= \sum_{i=1}^{S} \sum_{j=1}^{K} \Theta_{ij} (v_{ij} - v'_{ij})^2 \\ \Theta_{ij} &= \begin{cases} 0 & view_{ij}\ missing \\ 1 & otherwise \end{cases} \end{aligned} \tag{8}$$

where v_{ij} is the i^{th} sample's j^{th} origin view and v'_{ij} is the i^{th} sample's j^{th} generated view. Θ_{ij} represents whether the corresponding view is missing and S is the number of samples.

After generating the three views, we fuse them together to one view v_{final}, by feeding them into an attention layer and computing the weight summation of each generated view. In order to prevent the gradient vanishing, we use residual mechanism to connect the v_{origin} and v_{final} by Equation (9). Finally, the v_{final} is feed to the output layer and the probability distribution of the loan fraud status is obtained by Equation (10).

$$v_{final} = v_{final} + v_{origin} \tag{9}$$

$$p(y_i | v_{final}) = \frac{e^{W_i^T v_{final}}}{\sum_{i=1}^{C} e^{W_i^T v_{final}}}, \forall i = 1, 2, ..., C \tag{10}$$

Joint Training

In the training procedure, generating missing views and classifying fraud users are jointly trained together by the same one objective function, which consists of two different parts, i.e. generation loss and classification loss, described in Equation (11). Here, we use Cross entropy loss $loss_{classification}$ for the classification task.

$$loss = \alpha loss_{classification} + (1 - \alpha) loss_{generation} \tag{11}$$

where α is hype-parameter, we set it to be 0.5 in practice.

Experimental Results

Experiment Setup

Our dataset contains 401,978 loan records of 228,117 users from January 1, 2019, to September 30, 2019. We first split the dataset into training set and testing set. In order to avoid data leakage, we split training set and testing set by loan date. More specially, we used the loan records before September 1, 2019 as training set and the ones after September 1, 2019 as testing set. As such, we have 326,082 training samples and 75,896 testing ones. The optimizer we used was ADAMW and the learning rate was set as 0.001. The batch size we used was 256 and the dimensions of different views were set to be 256.

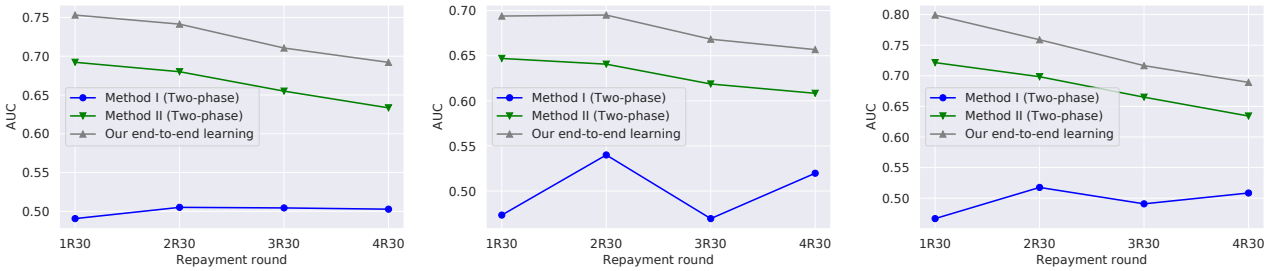
We used AUC (area under curve) as the metric to measure our model. Our models predict the loan fraud status of four targets $1R30$, $2R30$, $3R30$ and $4R30$. Besides, there are new customers (who loan for the first time) and regular customers (who loan more than once) in dataset. According to our observation, they usually have different characteristics in loan behaviors. Therefore, we reported the results for new customers and regular customers, respectively.

Results

Loan fraud users detection results The loan fraud users detection results are shown in Tab. 1. We first tested the performance when we used only one data view by each subnetwork, respectively. We then generated the missing views and fused all the views together for classification. As we can see from Tab. 1, it can be found: **1)** When we used only one kind of data view, the app-in logs and installation behaviors are the most powerful features for the tasks, and the user

	All Customers				New Customers				Regular Customers			
	1R30	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**
User attributes	0.61	0.61	0.59	0.60	0.61	0.62	0.60	0.60	0.64	0.61	0.60	0.61
App lists	0.66	0.65	0.64	0.63	0.61	0.62	0.60	0.59	0.67	0.64	0.63	0.61
InstallBehaviors	0.67	0.65	0.64	0.62	0.63	0.61	0.60	0.59	0.70	0.65	0.64	0.62
App-in logs	0.71	0.70	0.66	0.64	0.68	0.68	0.65	0.63	0.74	0.72	0.66	0.63
Ours	**0.75**	**0.74**	**0.71**	**0.69**	**0.69**	**0.70**	**0.67**	**0.66**	**0.80**	**0.76**	**0.72**	**0.69**

Table 1: Loan fraud users detection performance using single view and multiple views.



(a) All customers (b) New customers (c) Regular customers

Figure 3: Performance comparison between two-phase learning and end-to-end learning.

attributes are the weakest ones. The app lists and app installation behaviors perform similarly. For example, for classifying the all customers of 1R30, we obtained AUC of 0.71, 0.67, 0.66, and 0.61 when only used the app-in logs, app installation behaviors, and app installed lists, and user attributes. **2)** When we generated missing views and fused all the views together for all the samples, we obtained the best results for all the classification tasks, indicating the advantages in generating missing views and fusing all the views of our model. **3)** For different payment rounds, the AUC of the 1R30 and 2R30 is much higher than that of the 3R30 and 4R30. It is relatively easy to predict the loan fraud status in a shorter period. **4)** For new and regular customers, the AUC of regular ones is usually higher than that of new ones.

Performance comparison between two-phase learning and end-to-end learning To investigate the effectiveness of our end-to-end solution, we compared 2 two-phase solutions on the basis of our framework, both of which first generate missing views in the first phase, and then classify in the second phase: **Method I** (two-phase): In the first phase for missing views generation, we only kept the generation loss and removed the classification loss in the training procedure. Then, we used the generated vectors v_{final} for classification. **Method II** (two-phase): In the first phase for missing views generation, we kept both of generation loss and classification loss in the training procedure. Then, the generated vectors v_{final} were input for classification.

As we can see from Fig. 3, the views generated by the end-to-end learning has a distinct advantage *for all the classification tasks for all kinds of customers*, significantly improving the AUC, 5-10% higher than Method II and around 20% higher than Method I. The Method I performs the worst. If the classification loss is not considered when we generate missing views, the semantics may be missed and the generated views are not so helpful for the specific classification tasks. Meanwhile, the two-phase solution causes cumulative error which degrades the classification performance.

Comparison with other view generation methods **0-vectors**: the missing views are encoded as vectors with zeros, and then concatenated with other views. **0-attentions**: the v_{origin} in Equation (6) is directly feed to the attention layer without generating the views, and the attention score of the missing views is set to 0. **Rule Mean/ Rule Max**: the mean/maximum values of all the existing views are set as the final view vectors for classification. **GAN (Generative Adversarial Networks) (Goodfellow et al. 2014)**: we first obtained the hidden status of each view, generated the missing views in two phases, and then input the views for classification. The generator input the 4 views of each sample (if one view is missing, the vector is 0), and output the 4 complete views. The discriminator determines whether one view is generated. **MVL-IV (Xu, Tao, and Xu 2015)**: A state-of-the-art multi-view learning method based on matrix co-factorization. It embeds different views into a shared space, such that the incomplete views are restored by a coefficient matrix with the information of the observed views. **The method in (Zhang et al. 2020)**: projects the missing views to a common latent subspace. They designed an encoding network to degrade the complete latent representation into the available views, and then learned multi-view representation according to the distributions of observations and classes. **The method in (Arya and Saha 2021)**: incorporates the multi-view encoder networks and the bi-modal attention scheme to learn common latent space representations, and then generates missing view data using GANs.

As we can see from Tab. 2, our view generation and

Methods	All Customers				New Customers				Regular Customers			
	1R30	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**
0-vectors	0.74	0.73	0.70	0.68	0.68	0.69	0.65	0.65	0.78	0.75	0.71	0.68
0-attentions	0.74	0.73	0.70	0.68	0.69	0.69	0.66	0.65	0.78	0.75	0.70	0.67
Rule Mean	0.72	0.71	0.68	0.66	0.64	0.64	0.64	0.64	0.65	0.65	0.66	0.66
Rule Max	0.66	0.66	0.64	0.63	0.60	0.62	0.61	0.62	0.60	0.62	0.63	0.63
GAN	0.69	0.68	0.65	0.64	0.65	0.64	0.61	0.61	0.73	0.70	0.66	0.64
MLV-IV	0.69	0.69	0.66	0.65	0.65	0.66	0.63	0.62	0.72	0.71	0.67	0.65
Zhang 2020	0.67	0.68	0.65	0.64	0.63	0.64	0.61	0.61	0.71	0.70	0.66	0.64
Arya 2021	0.68	0.68	0.65	0.64	0.64	0.64	0.61	0.61	0.72	0.70	0.66	0.64
Ours	**0.75**	**0.74**	**0.71**	**0.69**	**0.69**	**0.70**	**0.67**	**0.66**	**0.80**	**0.76**	**0.72**	**0.69**

Table 2: Performance comparison with other generation approaches.

		All Customers				New Customers				Regular Customers			
		1R30	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**	**1R30**	**2R30**	**3R30**	**4R30**
Installation behaviors	ALL	0.69	0.66	0.65	0.63	0.65	0.62	0.62	0.60	0.74	0.67	0.66	0.63
	SIN	0.67	0.65	0.64	0.62	0.63	0.61	0.60	0.59	0.70	0.65	0.64	0.62
App lists	ALL	0.64	0.63	0.61	0.60	0.60	0.60	0.58	0.58	0.66	0.63	0.61	0.59
	SIN	0.66	0.65	0.64	0.63	0.61	0.62	0.60	0.59	0.67	0.64	0.63	0.61
User attributes	ALL	0.61	0.61	0.59	0.60	0.61	0.61	0.60	0.60	0.63	0.61	0.59	0.60
	SIN	0.66	0.65	0.64	0.63	0.61	0.62	0.60	0.59	0.67	0.64	0.63	0.61
App-in logs	ALL	0.73	0.72	0.69	0.66	0.70	0.70	0.67	0.65	0.75	0.73	0.70	0.66
	SIN	0.70	0.70	0.65	0.63	0.68	0.68	0.65	0.63	0.74	0.71	0.65	0.61

ALL refers to ALL-view training, and **SIN** refers to SINGLE-view training.

Table 3: Performance study w.r.t. ALL-view training and Single-view training.

fusion network produces superior performance for all the classification tasks, indicating its advantages in generating missing views. The "0-vectors" and "0-attentions" perform worse and the results are difficult to interpret. The Rule-based methods of Rule Mean and Rule Max perform much worse than ours, even though it is easy to operate. GAN and the methods in (Xu, Tao, and Xu 2015; Zhang et al. 2020; Arya and Saha 2021) perform much worse than ours. It is probably because that all of them are implemented in a two-phase manner, and the views generated in the first phase contain noises which cannot be well optimized in the second phase, causing cumulative error and degrading the performance. Besides, the method in (Xu, Tao, and Xu 2015) is not suitable for recovering the column-wise (row-wise) missing variables, and such cases frequently appear in our dataset.

Robustness check We explored the model robustness by studying the performance when one user only has one view. Here, *we assumed that users have only one out of the four views,* and we designed two training ways. **All-view training:** Inspired by the work (Ngiam et al. 2011), *we trained the view generation and fusion network using all the views in our dataset. In the testing phase, only one type of view is input for test*, since we assume new users have only one type of view. For example, if one only has the user attribute, her user attribute vector will be input to the trained model for test. **Single-view training**: *We trained the model using only one type of view*. *In the testing phase, one has one type of view, and the view vector will be input to the corresponding subnetwork*. If one has user attribute, her user attribute vector will be input to the trained user attribute network.

As shown in Tab. 3, our model trained by all views still performs well, even for the users with only one kind of view. There are not significant differences between the AUC of All-view training and Single-view training, indicating the model robustness. In particular, for installation behaviors and app-in logs, the performance with the All-view training is better than Single-view training. For example, for the users who only have app-in logs, we obtain an AUC of 0.73 for the 1R30 by the All-view training, higher than the Single-view training (0.73 v.s. 0.70). But for the views of app lists and user attributes, the performance of the All-view training is a little worse than that of the Single-view training. It may be because user attributes and app lists are not so powerful as installation behaviors and app-in logs.

Conclusion

In this work, we have successfully detected loan fraud users by proposing a novel deep multiview learning approach. In particular, it is designed in an end-to-end fashion, where view encoding, missing view generation and fusion, and classification are jointly learned, significantly improving the performance. We also propose a view generation and fusion network, which models the relationship among views to generate missing views and fuses all the views. The extensive experiments conducted on a real-world large-scale dataset showed the effectiveness, and robustness of our approach.

Acknowledgments

This work is supported by Natural Science Foundation of China (No. 61925603). The corresponding author is Dr. Gang Pan.

References

Arya, N.; and Saha, S. 2021. Generative Incomplete Multi-View Prognosis Predictor for Breast Cancer: GIMPP. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Attigeri, G.; Pai MM, M.; and Pai, R. M. 2021. Supervised Models for Loan Fraud Analysis using Big Data Approach. *Engineering Letters*, 29(4).

Awotunde, J. B.; Misra, S.; Ayeni, F.; Maskeliunas, R.; and Damasevicius, R. 2021. Artificial Intelligence Based System for Bank Loan Fraud Prediction. In *International Conference on Hybrid Intelligent Systems*, 463–472. Springer.

Baklouti, I.; and Baccar, A. 2013. Evaluating the predictive accuracy of microloan officers subjective judgment. *International Journal of Research Studies in Management*, 2(2): 21–34.

Cheng, D.; Wang, X.; Zhang, Y.; and Zhang, L. 2020. Graph neural network for fraud detection via spatial-temporal attention. *IEEE Transactions on Knowledge and Data Engineering*.

Cheng, D.; Zhang, Y.; Yang, F.; Tu, Y.; Niu, Z.; and Zhang, L. 2019. A dynamic default prediction framework for networked-guarantee loans. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2547–2555.

Ge, R.; Feng, J.; Gu, B.; and Zhang, P. 2017. Predicting and deterring default with social media information in peer-to-peer lending. *Journal of Management Information Systems*, 34(2): 401–424.

Gong, Y.; Li, Z.; Zhang, J.; Liu, W.; Chen, B.; and Dong, X. 2021. A spatial missing value imputation method for multi-view urban statistical data. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI)*, 1310–1316.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Hu, B.; Zhang, Z.; Shi, C.; Zhou, J.; Li, X.; and Qi, Y. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 946–953.

Jiang, J.; Ni, B.; and Wang, C. 2021. Financial Fraud Detection on Micro-credit Loan Scenario via Fuller Location Information Embedding. In *Companion Proceedings of the Web Conference 2021*, 238–246.

Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; and Song, L. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2077–2085.

Ma, L.; Zhao, X.; Zhou, Z.; and Liu, Y. 2018. A new aspect on P2P online lending default prediction using meta-level phone usage data in China. *Decision Support Systems*, 111: 60–71.

Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; and Ng, A. Y. 2011. Multimodal deep learning. In *ICML*.

Qian, H.; Zhang, S.; Wang, B.; Peng, L.; Gao, S.; and Song, Y. 2021. A comparative study on machine learning models combining with outlier detection and balanced sampling methods for credit scoring. *arXiv preprint arXiv:2112.13196*.

Tolstyakov, N.; and Mamedova, N. 2021. Development of a methodology for detecting fraud with bank loans for legal entities. In *SHS Web of Conferences*, volume 106. EDP Sciences.

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019. A Semi-supervised Graph Attentive Network for Financial Fraud Detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, 598–607. IEEE.

Wang, D.; Zhang, Z.; Zhou, J.; Cui, P.; Fang, J.; Jia, Q.; Fang, Y.; and Qi, Y. 2021. Temporal-aware graph neural network for credit risk prediction. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 702–710. SIAM.

Xu, B.; Shen, H.; Sun, B.; An, R.; Cao, Q.; and Cheng, X. 2021. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4537–4545.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12): 5812–5825.

Yang, L.; Li, J.; Dong, R.; Zhang, Y.; and Smyth, B. 2022. NumHTML: Numeric-Oriented Hierarchical Transformer Model for Multi-task Financial Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yang, R.; Zhang, J.; Gao, X.; Ji, F.; and Chen, H. 2019a. Simple and Effective Text Matching with Richer Alignment Features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4699–4709. Florence, Italy: Association for Computational Linguistics.

Yang, Y.; Xu, Y.; Wang, C.; Sun, Y.; Wu, F.; Zhuang, Y.; and Gu, M. 2019b. Understanding Default Behavior in Online Lending. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2043–2052.

Zhan, Q.; and Yin, H. 2018. A loan application fraud detection method based on knowledge graph and neural network. In *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*, 111–115.

Zhang, C.; Cui, Y.; Han, Z.; Zhou, J. T.; Fu, H.; and Hu, Q. 2020. Deep partial multi-view learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, C.; Han, Z.; Fu, H.; Zhou, J. T.; Hu, Q.; et al. 2019. CPM-Nets: Cross Partial Multi-View Networks. In *Advances in Neural Information Processing Systems*, 559–569.

Zhang, Z.; Ji, Y.; Shen, J.; Zhang, X.; and Yang, G. 2022. Heterogeneous Information Network based Default Analysis on Banking Micro and Small Enterprise Users. *arXiv preprint arXiv:2204.11849*.

Zhao, S.; Li, S.; Ramos, J.; Luo, Z.; Jiang, Z.; Dey, A. K.; and Pan, G. 2019. User profiling from their use of smartphone applications: A survey. *Pervasive and Mobile Computing*, 59: 101052.

Zhong, Q.; Liu, Y.; Ao, X.; Hu, B.; Feng, J.; Tang, J.; and He, Q. 2020. Financial Defaulter Detection on Online Credit Payment via Multi-view Attributed Heterogeneous Information Network. In *Proceedings of The Web Conference 2020*, 785–795.