IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received August 27, 2020, accepted September 21, 2020, date of publication September 28, 2020, date of current version October 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027337

A Deep Learning Based Online Credit Scoring Model for P2P Lending

ZAIMEI ZHANG[1], KUN NIU[2,3], AND YAN LIU[2,3]
[1]College of Economics and Management, Changsha University of Science and Technology, Changsha 410114, China
[2]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[3]Key Laboratory for Embedded and Network Computing of Hunan Province, Hunan University, Changsha 410082, China

Corresponding author: Yan Liu (liuyan@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702053, in part by the Key Program of Social Science Foundation of Hunan Province under Grant 16ZDB52, and in part by the Open Fund Project for Innovation Platform of universities in Hunan Province under Grant 11K002.

ABSTRACT Credit scoring models have been widely used in traditional financial institutions for many years. Using these models in P2P Lending have limitations. First, the credit data of P2P usually contains dense numerical features and sparse categorical features. Second, the existing credit scoring models are generally cannot be updated online. The loan transaction of P2P lending is very frequent and the new data leads data distribution to change. A credit scoring model without considering data update causes a serious deviation or even failure in subsequent credit assessment. In this paper, we propose a new online integrated credit scoring model (OICSM) for P2P Lending. OICSM integrates gradient boosting decision tree and neural network to make the credit scoring model can handle two types of features more effectively and update online. Offline and online experiments based on real and representative credit datasets are conducted to verify the effectiveness and superiority of the proposed model. Experimental results demonstrate that OICSM can significantly improve performance due to its advantage in deep learning over two features, and it can further correct model deterioration due to its online dynamic update capability.

INDEX TERMS Online P2P lending, deep learning, credit scoring model, machine learning, online update.

I. INTRODUCTION

With the deep integration of network technology and finance applications, the internet finance has developed rapidly. P2P Lending is a very typical application in it. P2P Lending provides a financing channel for many people who cannot obtain loans from traditional financial institutions, and also brings a new and convenient experience to borrowers and investors. However, many default events have greatly damaged the interests of P2P platforms and investors due to the immaturity of the credit assessment technology. In order to ensure the sustainable and healthy development of P2P Lending, it is very important to select high quality borrowers by using more effective personal credit assessment technology.

Credit scoring, as the main method of personal credit reporting, is an automatic assessment tool for rejecting or accepting loan requests. It distinguishes the borrower into two types of good and poor credit based on the characteristics of personal information, and then decides whether to provide loan [1]. Credit scoring method has been widely used in traditional financial institutions for many years. Considering the different characteristics of P2P Lending, the credit scoring models and their applications in P2P Lending are not yet mature.

According to the current research, the main methods of credit scoring are based on data mining and machine learning [1]–[7]. There are still two limitations.

First, complex data types lead to poor classification. The feature space of P2P credit data usually contains two types: dense numerical features (e.g. loan amount, asset-liability ratio) and sparse categorical features (e.g. gender, credit rating). However, existing classifiers are generally only good at processing one data type alone [8]. For example, a tree classifier is good at processing dense numerical features, and a neural network model has better performance on sparse categorical features. We need to design an effective model for P2P lending credit datasets containing multiple data types while guarantee its high performance simultaneously.

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

Second, existing credit scoring models require offline training, which makes it difficult to realize online learning and updating of the models. These models are generally constructed and verified offline. They cannot be updated online when they are running. They usually are retrained offline with new data after a period time (such as one month, one quarter or even longer). However, especially for P2P lending, the loan transaction is very frequent. A large number of new loan transactions will be generated which will cause the data distribution of lending to change before retraining the model. Lack of the latest data will affect the effectiveness of a updated credit scoring model. A credit scoring model must be able to be trained and updated online to be suitable for scenarios where P2P loan data grows rapidly and changes frequently.

In order to solve the above problems, we propose an online integrated credit scoring model (OICSM) for P2P lending based on machine learning methods. OICSM integrates gradient boosting decision tree (GBDT) and neural network (NN) to make the credit scoring model has online training and update capabilities, and can handle multiple types of features. GBDT has a good performance in learning over dense numerical data [2], [4] and NN method is better at learning over sparse categorical data [3], [5]. The proposed OICSM can effectively process dense numerical features and sparse categorical features at the same time. Furthermore, because GBDT cannot process the batch data, OICSM uses a neural network to perform knowledge distillation on the knowledge learned by GBDT to realize the batch processing, so OICSM can be updated online dynamically.

To verify the effectiveness of OICSM, we select two real and representative credit datasets of P2P Lending platform, Lending Club (LC) in the United States and Paipaidai (PPD) in China. Experimental results show that the OICSM not only can solve the above two problem effectively, but also has better performance than existing credit scoring models.

This paper makes the following contributions:

- We study the P2P lending credit scoring model from a new perspective of online update. To the best of our knowledge, the problem of credit scoring model online update has not attracted the attention of researchers. With the generation of new loan data, we verified that the unupdated model will have a great impact on the performance of the credit assessment.
- We propose an online integrated credit scoring model for P2P lending based on deep learning. OICSM can not only learn over both categorical and numerical credit data effectively, but also update dynamically using the newly generated loan data to avoid prediction deviations. It is especially applicable to P2P lending, which generally has massive and various borrowers, and with very frequent loan transactions.
- We select two real and representative credit datasets of P2P lending platforms and several representative baseline models for comparison. Offline and online experiments are conducted to verify the effectiveness of OICSM. Experimental results illustrate that OICSM can significantly improves performance of credit scoring and has the ability to update model online.

The rest of this paper is organized as follows. We introduce the related work in Section II. We describe the design and implementation of OICSM in Section III. Performance evaluations and results discussion of OICSM are presented in Sections IV and V. Section VI concludes this study.

II. RELATED WORK

Credit scoring is essentially a binary classification method. It is generally used to predict the default probability of loan applicants, and accordingly divides loan applicants into defaulters and non-defaulters [1]. Corresponding models of credit scoring have roughly gone through three development stages: linear discriminant method, statistical method, and machine learning method.

The linear discriminant method is first adopted by Durand and used to discriminate between benign and non-performing loans [9]. So far, it is still used as a benchmark method in a certain range and can be well applied. In 1970, Orgler first introduces a linear regression model in credit scoring [10]. Since this method is later proved to be flawed, the non-linear statistical methods (e.g. logistic regression) [11], [12], and nonparametric statistical methods (e.g. decision tree, bayesian network model) [13], [14] been introduced successively.

With the great development of optimization theory and computer technology, machine learning method has gradually become the mainstream of personal credit assessment research, and the performance has been greatly improved. Typical methods include neural network [15]–[17], genetic algorithm [18], support vector machine [19], refusal inference algorithm [6], gradient boosting decision tree [2], [4], *et al.* There are also some works to improve the performance through ensemble models [7], [20]–[22]. These methods solve problems such as increased data size and unbalanced data structure from different angles, and greatly promoted the development of personal credit assessment. According to the current research, GBDT and NN are particularly outstanding in the field of credit scoring due to their good performance [2]–[5], but they also have weaknesses.

Gradient boosting decision tree (GBDT) is a very popular integrated learning algorithm in recent years. It performs well in various machine learning tasks, such as click prediction [23], learning to rank [24]. In the field of credit scoring, Chang *et al.* [4] use eXtreme gradient boosting tree (XGBoost) and Ma *et al.* [2] use LightGBM to build credit scoring models respectively. XGBoost and LightGBM are two most popular variants of GBDT. The significant advantage of GBDT depends on its superiority in processing dense numerical features [25], [26]. But meanwhile, it has two limitations [8]. First, it is difficult to update the GBDT model online because the basic learned trees are not differentiable. This weakness makes the credit scoring model can only be updated offline after a fixed period. The update interval will cause the data distribution to change which will cause the model to be biased or even invalid. In addition, GBDT does not work well when used for sparse categorical features, and it usually fails to generate trees effectively. Although some variants of GBDT can convert categorical features into dense numeric features, the raw information may be hurt during the conversion process and resulting in the reduction of model accuracy. Some variants of GBDT [27] can also directly use the categorical feature in tree learning, but these models usually over-fits since the data in each category is too little.

Neural network can learn complex and non-linear knowledge from massive data. When applying it to the field of credit scoring, its two advantages can help construct models effectively. First, the batch-mode backpropagation algorithm makes it can not only learning over large scale data efficiently, but also use the newly generated loan data to update model dynamically while does not need to train the model from scratch. Second, it is excellent at learning over sparse categorical features by embedding structure [28], [29]. However, its inefficiency in learning over dense numerical features is a weakness [8]. Currently, NN has been well applied to the field of credit scoring, such as the wide & deep Learning model [1] and RNN model [3], but the weakness above has not been completely overcome. Although a fully connected neural network (FCNN) can be used to learn over numerical features, it easily leads to local optimization due to its complex structure [30]. Therefore, in learning with numerical data, the performance of NN does not exceed GBDT [25].

P2P credit data mainly includes two data types: sparse categorical data and dense numerical data. In addition, a P2P lending platform generally has a huge number of users and very frequent transactions. With the rapid increase of users and transactions, new loan data also accumulates rapidly, which will change data distribution. If a credit scoring model cannot updated in time, the prediction results are likely to deviate or fail. GBDT and NN have advantages, but using either method alone cannot meet the above-mentioned requirements of P2P lending credit scoring.

Currently, some papers try to combine the two methods. Some researchers construct tree-like NN models [31], but these works are mainly for computer vision tasks. Humbird *et al.* try to convert the decision trees to NN [32], but it consume many computing resources. [33], [34] use GBDT and NN together directly, but they cannot be used online efficiently due to the inherent weakness of GBDT. In addition, DeepGBM [8] framework is designed for online prediction tasks such as flight delay prediction by integrating GBDT and NN. Chen *et al.* [35] proposes a credit assessment model based on DeepGBM for home credit default risk of bank, but it does not consider the problem of deviations in the model caused by changes in data distribution and cannot online update. Although the above works have made great progress, no similar attempts has been made for credit scoring in P2P Lending.

III. METHODOLOGY

We present the design of online integrated credit scoring model (OICSM) for P2P lending based on the framework of DeepGBM [8]. OICSM integrates the advantages of GBDT and NN. It not only can learn over different two data types of P2P lending data, but also be updated online.

A. GENERAL DESCRIPTION OF OICSM

In the data warehouse of P2P lending platforms, there are mainly three categories of data: pre-loan data, unfinished loan data, and finished loan data (i.e. the P2P credit data used in our model). And the state of each corresponding loan datasets is constantly changing over time, as shown in Figure 1. Specifically, at the pre-loan stage, the new loan applications are divided into two categories after credit assessment, namely accepted loan and rejected loan. An accepted loan first forms an unfinished loan after the loan is obtained by applicant. Then it finally forms a finished loan after the repayment period ends. In other words, the state of a loan data is in a progressive relationship. The variable that controls the progressive relationship is time and the data in each state is gradually completed. Due to P2P lending's wide customer coverage and large number of users, the number of loan transactions is huge and the data updated frequently. Thus, these have become the two most significant characteristics of the P2P loan data.

Figure 2 shows the framework of OICSM proposed in this paper. In this model, the finished loan data in data warehouse is preprocesses firstly. We divide the data into two types with numerical and categorical features, and encode them separately. Next, the two types of data are imported into the ''learning over two features'' module for offline training to generate an initial credit scoring model. The method of learning over two features will be detailed in Section 3.2.

When a new loan application appears, the corresponding applicant will be assessed by the trained credit scoring model. If the application is accepted, the applicant will get a loan. After the repayment period ends, a new finished loan data is generated. The state of the three categories of data (new loan application data, unfinished loan data, and new finished loan data) are dynamically changing. As time goes on, more and more new finished loan data are accumulated as a new dataset. In OICSM, when a predetermined model update period or new dataset size is reached, the credit scoring model can be updated online with new data, and without offline retraining. The above process is continuously executed in a loop to form a dynamic update P2P credit scoring model using online data. The offline training and online dynamic update of OICSM will be described in Section 3.3.

B. DEEP LEARNING METHOD BASED ON TWO FEATURES

As mentioned earlier, the P2P credit data mainly includes sparse categorical and dense numerical features. In this section, we show the method to learn over the two different data types simultaneously for P2P lending.

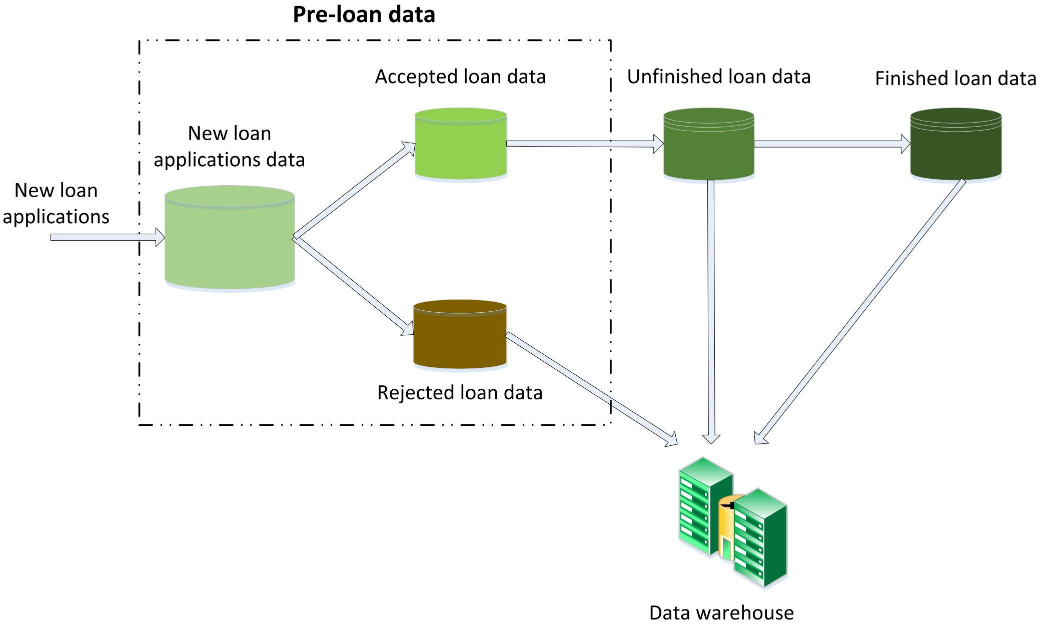

FIGURE 1. The state change of loan data.

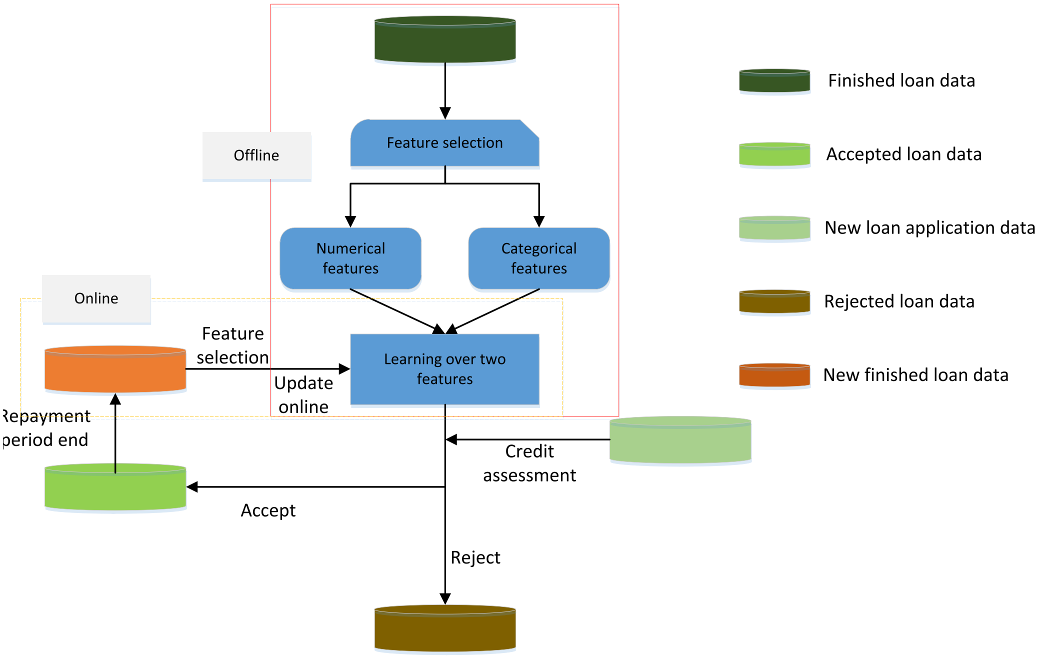

FIGURE 2. The framework of OICSM.

The learning method in this paper contains two modules: CatNN and GBDT2NN [8]. CatNN is a neural network structure performs better on learning over sparse categorical data and GBDT2NN is also a neural network structure distilled from GBDT and performs better on processing dense numerical data.

1) CatNN FOR CATEGORICAL DATA

Neural networks are widely used to construct prediction models over sparse categorical data. We directly use the existing neural network structures that already proven successful to play as the CatNN. CatNN can convert the high dimensional sparse categorical vectors into dense vectors effectively using embedding technology, as shown in Equation 1:

$$E_{V_i}(x_i) = embedding\_lookup(V_i, x_i), \tag{1}$$

where x_i is the value of i^{th} feature of sample $\boldsymbol{x}$, V_i stores all embeddings of the i^{th} feature, $E_{V_i}(x_i)$ denotes the embedding vector for x_i.

In addition, FM and Deep components are used to learn interactions over the features like [8]:

$$y_{FM}(\boldsymbol{x}) = w_0 + \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle E_{V_i}(x_i), E_{V_j}(x_j) \rangle x_i x_j, \quad (2)$$

$$y_{Deep}(\boldsymbol{x}) = \mathcal{N}\left(\left[E_{V_1}(x_1)^T, E_{V_2}(x_2)^T, \ldots, E_{V_d}(x_d)^T\right]^T; \theta\right), \quad (3)$$

where $y_{FM}(\boldsymbol{x})$ is used to learn low order feature interactions, $y_{Deep}(\boldsymbol{x})$ is used to learn high-order feature interactions. w_0 and $\boldsymbol{w}$ are the parameters of linear part, d is the number of features, $\langle \cdot, \cdot \rangle$ denotes the inner product operation, and $\mathcal{N}(x; \theta)$ denotes a multi-layered NN model with input $\boldsymbol{x}$ and parameter θ.

The final output of CatNN is the combination of these two components, which can be denoted as Equation 4:

$$y_{Cat}(\boldsymbol{x}) = y_{FM}(\boldsymbol{x}) + y_{Deep}(\boldsymbol{x}). \quad (4)$$

2) GBDT2NN FOR NUMERICAL DATA

GBDT has the advantage of learning over numerical data, but there are two shortcomings that it cannot be updated online in time and is not suitable for massive data. Next we will introduce how to distill the knowledge from GBDT into NN. The knowledge distillation process consists of three parts: tree-selected features, leaf embedding distillation, and tree grouping [8].

a: TREE-SELECTED FEATURES

one of the characteristics of tree-based model is that it does not need to use all the data features. It selects some useful features based on statistical information to fit the training target. The efficiency of the NN model can be improved by applying tree-selected technology. Thus, the tree-based selected features are used to be the input of NN model, rather than inputting all the features. In this paper, $\mathbb{I}^t$ denotes the features used in the tree t, $x[\mathbb{I}^t]$ denotes the input of NN.

b: LEAF EMBEDDING DISTILLATION

the essential difference of structures between tree-based model and NN model makes it difficult to transform between them directly. But a NN model can be used to fit functions of the tree model approximately to realize the knowledge distillation. We use NN to fit the result clusters of decision tree, so that it is close to the structure functions of the decision tree.

The structure function of a tree t is denoted as $C^t(\boldsymbol{x})$, and its return value is the output leaf index of sample $\boldsymbol{x}$. For the leaf index $C^t(\boldsymbol{x}^i)$ of the i^{th} training sample $\boldsymbol{x}^i$ on the tree t, it is denoted by one-hot encoded $L^{t,i}$. Then, embedding technology is used to reduce the dimension of $L^{t,i}$. In addition, due to the existence of the bijection relations between leaf indexes and values, leaf values can be used to learn embedding. The embedding learning process can be denoted by Equation 5:

$$\min_{\boldsymbol{w}, w_0, \omega^t} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\boldsymbol{w}^T \mathcal{H}(L^{t,i}; \omega^t) + w_0, p^{t,i}\right), \quad (5)$$

where n denotes the number of training samples, $H^{t,i} = \mathcal{H}(L^{t,i}; \omega^t)$ is an one-layered fully connected NN with parameter ω^t. It is used to convert the one-hot encoded leaf index $L^{t,i}$ to dense embedding $H^{t,i}$. $p^{t,i}$ is the leaf value predicted by tree t of sample $\boldsymbol{x}^i$. $\mathcal{L}$ is the cross-entropy loss function, and $\boldsymbol{w}$ and w_0 are the parameters for mapping embedding to leaf values. After this, dense embedding can be used as the targets to fit the function of tree structure approximately. This new learning process can be denoted by Equation 6:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}'\left(\mathcal{N}\left(\boldsymbol{x}^i[\mathbb{I}^t]; \theta\right), H^{t,i}\right), \quad (6)$$

where $\mathcal{L}'$ is the L2 loss function used for fitting dense embedding.

c: TREE GROUPING

in principle, each tree in GBDT needs a NN model to fit, but this is very inefficient. Therefore, [8] groups trees first, and then use a NN model to distill knowledge from a group of trees. To simplify the model, equally randomly grouping is used here. That is, assuming m trees in GBDT and can be divided into k groups, then the number of trees in each group is $s = \lceil m/k \rceil$, and the trees in each group are randomly obtained from GBDT. j^{th} group is denoted by $\mathbb{T}_j$. Correspondingly, the leaf embedding distillation also needs to be extended for a group of trees. Specifically, Equation 7 is to realize the knowledge distillation on a group of trees $\mathbb{T}$ based on Equation 5.

$$\min_{\boldsymbol{w}, w_0, \omega^{\mathbb{T}}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\boldsymbol{w}^T \mathcal{H}\left(||_{t \in \mathbb{T}}(L^{t,i}); \omega^{\mathbb{T}}\right) + w_0, \sum_{t \in \mathbb{T}} p^{t,i}\right), \quad (7)$$

where $||(\cdot)$ is a connection operation, which connects one-hot encoded leaf index vectors of multiple trees in tree group $\mathbb{T}$ into a multi-hot vector. $G^{\mathbb{T},i} = \mathcal{H}\left(||_{t \in \mathbb{T}}(L^{t,i}); \omega^{\mathbb{T}}\right)$ is one-layered fully connected NN that converts multi-hot vectors to a dense embedding $G^{\mathbb{T},i}$. Correspondingly, we need to use this new embedding as the distillation target of NN model, and the learning process can be extended from Equation 6 to 8.

$$\mathcal{L}^{\mathbb{T}} = \min_{\theta^{\mathbb{T}}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}'\left(\mathcal{N}\left(\boldsymbol{x}^i[\mathbb{I}^{\mathbb{T}}]; \theta^{\mathbb{T}}\right), G^{\mathbb{T},i}\right), \quad (8)$$

where $\mathbb{I}^{\mathbb{T}}$ is features used in $\mathbb{T}$. If the number of trees in $\mathbb{T}$ is large, we only select the top features in $\mathbb{I}^{\mathbb{T}}$ according to their importance.

Through above procedure, the output of the NN model obtained by knowledge distillation from $\mathbb{T}$ is:

$$y_{\mathbb{T}}(\boldsymbol{x}) = \boldsymbol{w}^T \times \mathcal{N}\left(\boldsymbol{x}[\mathbb{I}^{\mathbb{T}}]; \theta^{\mathbb{T}}\right) + w_0. \tag{9}$$

Because a GBDT uses k tree groups, the final output of the GBDT2NN is:

$$y_{GBDT2NN}(\boldsymbol{x}) = \sum_{j=1}^{k} y_{\mathbb{T}_j}(\boldsymbol{x}). \tag{10}$$

C. MODEL TRAINING AND ONLINE DYNAMIC UPDATE

In this section, we present the offline training and online dynamic update of OICSM. The implementation process is showed in Algorithm 1.

Algorithm 1 The Implementation Process of OICSM

```
Require: Offline credit data $D_{off}$; Batch training data (newly generated credit data) $D_{batch}$; Initialization trainable parameters $w_1, w_2$; Hyper-parameters $\alpha, \beta$;
Ensure: 0 (non-defaulter), 1 (defaulter).
// Offline Training
Train GBDT with $D_{off}$;
Use Eqn.(7) to obtain the leaf embedding for the tree groups;
Use Eqn.(8) to learn GBDT2NN;
Train CatNN with $D_{off}$;
Combine CatNN and GBDT2NN to get offline OICSM.
Using loss function $\mathcal{L}_{offline} = \alpha\mathcal{L}(\hat{y}(\boldsymbol{x}), y) + \beta \sum_{j=1}^{k} \mathcal{L}^{\mathbb{T}_j}$ to train OICSM;
// Online Update
When a predetermined model update period or newly generated dataset size is reached, instead of retraining from scratch, we only need $D_{batch}$ to update OICSM by loss function $\mathcal{L}_{online} = \mathcal{L}(\hat{y}(\boldsymbol{x}), y)$
```

1) OFFLINE TRAINING

Assume that the credit data (finished loan data) used in offline training is denoted by D_{off}. $D_{off} = (x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n), x_i = (f_{Num}, f_{Cat}), f_{Num}$ and f_{Cat} denote numerical and categorical features respectively. $y_i \in \{0, 1\}$, 1 means default and 0 means no default. For the GBDT2NN module in OICSM, we use $D_{off} = \{(f_{Num}, f_{CatToNum}, \boldsymbol{y})\}$ to train GBDT model, where $f_{CatToNum}$ denotes numeric features converted by categorical features through a certain feature engineering method. The feature engineering method used here will be described in section IV. For a trained GBDT model, we use Equation 7 to obtain the leaf embedding of tree groups. Thus, a GBDT2NN model can be learned by using Equation 8.

For the CatNN module, unlike GBDT2NN, a feature engineering method is used to convert numeric features into categorical features. $D_{off} = \{(f_{NumToCat}, f_{Cat}, \boldsymbol{y})\}$ is used to train the CatNN model. Finally, combine GBDT2NN and CatNN through Equation 11 to get the offline OICSM model:

$$\hat{y}(\boldsymbol{x}) = \sigma'(w_1 \times y_{GBDT2NN}(\boldsymbol{x}) + w_2 \times y_{Cat}(\boldsymbol{x})), \tag{11}$$

where w_1, w_2 are trainable parameters, σ' is *sigmoid* function. The loss function is shown in Equation 12.

$$\mathcal{L}_{offline} = \alpha\mathcal{L}(\hat{y}(\boldsymbol{x}), y) + \beta \sum_{j=1}^{k} \mathcal{L}^{\mathbb{T}_j}, \tag{12}$$

where $\mathcal{L}$ is the cross-entropy loss function. $\mathcal{L}^{\mathbb{T}}$ is the embedding loss defined by Equation 8. α, β are the weight hyper-parameters used to adjust the importance of the two losses.

2) ONLINE UPDATE

In the update period of the traditional credit scoring model, a large number of new loan transactions are finished, which make the distribution of new data to be inconsistent with the data used to train offline models. Thus, the prediction of the offline model will be biased or even invalid. This problem can be solved by manually updating. However, during this period, with a large number of new loan data coming, if the model predictions are biased, it may cause serious loss to the platforms and investors of P2P Lending. A credit scoring model should has the ability to update online dynamically with newly generated data to adapt to the changes in data distribution.

The proposed OICSM can not only process two different features effectively, but its batch processing mode can process massive data and update online dynamically in a timely manner. The update process is that, when a predetermined model update period or new dataset size is reached, to input the newly generated credit data as batch training data into the currently running model, then the model parameters are updated accordingly, to better adapt to changes in data distribution.

Due to the need for dynamic update, the loss function of the online model is different from the offline model. Let $\alpha = 1, \beta = 0$ in Equation 12, we can obtain the loss function of the online model as shown in Equation 13.

$$\mathcal{L}_{online} = \mathcal{L}(\hat{y}(\boldsymbol{x}), y), \tag{13}$$

IV. EXPERIMENT SETUP

In this section, we describe the experimental settings in detail, including compared models, data description, and specific experimental design.

A. COMPARED MODELS

In our experiments, we select the following baseline models to compare with OICSM:

- Logistic Regression (LR): LR is widely used in the construction of credit scoring models [11], [12].
- GBDT: GBDT is a very popular tree-based algorithm with good performance, and is widely used in the construction of credit scoring models [2], [4]. There are multiple variants of GBDT, we select LightGBM [27] in this

paper. It is good at learning over numeric features, but cannot process categorical features well, and it cannot be updated online.

- Wide&Deep: Wide&Deep [36] is designed for recommender systems and [1] builts a credit scoring model based on it. It is a deep learning framework composed of a shallow linear model and a deep natural network.
- DeepFM: DeepFM [37] improves the Wide&Deep learning framework by adding an additional FM component. In this paper, we use DeepFM as basic CatNN. Both it and Wide&Deep are good at learning over categorical features and can be updated online, but they cannot process numerical features well.
- GBDT2NN: GBDT2NN is a part of the OICSM proposed in this paper. Similar to GBDT, it can learn over numerical features and not good at process categorical features. But different from GBDT, it can be updated online.

B. DATA DESCRIPTION

To verify the effectiveness of OICSM, lending club (LC) in the United States and Paipaidai (PPD) in China are chose as the test datasets.

For LC, we select its published credit data from 2015 to 2017. This dataset contains more than 800,000 items and more than 100 features. For PPD, we select its published credit data from 2013-11 to 2014-11. This dataset contains more than 80,000 items and more than 200 features.

Since the original datasets contain a lot of post-loan features and noise data, we execute pre-processing operations including deleting post-loan features, deleting features with smaller variances, deleting items and features with a lot of missing values, and ignoring unfinished loan items. After then, the details of the two datasets used in our experiments are shown in Table 1.

In addition, we execute different feature engineerings for different baseline models to improve their performance and increase the credibility of comparative experiments. Specifically, for models that cannot learn over categorical features such as GBDT, we use label-encoding [38] and binary encoding methods to convert categorical features to numerical features. For models cannot learn over numerical features (LR, DeepFM and Wide&Deep), we discretize the numerical features into categorical features. After above processing, each baseline model can use the information of all features.

C. EXPERIMENTAL DESIGN

We design two experiments, execute offline and online respectively, to verify the effectiveness and superiority of proposed OICSM.

1) OFFLINE EXPERIMENT

The purpose of this experiment is to verify the offline performance of OICSM, that is, the effectiveness on learning over two different features. To imitate real business scenarios, we divide each dataset into two parts based on time stamps. For LC credit dataset, the data in 2015 is used as training set, and the data in 2016Q1-2016Q2 (Qn is n^{th} quarter) is used as test set. For PPD credit dataset, the data in 2013.11-2014.08 is used as training set, and the remaining data is used as test set. The details of divided datasets are shown in Table 2.

TABLE 1. Details of datasets used in experiments. Sample is the number of samples. Num and Cat is the number of numerical and categorical features, respectively. Qn is n^{th} quarter.

Datasets	Public Date	Sample	Num	Cat
LC	2015-2017Q1	0.8M	14	8
PPD	2013.11-2014.11	80K	204	18

TABLE 2. Details of the credit datasets used in offline experiments.

Datasets	Training		Test	
	Public Date	Sample	Public Date	Sample
LC	2015	366466	2016Q1-2016Q2	197748
PPD	2013.11-2014.08	53819	2014.09-2014.11	26180

TABLE 3. Details of batch data division for LC and PPD credit datasets. Sample is the number of samples. Qn is n^{th} quarter.

Datasets	LC		PPD	
	Public Date	Sample	Public Date	Sample
Batch 0	2015	366466	2013.11-2014.05	26746
Batch 1	2016Q1	113969	2014.06	8391
Batch 2	2016Q2	83779	2014.07	9378
Batch 3	2016Q3	84892	2014.08	9304
Batch 4	2016Q4	78268	2014.09	12413
Batch 5	2017Q1	72918	2014.10	13162

2) ONLINE EXPERIMENT

There are two purpose of this experiment, to verify whether the models that can be updated online are better than the models that cannot, and to verify whether our model is superior to other baseline models that can be updated online. Next, we will detail the experimental design from two aspects: the division of batch data, and the model training approach.

First, in terms of batch data division, we divide each credit dataset into 6 consecutive batches (Batches 0 to 5) according to the time slice. Specifically, for LC credit dataset, we use the data in 2015 as Batch 0, and the data in 2016Q1-2017Q1 are divided into 5 consecutive Batches 1 to 5, which use quarter (Q) as time slice. For PPD credit dataset, the data in 2013.11-2014.05 is used as Batch 0, and the remaining data is divided into 5 consecutive batches which use month (M) as time slice. The specific details of the divided datasets are shown in Table 3.

Here, we use quarter (Q) and month (M) as time slices because it is more convenient to observe the online update process and performance changes of the models along with the time. In real application, smaller time slices can be used as required. Moreover, the smaller the time slices, the better the performance will be. To verify this, we also design a comparison experiment by introducing smaller time slices, specifically, for LC credit dataset, we use quarter (Q) and month (M) as time slices respectively for comparison; and for

TABLE 4. AUC scores of offline experiment on LC and PPD datasets.

Model	LC	PPD
LR	$0.5001 \pm 1e-4$	$0.5005 \pm 1e-4$
GBDT	$0.7021 \pm 4e-4$	$0.6746 \pm 7e-3$
Wide&Deep	$0.6787 \pm 3e-2$	$0.6555 \pm 2e-2$
DeepFM	$0.6932 \pm 4e-2$	$0.6732 \pm 1e-2$
GBDT2NN	$0.7109 \pm 2e-4$	$0.7127 \pm 3e-3$
OICSM	$0.7208 \pm 7e-4$	$0.7316 \pm 3e-3$

PPD credit dataset, we use month (M) and half month (HM) as time slices respectively for comparison.

Second, in terms of model training, we distinguish two types of models based on whether they can be updated online.

For updatable models, including Wide&Deep, DeepFM, GBDT2NN and OICSM, they are trained by using the data of each batch along with time. Specifically, for i^{th} batch, we use the samples only in that batch to train or update the model. Samples in $(i+1)^{th}$ are used for evaluation.

For non-updatable models, including GBDT and the offline version of OICSM (denote as OICSM-off), we only use Batch 0 to train them, and then use the trained models to predict Bathc 1-Batch 5 separately, without updating the model.

V. RESULTS DISCUSSION

The experimental results are analyzed and discussed in this section. Considering that the credit data is unbalanced and the overall accuracy is not appropriate to evaluate the models, we use area under roc curve (AUC) [7] as a performance evaluation indicator. All experiments run 5 times, each time using a different random number seed.

A. OFFLINE EXPERIMENT RESULTS

The offline experimental results of all models on two credit datasets are shown in Table 4. The results show that:

- LR has the worst performance. Because LR is difficult to fit the true distribution of massive and complex credit data.
- GBDT performs better than Wide&Deep and DeepFM on both datasets. We can see that the number of numerical features is significantly more than categorical features in both LC and PPD credit datasets. GBDT performs better than NN models (Wide&Deep and DeepFM) in learning tasks with more numerical features.
- GBDT2NN, as an integral part of OICSM, is distilled by GBDT. The experimental results show that GBDT2NN is superior to GBDT in both credit datasets. This indicates that, for credit datasets that contain both numerical and categorical features, GBDT2NN can improve the performance of GBDT through knowledge distillation.
- OICSM is superior to all other baseline models. The results show that OICSM increases AUC by 1%-7% compared to other four baseline models and even more to LR. Because OICSM combines the advantages of GBDT and NN, it has the ability to deal with categorical and numerical features at the same time effectively. This superiority of OICSM shows that it is very suitable for P2P lending credit scoring.

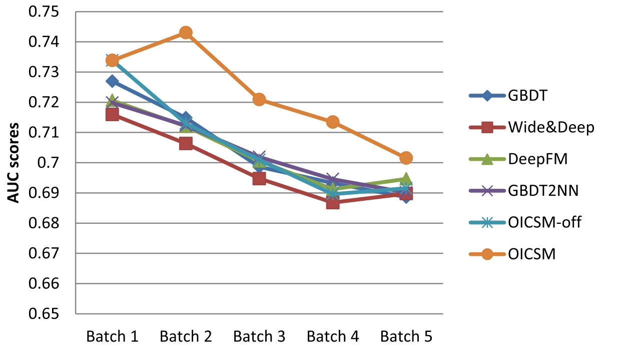

FIGURE 3. Online performance comparison on LC dataset.

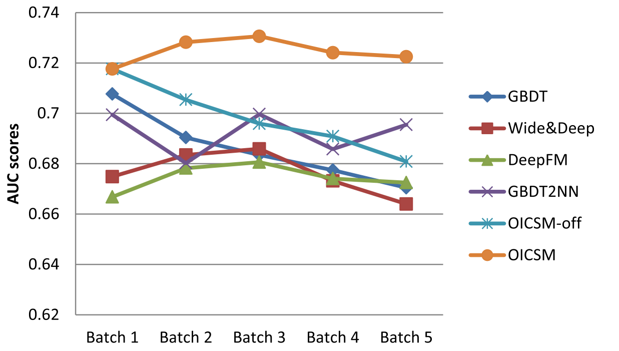

FIGURE 4. Online performance comparison on PPD dataset.

B. ONLINE EXPERIMENT RESULTS

The online experimental results of all models on LC and PPD datasets are shown in Tables 5 to 6. From the results of AUC scores, we can see that, with the addition of each batch data, the performance of all models changes accordingly. To show these changes more clearly, the AUC scores in Table 5 and Table 6 are plotted as figures in Figure 4 and Figure 5 respectively. From the results we can see that:

- For non-updatable models of GBDT and OICSM-off, on Batch 1, they have good performance on both the two credit datasets. However, because they cannot be updated online, their performance drops rapidly after Batch 1. when Batch 5 arrives, the performance of GBDT become the worst.
- For updatable baseline models of Wide&Deep, DeepFM and GBDT2NN, their performance on the LC credit dataset also gradually decline after Batch 1, but the decline rate is slower than GBDT and OICSM-off. On the PPD credit dataset, their performance are relatively stable. This proves that these updatable models have obvious advantages than non-updatable models. It is necessary to update credit scoring model online in time with the newly generated data.

TABLE 5. The AUC scores of online experiment on LC dataset.

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5
GBDT	$0.7269 \pm 4e-4$	$0.7148 \pm 3e-3$	$0.6986 \pm 4e-3$	$0.6933 \pm 7e-3$	$0.6886 \pm 6e-3$
Wide&Deep	$0.7159 \pm 8e-3$	$0.7064 \pm 9e-3$	$0.6947 \pm 9e-3$	$0.6868 \pm 8e-3$	$0.6897 \pm 8e-3$
DeepFM	$0.7206 \pm 7e-3$	$0.7120 \pm 8e-3$	$0.7006 \pm 8e-3$	$0.6912 \pm 8e-3$	$0.6947 \pm 7e-3$
GBDT2NN	$0.7198 \pm 5e-4$	$0.7123 \pm 8e-4$	$0.7019 \pm 2e-3$	$0.6946 \pm 5e-4$	$0.6901 \pm 1e-3$
OICSM-off	$0.7339 \pm 6e-3$	$0.7130 \pm 2e-3$	$0.7009 \pm 4e-3$	$0.6896 \pm 3e-3$	$0.6915 \pm 5e-4$
OICSM	$0.7339 \pm 8e-4$	$0.7430 \pm 2e-3$	$0.7209 \pm 3e-3$	$0.7134 \pm 3e-3$	$0.7015 \pm 5e-3$

TABLE 6. The AUC scores of online experiment on PPD dataset.

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5
GBDT	$0.7076 \pm 5e-4$	$0.6903 \pm 2e-3$	$0.6834 \pm 1e-3$	$0.6774 \pm 4e-3$	$0.6704 \pm 2e-3$
Wide&Deep	$0.6748 \pm 4e-3$	$0.6834 \pm 2e-3$	$0.6858 \pm 5e-3$	$0.6731 \pm 8e-3$	$0.6640 \pm 8e-3$
DeepFM	$0.6668 \pm 7e-3$	$0.6782 \pm 5e-3$	$0.6806 \pm 8e-3$	$0.6741 \pm 3e-3$	$0.6724 \pm 7e-3$
GBDT2NN	$0.6994 \pm 5e-4$	$0.6803 \pm 4e-4$	$0.6997 \pm 2e-3$	$0.6858 \pm 5e-4$	$0.6955 \pm 2e-3$
OICSM-off	$0.7176 \pm 2e-3$	$0.7054 \pm 5e-3$	$0.6959 \pm 1e-3$	$0.6908 \pm 3e-3$	$0.6808 \pm 2e-4$
OICSM	$0.7176 \pm 3e-4$	$0.7282 \pm 4e-3$	$0.7306 \pm 2e-3$	$0.7241 \pm 3e-3$	$0.7224 \pm 4e-3$

TABLE 7. The AUC scores of OICSM using different time slices on LC dataset. Q is quarter and M is month.

	2016Q1	2016Q2	2016Q3	2016Q4
OICSM (Q)	$0.7339 \pm 8e-4$	$0.7430 \pm 2e-3$	$0.7209 \pm 3e-3$	$0.7134 \pm 3e-3$
OICSM(M)	$0.7360 \pm 2e-4$	$0.7467 \pm 4e-3$	$0.7322 \pm 3e-3$	$0.7201 \pm 2e-3$

TABLE 8. The AUC scores of OICSM using different time slices on PPD dataset. M is month and HM is half month.

	2014-06	2014-07	2014-08	2014-09	2014-10
OICSM (M)	$0.7176 \pm 3e-4$	$0.7282 \pm 4e-3$	$0.7306 \pm 2e-3$	$0.7251 \pm 3e-3$	$0.7224 \pm 4e-3$
OICSM (HM)	$0.7211 \pm 5e-4$	$0.7301 \pm 2e-3$	$0.7344 \pm 1e-3$	$0.7321 \pm 3e-3$	$0.7250 \pm 2e-3$

- In addition, on these two credit datasets, the performance of GBDT2NN is better than Wide&Deep and DeepFM. This indicates that, for credit datasets that contain more numerical features than categorical features, GBDT2NN is superior than other models with NN structures.
- Finally, the performance of OICSM is better than all baseline models. Because it can not only use the newly generated batch data to update the model online dynamically, but also learn over the categorical and numerical features at the same time effectively.

In addition, the smaller the time slices, the better the performance of models. We design a comparison experiment. The experimental results on LC and PPD datasets are shown in Tables 7 to 8 respectively. OICSM (Q) and OICSM (M) in Table 7 denote the results using quarter (Q) and month (M) as time slices respectively. OICSM (M) and OICSM (HM) in Table 8 denote the results using month (M) and half month (HM) as time slices respectively.

Please note that in order to show the comparison results more clearly, for the results using smaller time slices, we take the average value to compare. For example, for the 2016Q1 credit dataset of LC, we first divide it into three subsets as 2016-01, 2016-02 and 2016-03, then repeat the online experiments on these three subsets respectively to get three AUC scores. Finally, the average value of three AUC scores is used to present the performance of OICSM (M). Meanwhile, the AUC score of OICSM (Q) on 2016Q1 can be obtained from Table 5. Through above processing,OICSM (Q) and OICSM (M) are comparable on same dataset.

As shown in Table 7, the AUC scores of OICSM (M) are higher than OICSM (Q) in all batch stages. Similarly in Table 8, the AUC scores of OICSM (HM) are higher than OICSM (M) in all batch stages. Thus, the performance based on the smaller time slice is better. This result further verifies that updating the credit scoring model in a timely manner can improve the classification performance and stability of the model, and avoid the model deviation caused by changes in the data distribution.

C. EXPERIMENT SUMMARY

In summary, the following conclusions can be drawn from the experimental results:

- In offline experiment, the performance of OICSM is better than all baseline models, which shows that an effective credit scoring model needs the capability to learning over both the categorical and numerical features simultaneously.
- In online experiment, the performance of updatable models is better than non-updatable models, which shows that it is necessary to use the newly generated data to update the model online dynamically to correct the model deviation caused by changes in the data distribution.
- Combining offline and online experiments, the performance of OICSM is better than all baseline models, which shows that our model is effective, and can simultaneously solve the two problems in existing models.

VI. CONCLUSION

In this paper, we propose a new credit scoring model OICSM for P2P lending. OICSM is composed of two parts. This integration can not only learning over two features simultaneously, but also update online dynamically using the batch processing capability of its NN structure. In order to verify the effectiveness and superiority of proposed OICSM, we select two real and representative credit datasets of P2P Lending and design offline and online experiments. Experimental results demonstrate that OICSM outperforms all other baseline models. This method has a cold start problem. To solve this problem, we will try to use the transfer learning method in the future. OICSM can make a more accurate assessment of loan applicant's credit and is especially suitable for P2P Lending with very frequent transactions and massive users.

REFERENCES

[1] K. Bastani, E. Asgari, and H. Namavari, "Wide and deep learning for peer-to-peer lending," *Expert Syst. Appl.*, vol. 134, pp. 209–224, Nov. 2019.

[2] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, "Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning," *Electron. Commerce Res. Appl.*, vol. 31, pp. 24–39, Sep. 2018.

[3] D. Babaev, M. Savchenko, A. Tuzhilin, and D. Umerenkov, "ET-RNN: Applying deep learning to credit loan applications," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2183–2190.

[4] Y.-C. Chang, K.-H. Chang, and G.-J. Wu, "Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions," *Appl. Soft Comput.*, vol. 73, pp. 914–920, Dec. 2018.

[5] C. Wang, D. Han, Q. Liu, and S. Luo, "A deep learning approach for credit scoring of Peer-to-Peer lending using attention mechanism LSTM," *IEEE Access*, vol. 7, pp. 2161–2168, 2019.

[6] Y. Liu, X. Li, and Z. Zhang, "A new approach in reject inference of using ensemble learning based on global semi-supervised framework," *Future Gener. Comput. Syst.*, vol. 109, pp. 382–391, Aug. 2020.

[7] K. Niu, Z. Zhang, Y. Liu, and R. Li, "Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending," *Inf. Sci.*, vol. 536, pp. 120–134, Oct. 2020.

[8] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu, "DeepGBM: A deep learning framework distilled by GBDT for online prediction tasks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 384–394.

[9] D. Durand, "Risk elements in consumer instalment financing," in *NBER Books*. Cambridge, MA, USA: National Bureau of Economic Research, Inc., 1941.

[10] Y. E. Orgler, "A credit scoring model for commercial loans," *J. Money, Credit Banking*, vol. 2, no. 4, pp. 435–445, 1970.

[11] E. I. Altman and G. Sabato, "Modeling credit risk for SMEs: Evidence from the US market," in *World Scientific Book Chapters*. Singapore: World Scientific, 2013, pp. 251–279.

[12] S. Y. Sohn, D. H. Kim, and J. H. Yoon, "Technology credit scoring model with fuzzy logistic regression," *Appl. Soft Comput.*, vol. 43, pp. 150–158, Jun. 2016.

[13] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Syst. Appl.*, vol. 78, pp. 225–241, Jul. 2017.

[14] N.-C. Hsieh and L.-P. Hung, "A data driven ensemble classifier for credit scoring analysis," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 534–545, Jan. 2010.

[15] Z. Zhao, S. Xu, B. H. Kang, M. M. J. Kabir, Y. Liu, and R. Wasinger, "Investigation and improvement of multi-layer perceptron neural networks for credit scoring," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3508–3516, May 2015.

[16] L. Ma, X. Huo, X. Zhao, and G. D. Zong, "Observer-based adaptive neural tracking control for output-constrained switched MIMO nonstrict-feedback nonlinear systems with unknown dead zone," *Nonlinear Dyn.*, vol. 99, no. 2, pp. 1019–1036, Jan. 2020.

[17] C. Deng, W.-W. Che, and P. Shi, "Cooperative fault-tolerant output regulation for multiagent systems by distributed learning control approach," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2020, doi: 10.1109/TNNLS.2019.2958151.

[18] V. Kozeny, "Genetic algorithms for credit scoring: Alternative fitness function performance comparison," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 2998–3004, 2015.

[19] S. Maldonado, J. Pérez, and C. Bravo, "Cost-based feature selection for support vector machines: An application in credit scoring," *Eur. J. Oper. Res.*, vol. 261, no. 2, pp. 656–665, Sep. 2017.

[20] S. Finlay, "Multiple classifier architectures and their application to credit risk assessment," *Eur. J. Oper. Res.*, vol. 210, no. 2, pp. 368–378, Apr. 2011.

[21] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 223–230, Jan. 2011.

[22] Y. Xia, C. Liu, B. Da, and F. Xie, "A novel heterogeneous ensemble credit scoring model based on bstacking approach," *Expert Syst. Appl.*, vol. 93, pp. 182–199, Mar. 2018.

[23] X. Qiu, Y. Zuo, and G. Liu, "ETCF: An ensemble model for CTR prediction," in *Proc. 15th Int. Conf. Service Syst. Service Manage. (ICSSSM)*, Jul. 2018, pp. 1–5.

[24] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2010-82, 2010.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[26] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran, "Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing," *Mech. Syst. Signal Process.*, vol. 21, no. 2, pp. 930–942, Feb. 2007.

[27] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3149–3157.

[28] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR (Workshop Poster)*, 2013, pp. 1–12.

[29] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.

[30] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[32] K. D. Humbird, J. L. Peterson, and R. G. Mcclarren, "Deep neural network initialization with decision trees," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1286–1295, May 2019.

[33] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, and F. Sun, "Model ensemble for click prediction in bing search ads," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW Companion)*, 2017, pp. 689–698.

[34] J. Zhu, Y. Shan, J. Mao, D. Yu, H. Rahmanian, and Y. Zhang, "Deep embedding forest: Forest-based serving with deep embedding features," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1703–1711.

[35] X. Chen, Z. Liu, M. Zhong, X. Liu, and P. Song, "A deep learning approach using DeepGBM for credit assessment," in *Proc. Int. Conf. Robot., Intell. Control Artif. Intell. (RICAI)*, 2019, pp. 774–779.

[36] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[37] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1725–1731.

[38] A. Veronika Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," 2018, *arXiv:1810.11363*. [Online]. Available: http://arxiv.org/abs/1810.11363

ZAIMEI ZHANG received the Ph.D. degree in management science and engineering from Hunan University, China, in 2011. She is currently an Assistant Professor with the College of Economics and Management, Changsha University of Science and Technology, China. Her research interests include financial engineering, big data, and artificial intelligence.

KUN NIU is currently pursuing the master's degree with Hunan University. His research interests include data mining and big data.

YAN LIU received the Ph.D. degree in computer science and technology from Hunan University, China, in 2010. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, China. His research areas include big data, artificial intelligence, and parallel and distributed systems.