DeRisk: An Effective Deep Learning Framework for Credit Risk Prediction over Real-World Financial Data

Yancheng Liang[13*]   **Jiajie Zhang**[1*]   **Hui Li**[2]   **Xiaochen Liu**[2]
Yi Hu[2]   **Yong Wu**[2]   **Jinyao Zhang**[2]   **Yongyan Liu**[2]   **Yi Wu**[13]
[1] Tsinghua University   [2] Fintopia Group   [3] Shanghai Qi Zhi Institute

Abstract

Despite the tremendous advances achieved over the past years by deep learning techniques, the latest risk prediction models for industrial applications still rely on highly hand-tuned stage-wised statistical learning tools, such as gradient boosting and random forest methods. Different from images or languages, real-world financial data are high-dimensional, sparse, noisy and extremely imbalanced, which makes deep neural network models particularly challenging to train and fragile in practice. In this work, we propose *DeRisk*, an effective deep learning risk prediction framework for credit risk prediction on real-world financial data. *DeRisk* is the first deep risk prediction model that outperforms statistical learning approaches deployed in our company's production system. We also perform extensive ablation studies on our method to present the most critical factors for the empirical success of *DeRisk*.

1 Introduction

Credit risk is the risk of loan default or loan delinquency when a borrower fails to repay on time. Credit risk prediction is an analytical problem that is vital for financial institutions when they are formulating lending strategies for loan applications. It helps make lending decisions by assessing the solvency of the applicants from their credit information. Accurate prediction keeps bad debts at a low level, which directly saves substantial financial loss for the multi-billion dollar credit loan industry (Malekipirbazari and Aksakalli, 2015; Tan et al., 2018). As credit risk is one major threat to financial institutions (Buehler et al., 2008; Li et al., 2015; Ma et al., 2018; Tan et al., 2018), better credit risk prediction also improves the risk management capacity of banks and financial technology companies.

Although credit scores, such as FICO Score, have been widely used as mainstream risk indicators by many financial institutions, data-driven

methods have recently shown their great potential and superior practical performances (Xu et al., 2021). Deep learning (DL), the dominating modeling technique in various domains such as computer vision, natural language processing, and recommendation system, has been a promising and increasingly popular tool considered to tackle financial problems. Recent attempts include market prediction (Ding et al., 2015; Minh et al., 2018), stock trading (Sezer et al., 2017) and exchange rate prediction (Shen et al., 2015). Despite the recent trend of using deep models, non-DL methods, such as XGBoost and logistic regression, remain the most effective techniques so far for credit risk prediction in the financial industry. Many existing studies have shown that neural network models lead to similar or even worse performances than non-DL methods (Fu, 2017; Kvamme et al., 2018; Varmedja et al., 2019; Li et al., 2020; Moscato et al., 2021).

Credit risk prediction can be formulated as a binary classification problem, where the goal is to learn a function $f_\theta : \mathcal{X} \rightarrow [0, 1]$ to map the credit information $x \in \mathcal{X}$ of an applicant to a risk score $y \in [0, 1]$ that represents the probability of default.

Despite such a simple problem formulation, credit risk prediction can be particularly challenging. Existing deep-learning-based solutions mainly focus on e-commerce consumer data (Liang et al., 2021), which typically include dense features and highly frequent user activities, such as clicks and payments, on e-commerce platforms. However, these fine-grained data are not commonly available to financial institutions. Specifically, in our application, we adopt the official credit reports provided by the Credit Reference Center (CRC) of the People's Bank of China. These financial data are of much lower quality, i.e., containing much higher dimensions (over 4k) with a large portion of missing entries and extreme values, due to low-frequency credit records. End-to-end training neural networks on these data can be substantially more challeng-

---

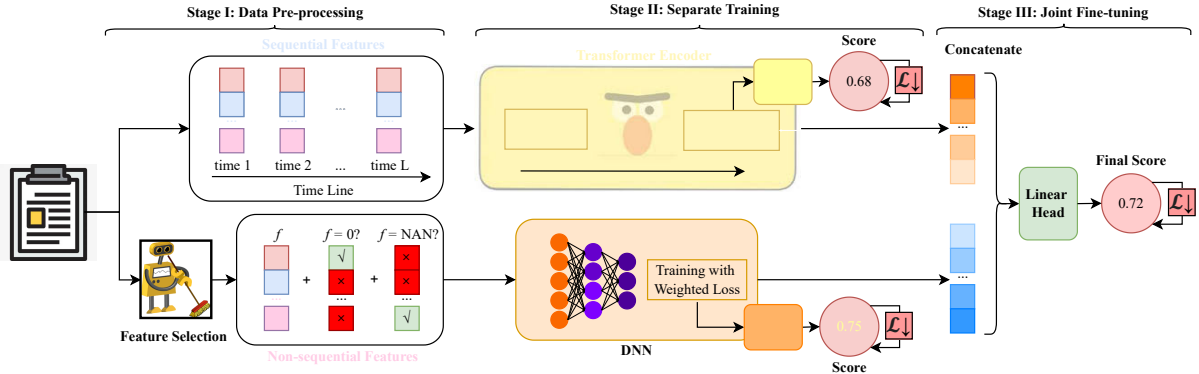*Both authors contributed equally to this research.

Figure 1: The three-stage pipeline of our *DeRisk* framework. First, in data pre-processing, feature selection and DL-specific data argumentation are adopted to benefit the optimization of DL models. Then we separately train two models for non-sequential data and sequential data, respectively. Finally, we combine them and fine-tune the joint model on the whole multi-format data. In contrast to the end-to-end paradigm in conventional DL applications, we remark the multi-stage process is critical to the overall success of *DeRisk* on real-world financial data.

ing and brittle (Poole et al., 2016; Borisov et al., 2021). Therefore, to the best of our knowledge, most financial institutions (e.g., banks) still adopt non-DL-based methods.

In this work, we present a successful industrial case study by developing an effective deep learning framework, *DeRisk*, which outperforms our production decision-tree-based system, on real-world financial data. Our *DeRisk* framework consists of three major stages including data pre-processing, separate training of non-sequential and sequential models, and joint fine-tuning. We also design a collection of practical techniques to stabilize deep neural network training under the aforementioned challenges. Specifically for the low-quality real-world financial data, we observe that a multi-stage process with feature selection and DL-specific engineering processing can be critical to the overall success of our framework.

Main contributions. (1) We develop a **comprehensive workflow** that *considers all the model training aspects* for risk prediction. (2) We implement *DeRisk*, the *first* deep risk prediction model that **outperforms statistical learning approaches** on *real-world financial data*. (3) We conduct **extensive ablation studies** on the effect of different technical components of *DeRisk*, which *provides useful insights and practical suggestions* for the research community and relevant practitioners.

2 Related Work

There have been extensive studies using machine learning techniques for credit risk prediction, in-

cluding linear regression (Puro et al., 2010; Guo et al., 2016), SVM (Jadhav et al., 2018; Kim and Cho, 2019), decision tree based methods like Random Forest (RF) (Malekipirbazari and Aksakalli, 2015; Varmedja et al., 2019; Xu et al., 2021) or Gradient Boost Decision Tree (GBDT) (Xia et al., 2017a; He et al., 2018), deep learning (Byanjankar et al., 2015; Kvamme et al., 2018; Yang et al., 2018; Yotsawat et al., 2021), or an ensemble of them (Fu, 2017; Li et al., 2020). Most of these works use data with non-sequential features. Although deep learning is applied, empirical results find that XGBoost or other GBDT approaches usually outperforms deep learning (Fu, 2017; Kvamme et al., 2018; Varmedja et al., 2019; Xu et al., 2021).

On the other hand, deep learning has shown its superiority beyond tabular data through the flexibility of deep neural networks. Convolutional Neural Network (CNN) (Kvamme et al., 2018), Long Short-Term Memory (LSTM) (Yang et al., 2018) and Graph Neural Network (GNN) (Wang et al., 2021a) are adopted for sequential data or graph data since other machine learning techniques like GBDT fail to properly model non-tabular data. According to (Liang et al., 2021), deep learning outperforms conventional methods on multimodal e-commerce data for credit risk prediction.

Many data challenges in financial applications are also common in other machine learning fields. (1) For high-dimensional data, many feature selection methods have been proposed, including filter methods (Gu et al., 2011), wrapper methods (Yamada et al., 2014) and embedded methods (Feng and Simon, 2017). Many risk prediction works

have adopted feature selection for better performance (Xia et al., 2017a; Ha et al., 2019; Li et al., 2020) or interpretability (Ma et al., 2018; Xu et al., 2021). (2) Handling multiple data formats and feature types is related to the field of deep learning for tabular data (Gorishniy et al., 2021; Borisov et al., 2021). There are typical three popular deep neural network architectures for tabular data (Klambauer et al., 2017; Huang et al., 2020; Arık and Pfister, 2021), including Multi-Layer Perception (MLP), Residual Network (ResNet) (He et al., 2016) and Transformer (Vaswani et al., 2017). Similar to the financial domain, it is also reported that deep models are not universally superior to GBDT models (Gorishniy et al., 2021) on tabular data. (3) For the out-of-time distribution shift issue, it is common to split training and test data according to the temporal order (Kvamme et al., 2018; Jiang et al., 2021). (4) Furthermore, data imbalance is also a long-standing problem in machine learning research. Among the popular over-sampling and under-sampling strategies (He et al., 2018; Bastani et al., 2019; Mahbobi et al., 2021), Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) is a widespread technique for synthetic minority data, which is also reported be effective for credit risk prediction (Bastani et al., 2019). Generative adversarial networks can also be used to generate additional minority data (Mariani et al., 2018) and this method can be applied to financial data (Liu et al., 2020) for risk prediction. However, these methods are limited to non-sequential data generation, while our financial data has multiple formats. Class-balanced loss is another method to make the model attend more to the minority samples (Lin et al., 2017; Xia et al., 2017b; Cui et al., 2019; Ren et al., 2022). Comparative experiments (Kaur et al., 2019; Moscato et al., 2021) show that all strategies have their pros and cons. In our work, we use a class-balanced loss to mitigate the problem of data imbalance, and different strategies are used for non-sequential data and sequential data thanks to their great difference in data dimension.

3 Preliminary

In this section, we first present the problem statement for the credit risk prediction task, and then introduce the credit information and labels used in the task.

3.1 Task Formulation

The credit risk prediction task aims to decide whether a loan can be granted to the applicant according to his/her credit information. To be more specific, the risk prediction model needs to learn a function $f_\theta : \mathcal{X} \rightarrow [0, 1]$, which takes the credit information $x \in \mathcal{X}$ of an applicant as input and produces a risk score $y \in [0, 1]$ that represents the probability of delinquent on the applicant's payments.

3.2 Multi-format Credit Information.

In this work, we adopt the credit information in the credit report data that is generally available in financial institutions. The credit report data of an applicant consists of two parts: non-sequential features and sequential features. Specifically, the non-sequential part usually contains thousands of stable profiles of the applicant, including age, marital status, industry, property status, etc. We remark that the non-sequential data of a credit report can be extremely high-dimensional and sparse, which requires further processing to successfully train deep neural network models. The sequential part contains dozens of features and consists of three components of the applicant's financial behavior organized by time: (1) applicant's past loan information (**loan**), including the date of loan issuing, type of lending institution, loan amount, etc.; (2) the records that applicant's credit report was inquired in the past (**inquiry**), including inquiry time, inquiry institutions, inquiry reasons, etc.; (3) applicant's credit card information (**card**), including card application date, credit card type, currency, etc. Note that the number of sequential features is much smaller than non-sequential features.

3.3 Multiple Labels and Imbalanced Data

Loan repayments naturally generate multiple labels because of installment (e.g., the first or the second month to pay back) and different degrees of delinquency (e.g., one-week or one-month delay). These labels are roughly categorized into short-term labels (e.g., the first/second/third installment is more than 30 days overdue) and long-term labels (e.g., any installment in recent 12 months is more than 5/15/30 days overdue). Due to the general priority of short-term benefits and the convenience of subse-

quent collection, financial institutions typically use short-term labels for evaluation. However, directly using this short-term evaluation label as the training label can be suboptimal. The choice of training label needs careful consideration for the best practice. Note that all these labels are particularly imbalanced (10% or even 1% for minority samples) because applicants who pay on time are much more than applicants who are overdue. Therefore, different choices of labels may lead to drastically different model performances in practice, as shown in our ablation study in Section 7.2.

4 Methodology

4.1 Overall Pipeline

The overall pipeline of our *DeRisk* framework is shown in Figure 1. Firstly, we apply careful data processing to turn noisy and irregular input features into a neatly structured format, which is indispensable for training deep networks. Secondly, to well utilize both sequential and non-sequential features, we design two main sub-models: a DNN model for processing non-sequential features and a Transformer-based model $\mathcal{M}_S$ for processing sequential features. We train them separately in the second stage. In the last stage, we fuse $\mathcal{M}_{NS}$ and $\mathcal{M}_S$ by concatenating the final hidden layers from both models and applying another linear head to give the final prediction score. We jointly fine-tune this whole model to get improved performance.

4.2 Selection of Training Label

As we mentioned in Sec. 3, there are multiple labels in risk prediction tasks that record an applicant's repayment behavior in different time periods. Among these labels, we choose a long-term label to train our model for two reasons. First, long-term labels are more balanced than short-term labels. Second, the data distribution (e.g., the ratio of negative and positive data) varies over time (see Appendix A.3) because of economic changes and the continual improvement of our deployed model. The long-term label is less sensitive to these influences and is more stable because it summarizes an applicant's behavior in the last 12 months, conceptually performing a smoothing operator over the timeline. We believe this will make our model more generalizable and perform better on the out-of-time test set, though predicting long-term risk is inherently more difficult.

4.3 Data Pre-Processing

The credit report data, especially the non-sequential data, is extremely complex and noisy, as it contains many missing values and outlier values. This low-quality input can make the learning process unstable and hurts the final performance. Therefore, proper data pre-preprocessing can be significantly beneficial for the optimization of DL models.

Both sequential and non-sequential features can be divided into three types: time features (i.e., features about time such as credit card issue date), real-value features (e.g., age, loan amount), and category features (e.g., industry, type of lending institution). For the time features, we always use a relative date difference to avoid the models memorizing input data according to the date. We also apply normalization for the numerical time features and real-value features, and discard minor classes in the category features.

In addition, we adopt specific techniques for non-sequential features. We found that lots of non-sequential real-value features are useless noise and even harmful for training. Hence we adopt a commonly-used feature selection technique that utilizes XGBoost (Chen et al., 2015) to select the most important 500 features among thousands of non-sequential real-value features and discard the others. Besides, most non-sequential features have many 0s and missing values (NAN) that naturally arise from the financial behaviors and data collection processes, which makes non-sequential data sparse, noisy, and problematic for DL training. These 0s and NANs are not necessarily meaningless, e.g., a NAN in "The time of first application for a mortgage" may imply that this applicant has never applied for a mortgage. Besides, if we simply fill these entries with a constant c, it will influence those true entries close to c and significantly influence the learned model. So, we treat these 0s and NANs carefully. For every category feature, we add a category $\langle \text{NAN} \rangle$, and for every real-value and time feature, besides replacing all NANs with 0s, we also create two indicators that directly tell whether a value is 0 and is NAN. With explicit indicators, DL models can therefore directly utilize the information implied by meaningful 0s and NANs and learn to ignore those 0s and NANs that are harmful to training.
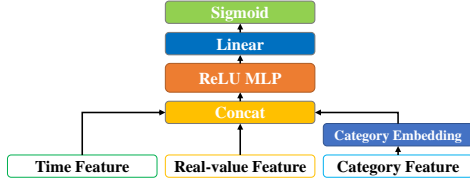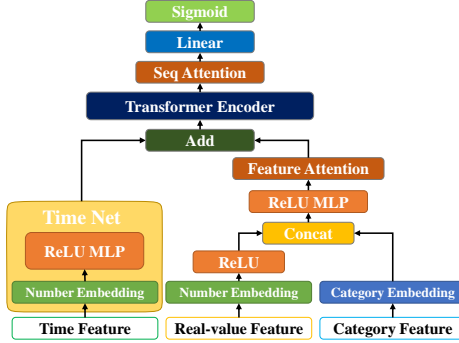
Figure 2: Non-sequential DNN model.

Figure 3: Sequential Transformer-based model.

4.4 Modeling Non-sequential Features

We adopt a simple but effective neural network for non-sequential features. The architecture is shown in Figure 2. Firstly it uses an embedding layer to convert category features into dense vectors and concatenate them with time and real-value features to the dense input $x_{\text{dense}}^{\text{NS}} \in \mathbb{R}^{m_1}$. Then $x_{\text{dense}}^{\text{NS}}$ is fed into a MLP (multi-layer perceptron) with ReLU activation function to get the non-sequential output hidden state $x_{\text{final}}^{\text{NS}} \in \mathbb{R}^{m_2}$. And the final prediction $\hat{y}^{\text{NS}}$ is computed as: $\hat{y}^{\text{NS}} = \sigma((\text{w}_{\text{logit}}^{\text{NS}})^T x_{\text{final}}^{\text{NS}} + \text{b}_{\text{logit}}^{\text{NS}})$, where $\text{w}_{\text{logit}}^{\text{NS}} \in \mathbb{R}^{m_2}$ and $\text{b}_{\text{logit}}^{\text{NS}}$ are the weight vector and bias for the logit, respectively, $z^{\text{NS}} = (\text{w}_{\text{logit}}^{\text{NS}})^T x_{\text{final}}^{\text{NS}} + \text{b}_{\text{logit}}^{\text{NS}}$ is the logit, and $\sigma(x) = 1/(1 + \exp(-x))$ is sigmoid activation.

4.5 Modeling Sequential Features

4.5.1 Architecture

We adopt a Transformer (Vaswani et al., 2017)-based model for its strong modeling capacity. The architecture is shown in Figure 3. Three such models, $\mathcal{M}_{\text{card}}$, $\mathcal{M}_{\text{inquiry}}$, and $\mathcal{M}_{\text{loan}}$, are used for card, inquiry, and loan features, respectively. Suppose the sequence length is l and the embedding size is e. Firstly a time net will convert the time feature into time embedding $E_t \in \mathbb{R}^{l \times e}$, which plays a role of position embedding, and attention is used to merge different feature embeddings into one, i.e., $E_f \in \mathbb{R}^{l \times e}$. Then a Transformer encoder will encode the sequential embeddings $E = E_t + E_f$ into

hidden feature $x_h \in \mathbb{R}^{l \times e}$, which will be pooled by another attention into output feature $x_{\text{final}}^* \in \mathbb{R}^e$, where $*$ refer to card, inquiry or loan. We concatenate $x_{\text{final}}^{\text{card}}$, $x_{\text{final}}^{\text{inquiry}}$, and $x_{\text{final}}^{\text{loan}}$ to obtain $x_{\text{final}}^{\text{S}} \in \mathbb{R}^{3e}$. At last, similar to non-sequential case, we have logit $z^{\text{S}} = (\text{w}_{\text{logit}}^{\text{S}})^T x_{\text{final}}^{\text{S}} + \text{b}_{\text{logit}}^{\text{S}}$ and final predication $\hat{y}^{\text{S}} = \sigma(z^{\text{S}})$. To improve the generalization ability of the sequential model, we share the time net and Transformer encoder among $\mathcal{M}_{\text{card}}$, $\mathcal{M}_{\text{inquiry}}$, and $\mathcal{M}_{\text{loan}}$.

4.5.2 Mask Language Model Pre-training

During training, we found that optimization of the sequential model is much harder than the non-sequential model (the left part of Figure 4) due to the scarcity of sequential features compared with non-sequential features. To ease the training of the sequential model, we adopt mask language model (MLM) pre-training as BERT (Devlin et al., 2019) to make the model first learn informative and general features from sequential data. We randomly mask the input sequential features, where 80% of masked value are replaced with token $\langle\text{MASK}\rangle$ (for category features) or 0 (for time and real-value features), 10% are replaced with a random value, and 10% remain unchanged. The three output hidden features of Transformer encoder, i.e., x_h^{card}, x_h^{inquiry}, and x_h^{loan}, will be input into different classification heads to predict different type of origin value at the masked position. After pre-training, we fine-tune $\mathcal{M}_{\text{NS}}$ on the downstream classification task.

4.6 Weighted BCE Loss

We also adopt weighted BCE loss to deal with data imbalance. Firstly, BCE (binary cross entropy) loss function is commonly used in binary classification tasks:

$$\text{BCE} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)].$$

However, when negative samples are much more than positive samples, naive BCE loss will induce the model to output $\hat{y}_i = 0$. To avoid this, we can give more weight to positive samples by using weighted BCE loss:

$$\text{WBCE} = -\frac{1}{|\mathcal{D}^-|}\sum_{i \in \mathcal{D}^-}\log(1 - \hat{y}_i) - \frac{1}{|\mathcal{D}^+|}\sum_{i \in \mathcal{D}^+}\log(\hat{y}_i),$$

where $\mathcal{D}^+ = \{i : y_i = 1\}$ is the set of positive samples and $\mathcal{D}^- = \{i : y_i = 0\}$ is the set of negative samples. Another implementation of the
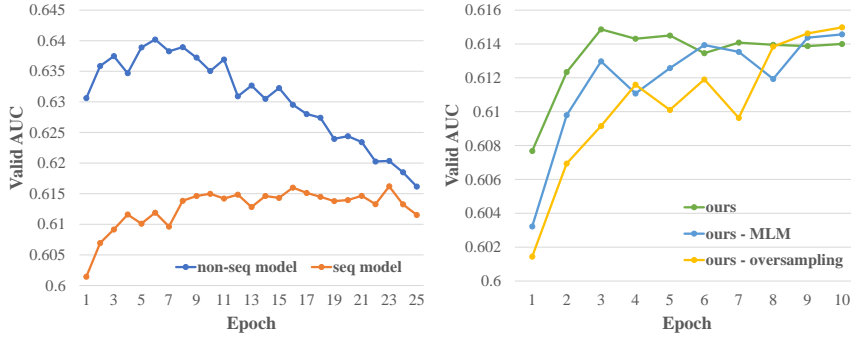
Figure 4: Left: valid AUC of non-sequential and sequential models with training epoch. Right: change of valid AUC of the sequential model with and without MLM pre-training or oversampling.

above-mentioned weighted loss is oversampling, i.e., adjusting the ratio $|\mathcal{D}^-| : |\mathcal{D}^+|$ to $1 : 1$ by re-sampling positive samples.

We use oversampling on the sequential model and use normal weighted BCE loss on the non-sequential model and joint fine-tuning stage. This is because the optimization of the non-sequential model is much harder and slower than that of the non-sequential model due to the small number of non-sequential features, while oversampling enables the model to see rare samples multiple times in one epoch and thus accelerates optimization. On the other hand, the number of non-sequential features is large and the optimization of the non-sequential model is already fast enough, oversampling may lead to overfitting on the minority samples instead.

4.7 Separate Training & Joint Fine-tuning

To fuse the sequential and non-sequential features, we use a concatenation layer (**Concat Net**) on the top of them to concatenate their output hidden states and to predict the final score, i.e., $\hat{y} = \sigma((\mathrm{w}_{\mathrm{logit}})^T x_{\mathrm{final}} + \mathrm{b}_{\mathrm{logit}})$, where $x_{\mathrm{final}}^{\mathrm{NS}} = [x_{\mathrm{final}}^{\mathrm{NS}}, x_{\mathrm{final}}^{\mathrm{S}}]$. Note that the hardness of optimization non-sequential and sequential models is different, so if we train them with the concatenation layer together from scratch, the overall model will totally rely on the non-sequential outputs, which are easier to train on, while ignoring the output of the sequential model. To avoid this and to utilize the sequential features better, we adopt a two-stage training strategy: separately train sequential and non-sequential models first and then jointly fine-tune them with the Concat Net.

Data	Time	Real-value	Category
card	1	2	5
inquiry	1	0	2
loan	1	4	5
non-sequential	13	4098	9

Table 1: Number of time, real-value, and category features in each sample of sequential and non-sequential data. The card, inquiry, and loan sequences for each user are clipped with lengths 32, 64, and 128, respectively.

5 Experiment Setup

5.1 Notation

We mainly use a long-term label $\mathcal{Y}_{\mathrm{long}}$ for training and a short-term label $\mathcal{Y}_{\mathrm{short}}^{\mathrm{eval}}$ for evaluation. $\mathcal{Y}_{\mathrm{short}}^{\mathrm{other1}}, \mathcal{Y}_{\mathrm{short}}^{\mathrm{other2}}, \mathcal{Y}_{\mathrm{short}}^{\mathrm{other3}}$, There are also three other short-term labels used in our experiments. The description of these labels are in Sec A.2.

5.2 Dataset Statistics

We sample 582,996 Yanqianguan Users and use their credit report data and repayment behavior from August 2020 to July 2021 as the dataset. To simulate the out-of-time prediction in real business scenarios, we take the 430,865 data pieces from August 2020 to May 2021 as the training set and 152,131 data pieces from June 2021 to July 2021 as the test set. The ratio of negative and positive samples is about $50 : 1$ according to the short-term label used for evaluation and is about $10 : 1$ according to the long-term label used for training.[1] For sequential data, we set the maximum sequence length of card, query, and loan data to be 32, 64, and 128, respectively, according to the distribution

---

[1]We keep the exact ratio numbers confidential due to commercial and security concerns.

Model	$\mathcal{Y}_{short}^{eval}$	$\mathcal{Y}_{short}^{other2}$	$\mathcal{Y}_{short}^{other3}$
non-seq model over non-seq data only			
XGBoost	0.6418	0.6282	0.6187
DeepFM	0.5700	0.5508	0.5478
SDCN	0.6450	0.6319	0.6236
PDCN	0.6483	0.6343	**0.6254**
AutoInt	0.6454	0.6325	0.6238
DNN	**0.6499**	**0.6349**	**0.6254**
seq model over sqe data only			
Pooled MLP	0.5996	0.5821	0.5749
LSTM	0.6108	0.5936	0.5859
Transformer	0.6132	0.5941	0.5871
Transformer+MLM	**0.6156**	**0.5971**	**0.5885**
joint model over the entire data			
Add-Attn Net	0.6504	0.6369	0.6285
Mul-Attn Net	0.6520	0.6377	0.6278
DeRisk(ours)	**0.6546**	**0.6398**	**0.6297**

Table 2: All models are evaluated by AUC scores on three different short-term labels.

of data length. Only the latest data will be included for training and evaluation. Some statistics are summarized in Table 1.

Note that all above data are definitely authorized by the customers since they hope to apply for loan in our platform and they should provide the access to their credit report. We also anonymized the names of people and organizations on credit reports to protect customers' privacy.

5.3 Evaluation Metric

The metric commonly used to evaluate credit risk prediction models $\mathcal{M}$ is AUC (Area Under the ROC Curve) score. We remark that this is a challenging task and an increment of 0.01 in AUC can be significant in performance as this results in a roughly 5% decrement of real-world bad debts.

6 Main Results

6.1 Baselines

For non-sequential model, the baselines include (1) current popular traditional ML model **XG-Boost** (Chen et al., 2015) (main baseline) and several more complicated deep models including (2) **DeepFM** (Guo et al., 2017): the final score is $y^{NS} = \sigma(z_{DNN}^{NS} + z_{FM}^{NS})$, where z_{DNN}^{NS} is the logit of DNN and z_{FM}^{NS} is the logit gotten by a FM (factorization machine (Rendle, 2010)) layer. (3) **DCNv2** (Wang et al., 2021b): use cross-network

(multiple cross layers) to obtain high-order cross feature. A DNN can be stacked on top of the cross-network (SDCN); we could also place them in parallel (PDCN). (4) **AutoInt** (Song et al., 2019): use a multi-head self-attention to learn interacted features.

For the sequential model, our baselines are **pooled MLP** (which uses a pooling layer to average hidden states of different times that are individually produced by the MLP) and **LSTM** (Hochreiter and Schmidhuber, 1997).

For the final module that fuses the output of the hidden state by the non-sequential model and sequential model, we compare our simple Concat Net with an additive attention layer (**Add-Attn Net**) and a multiplicative attention layer (**Mul-Attn Net**) that use x_{final}^{NS} as a query vector to pool output hidden feature of Transformer Encoder x_h by additive and multiplicative attention, respectively.

6.2 Evaluation and Analysis

Since our dataset has multiple formats, we first test separated models for single-format data modeling. For non-sequential data, we compare the DNN module in *DeRisk* with XGboost, a widely-used decision-tree model in our production system. We aim to show whether our *DeRisk* system and techniques can make its DNN module outperform other non-DL methods on real-world financial data. Other popular models in recommendation systems like DeepFM, DCN, and AutoInt are also tested as DL competitors. For sequential data, we consider different sequential models including Pooled MLP, LSTM, and Transformer for evaluation. Our *DeRisk* adopts Transformer and additionally adopts MLM-pretraining to accelerate training.

Finally, we consider joint models trained over the entire dataset with both formats by fusing the best non-sequential model, DNN, and the best sequential model MLM-pretrained Transfomer, to obtain joint models for the best evaluation results. With more data, the joint models outperform either separated models, but we also find different fusing techniques lead to different performances. We compare our Concat Net with two different attention-based methods.

Table 2 summarizes the main results. All models are evaluated by three different labels to show consistent results. From the results we can see that:

(1) Our non-sequential model DNN and sequential model

Change	AUC
No (ours)	**0.6546**
w/o Separate Training (end-to-end)	0.6487
w/ Freeze Sub-models	0.6512

Table 3: $\mathcal{Y}_{\text{short}}^{\text{eval}}$ AUC scores with different training strategies.

MLM+Transformer outperform all baselines, respectively. Specifically, compared with current popular XGBoost model, our DNN model $\mathcal{M}_{\text{NS}}$ and best joint model *DeRisk* (with Concat Net) improve $\mathcal{Y}_{\text{short}}^{\text{eval}}$ AUC score by 0.0081 and 0.0128, respectively.

(2) Joint fine-tuning of non-sequential and sequential models can achieve better results than only using a single non-sequential or sequential model.

(3) Complex models do not necessarily perform better: simplest DNN and Concat Net outperform other more complicated models. This indicates that the high-order features created by those additional networks such as FM and cross layers are not that helpful for the credit risk prediction task.

7 Ablation Study

In this section, we conduct a series of experiments to demonstrate the effect of each part of our *DeRisk* framework. We mainly use $\mathcal{Y}_{\text{short}}^{\text{eval}}$ for evaluation since we find it shows a consistent result with other short-term labels as in Table 2. We test the effectiveness of different modules in our multi-stage process, including separate training & joint fine-tuning, feature selection, indicator features, and MLM-pretraining. Many different techniques for data imbalance are also studied in this section. With our ablation studies, we also present best practices for training deep neural network models over real-world financial data.

7.1 Effect of Multi-stage Training

Because the hardness of optimization on non-sequential data and sequential data is different as shown in Figure 4, we first separately train $\mathcal{M}_{\text{NS}}$ and $\mathcal{M}_{\text{S}}$ and then joint fine-tune them. We also tried joint training them from scratch (end-to-end), or freezing $\mathcal{M}_{\text{NS}}$ and $\mathcal{M}_{\text{S}}$ and only tuning the concatenating layer during joint fine-tuning. The results are reported in Table 3. We can see that separate training outperforms the other two training strategies.

Training Label	Test Label	AUC
non-seq model		
$\mathcal{Y}_{\text{long}}$ (Ours)	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	**0.6499**
$\mathcal{Y}_{\text{short}}^{\text{other1}}$	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	0.6392
$\mathcal{Y}_{\text{short}}^{\text{eval}}$	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	0.6363
seq model		
$\mathcal{Y}_{\text{long}}$ (Ours)	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	**0.6156**
$\mathcal{Y}_{\text{short}}^{\text{other1}}$	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	0.6113
$\mathcal{Y}_{\text{short}}^{\text{eval}}$	$\mathcal{Y}_{\text{short}}^{\text{eval}}$	0.6105

Table 4: Experiment results of selecting different training labels on non-sequential and sequential models.

Change	AUC
No (Ours)	**0.6499**
$|\mathcal{F}_R| = 4098$	0.6415
$|\mathcal{F}_R| = 100$	0.6390
w/o Indicator	0.6426
w/ BCE Loss	0.6454
w/ Focal Loss	0.6403
w/ Oversample	0.6458

Table 5: Analysis experiment results on non-sequential DNN model, where $|\mathcal{F}_R|$ is the number of selected features.

Suggestion#1: It is beneficial to first perform separate training and then joint tuning for multi-format data. The additional tunable parameters introduced in the fine-tuning process should be sufficiently large for effective multi-format fusion.

7.2 Effect of Different Training Labels

We tried taking two short-term labels ($\mathcal{Y}_{\text{short}}^{\text{other1}}$ and $\mathcal{Y}_{\text{short}}^{\text{eval}}$) and a long-term label ($\mathcal{Y}_{\text{long}}$) as the training label, respectively. The results in Table 4 demonstrate that the long-term label is the best choice for both non-sequential and sequential models, even when the model is evaluated on a short-term label. **Suggestion#2: It is better to choose a balanced and stable signal that measures the long-term behaviors as the training label.**

7.3 Effect of Real-value Feature Selection

To show the effect of selecting real-value features with XGBoost, we compare the following three

Change	AUC
No (Ours)	**0.6156**
w/o MLM Pre-training	0.6132
w/o Oversampling	0.6153

Table 6: Analysis experiment results on sequential Transformer-based model.

cases: no selection, selecting 500 real-value features (Ours), and selecting 100 real-value features. The results in Table 5 show that selecting 500 features performs the best. This indicates that (1) by selecting real-value features with XGBoost, we can drop useful fewer features and improve the performance. (2) dropping too many features would lead to worse predictions.

Suggestion#3: It is important to perform feature selection before deep learning training. The dimension of selected features should be chosen carefully.

7.4 Effect of Indicator Features

To show the effect of NAN and zero indicators, we compare the case with and without them. As shown in Table 5, after removing indicators, the AUC score decreases by 0.0073.

Suggestion#4: Some NANs and 0s can be meaningful and it is better to use indicator features rather than simply filling these missing values with a constant or discarding them.

7.5 Comparison of Different Loss Functions

We compared the performance of using weighted BCE loss (Ours) with using naive BCE loss on the DNN model. In addition, we also tried Focal loss (Lin et al., 2017) which is designed for the data imbalance case, but the result in Table 5 shows that it is not helpful for our task and weight BCE achieves the best performance.

Suggestion#5: Adding more weight to rare positive samples is critical to prevent the model from biasing to the overwhelming negative outputs.

7.6 Effect of Oversampling

We compared the cases with and without oversampling on both the non-sequential model and sequential model to demonstrate the effect of oversampling. We can see from Table 6 and the right of Figure 4 that for sequential model, oversampling (1) improves AUC. (2) accelerates optimization. By enabling the model to see rare positive sam-

ples more times in each epoch, oversampling reduces the training difficulty of the sequential model. On the other hand, oversampling also makes the non-sequential model, the one easier to optimize, overfits more quickly on the training data and thus cannot achieve good performance as shown in Table 5. In practice, DNN with oversampling usually overfits after the first epoch.

Suggestion#6: Oversampling makes optimization of the sequential model easier and improves performance. And considering the difference between non-sequential data and sequential data, each separated model should be optimized with different sampling strategies.

7.7 Effect of MLM Pre-training of Sequential Model

From Table 6 and the right of Figure 4 that MLM pre-training of the sequential model (1) improves performance. (2) accelerates optimization. This indicates that the pre-trained model has learned some knowledge of sequential data that are useful for the risk prediction task.

Suggestion#7: MLM pre-training benefits the optimization of the sequential model on credit risk prediction.

8 Conclusion

In this work, we proposed an effective deep learning framework, *DeRisk*, which utilizes both sequential and non-sequential features for credit risk prediction. We apply careful data pre-processing to obtain clean and useful data for deep models, use MLM to pre-train the sequential model, adopt weighted BCE loss and oversampling to deal with the data imbalance problem, and select generalizable and stable training labels for better performance. The overall performance of *DeRisk* largely outperforms existing approaches on real-world financial data. We remark that it is unnecessary that a more complicated network always performs better. In our analysis, every components of the training framework including data pre-processing and a carefully designed optimization process are all critical to make deep learning models perform well on a real-world financial application. We hope our framework and analysis can bring insights for a wide range of important commercial applications and inspire future research on developing more powerful deep learning tools for real-world industrial data.

References

Sercan O Arık and Tomas Pfister. 2021. Tabnet: Attentive interpretable tabular learning. In *AAAI*, volume 35, pages 6679–6687.

Kaveh Bastani, Elham Asgari, and Hamed Namavari. 2019. Wide and deep learning for peer-to-peer lending. *Expert Systems with Applications*, 134:209–224.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2021. Deep neural networks and tabular data: A survey. *arXiv preprint arXiv:2110.01889*.

Kevin Buehler, Andrew Freeman, and Ron Hulme. 2008. The new arsenal of risk management. *Harvard Business Review*, 86(9):93–100.

Ajay Byanjankar, Markku Heikkilä, and Jozsef Mezei. 2015. Predicting credit risk in peer-to-peer lending: A neural network approach. In *2015 IEEE symposium series on computational intelligence*, pages 719–725. IEEE.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

Jean Feng and Noah Simon. 2017. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*.

Yijie Fu. 2017. Combination of random forests and neural networks in social lending. *Journal of Financial Risk Management*, 6(4):418–426.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34.

Quanquan Gu, Zhenhui Li, and Jiawei Han. 2011. Generalized fisher score for feature selection. In *27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, pages 266–273.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731.

Yanhong Guo, Wenjun Zhou, Chunyu Luo, Chuanren Liu, and Hui Xiong. 2016. Instance-based credit risk assessment for investment decisions in p2p lending. *European Journal of Operational Research*, 249(2):417–426.

Van-Sang Ha, Dang-Nhac Lu, Gyoo Seok Choi, Ha-Nam Nguyen, and Byeongnam Yoon. 2019. Improving credit risk prediction in online peer-to-peer (p2p) lending using feature selection with deep learning. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 511–515. IEEE.

Hongliang He, Wenyu Zhang, and Shuai Zhang. 2018. A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, 98:105–117.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*.

Swati Jadhav, Hongmei He, and Karl Jenkins. 2018. Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing*, 69:541–553.

Junxiang Jiang, Boyi Ni, and Chunping Wang. 2021. Financial fraud detection on micro-credit loan scenario via fuller location information embedding. In *Companion Proceedings of the Web Conference 2021*, pages 238–246.

Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. 2019. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, 52(4):1–36.

Aleum Kim and Sung-Bae Cho. 2019. An ensemble semi-supervised learning method for predicting defaults in social lending. *Engineering Applications of Artificial Intelligence*, 81:193–199.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. *Advances in neural information processing systems*, 30.

Håvard Kvamme, Nikolai Sellereite, Kjersti Aas, and Steffen Sjursen. 2018. Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, 102:207–217.

Jianping Li, Xiaoqian Zhu, Cheng-Few Lee, Dengsheng Wu, Jichuang Feng, and Yong Shi. 2015. On the aggregation of credit, market and operational risks. *Review of Quantitative Finance and Accounting*, 44(1):161–189.

Wei Li, Shuai Ding, Hao Wang, Yi Chen, and Shanlin Yang. 2020. Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in china. *World Wide Web*, 23(1):23–45.

Ting Liang, Guanxiong Zeng, Qiwei Zhong, Jianfeng Chi, Jinghua Feng, Xiang Ao, and Jiayu Tang. 2021. Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 229–237.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Yang Liu, Xiang Ao, Qiwei Zhong, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Alike and unlike: Resolving class imbalance problem in financial credit risk assessment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2125–2128.

Xiaojun Ma, Jinglan Sha, Dehua Wang, Yuanbo Yu, Qian Yang, and Xueqi Niu. 2018. 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. *Electronic Commerce Research and Applications*, 31:24–39.

Mohammad Mahbobi, Salman Kimiagari, and Marriappan Vasudevan. 2021. Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*, pages 1–29.

Milad Malekipirbazari and Vural Aksakalli. 2015. Risk assessment in social lending via random forests. *Expert Systems with Applications*, 42(10):4621–4631.

Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. 2018. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*.

Dang Lien Minh, Abolghasem Sadeghi-Niaraki, Huynh Duc Huy, Kyungbok Min, and Hyeonjoon Moon. 2018. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *Ieee Access*, 6:55392–55404.

Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperlí. 2021. A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165:113986.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. 2016. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29.

Lauri Puro, Jeffrey E Teich, Hannele Wallenius, and Jyrki Wallenius. 2010. Borrower decision aid for people-to-people lending. *Decision Support Systems*, 49(1):52–60.

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. 2022. Balanced mse for imbalanced visual regression. *arXiv preprint arXiv:2203.16427*.

Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE.

Omer Berat Sezer, Murat Ozbayoglu, and Erdogan Dogdu. 2017. A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters. *Procedia computer science*, 114:473–480.

Furao Shen, Jing Chao, and Jinxi Zhao. 2015. Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing*, 167:243–253.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170.

Fei Tan, Xiurui Hou, Jie Zhang, Zhi Wei, and Zhenyu Yan. 2018. A deep learning approach to competing risks representation in peer-to-peer lending. *IEEE transactions on neural networks and learning systems*, 30(5):1565–1574.

Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. 2019. Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Daixin Wang, Zhiqiang Zhang, Jun Zhou, Peng Cui, Jingli Fang, Quanhui Jia, Yanming Fang, and Yuan Qi. 2021a. Temporal-aware graph neural network for credit risk prediction. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 702–710. SIAM.

Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021b. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, pages 1785–1797.

Yufei Xia, Chuanzhe Liu, YuYing Li, and Nana Liu. 2017a. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241.

Yufei Xia, Chuanzhe Liu, and Nana Liu. 2017b. Cost-sensitive boosted tree for loan evaluation in peer-to-peer lending. *Electronic Commerce Research and Applications*, 24:30–49.

Junhui Xu, Zekai Lu, and Ying Xie. 2021. Loan default prediction of chinese p2p market: a machine learning methodology. *Scientific Reports*, 11(1):1–19.

Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. 2014. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207.

Zhi Yang, Yusi Zhang, Binghui Guo, Ben Y Zhao, and Yafei Dai. 2018. Deepcredit: Exploiting user cickstream for loan risk prediction in p2p lending. In *Twelfth International AAAI Conference on Web and Social Media*.

Wirot Yotsawat, Pakaket Wattuya, and Anongnart Srivihok. 2021. A novel method for credit scoring based on cost-sensitive neural network ensemble. *IEEE Access*, 9:78521–78537.

A Appendix

A.1 Training Details

For our model and all the deep-learning baselines, we use Adam (Kingma and Ba, 2014) optimizer with learning rate 5×10^{-4} and weight decay 1×10^{-4}. We set the batch size to $1,000$. For non-sequential model DNN, we set the embedding size to 16, use three-layer MLP, and set the hidden size to 1028, 256, and 128, respectively. For sequential models, we use a one-layer Transformer encoder, set the embedding size to 128, the number of heads

to 8, and the dropout probability to be 0.1. We adopt a 5-fold cross-validation on the training set and evaluate the ensembled model on the test set.

Both sequential and non-sequential features are composed of time features (i.e., features about time such as date), real-valued features, and category features. For every time feature in date format, we subtract it by the date at which the credit report is used for prediction. That is, the time feature indicates the number of days between when the financial activity happens and when the credit report is called. Then for every time and real-value feature, we do zero-mean and one-std normalization and clip all values into $[-4, 4]$ to make the distribution easier for DL models to learn. For every category feature, we merge all the categories outside the top 30 into one category $\langle \text{UNK} \rangle$.

We utilize XGBoost (Chen et al., 2015) to select the most important 500 features of the non-sequential real-value features and discard the rest of them. We simply train an XGBoost model on the same task of risk prediction. After that, we choose 500 features with the highest feature importance value to feed the non-sequential DL model. For every category feature, we add a category $\langle \text{NAN} \rangle$, and for every real-value and time feature, besides replacing all NANs with 0s, we also create two indicators $[x = 0]$ and $[x = NAN]$. Therefore, for every real-value and time feature, there will be three corresponding features after this process. Thus, the 500 features we selected above become 1500 features.

A.2 Label Notation

The dataset mainly contains two types of labels: 1) short-term label i$x$label$y$, which means the user fails to pay back y days after the xth-month's repayment deadline; 2) long-term label overdue$y$, which means the user has at least one y-day overdue behavior in the last year.

In the following experimental parts we mainly use the following labels: $\mathcal{Y}_{\text{long}}$, the long-term label overdue15; $\mathcal{Y}_{\text{short}}^{\text{eval}}$, the main short-term label i1label30 used for evaluation; $\mathcal{Y}_{\text{short}}^{\text{other1}}, \mathcal{Y}_{\text{short}}^{\text{other2}}, \mathcal{Y}_{\text{short}}^{\text{other3}}$, denoting another three short-term labels i1label15, i2label30, i3label30, for training and evaluation.

A.3 Dataset Analysis

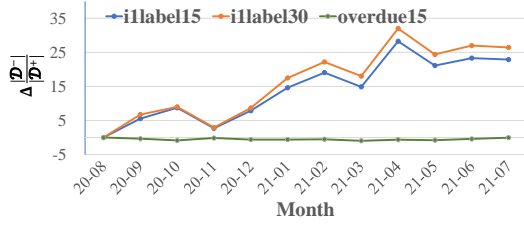We show in Figure 5 that the input data distribution, i.e., the ratio of negative and positive data,

Figure 5: The change of imbalance ratio $\Delta\frac{|\mathcal{D}^-|}{|\mathcal{D}^+|}$ from August 2020 to July 2021. Compared with i1label15 and i1label30, the ratio of overdue15 is more stable.

varies over time. Besides the changes of the economic environment, the data distribution changes also because the consumers are first filtered by a basic decision model in practice, which keeps being optimized over time. As a result of a better filtering process, fewer applicants default and the data becomes more imbalanced. (e.g., see Jan-2021 and May-2021 for i1label15 and i1label30). Empirically, compared to the short-term label, we notice that the long-term label overdue15 is less sensitive to economic environment influence and optimization of the basic decision model. It is more stable because it summarizes a customer's behavior in the last 12 months, which is conceptually performing a smoothing operator over the timeline. In addition, the prediction of long-term risk is more difficult and thus is less affected by the basic decision model.