IEICE TRANS. INF. & SYST., VOL.E105–D, NO.7 JULY 2022

LETTER

# Loan Default Prediction with Deep Learning and Muddling Label Regularization

**Weiwei JIANG**[†a)], ***Member***

**SUMMARY** Loan default prediction has been a significant problem in the financial domain because overdue loans may incur significant losses. Machine learning methods have been introduced to solve this problem, but there are still many challenges including feature multicollinearity, imbalanced labels, and small data sample problems. To replicate the success of deep learning in many areas, an effective regularization technique named muddling label regularization is introduced in this letter, and an ensemble of feed-forward neural networks is proposed, which outperforms machine learning and deep learning baselines in a real-world dataset.

***key words:*** *loan default prediction, deep learning, muddling label regularization*

## 1. Introduction

Different kinds of loans have been a major source of income for financial institutions. However, the default of loans would incur significant losses. The loan default prediction problem is thus proposed by collecting various data as the input features, e.g., personal and behavior information. While the loan default prediction problem has drawn attention from researchers in different fields and machine learning methods are already used [1], it is not fully resolved for the following challenges. The first challenge is the multicollinearity in high-dimensional input features, which introduces many highly-correlated features that are not helpful for building an efficient classifier. The second challenge is the imbalanced label problem, which is caused by the fact that overdue cases rarely occur in reality. The third challenge is the small sample problem. Unlike image or text data, tabular data used in loan default prediction are difficult to collect due to both high cost and privacy concerns.

While deep learning has been successful in the financial domain [2] and computer vision domain [3], it is not the panacea for problems with tabular data and often fails behind machine learning models, e.g., support vector machine, random forest, XGBoost and LightGBM [4]. With the small sample problem in loan default prediction, deep learning models are prone to overfitting. In this letter, we leverage the recently proposed muddling label regularization technique [5] for handling overfitting concerns and successfully train an ensemble of FFNNs that outperforms machine learning baselines, on a real-world dataset for loan default prediction.

The contributions of this letter are summarized as follows. First, we use a simple yet general workflow of applying machine learning technologies to the default prediction problem. Second, we propose the usage of muddling label regularization as an effective training method for an ensemble of standard feed-forward neural networks as the classifier for default prediction. Last, the proposed deep learning model outperforms all competitive machine learning and deep learning baselines in a real-world dataset, in terms of the F1 score.

## 2. Methodology

The overall workflow for loan default prediction is shown in Fig. 1. Common data preprocessing operations include feature deletion, missing data filling, and feature scaling. The feature reduction technique used in this letter is PCA (principal component analysis). The oversampling techniques used in this letter include SMOTE (synthetic minority oversampling technique) and ADASYN (adaptive synthetic sampling approach), both of which are widely used in the literature. SMOTE applies kNN to choose k nearest neighbors to create the synthetic samples and ADASYN adaptively generates minority data samples according to the density distribution using k nearest neighbors.

The base classifier used in this letter is FFNN (feed-forward neural network). Denote the tabular dataset with $N$ samples as $\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{(X_i, y_i)\}$, where $X_i \in \mathcal{R}^d$, $y_i \in \{0, 1\}$, and $d$ is the feature number. Denote $\mathbf{X} \in \mathcal{R}^{n\times d}$ as a batch of $n$ samples, $\mathbf{H}^\ell \in \mathcal{R}^{n\times h}$ as the output of hidden layer $\ell$ with $h$ neurons, and $\mathbf{O} \in \mathcal{R}^n$ as the output, then the mapping from $\mathbf{X}$ to $\mathbf{O}$ in a FFNN with $L$ layers can be denoted as follows: $\mathbf{H}^1 = \sigma(\mathbf{X}\mathbf{W}^1 + \mathbf{b}^1)$, $\mathbf{H}^\ell = \sigma(\mathbf{H}^{\ell-1}\mathbf{W}^\ell + \mathbf{b}^\ell)$, and $\mathbf{O} = \mathbf{H}^L\mathbf{W}^{L+1} + \mathbf{b}^{L+1}$, where $\mathbf{W}$ and $\mathbf{b}$ are model parameters to be

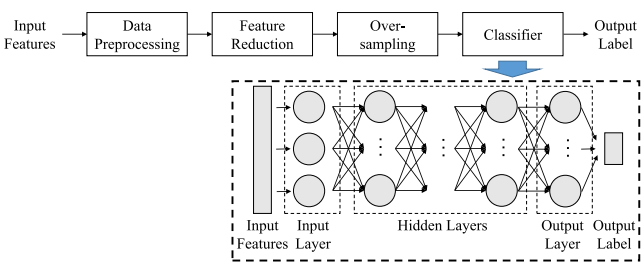



**Fig. 1** The overall workflow for loan default prediction.

Manuscript received January 13, 2022.
Manuscript revised March 15, 2022.
Manuscript publicized April 4, 2022.
[†]The author is with School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China.
a) E-mail: jww@bupt.edu.cn
DOI: 10.1587/transinf.2022EDL8003

trained from data, $\sigma(*)$ is the activation function, e.g., ReLU used in this letter. Denote $\theta$ as $\{\mathbf{W}^i, \mathbf{b}^i\}$, $i = 1, 2, \ldots, L+1$, and the above process can be denoted as a function $f$, i.e., $O = f(\theta, \mathbf{X})$.

The loss function used to train the FFNN is extended from the common BCE (binary cross entropy) function for binary classification with muddling label regularization (BCE-MLR) in this letter. Denoting $\mathbf{Y} \in \mathcal{R}^n$ as the true labels, the BCE loss is defined as $BCE(\mathbf{Y}, \mathbf{O}) = -\frac{1}{n}[\mathbf{Y}^\top \log(Sig(\mathbf{O})) + (\mathcal{I}_n - \mathbf{Y})^\top \log(\mathcal{I}_n - Sig(\mathbf{O}))]$, where $\mathcal{I}$ is the identity matrix, $Sig(\cdot)$ is the sigmoid function and $\top$ is the matrix transpose.

To increase the generalization ability of FFNNs, ridge regularization and random permutations are introduced in BCE-MLR. For a ridge regularization term $\lambda > 0$, we define $\mathbf{P} = \mathbf{P}(\theta, \lambda, \mathbf{X}) = [(\mathbf{H}^L)^\top \mathbf{H}^L + \lambda \mathcal{I}_h]^{-1}(\mathbf{H}^L)^\top \in \mathcal{R}^{h \times n}$. For a permutation $\pi$ of $n$ elements, the label permulation operator of $\mathbf{Y}$ is defined as $\pi(\mathbf{Y}) = (y_{\pi(1)}, \ldots, y_{\pi(n)})$. For a given number $T \geq 1$, $(\pi^t(\mathbf{Y}))_{t=1}^T$ are $T$ label permutation operators that are drawn uniformly at random in the set of all possible label permutations.

Then, the BCE-MLR function is defined as follows: BCE-MLR$(\theta, \lambda) = BCE(\mathbf{Y}, \mathbf{Y}^* + (\mathcal{I}_n - \mathbf{H}^L\mathbf{P})\xi + \mathbf{H}^L\mathbf{P}\mathbf{Y}^*) + \frac{1}{T}\sum_{t=1}^T |BCE(\mathbf{Y}, \overline{\mathbf{Y}}\mathcal{I}_n) - BCE(\pi^t(\mathbf{Y}^*), \pi^t(\mathbf{Y}^*) + (\mathcal{I}_n - \mathbf{H}^L\mathbf{P})\xi_t + \mathbf{H}^L\mathbf{P}\pi^t(\mathbf{Y}^*))|$, where $\xi$ and $(\xi_t)_{t=1}^T$ are *i.i.d.* $\mathcal{N}(0_n, \mathcal{I})$ vectors, $\mathbf{Y}^* = 2\mathbf{Y} - 1$ and $\overline{\mathbf{Y}} = mean(\mathbf{Y})$.

## 3. Dataset Description

The real-world dataset used in this letter comes from a data competition[†] hosted by China UnionPay Merchant Services, which provides nationwide payment services for China UnionPay-labeled cards. The input features include personal information and property status (feature 1 to feature 19, including gender, age, house, car, etc.), bank card holding information (feature 20 to feature 40, including card type and number, bank types and locations, etc.), transaction information (feature 41 to feature 130, including the bank account balance, trading volume, etc.), lending information (feature 131 to feature 146, including the lending number and amount in different time periods, etc.), repayment information (feature 147 to feature 187, including the repayment number and amount in different time periods, etc.), and loan application information (feature 188 to feature 199, including the loan number and amount in different time periods, etc.). The target is to predict the loan default case, which is a binary classification problem with label 1 as overdue and 0 as no overdue. The labels are highly imbalanced, with 2,144 samples with label 1 and 8,873 samples with label 0.

## 4. Experiment and Discussion

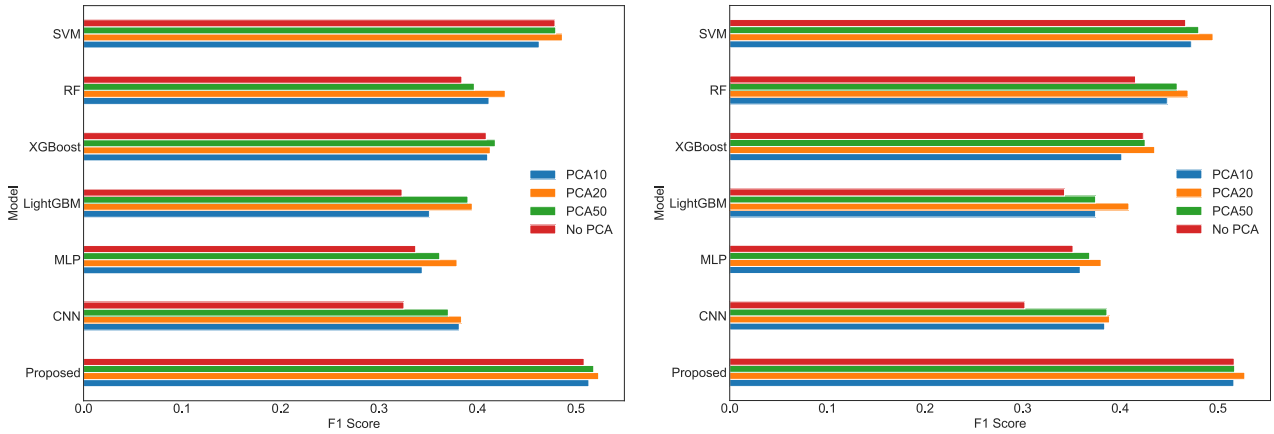
To evaluate the proposed deep learning approach and compare it with machine learning baselines, the dataset is split into a training set and a test set with a split ratio of 80% : 20%. The input features with a missing rate higher than 40% are deleted and the remaining 117 features are fulfilled with zero values before being used as the input of the PCA module. Min-max normalization is used as the feature scaling technique. Different numbers of components to keep in PCA are set as 10, 20, and 50. Since accuracy is not a suitable evaluation metric for imbalanced classification, the F1 score is adopted in this study, which is the harmonic mean of the precision and recall. The baseline models include four traditional machine learning models, i.e., support vector machine (SVM), random forest (RF), XGBoost and LightGBM, which are implemented and fine-tuned with scikit-learn. The baseline models also include two recent deep learning models, i.e., multilayer perceptron (MLP) [6] and convolutional neural network (CNN) [7]. A voting ensemble of three FFNNs based on the majority rule is implemented with PyTorch, in which 1, 2 or 3 hidden layers are used in FFNNs, with 1024 neurons in each layer. Each FFNN is trained for 200 epochs, with a batch size of 1 and a learning rate of 1e-3.

The results are summarized in Table 1. The main finding is that our proposed deep learning approach with the FFNN ensemble model and BCE-MLR loss function manages to achieve the highest F1 score on the test set with ADASYN as the over-sampling technique and 20 PCA components to keep. Our results reveal a promising research direction by applying the proposed deep learning approach for similar problems with tabular data. There are some other observations from our results. The first observation is that almost all evaluated models perform poorly without performing over-sampling techniques, with XGBoost as an exception. The second observation is that PCA in this case study brings a performance improvement that exists but is not so impressive. The third observation is that the best choice for the number of PCA components to keep is 20, as indicated

**Table 1** Experimental results with F1 scores for different models.

| Model | Over-sampling | PCA | | | |
|---|---|---|---|---|---|
| | | N/A | PCA50 | PCA20 | PCA10 |
| SVM | N/A | 0.0744 | 0.1116 | 0.1028 | 0.0274 |
| | SMOTE | 0.4787 | 0.4794 | 0.4862 | 0.4627 |
| | ADASYN | 0.4669 | 0.4803 | 0.4950 | 0.4728 |
| RF | N/A | 0.0583 | 0.0046 | 0.0405 | 0.0573 |
| | SMOTE | 0.3841 | 0.3968 | 0.4282 | 0.4117 |
| | ADASYN | 0.4156 | 0.4583 | 0.4694 | 0.4482 |
| XGBoost | N/A | 0.3887 | 0.4134 | 0.4059 | 0.3472 |
| | SMOTE | 0.4089 | 0.4181 | 0.4130 | 0.4103 |
| | ADASYN | 0.4236 | 0.4255 | 0.4352 | 0.4015 |
| LightGBM | N/A | 0.2275 | 0.3130 | 0.0823 | 0.0938 |
| | SMOTE | 0.3234 | 0.3903 | 0.3947 | 0.3512 |
| | ADASYN | 0.3429 | 0.3747 | 0.4088 | 0.3747 |
| MLP [6] | N/A | 0.3344 | 0.2678 | 0.1943 | 0.1013 |
| | SMOTE | 0.3372 | 0.3616 | 0.3792 | 0.3439 |
| | ADASYN | 0.3516 | 0.3686 | 0.3804 | 0.3590 |
| CNN [7] | N/A | 0.2366 | 0.2275 | 0.0938 | 0.0823 |
| | SMOTE | 0.3253 | 0.3704 | 0.3837 | 0.3815 |
| | ADASYN | 0.3023 | 0.3863 | 0.3889 | 0.3840 |
| Proposed | N/A | 0.2123 | 0.1768 | 0.1396 | 0.1213 |
| | SMOTE | 0.5083 | 0.5180 | 0.5230 | 0.5131 |
| | ADASYN | 0.5167 | 0.5171 | **0.5275** | 0.5163 |

[†]https://open.chinaums.com/intro




(a) (b)

**Fig. 2** F1 scores with different PCA components. (a) SMOTE as the over-sampling technique; (b) ADASYN as the over-sampling technique.

from different models as a universal result. The last observation is that in most cases, ADASYN is a better choice than SMOTE, but the performance gap is minimal.

For a better illustration, F1 scores for evaluated models with different PCA components are shown in Fig. 2, when SMOTE and ADASYN are applied as the over-sampling technique respectively. In both cases, our proposed approach demonstrates a better performance than the baselines.

## 5. Conclusion

This letter proposes a novel loan default prediction framework with a muddling label regularization and an ensemble of feed-forward neural networks, which is proven effective with a real-world dataset. Two further research directions are considered. The first direction is the potential extension with other deep learning structures. The other direction is the extension from binary classification to multi-class classification, e.g., high/medium/low default risks.

## References

[1] L. Coenen, W. Verbeke, and T. Guns, "Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods," Journal of the Operational Research Society, vol.73, no.1, pp.191–206, 2022.

[2] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," Expert Systems with Applications, vol.184, 115537, 2021.

[3] W. Jiang and L. Zhang, "Edge-SiamNet and Edge-TripleNet: New deep learning models for handwritten numeral recognition," IEICE Trans. Inf. & Syst., vol.103, no.3, pp.720–723, March 2020.

[4] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," Information Fusion, vol.81, pp.84–90, 2022.

[5] K. Lounici, K. Meziani, and B. Riu, "Muddling label regularization: Deep learning for tabular datasets," arXiv preprint arXiv:2106.04462, 2021.

[6] H. He and Y. Fan, "A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction," Expert Systems with Applications, vol.176, 114899, 2021.

[7] M. Li, C. Yan, and W. Liu, "The network loan risk prediction model based on convolutional neural network and stacking fusion model," Applied Soft Computing, vol.113, part B, 107961, 2021.