IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 September 2023, accepted 3 October 2023, date of publication 6 October 2023, date of current version 16 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322429

APPLIED RESEARCH

Incorporating Feature Interactions and Contrastive Learning for Credit Prediction

LISI ZHANG[1], **QIANCHENG YU**[1,2], **BEIJING ZHOU**[1], **YIFAN ZHANG**[1], **AND ZHIYONG HU**[1]

[1]College of Computer Science and Engineering, North Minzu University, Ningxia 750002, China
[2]Laboratory of Graphics and Images of the State Ethnic Affairs Commission, Ningxia 750002, China

Corresponding author: Qiancheng Yu (1999019@nmu.edu.cn)

This work was supported in part by the Ningxia Key Research and Development Plan (Key Project) under Grant 2023BDE02001, in part by the 2022 Ningxia Autonomous Region Key Research and Development Plan (Talent Introduction Special) under Project 2022YCZX0013, in part by the 2022 University Research Platform ‘‘Digital Agriculture Empowering Ningxia Rural Revitalization Innovation Team’’ of North Minzu University under Grant 2022PT_S10, in part by the Major Key Project of School-Enterprise Joint Innovation in Yinchuan 2022 under Grant 2022XQZD009, and in part by the ‘‘Innovation Team for Image and Intelligent Information Processing’’ under the National Ethnic Affairs Commission.

ABSTRACT The efficacy of credit risk assessment models is pivotal to the risk management capacity of financial institutions. Traditional credit risk models often suffer from inadequate predictive accuracy due to overlooked feature combinations and weak supervisory signals. Addressing these limitations, we present a novel approach for credit default prediction that integrates feature interactions and contrastive learning. Specifically, we introduce second-order interactions atop standard linear models to achieve low-order feature interplay. Concurrently, the integration of deep neural networks and attention mechanisms facilitates the learning of concealed high-order features, thus enhancing the model's non-linear modeling capabilities and illuminating latent feature associations. Further, to ameliorate the issues of noise and diminished supervisory signals, we embed slight noise in feature embeddings for data augmentation and construct contrastive views, ultimately refining feature quality. To attest to the effectiveness of our approach, we conducted experiments on two real-world datasets, benchmarking against eight predictive methods including LR, XGBoost, and FiBiNET. The results unequivocally demonstrate the superior performance of our method across various metrics, underscoring its promise and excellence in the realm of credit risk assessment.

INDEX TERMS Credit appraisal, second-order crossover, attention mechanisms, deep neural networks, contrastive learning.

I. INTRODUCTION

Credit lending, with its distinctive and convenient financing approach, has emerged as one of the predominant channels for financing [1]. However, the intrinsic default risks associated with credit loans can significantly impede the growth of financial platforms. Establishing an effective credit loan prediction model is of paramount importance for both individual consumers and the evolutionary trajectory of financial platforms.

Credit loan prediction is a financial task that utilizes mathematical models and historical credit data to forecast whether borrowers are likely to fail in repaying their loans on time. Borrowers are typically categorized into two groups: those who default and those who do not. Currently, common credit loan prediction methods fall into two categories: Credit loan prediction techniques can be bifurcated into methods based on machine learning [2], [3], [4], [5], [6], [7] [8], [9] and those rooted in deep learning [11], [12], [13], [14]. While both categories have demonstrated commendable predictive outcomes, practical lending operations necessitate manual extraction of latent relationships between attributes, limiting the ability to generalize combination features not present in the original dataset [15].

In recent years, numerous researchers have delved into feature-generation techniques. For instance, the Factorization Machines series of models [16], [17], [18], [19], [20], by factorizing the relationships between features, captures the

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

interaction information among features, and can effectively improve the prediction accuracy of the model. However, these methods do not consider the issue of integrating both low-order and high-order feature importance at the same time. Furthermore, most of these interaction models are applied in domains such as ecommerce, and are seldom used for predicting credit defaults. Additionally, many deep learning prediction efforts, such as DeepFM [19], FiBiNet [20], etc., emphasize designing advanced frameworks to simulate complex feature interactions and mine hidden association rules. In these methods, the issues of noise in the data and weak supervised signals are overlooked, neglecting the importance of high-order feature interaction representation learning, leading to subpar model prediction accuracy.

Based on the above analysis, to address the issues in traditional credit assessment models, such as neglecting the importance of low-order and high-order interactions in feature importance, and poor prediction accuracy due to weak supervised signals, this paper proposes a credit default prediction method that integrates feature interaction and contrastive learning (FI-CL). Specifically, the initial embedding of raw features is performed via an Embedding layer, and pairwise crossing is employed to generate second-order feature combinations, facilitating low-order feature interaction. Concurrently, a Deep Neural Network (DNN) is applied to the features of the Embedding layer to decipher the profound latent relations associated with high-order feature interactions, thereby mining behavior patterns correlating user features with loan delinquency outcomes. Subsequently, an attention network is utilized to adaptively discern the significance of both low and high-order feature interactions. Ultimately, we incorporate Contrastive Learning (CL) [21], [22] to mitigate noise encapsulated in latent features, enhancing the quality of feature representation. This is further augmented by leveraging multi-task learning to bolster supervisory information.

The principal contributions of this study are multifaceted and can be summarized as follows:

- We have proposed a FI-CL credit default prediction method, which not only relies on the FI feature interaction technique to uncover deep hidden relationships among features, but also enhances the representation quality of the original features through the CL method.
- We introduced a new feature interaction approach (FI) that integrates second-order feature interaction, DNN, and the attention mechanism. This enables the model to adaptively learn the weights of both low-order and high-order combined features, uncover hidden associative rules among features, and enhance the model's nonlinear modeling capability.
- By integrating subtle noise into the original features, we have augmented their robustness. Furthermore, through contrastive learning, we enforced consistency across dual views, aiming to mitigate noise-related challenges and attenuate the issues arising from weak supervision.
- Empirical evaluations underscore the superiority of the FI-CL approach. When benchmarked against prominent existing methods, it manifests an improvement of 1-3% in AUC performance, 1-7% in KS, and 2-4% in G-mean values.

The rest of the paper is arranged as follows. Related works are summarized in section II. The proposed method is shown in section III. We introduce the experimental settings and analyze the results in sections IV and V. The conclusion is drawn in section VI.

II. RELATED WORK

From a technical perspective, this paper reviews the current mainstream methods for credit default prediction, which are methods based on machine learning and those based on deep learning. Given the inability of existing default prediction methods to generalize combination features not present in the original features, this study also outlines methods based on feature generation.

A. CREDIT DEFAULT PREDICTION METHOD BASED ON MACHINE LEARNING

Credit default prediction methods based on machine learning offer data-driven solutions by automatically learning and recognizing complex non-linear patterns from vast amounts of data, aiding in the accurate and timely assessment of borrowers' default risks. Traditional machine learning models are relatively simple and easy to implement. Techniques like Logistic Regression (LR) [2], Support Vector Machine (SVM) [3], and Decision Tree (DT) [4] using individual classifiers have been proven effective in credit-related classification prediction tasks.

However, singular classification models possess issues like lower prediction accuracy and poor robustness. Therefore, scholars have adopted methods such as ensemble learning [8] and multi-model fusion [9]. By leveraging techniques like voting, weighted averaging, and stacking, they blend results from various models to achieve enhanced predictive performance. Yet, these methods require multiple different classifiers, leading to an increase in model complexity and computational costs.

B. CREDIT DEFAULT PREDICTION METHOD BASED ON DEEP LEARNING

With the gradual rise of deep learning technology, deep learning has been widely applied in various fields [10]. Credit loan default prediction tasks are one typical application. Compared to default prediction models based on machine learning, credit default prediction methods based on deep learning can capture complex data relationships, providing a deep model framework for high-precision default risk assessment, and achieving significant breakthroughs in prediction accuracy and speed.

Brenes et al. [11] investigated the discriminative capability of the Multi-Layer Perceptron (MLP) in the context of bankruptcy prediction. After parameter optimization, the proposed MLP exhibited outstanding performance. Guo [12] analyzed traditional financial risks and information technology risks on P2P lending platforms and introduced a loan risk assessment algorithm based on the BP neural network. Experiments indicated that this method could effectively reduce investor risks. Nguyen et al. [13] utilized auto-encoders (AE) and Long Short-Term Memory networks (LSTM) to construct models, addressing issues in traditional fraud detection methods, such as the need for manual rule definition and the difficulty in handling complex fraudulent behaviors. Experiments showed that deep learning methods performed excellently. He [14] introduced the Bidirectional Recurrent Neural Network (BRNN) for the problem of customer credit default prediction and integrated the LR, Extreme Gradient Boosting (XGB), and BRNN methods. Predictive results were obtained by assigning weights. Although deep learning models have achieved satisfactory results in the credit field, actual loan operations still require manual extraction of implicit relationships between attributes, and cannot generalize interaction terms that are not present in the original features.

C. PREDICTION METHOD BASED ON FEATURE GENERATION

Prediction methods based on feature generation innovatively construct or transform original data features, extracting more valuable information, thereby enhancing the model's expressive power and prediction accuracy. Rendle [16] introduced the Factorization Machine (FM) model, which learns the relationships between features through second-order interactions. These interactions can generalize to unobserved feature combinations, thereby eliminating the need for manual feature relationship discovery. Juan et al. [17] proposed the Field-aware Factorization Machine (FFM) model, which incorporates domain information of features. Compared to FM, FFM has a more potent capability in modeling feature interactions but introduces additional computational complexity and parameter size, resulting in a higher time complexity during model training. Xiao et al. [18] proposed the Attentional Factorization Machine (AFM) model, which incorporates attention mechanisms into the FM model to dynamically adjust the importance of feature interactions, enhancing the modeling capability for key feature interactions. However, like FM, AFM only considers second-order feature interactions and neglects deeper high-order hidden feature relationships.

Subsequently, models emerged that leverage DNN to learn complex feature interactions. For instance, Guo et al. [19] combined the strengths of FM and DNN. They integrated DNN to learn high-order feature interactions while retaining the FM model's second-order interaction capabilities. With the nonlinear fitting capability of DNN, the model extracts more intricate feature interaction patterns, enhancing the model's expressiveness and prediction accuracy. Huang et al. [20], building on the FM model, introduced the FiBiNET model. It utilizes the SENET mechanism to adaptively learn the importance of features and effectively learns feature interactions through a bilinear function. However, due to learning at a finer granularity of features, the model's time complexity significantly increases.

III. METHODOLOGY

The core processing modules of the proposed method (FI-CL) include the Feature Interaction module (FI) and the Contrastive Learning module (CL), as illustrated in Figure 1. Specifically, the FI module can adaptively mine valid association rules present in both low-order and high-order feature interactions. The CL module learns the consistency between feature views, thereby enhancing feature quality. More precisely, firstly, raw features are passed through an embedding layer, mapping them to a lower-dimensional space, yielding their corresponding dense Embedding vectors, facilitating computation and learning by the model. Subsequently, the features from this Embedding serve as input for both the FI and CL modules, thoroughly extracting interactive information from the features and enhancing their quality. Lastly, the model is optimized using a multi-task joint cross-entropy loss function and a contrastive loss function.

A. FI MODULE

The FI module consists of a second-order cross layer, a Deep layer, and an attention network layer, as shown in Figure 1(a). The second-order cross layer pairs up original features for combination to capture low-order feature information. The Deep layer learns high-order feature information between different features using a DNN. The attention network layer adaptively combines the importance of original features, low-order features, and high-order features, enhancing the model's nonlinear modeling capability and subsequently extracting deeper implicit information.

1) SECOND-ORDER CROSS LAYER

The second-order cross is used to fit interactions between features by implementing pairwise combinations of the original features, obtaining low-order cross terms to describe the associations between features. In this paper, polynomial combinations are utilized to learn first-order features and second-order combined features, capturing the interactions between them. That is, on top of the general linear model, we consider the relationships between features and learn combined features not present in the original set through the form of inner products of latent vectors. This approach boasts strong generalization capabilities, as shown in the following formula:

$$E_{2CN} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j + \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

In the above, n denotes the number of features in the sample, x_i represents the i -th feature, b and w_i respectively represent the bias term and the weight, and $< v_i, v_j >$ denotes the weight

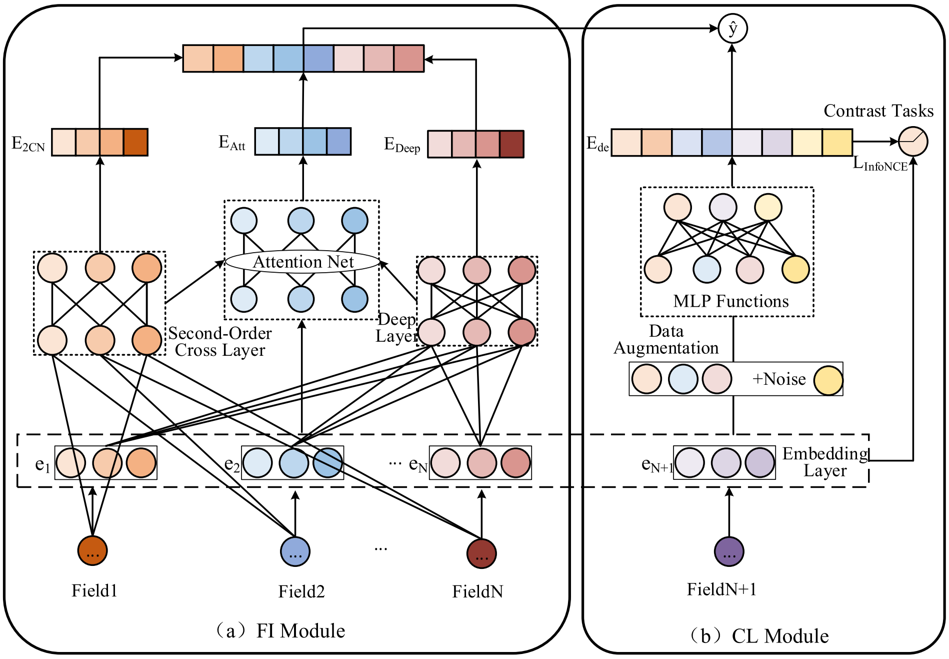

FIGURE 1. **The overall framework of FI-CL. FI-CL consists of (a) the FI module and (b) the CL model. The FI module uses a second-order interaction layer to learn low-order feature interactions and a Deep layer to learn high-order feature interactions. By incorporating attention mechanisms, it achieves adaptive learning of both low-order and high-order feature interactions. The CL model enhances representation quality through data augmentation.**

of the second-order cross feature, which is used to learn combined features, enhancing the linear modeling capability of the model.

To reduce the computational load of $< v_i, v_j >$ and decrease time complexity, we leverage the properties of diagonal matrices to simplify the second-order cross term in formula (1), leading to the following expression:

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} < v_i, v_j > x_i x_j$$
$$= \frac{1}{2}^{*} \sum_{f=1}^{k} \left\{ \left(\sum_{i=1}^{n} v_{if} x_i \right)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2 \right\} \quad (2)$$

2) DEEP LAYER

The second-order cross layer adeptly captures both primary (first-order) and secondary (second-order) feature information. However, they fall short when it comes to discerning higher-order features, such as ternary (third-order) interactions. Inspired by the paradigm presented in the Wide & Deep model [23], we deftly amalgamate linear modeling with deep learning approaches. By incorporating a DNN into the deep layer, we achieve an end-to-end learning mechanism, enabling the model to handle both low-order and high-order feature combinations simultaneously. As shown in Figure 1(a), the output of the embedding layer serves as the input to the Deep layer, that is:

$$a^{(0)} = E = [e_1, e_2, \ldots, e_i, \ldots, e_N] \quad (3)$$

where N denotes the number of original discrete feature domains, $e_i \epsilon R^k$ represents the embedding of the i-th feature, and k is the embedding dimension. The process of inputting $a^{(0)}$ into the Deep layer is as follows:

$$a^{(l+1)} = \sigma \left(W^{(l)} a^{(l)} + b^{(l)} \right) \quad (4)$$

where σ denotes the Relu activation function, l is the depth of the layer, $W^{(l)}$ represents the weights of the l-th layer, $a^{(l)}$ is the output of the l-th layer, and $b^{(l)}$ is the bias term of the l-th layer. Lastly, the outputs from each layer are fed into the Sigmoid function to obtain the output of the Deep layer as:

$$E_{Deep} = Sigmoid(W^{h+1} a^h + b^{h+1}) \quad (5)$$

where h represents the number of hidden layers in the Deep layer.

3) ATTENTION NETWORK LAYER

Both the second-order interaction layer and the Deep layer engage in feature interactions by utilizing fixed vector values in conjunction with other features. However, it's paramount to note that in real-world scenarios, the contribution of diverse features to the final predictive outcome can vary significantly. Neglecting to distinguish these features during training might potentially compromise the model's predictive capabilities and training efficiency. Consequently, we've incorporated an attention mechanism, enabling the model to adaptively learn the weights of various feature interactions and their combinations. This is instrumental in augmenting the model's predictive performance and interpretability. The attention network function introduces weights between features during the feature crossing process. The calculation process is as follows:

$$E_{Att} = W^T \sum_{i=1}^{n} \sum_{j=i+1}^{n} \alpha_{ij} \left(v_i \odot v_j\right) x_i x_j + \sum_{i=1}^{n} w_i x_i + b \tag{6}$$

where W represents the weight matrix, $\odot$ represents the element-wise product, which multiplies each element of two vectors of the same dimensionality, and α_{ij} is the attention score, which serves as the weight coefficient of the combination feature and represents the contribution of different combination features to the final prediction. α_{ij} is obtained by normalizing α'_{ij} through the softmax function, as shown in the following equation:

$$\alpha_{ij} = \frac{\exp\left(\alpha'{}_{ij}\right)}{\sum_{(i,j)\in R_X} \exp\left(\alpha'_{ij}\right)} \tag{7}$$

Here, α'_{ij} is obtained by adding the output of the attention network's hidden layer and the weight vector, with the ReLU activation function applied, as shown below:

$$\alpha'_{ij} = h^T ReLU(W(v_i \odot v_j)x_i x_j + b) \tag{8}$$

B. CL MODULE

Contrastive learning operates as an unsupervised learning paradigm, circumventing the reliance on labeled data intrinsic to supervised learning, and emphasizing intrinsic data features. To address issues of poor feature representation quality, we employ the CL approach, as illustrated in Figure 1(b). The CL module encompasses data augmentation and a contrastive loss function. Inspired by the XSimGCL methodology [22], we introduce subtle noise to the features of the Embedding layer as an initial interference to the original characteristics, followed by the application of an MLP for enhanced feature representation. Subsequently, a comparison between the augmented features and those of the Embedding layer is executed, aiming to mitigate feature noise and alleviate the shortcomings of weak supervision, thereby enhancing the overall feature representation quality.

1) DATA AUGMENTATION

CL enhances the model performance by implementing biased data augmentation. In this paper, we introduced noise that follows a uniform distribution into the features of the Embedding layer, leading to subtle feature interference. The features are further enhanced using an MLP. Given a feature x_i and its representation e_{xi} in a d-dimensional embedding space, with $e_{xi} \epsilon R^d$, the augmented data representation for e_{xi} is denoted as e'_{xi}, as illustrated in the following equation:

$$e'_{xi} = e_{xi} + n_{xi} \tag{9}$$

Herein, $n_{xi} = X \odot sign(e_{xi})$ represents the noise added stochastically, where $||n_{xi}|| = \varepsilon$ is a relatively small constant (typically, $\varepsilon < 0.3$), and X is a function of the normal distribution. The e'_{xi} constructed using n_{xi} retains most of the original features while introducing some variations.

Compared to the traditional CL approach, which utilizes dropout techniques in embeddings and consequently sacrifices some features, the data-augmented embedding introduces noise that maintains feature uniformity, without any feature loss. However, the features from data augmentation are of higher dimensionality and sparsity. For models with a large volume of data, this might adversely impact training stability. As such, multi-layer MLP projection functions are employed to reduce the feature dimensions, as elaborated below:

$$F_{zi} = \sigma\left(wz_i + b\right) \tag{10}$$

wherein, Ede represents the augmented feature, and e'_{xi} is the representation for the i -th data augmentation.

2) CONTRASTIVE LOSS

Contrastive loss functions are used to minimize the distance between samples of the same class, making similar samples more clustered in the feature space and pushing dissimilar samples farther apart [24]. This allows the model to adaptively adjust the uniformity of learning representations, enhancing generalization performance. In this paper, we utilize the InfoNCE loss function [25] to minimize the distance between the original feature embeddings in the Embedding layer:

$$L_{\text{infonce}} = -\log \frac{\exp\left(Z'_{iT} \cdot Z_i/\tau\right)}{\sum\limits_{i\in B} \exp\left(Z'_{iT} \cdot Z_j/\tau\right)} \tag{11}$$

where Z'_{iT} represents the feature embedding after data augmentation, Z_i denotes the original feature embedding, $Z'_{iT} \cdot Z_i$ signifies the similarity between positive samples, $Z'_{iT} \cdot Z_j$ represents the similarity between a positive sample and a negative sample, τ is the temperature that controls the model's ability to differentiate negative samples, and $B \epsilon (0, k)$ is the batch size.

C. MODEL PREDICTION AND MULTI-TASK TRAINING

To effectively carry out credit prediction, we concatenate the second-order cross layer, Deep layer, the attention network layer of the FI module, and the data-enhanced output of the CL module, and use the activation function sigmoid to obtain the prediction result:

$$\hat{y} = sigmoid\left(E_{2CN}||E_{Att}||E_{Deep}||E_{de}\right) \tag{12}$$

Herein, the symbol || denotes the concatenation procedure. Furthermore, our methodology incorporates the cross-entropy loss function [25] to determine the associated loss, as articulated in the subsequent equation:

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i * logp\left(y_i\right) + \left(1-y_i\right)^* log\left(1-p\left(y_i\right)\right)\right] \tag{13}$$

Here, y_i denotes the true value, and $p(y_i)$ represents the predicted value.

To integrate the contrastive loss function into the prediction task and optimize the model, we adopted a multi-task training approach, amalgamating both the cross-entropy loss function and the contrastive loss function. The comprehensive loss function is formalized as follows:

$$L = L_{\text{BCE}} + \lambda L_{InfoNCE} \tag{14}$$

where λ is the hyperparameter that controls the strength of the contrastive loss.

IV. EXPERIMENTAL DESIGN

To validate the performance of the FI-CL model, in this section, we first introduce the sources of the datasets utilized in this study along with basic data preprocessing steps. Following this, we outline the experimental environment and model parameter settings, as well as the evaluation metrics. Lastly, we present the baselines used for performance assessment.

A. DATA SOURCES AND PROCESSING

To gauge the predictive efficacy of the FI-CL method, we conducted experiments on two authentic datasets. The first dataset, denoted as Dataset A[1], stems from a loan default prediction dataset released on the Tianchi platform. This dataset encompasses 800,000 records, each containing 47 primal features. The second dataset, referred to as Dataset B[2], constitutes data from the Lending Club platform pertaining to the borrower's data for the first quarter of 2018. It comprises 107,866 individual samples, each endowed with 145 original features. A comprehensive overview of the datasets is delineated in Table 1.

TABLE 1. Descriptive analyses of the two utilized datasets in this paper.

Dataset Description	Dataset A	Dataset B
Sample size	800000	107866
Number of defaults	640390	105804
Number of non-defaults	159610	2060
Imbalance rate	4.01	50.36
Number of numerical features	42	114
Number of category features	5	31
Number of total features	47	145
Label	isDefault	loan_status

Prior to model instantiation, an intricate analysis of the raw data was undertaken to refine it to a standardized high-quality format. The data preprocessing adopted in this study was multistep in nature, encompassing a myriad of techniques. Notably, these included feature binning (utilizing quantile-based binning), feature engineering (resulting in the derivation of 19 business features), and feature selection (employing a combined methodology of Information Value (IV), variance analysis, Pearson correlation coefficient, and Gini coefficient importance assessment from Random Forests. As a result, the refined Dataset A for training comprised 32 features, whereas Dataset B encompassed 59 features). Numerical encoding and Synthetic Minority Over-sampling Technique (SMOTE) [26] were further applied.

B. EXPERIMENTAL ENVIRONMENT

The experimental infrastructure utilized in this study consisted of an Intel(R) Xeon(R) Gold 6154 processor, NVIDIA TITAN V 24G graphics card, and 128G of memory. The software environment was structured around the CentOS 7 operating system, the deep learning framework PyTorch, and Python 3.7 as the programming language.

For this study, the partitioning of the dataset was configured with an 80% training and 20% testing split. Model optimization was entrusted to the Adam optimizer [27], with a batch size stipulated at 32 and the iteration epoch set at 100. Comprehensive details of other salient parameter configurations are presented in Table 2.

TABLE 2. Parameter settings in FI-CL.

Parameter	Dataset A	Dataset B
embed-dim	8	16
dnn-dropout	0.5	0.5
att-vector	4	8
learning-rate	0.001	0.001
Noise parameter (ε)	0.15	0.1
Temperature (τ) in InfoNCE loss function	0.2	0.4
Contrastive Loss Intensity (λ)	0.01	0.01

C. EVALUATION INDICATORS

In this study, the predictive efficacy of the model was ascertained via four evaluation metrics: AUC (Area Under the ROC Curve), ACC (Accuracy), KS (Kolmogorov-Smirnov) value, and G-mean. All these metrics are derived from the confusion matrix.

The AUC is among the most frequently employed performance evaluation metrics for classification tasks. The ROC curve is plotted with the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis. The formulations for FPR and TPR are delineated as follows:

$$FPR = \frac{FP}{TN + FP} \tag{15}$$

$$TPR = Sensitivity = \text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

[1] https://tianchi.aliyun.com/competition/entrance/531830/information
[2] https://www.lendingclub.com/statistics/additional-statistics

Within this context, TP stands for True Positives, TN for True Negatives, FP for False Positives, and FN for False Negatives. A larger AUC is indicative of the superior predictive capability of the model. Notably, an AUC value below 0.5 signifies a performance inferior to random classification, rendering the model devoid of practical merit.

The ACC delineates the proportion of correctly classified samples to the total number of samples. It remains a commonly utilized metric in classification tasks. However, it's crucial to note that, for datasets characterized by a pronounced class imbalance, the accuracy might not reflect a genuine representation of the model's efficacy. In such scenarios, accuracy can be misleadingly high. The formulation for ACC is presented as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

The KS statistic serves as a valuable metric for assessing model discrimination and is frequently deployed in the realm of financial risk control. Analogous to the ROC curve, the KS curve plots both TPR and FPR on the y-axis, with the selected threshold forming the x-axis. The formulation for the KS statistic is articulated as follows:

$$KS = max\,(TPR - FPR) \tag{18}$$

A larger KS value signifies a heightened discriminatory capacity of the predictive model. Notably, when the KS value exceeds 0.4, it is indicative of the model possessing robust discriminative prowess.

The G-Mean stands as an aggregate evaluation method constructed from sensitivity and specificity. It is commonly employed as an analytical metric for classification problems dealing with imbalanced datasets. The G-Mean is computed as given in Equation (19), with sensitivity and specificity defined respectively in Equations (16) and (20).

$$G - mean = \sqrt{Sensitivity{*}Specificity} \tag{19}$$

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{20}$$

D. BASELINES

For performance assessment, FI-CL was juxtaposed with a spectrum of baseline methodologies: standalone machine learning approaches (LR [2], SVM [3], DT [4]), ensemble algorithms (RF [5], GBDT [6], XGB [7]), and advanced deep learning models (MLP [11] and FiBiNET [19]). This comprehensive comparison serves to underscore the robustness and efficacy of FI-CL.

LR [2]: LR stands as one of the most ubiquitously employed linear models in industrial applications. Serving as a supervised learning algorithm, it predicts continuous variables by fitting an optimal linear relation. Though facile in its comprehension and operationalization, LR's inherent limitation lies in its inability to capture intricate nonlinear relationships.

SVM [3]: SVM prioritizes maximizing the classification margin by identifying the optimal boundary (specifically, the maximal margin hyperplane) between two classes. SVM is adept at managing nonlinearities and high-dimensional spaces, yet its sensitivity to large datasets and parameter tuning is evident.

DT [4]: DT derives target values via a succession of feature-based queries. DT conveniently elucidates feature relationships and decision pathways. However, its propensity for overfitting and sensitivity to noise remain drawbacks.

RF [5]: Random Forests (RF) typify an ensemble learning methodology, comprising multiple decision trees. Aggregating predictions from individual trees—either by voting or averaging—RF enhances predictive precision and robustness. Notably, its overfitting tendency surfaces in instances with significant noise.

GBDT [6]: Gradient Boosted Decision Trees (GBDT) represent an ensemble approach where decision trees are incrementally added and adjusted to minimize a given loss function, optimizing model performance. Their adeptness with imbalanced datasets is noteworthy. Yet, they demand extensive training durations, a potential overfitting issue, and meticulous parameter tuning.

XGB [7]: XGB embodies an optimized gradient boosting algorithm, architected for amplified speed and performance. Its inclusion of regularization terms mitigates overfitting, rendering the model more robust. Nevertheless, compared to other ensemble techniques, it requires a more intricate tuning process.

MLP [11]: MLP signifies a feed-forward artificial neural network, consisting of three or more layers: an input layer, one or several hidden layers, and an output layer. It's adept at discerning intricate nonlinear relationships, though its training might be time-intensive.

FiBiNET [19]: FiBiNET integrates and blends factorized networks, merging linear regression and factor decomposition models at the feature interaction level. This versatile model adeptly captures intricate inter-feature interactions, albeit at the cost of increased computational intricacy.

V. EXPERIMENTAL FINDINGS AND DISCUSSION

In this section, we embarked on a comprehensive series of experiments to address the following research questions:

RQ1: How does the performance of our model compare to existing classic methods for credit default prediction?

RQ2: Which component is most pivotal in the FI-CL method?

RQ3: How do various parameter configurations within the FI-CL method influence the predictive performance of the model?

RQ4: How to validate the robustness of the FI-CL method?

A. COMPARISON OF EXPERIMENTAL RESULTS (RQ1)

In Table 3, we delineate the evaluation outcomes for all baseline methods across two datasets, employing metrics such as AUC, ACC, KS, and G-mean.

By comparing the performance of Dataset A and Dataset B in Table 3, we noticed that Dataset B generally

TABLE 3. Overall performance comparison. The best results among all methods are highlighted in bold, with the optimal values from the comparative models underlined. The column is denoted as %Improv. illustrates the percentage improvement of our proposed AFCC method over the best-performing benchmark model Higher values of AUC, ACC, KS, and G-mean denote superior predictive performance of the model.

Model	Data1				Data2			
	AUC	ACC	KS	G-mean	AUC	ACC	KS	G-mean
LR	0.6263	0.8025	0.3901	0.5742	0.5122	0.9813	0.3343	0.5058
SVM	0.7019	0.8030	0.3923	0.6901	0.6891	0.9773	0.3108	0.2260
DT	0.7055	0.8036	0.3969	0.4754	0.7808	0.9809	0.3972	0.6446
RF	0.7117	0.8052	0.4055	0.6637	0.9050	0.9824	0.6743	0.8911
GBDT	0.7305	0.8069	0.4326	0.6854	0.9354	0.9923	0.7228	0.8697
XGB	0.7302	0.8080	0.4345	0.6824	0.9415	**0.9942**	0.7975	0.8820
MLP	0.7272	0.8081	0.3297	0.5374	0.9323	0.9908	0.7279	0.7633
FiBiNet	0.7271	0.8072	0.5188	0.6701	0.8604	0.9863	0.6989	0.6462
FI-CL	**0.7526**	**0.8126**	**0.5308**	**0.7246**	**0.9533**	0.9931	**0.8672**	**0.9208**
%Improv.	2.21	0.45	1.20	3.45	1.18	-0.11	6.97	2.97

demonstrates higher predictive performance. This difference can be attributed to the divergences in the collection scope and environment of the two datasets. Firstly, we observed that Dataset B comprises a larger number of original features, totaling 145 compared to the 47 found in Dataset A. This not only avails a more comprehensive set of information for guiding feature selection but also enables a finer-grained analysis, potentially resulting in the inclusion of more useful features in the final model. Secondly, the BANI (Brittle, Anxious, Nonlinear, and Incomprehensible) environmental factors might have affected the data collection and feature selection process. In this scenario, the more comprehensive set of features in Dataset B may allow it to be more adaptive to the influences of the BANI environment, thereby improving its predictive performance.

Drawing insights from Table 3, it becomes manifest that our proposed method outperforms linear base models such as LR and SVM. Specifically, on Dataset A, we observed AUC enhancements of 12.64% and 5.07% respectively when compared with LR and SVM. In contrast, for Dataset B, these augmentations were even more pronounced, registering at 44.11% and 26.32% respectively. Such findings underscore the imperative nature of bolstering a model's capacity for nonlinear modeling. This is particularly salient for datasets with marked imbalances, akin to Dataset B, where reliance solely on linear classifiers results in suboptimal predictive outcomes.

Relative to the decision tree-based methods—namely DT, RF, GBDT, and XGB—our approach exhibits distinct advantages. When juxtaposed with the solitary decision tree method, DT, we register AUC elevations of 4.71% and 17.25% across the two datasets. In contrast, against the ensemble learning techniques, which utilize multiple decision trees—RF, GBDT, and XGB—our method manifests AUC enhancements ranging from 2.21% to 4.09% and from 1.18% to 4.83% on the datasets, respectively. These observations elucidate the efficacy of our methodology in discerning latent feature combinations, adaptively learning both low and high-order weights, and thereby refining representation quality.

When benchmarked against the deep learning models, specifically MLP and FiBiNet, our approach evidences a palpable enhancement in AUC by margins of 2.54%-2.55% and 2.1%-9.29%, respectively. Such results affirm the contention that the integration of high-order feature weight learning coupled with contrastive learning indeed augments the predictive prowess of the model.

In summation, the empirical outcomes of our proposed approach consistently outshine those of the referenced classical methodologies. When set against the most efficacious among these eight canonical methods, FI-CL demonstrates improvements on Dataset A in AUC, ACC, KS, and G-mean by 2.21%, 0.45%, 1.2%, and 3.45% respectively, thereby underscoring the merits of factoring in feature combinations and data augmentation to bolster model predictive capacity. For Dataset B, there's an uptick in AUC, KS, and G-mean by 1.18%, 6.97%, and 2.97% respectively. However, it is worth noting the underwhelming ACC metric performance in this dataset. This diminished performance can be attributed to the pronounced data imbalance inherent in Dataset B. The data sampling employed a rudimentary SMOTE oversampling technique without crafting a sampling strategy tailored to the dataset's unique features, culminating in a vacillating ACC metric performance.

B. ABLATION EXPERIMENTS (RQ2)

To rigorously gauge the salient contributions of the attention network, DNN, and CL, we executed a series of ablation studies. Specifically, we delineated six distinct configurations:

- w/o Att-dnn-cl: A scenario wherein the attention network, DNN, and CL are concurrently abrogated, relegating the model to rely solely on the second-order feature interactions for predictions.
- w/o Att-cl: The model is divested of the attention mechanism and CL, leaving it to discern both low and high-order combinatory scores.
- w/o dnn-cl: With the DNN and CL omitted, the model introduces the attention mechanism to the second-order interactions.
- w/o cl: Eliminating only the CL component, the model incorporates both the attention network and DNN over the second-order interactions.

TABLE 4. Ablation analysis. The best results are shown in bold.

Model	Data1				Data2			
	AUC	ACC	KS	G-mean	AUC	ACC	KS	G-mean
w/o Att-dnn-cl	0.7120	0.8008	0.3120	0.6945	0.8257	0.9815	0.5136	0.7788
w/o Att-cl	0.7124	0.8014	0.3201	0.3048	0.8534	0.9833	0.5896	0.7501
w/o dnn-cl	0.7297	0.8072	0.5188	0.6244	0.8839	0.9796	0.7358	0.7478
w/o cl	0.7436	0.8076	0.5226	0.5953	0.9354	0.9849	0.8474	0.9077
w/o den-cl	0.7296	0.8072	0.5223	0.6862	0.9281	0.9835	0.8601	0.8826
w/o spar-cl	0.7496	0.8077	0.5241	0.6635	0.9398	0.9846	0.8623	0.8999
FI-CL	**0.7526**	**0.8126**	**0.5308**	**0.7246**	**0.9533**	**0.9931**	**0.8672**	**0.9208**

- w/o den-cl: In this variant, we forsake the dense vector treatment, compelling the holistic model to utilize the low-order linear interactions for CL.
- w/o spar-cl: By eschewing the sparse vector processing, the model in its entirety hinges on high-order nonlinear interactions for CL.

The detailed outcomes of these ablation experiments are presented in Table 4.

Insights gleaned from Table 4 underscore a conspicuous performance deterioration when CL is expunged. This is attributable to the presence of model noise, attenuated supervisory signals, and sub-optimal feature representation quality. Further, the omission of either the DNN or attention mechanism culminates in a decline in model efficacy. This accentuates the prowess of DNN in discerning deep-rooted high-order feature interplay, while the attention mechanism adeptly mines salient low and high-order feature weights. Abrogating either the sparse or dense vector segments precipitates performance reduction, suggesting that a comprehensive consideration of all feature vectors in CL outperforms partial vector contemplation. Moreover, leveraging high-order nonlinear interactions for CL holds a marked advantage over the employment of low-order linear interactions. In summation, each facet of our proposed methodology is paramount. Stripping any singular component invariably results in diminished performance.

C. HYPERPARAMETRIC ANALYSIS (RQ3)

To enhance the reliability of the results obtained from the FI-CL method, this section thoroughly analyzes the hyperparameters involved, including: (1) the number of hidden neurons in the attention network denoted by att-layer, (2) the dropout parameter within the DNN, (3) the noise addition parameter ε in the CL model, (4) the temperature hyperparameter τ within the InfoNCE loss function, and (5) the loss weight λ in multitask training. Experimental outcomes are employed to individually demonstrate the influence of these parameters on the model, with detailed analysis for each parameter as follows.

1) THE IMPACT OF THE NUMBER OF HIDDEN LAYER NEURONS (ATT-LAYER) IN THE ATTENTION NETWORK ON THE MODEL

In attention networks, fewer neurons in the hidden layer might limit the model's expressive capacity, preventing it from capturing complex features and relationships. Conversely, having too many neurons can increase the model's capacity but might also lead to overfitting. An appropriate number of neurons in the hidden layer can help the model learn richer feature representations during training, enhancing its fitting and generalization capabilities. To investigate the impact of att-layer on the model, we adjusted the value of att-layer from {2, 4, 8, 16, 32}. For a fair comparison, other settings were kept constant. Figure 2 displays the effects of different att-layer values on the AUC of the FI-CL method on Dataset A and Dataset B.

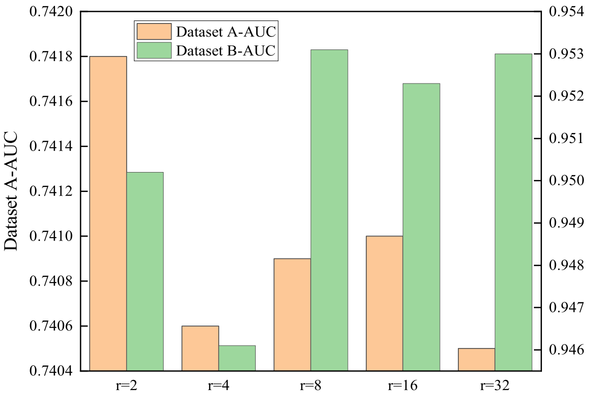

FIGURE 2. Effect of att-layer in attention network on FI-CL.

As shown in Figure 2, for Dataset A, the AUC performance of FI-CL is optimal when the att-layer in the attention network is set to 4. The reason behind this is that the training features for this dataset are limited, making the attention network's neuron parameters less sensitive to changes. A smaller att-layer value is already sufficient. For Dataset B, the best AUC performance of FI-CL occurs when the att-layer is set to 8. This is because this dataset has more training features. Although the dataset is smaller in size, the interaction between features is more complex. Increasing the dimensionality of the attention vector can better capture the intricate relationships between features, leading to improved performance. However, continuously increasing this value would result in increased model complexity and computational resource demands.

2) THE INFLUENCE OF DROPOUT PARAMETERS IN DNN ON THE MODEL

Using dropout in DNN can effectively enhance the model's generalization capability, robustness, and learning capacity.

TABLE 5. Effect of noise parameter ε on the performance of FI-CL method. The best results are shown in bold.

ε	Dataset A				Dataset B			
	AUC	ACC	KS	G-mean	AUC	ACC	KS	G-mean
0	0.7524	0.8122	0.5297	0.7240	0.9511	0.9917	0.8454	0.8998
0.05	0.7524	0.8123	0.5301	0.7241	0.9462	0.9930	0.8612	0.8994
0.1	0.7519	0.8123	0.5305	0.7243	**0.9533**	**0.9931**	**0.8672**	**0.9208**
0.15	**0.7526**	**0.8126**	**0.5308**	**0.7246**	0.9507	0.9928	0.8569	0.9026
0.2	0.7525	0.8125	0.5296	0.7240	0.9487	0.9929	0.8463	0.8964

By discarding a portion of the hidden nodes from the network, it can reduce the complexity of the neural network, effectively suppressing the overfitting phenomenon, and alleviating the impact of gradient vanishing, thereby improving the performance of the model. To investigate the influence of the dropout parameter on the model, we adjusted its value from {0.01, 0.5, 1}. For a fair comparison, all other settings were kept constant. Figure 3 displays the effects of different dropout values on the performance of the FI-CL method on Dataset A and Dataset B.

From the results in Figure 3, it can be observed that the model performs best on both datasets when the value of dropout is set at 0.5. With a dropout value of 0.01, it implies that each neuron has a very low probability (1%) of being discarded. As a result, most neurons are retained, leading the model's learning to be non-specific and causing poor performance. When the dropout value is set to 1, it signifies that all neurons are discarded during training, rendering the network incapable of learning any effective features. This lack of learning ability consequently results in extremely poor model performance.

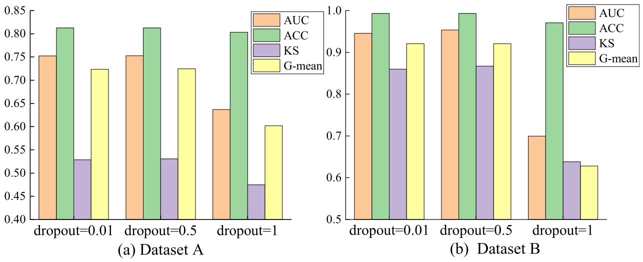

FIGURE 3. The influence of dropout parameters on the model in DNN.

3) THE EFFECT OF ADDING THE NOISE PARAMETER ε IN CL TO THE MODEL

To investigate the optimal value of randomly adding noise in CL, ε was adjusted from the set {0, 0.05, 0.1, 0.15, 0.2}. For a fair comparison, all other configurations were kept constant. Table 5 displays the impact of ε on the performance of the FI-CL method on Dataset A and Dataset B.

In Table 5, the model achieves optimal performance on Dataset A and Dataset B when selecting 0.15 and 0.1, respectively. Upon examination, when ε is set to 0, meaning no noise is added, there is a noticeable decline in performance. The model's performance also deteriorates when the noise is too minimal ($\varepsilon = 0.05$) or excessively high ($\varepsilon = 0.2$). This confirms that introducing an appropriate amount of noise can even out the data distribution, which is conducive to enhancing model performance.

4) INFLUENCE OF HYPERPARAMETER TEMPERATURE τ ON THE MODEL IN INFONCE LOSS FUNCTION

To further investigate the hyperparameter temperature τ in the InfoNCE loss function (as in Equation 11) of this chapter, we adjusted τ from the set {0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4}. For a fair comparison, all other configurations were kept constant. Figure 4 illustrates the effects of τ on the AUC of the FI-CL method on Dataset A and Dataset B.

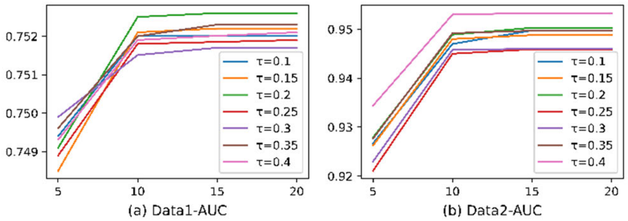

FIGURE 4. The effect of hyperparameter temperature τ on FI-CL.

In the InfoNCE loss function, the value of the temperature parameter τ influences the training process of the model by adjusting the output distribution of the activation function. Figure 4(a) indicates that in Dataset A, the model performs best when the temperature is set to 0.2. This suggests that for Dataset A, a temperature parameter of 0.2 can sharpen the distribution of the activation function's output, making it easier for the model to differentiate between similar sample pairs and dissimilar ones. Figure 4(b) shows that for Dataset B, the overall model performance peaks when the temperature is 0.4. This implies that during the oversampling process for Dataset B, a substantial imbalance ratio introduced a considerable amount of noise. A temperature parameter of 0.4 can mitigate the influence of noise and uncertainties on the model, enhancing its robustness.

5) EFFECT OF LOSS WEIGHT ON THE MODEL IN MULTI-TASK TRAINING

In a bid to fathom the repercussions of the loss of weight in multi-task training (denoted as λ in Equation 12) on the FI-CL modus operandi, we varied λ across a spectrum of values: {1, 0.1, 0.01, 0.001, 0.0001, 0}. Ensuring rigorous experimental consistency, all auxiliary parameters were anchored. Figure 5 provides a nuanced depiction of how these

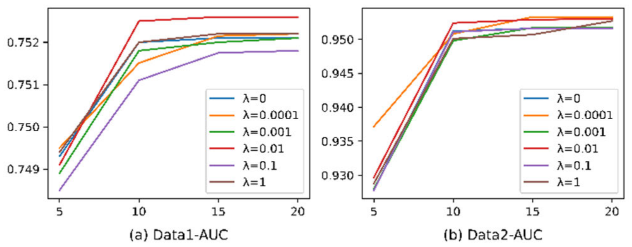

FIGURE 5. The effect of weight parameter λ on FI-CL.

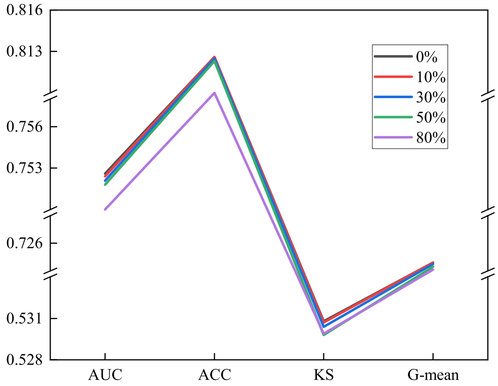

FIGURE 6. The effect of hyperparameter temperature τ on FI-CL.

λ variations influence the AUC of the FI-CL method across the dichotomy of Dataset A and Dataset B.

During multi-task training endeavors, striking a harmonious balance between disparate task outcomes is pivotal to obviate the overshadowing of the FI module's efficacy by the CL module's performance. As elucidated in Figure 5, an optimum performance nexus is achieved across both datasets when λ is configured at 0.01. This observation accentuates the notion that in the FI-CL framework, an excessively dominant or diminutive weight assigned to the contrastive module can deleteriously impinge on the model's over-arching predictive prowess.

D. PLACEBO TEST EXPERIMENT (RQ4)

To further understand the robustness and performance stability of the FI-CL model when faced with false data, we conducted a placebo test experiment using the test set of Dataset A. Specifically, we generated 10%, 30%, 50%, and 80% false test data for three features in the test dataset: "annualIncome"(Annual income), "dti" (debt-to-income ratio), and "close_Acc" (the total amount of remaining credit in the borrower's credit profile). We mixed the real test data with the generated false test data and conducted multiple experiments on datasets with different proportions of false data. To ensure a fair comparison, other settings were kept unchanged. Figure 6 shows the impact of different proportions of false data on the performance of the FI-CL model on Dataset A.

From the results of the placebo test experiment in Figure 6, it can be observed that the performance indicators of the FI-CL model on Dataset A vary only slightly across different false data proportions. Specifically, when the proportion of false data is 10%, the model's performance is very close to its performance without any false data. This indicates that the model exhibits some robustness against minor false information, effectively handling the influence of noisy data. This is of significance for real-world applications where small amounts of false information might exist. As the false data proportion gradually increases to 30% and 50%, the model's performance does experience a slight drop, but the difference is not substantial. This suggests that the model can, to a certain extent, handle moderate levels of false information and still maintain a relatively stable performance level. Even when the proportion of false data reaches as high as 80%, the performance degradation of the model remains relatively limited. Although there's a decline in performance, the model still retains a degree of predictive capability. In summary, the FI-CL model demonstrates a relatively stable performance across varying false data proportions, showcasing its robustness.

VI. CONCLUSION

We propose a credit prediction method that integrates feature interaction and contrastive learning. By introducing DNN and attention mechanisms, we aim to improve the issue of neglecting feature importance during second-order interactions and the non-linear combination problem of deep high-order features. In terms of feature representation learning, contrastive learning is used to enhance the quality and generalization of feature representations. Comparative experiments with other classical methods demonstrate that the FI-CL method improves AUC performance by 1-3%, KS by 1-7%, and G-mean by 2-4%. This confirms that our model can improve credit prediction performance.

In this study, we employed simple SMOTE sampling, resulting in unstable ACC performance of the model. In the future, we plan to consider improving various sampling techniques and establish the optimal sampling method for the best results across different threshold ranges of imbalance ratios. Additionally, this paper only explores the impact of DNN on credit default predictions. Moving forward, we can substitute DNN with a residual network to effectively construct deeper neural networks leveraging the unique propagation characteristics of residual networks. Lastly, to delve deeper into determining the influence of the BANI environment on the model, we contemplate incorporating a more extensive set of macroeconomic indicators, such as inflation rates, and examining time-related factors (for instance, economic cycles, seasonal variations, etc.) to analyze their effects on loan predictions.

AUTHOR CONTRIBUTIONS

Method research, Lisi Zhang, Qiancheng Yu, and Beijing Zhou; Writing—original draft, LISI Zhang; Paper revision, Lisi Zhang and Beijing Zhou; Data curation, Lisi Zhang, Beijing Zhou, Yifan Zhang and Zhiyong Hu; and Funding acquisition, Qiancheng Yu.

REFERENCES

[1] H. L. Chen, C. Yang, and M. Du, "Improving the credit risk prediction model of LightGBM based on boundary adaptive SMOTE and focal loss function," *Comput. Appl.*, vol. 42, no. 7, pp. 2256–2264, 2022.

[2] E. C. E. Silva, I. C. Lopes, A. Correia, and S. Faria, "A logistic regression model for consumer default risk," *J. Appl. Statist.*, vol. 47, nos. 13–15, pp. 2879–2894, Nov. 2020.

[3] G. Yao, X. Hu, and G. Wang, "A novel ensemble feature selection method by integrating multiple ranking information combined with an SVM ensemble model for enterprise credit risk prediction in the supply chain," *Expert Syst. Appl.*, vol. 200, Aug. 2022, Art. no. 117002.

[4] G. Yao, X. Hu, T. Zhou, and Y. Zhang, "Enterprise credit risk prediction using supply chain information: A decision tree ensemble model based on the differential sampling rate, synthetic minority oversampling technique and AdaBoost," *Expert Syst.*, vol. 39, no. 6, pp. 1–12, Jul. 2022.

[5] N. Arora and P. D. Kaur, "A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105936.

[6] S. Haoran and W. Boyang, "Research on credit risk assessment of online network credit based on GBDT," in *Proc. Int. Conf. Big Data Manage.*, May 2020, pp. 103–107.

[7] K. Wang, M. Li, J. Cheng, X. Zhou, and G. Li, "Research on personal credit risk evaluation based on XGBoost," *Proc. Comput. Sci.*, vol. 199, pp. 1128–1135, Jan. 2022.

[8] Y. Song, Y. Wang, X. Ye, D. Wang, Y. Yin, and Y. Wang, "Multi-view ensemble learning based on distance-to-model and adaptive clustering for imbalanced credit risk assessment in P2P lending," *Inf. Sci.*, vol. 525, pp. 182–204, Jul. 2020.

[9] M. A. Muslim, T. L. Nikmah, D. A. A. Pertiwi, J. Subhan, and Y. Dasril, "New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning," *Intell. Syst. Appl.*, vol. 18, May 2023, Art. no. 200204.

[10] C. Dhasarathan, M. Shanmugam, M. Kumar, D. Tripathi, S. Khapre, and A. Shankar, "A nomadic multi-agent based privacy metrics for e-health care: A deep learning approach," *Multimedia Tools Appl.*, vol. 2023, pp. 1–24, Jun. 2023.

[11] R. F. Brenes, A. Johannssen, and N. Chukhrova, "An intelligent bankruptcy prediction model using a multilayer perceptron," *Intell. Syst. Appl.*, vol. 16, Nov. 2022, Art. no. 200136.

[12] Y. Guo, "Credit risk assessment of P2P lending platform towards big data based on BP neural network," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102730.

[13] T. Thi Nguyen, H. Tahir, M. Abdelrazek, and A. Babar, "Deep learning methods for credit card fraud detection," 2020, *arXiv:2012.03754*.

[14] X. T. He, "Research on credit scoring integration model combined with deep learning optimization algorithm," Northwest Univ., Xian, China, Tech. Rep., 2021, no. 2.

[15] J. Y. Zhang, "Construction and application of credit risk control model based on deep neural network," Beijing Jiaotong Univ., Beijing, China, Tech. Rep., 2021, no. 3.

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

[17] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 43–50.

[18] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," 2017, *arXiv:1708.04617*.

[19] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*.

[20] T. Huang, Z. Zhang, and J. Zhang, "FiBiNET: Combining feature importance and bilinear feature interaction for click-through rate prediction," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 169–177.

[21] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? Simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1294–1303.

[22] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung, and H. Yin, "XSimGCL: Towards extremely simple graph contrastive learning for recommendation," 2022, *arXiv:2209.02544*.

[23] H. T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, and G. Anderson, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[24] F. Wang, Y. Wang, D. Li, H. Gu, T. Lu, P. Zhang, and N. Gu, "CL4CTR: A contrastive learning framework for CTR prediction," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 805–813.

[25] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," 2021, *arXiv:2106.11342*.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

LISI ZHANG was born in 1999. She is currently pursuing the M.S. degree with the College of Computer Science and Engineering, North Minzu University. Her research interests include machine learning and deep learning.

QIANCHENG YU was born in 1976. He received the Ph.D. degree. He is currently an associate professor and a M.S. supervisor. His research interests include social perception computing, social network analysis, and machine learning.

BEIJING ZHOU was born in 1997. He is currently pursuing the M.S. degree with the College of Computer Science and Engineering, North Minzu University. His research interests include knowledge graph and recommended systems.

YIFAN ZHANG was born in 1998. She is currently pursuing the M.S. degree with the College of Computer Science and Engineering, North Minzu University. Her research interests include machine learning and ensemble learning.

ZHIYONG HU was born in 1998. He is currently pursuing the M.S. degree with the College of Computer Science and Engineering, North Minzu University. His research interests include federated learning and algorithm optimization.