IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 May 2023, accepted 26 June 2023, date of publication 30 June 2023, date of current version 7 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290957

RESEARCH ARTICLE

AMWSPLAdaboost Credit Card Fraud Detection Method Based on Enhanced Base Classifier Diversity

WANG NING[1], **SILIANG CHEN**[2], **SONGYI LEI**[2], **AND XIONGBIN LIAO**[2]
[1]College of Computer and Communication, Hunan Institute of Engineering, Xiangtan 411104, China
[2]College of Computational Science and Electronics, Hunan Institute of Engineering, Xiangtan 411104, China

Corresponding author: Siliang Chen (2415035764@qq.com)

This work was supported by the Innovation and Entrepreneurship Training Program for College Students in Hunan Province, in 2022, under Grant 202211342053.

ABSTRACT With the popularity of online transactions, credit card fraud incidents are occurring more and more frequently, and adaptive enhancement (Adaboost) models are most often used in credit card fraud detection, so how to improve the robustness of the traditional Adaboost algorithm has become a hot issue. A large part of the reason for the poor robustness of the traditional Adaboost algorithm is that the base classifier is selected in a way that is uniquely oriented to the error rate. Therefore, this paper uses an adaptive hybrid weighted self-paced learning method to improve the objective function of the Adaboost algorithm, thus changing the strategy of base learner selection in the Adaboost algorithm, while the self-paced learning selected in this paper The self-adaptive threshold finding algorithm selected in this paper can well mitigate the influence of human experience on model training. This paper also selects a double-fault measure to calculate the degree of diversity among base categories from the perspective of generalization error, adds the influence coefficient of diversity to the weight calculation of weak learners, and gives the optimal range of influence coefficients through experiments. Finally, the proposed improved algorithm is applied to credit card fraud scenario, and the experiments are compared with several effective Adaboost improvement algorithms, which show that the combined performance of the proposed improved algorithm is better than other algorithms in terms of AUC value and F1 value.

INDEX TERMS Credit card fraud detection, adaboost, loss function, self-paced learning, diversity.

I. INTRODUCTION

With the advent of the big data era, people are increasingly using the internet to facilitate their daily lives, especially using credit cards for transactions. However, as the amount and frequency of credit card transactions increase, so do incidents of credit card fraud. Due to the severe data imbalance and noise in large datasets, as well as the high data dimensionality, traditional methods are unable to achieve higher accuracy. Therefore, how to better detect credit card fraud remains a hot issue in the era of big data. Nowadays, credit card online payment systems typically establish a predictive model to distinguish between fraudulent and non-fraudulent transactions [1]. In recent years, many effective predictive models have been proposed by experts and scholars: Chen et al. [2] proposed a financial fraud detection solution based on deep convolutional neural networks (DCNN) using deep learning algorithms. When dealing with large amounts of data, this technique can improve detection accuracy. Taha et al. [3] proposed an intelligent method for detecting credit card transaction fraud using an optimized gradient boosting machine (OLightGBM). By integrating a Bayesian-based hyperparameter optimization algorithm to adjust the parameters of LightGBM. Ileberi et al [4]. proposed a credit card fraud detection engine based on machine learning (ML) using a genetic algorithm (GA) for feature selection.

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

Through experimental verification, this approach was shown to be superior to conventional detection algorithms. This research mainly focuses on the processing of imbalanced data, using undersampling techniques and different machine learning algorithms to obtain more accurate and better results. Ahmad et al. [5] proposed a data set classification framework based on fuzzy C-means clustering to address the severe data imbalance issue in credit card fraud. The framework selects similar fraudulent behaviors and normal instances with the same characteristics to ensure the integrity of data features. Recently, some scholars have pointed out that the Adaboost improved algorithm can better meet practical business needs in the credit card fraud scenario compared to traditional classification algorithms [6], [7], [8].

As early as the proposal of the traditional Adaboost algorithm by Freund et al. [9], Adaboost has been widely applied in various fields, especially in transaction fraud detection [10], [11]. However, the traditional Adaboost algorithm suffers from problems such as overfitting and poor robustness. Therefore, when applying Adaboost to credit card fraud detection, it usually requires improvement. Wang et al. [12] proposed an AWTAdaboost algorithm that introduces a penalty factor to alleviate the problem of excessive training time for traditional Adaboost algorithm on massive transaction data. Ileberi et al. [13] combined Adaboost algorithm with some traditional machine learning algorithms using SMOTE sampling technique and applied it to credit card fraud detection. Mo et al. [14] proposed a new classification method for two-class imbalanced datasets in credit card fraud, called GAN-Adaboost-DT algorithm, which addresses the serious class imbalance problem in credit card fraud data. However, these improvements do not fundamentally solve the problem of overfitting in Adaboost algorithm. Therefore, this paper proposes a new Adaboost improvement method by utilizing self-paced learning (SPL) to reconstruct the objective function of Adaboost, aiming to improve the generalization performance of traditional Adaboost in credit card fraud detection. The main contributions of this paper are as follows:

1) In this paper, a new SPL regular term with adaptive hybrid weights is selected to embed into the objective function of Adaboost, which changes the traditional weak classifier selection strategy of Adaboost with the error rate as the only guide, and a generalized SPLAdaboost algorithm framework is proposed.
2) In updating the weak classifier weights, considering the generalization effect of diversity among base classifications on the final integrated model, this paper introduces the double-fault to measure the diversity among weak classifiers, reconstructs the weight calculation formula of base classifiers, and gives the range of values of diversity influence coefficients through experiments.

The rest of the paper is organized as follows: Chapter 2 briefly reviews the research results of SPL regular terms in recent years and focuses on an adaptive hybrid weight self-step learning that will be used in this paper. Chapter 3 introduces the optimization algorithm proposed in this paper, D-AMWSPLAdaboost. Chapter 4 compares the performance of the optimization algorithm proposed in this paper with the more effective Adaboost improvement algorithm in recent years under credit card fraud dataset as well as other datasets through experiments. And the conclusions of this paper are summarized in Chapter 5.

II. RELATED WORK

Inspired by the learning process of humans and animals, which generally starts with learning simple tasks before gradually progressing to relatively difficult ones, Kumar et al. [15] proposed self-paced learning (SPL), whose core is to optimize the objective function using regularization terms. Numerous studies have also shown that using SPL to optimize models can improve their robustness, enhance their ability to resist overfitting, and reduce the impact of noisy data in the training set [16], [17]. The typical form of SPL includes incorporating various forms of regularization terms into the learning objective, and has a gradually increasing step parameter.Let $D = \{(x_1, y_1), (x_2, y_2) \ldots\ldots.(x_n, y_n)\}$ denote the training set, and the objective function (1) consists of two parts: the weighted sample loss value and the self-step regularization term, where $V = [v_1, v_2 \ldots\ldots.v_n]$ is the introduced sample weight parameter to indicate the difficulty of the training sample;w is the model parameter; λ is the parameter of the SPL regularization term $f(V;\lambda)$ to control the step size of learning. $L(y_i, g(x_i, w))$ represents the loss value between the sample label y_i and the model prediction $g(x_i, w)$. The goal of SPL is to obtain the optimal parameters by optimizing the objective function (1) training w, V

$$\min_{w,V} E(w, V; \lambda) = \sum\nolimits_{\mathrm{i}=1^n} v_i L(y_i, g(x_i, w)) + f(V; \lambda) \quad (1)$$

In recent years, the improvement of self-step learning is mainly reflected in the proposal of different forms of self-step regular terms, which have different effects on the performance of the whole Self-paced Learning model.As shown in (2), (3).

$$f(V; \lambda) = \lambda \sum\nolimits_{i=1}^{n} v_i \quad (2)$$

$$v_i = \begin{cases} 1, L_i \le \lambda_2 \\ \dfrac{\varepsilon}{L_i} - \dfrac{\varepsilon}{\lambda_1}, \lambda_2 < L_i < \lambda_1 \\ 0, L_i \ge \lambda_1 \end{cases} \quad (3)$$

And then more effective SPL regular terms were proposed: mixed-weight SPL regular terms (MWSPL), which are expressed in the form shown in (4), (5), where $\varepsilon = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}$

$$f^m(V; \lambda) = -\varepsilon \sum\nolimits_{i=1}^{n} \log(v_i + \frac{1}{\lambda_i}\varepsilon) \quad (4)$$

$$v_i = \begin{cases} 1, L_i \le \lambda_2 \\ \dfrac{\varepsilon}{L_i} - \dfrac{\varepsilon}{\lambda_1}, \lambda_2 < L_i < \lambda_1 \\ 0, L_i \ge \lambda_1 \end{cases} \quad (5)$$

Compared to the traditional self-paced regularization term, MWSPL is more biased towards samples with smaller loss values, which makes it more reasonable. Gong et al. [18] proposed the MOSPLAdaboost algorithm using this self-paced regularization, while this paper only compares the improvement of the regularization term on the Adaboost algorithm. Therefore, we refer to the improved algorithm that only uses MWSPL to reconstruct the objective function of Adaboost as MWSPLAdaboost.

Adjustable polynomial soft weighted SPL regular term (PSWSPL), which is constructed in the form shown in (6), (7).

$$f^{p}(V;\lambda)=\lambda(\frac{1}{t}||v||_2^t-\sum\nolimits_{i=1}^{n} v_i) \tag{6}$$

$$v_i=\begin{cases}(1-\dfrac{L_i}{\lambda})^{\frac{1}{t-1}} & L_i<\lambda\\ 0 & L_i\geq\lambda\end{cases} \tag{7}$$

Meanwhile, Wang et al. [19] incorporated PSWSPL into Adaboost and demonstrated that this improvement method performs better than other popular SPLBoost methods when dealing with training data with outliers. We will refer to the improved algorithm that only uses PSWSPL to reconstruct the objective function of Adaboost as PSWSPLAdaboost.

Li et al. [20] proposed an adaptive mixture weight self-paced learning (AMWSPL) method, which has the following form:

$$f^{a}(V;\lambda)=\sum\nolimits_{i=1}^{n}(\frac{\lambda_1-\lambda_2}{t}\mathrm{v}_i^t-\lambda_1 v_i) \tag{8}$$

$$v_i=\begin{cases}1, L_i\leq\lambda_2\\ (\dfrac{\lambda_1-L_i}{\lambda_1-\lambda_2})^{\frac{1}{t-1}}, \lambda_2<L_i<\lambda_1\\ 0, L_i\geq\lambda_1\end{cases} \tag{9}$$

Meanwhile, since the values of parameters such as learning rate in SPL often depend on human experience to set, Li et al. used an iterative threshold adaptive parameter finding algorithm to determine the parameters t, λ_1, λ_2, and the general flow of this iterative threshold adaptive parameter finding algorithm is shown in Algorithm 1.

And each round of training calls the λ generated by Algorithm 1 to computationally generate the remaining 3 parameters used to update V

$$\lambda_1^m=\begin{cases}Iteration\_threshold(L^0) & m=0\\ \lambda_1^{m-1}+\dfrac{average(L^m)*r}{m} & m>0\end{cases} \tag{10}$$

$$\lambda_2^m=Iteration\_threshold(L^m_{\lambda_1^m}) \tag{11}$$

$$t=\tan((1-\frac{len(L^m_{\lambda_1^m})}{2*len(L^m)+1})*\frac{\pi}{2}) \tag{12}$$

where *Iteration_threshold* denotes Algorithm 1, $average(L^m)$ denotes the mean of the sample loss distribution under the round m, L_λ denotes the new distribution consisting of sample losses with loss values less than λ, and r denotes the learning rate of the regulated self-paced learning process. The method proposed by Li et al. [20], an adaptive mixture-weighted self-paced learning method (AMWSPL), is formulated as follows. This approach extends the mixture weight to a more general form by introducing a polynomial mixture-weighted regularization term. Additionally, the adaptive parameter-solving framework alleviates the importance of human experience in determining parameters. To differentiate it from the improved algorithm proposed in this paper, we will refer to the algorithm that only utilizes AMWSPL to reconstruct the objective function as AMWSPLAdaboost.

Algorithm 1 Iterative Threshold Adaptive Parameter Finding Algorithm

```
Input: sample loss value sequence L; maximum number of iterations M
Output: Parameter $\lambda$
1. Find the maximum, minimum loss values in $l_{\max}$, $l_{\min}$.
   And let the initial threshold value be $l^0=\frac{l_{\max}+l_{\min}}{2}$
2. for k=0 to M:
3. Divide L into two parts greater than $l^k$ and less than $l^k$ according to the threshold $l^k$
4. Calculate the quantity of loss value less than $l^k$ and greater than $l^k$ in L respectively: $num_{less}$, $num_{greater}$,
   and find the average grayscale value of these two components.
   $l_d=\frac{\sum_{L(i)<l^k}L(i)}{num_{less}}$
   $l_u=\frac{\sum_{L(i)>l^k}L(i)}{num_{greater}}$
5. Update the threshold for the next round
   $l^{k+1}=\frac{l_d+l_u}{2}$
6. End for
7. Get the final threshold value $\lambda=l^{M+1}$
```

III. THE METHOD PROPOSED IN THIS PAPER

A. SPLADABOOST

In this chapter, we will use the AMWSPL introduced in the previous chapter to optimize the objective function of the traditional Adaboost algorithm, and propose a general SPLAdaboost algorithm framework. Before that, we will first analyze the training principle of traditional Adaboost.

The traditional Adaboost classification algorithm uses an exponential loss function to approximate the 0/1 loss function instead.And the objective function for finding the best weak classifier G_m^* in the round m and for calculating the corresponding weak classifier weights α_m^* is shown in (13),(14).

$$\{\alpha_m^*, G_m^*\}=\arg\min_{\alpha,G}\sum\nolimits_{\mathrm{i}=1}^{N} w_{mi}\exp(-y_i\alpha_m G_m(x_i)) \tag{13}$$

where $w_{mi} = \exp(-y_i f_{m-1}(x_i))$ denotes the weight of the ith sample in the mth round, and f_{m-1} denotes the strong classifier integrated from the weak classifiers generated under the previous m-1 rounds at the round m.

Solving this objective equation yields: the selection strategy of the optimal weak classifier G_m^* for the round m is

$$G_m^* = \arg\min_G \sum_{i=1^N} w_{mi} I(y_i \neq G_m(x_i)) \tag{14}$$

$$I(y_i \neq G_m(x_i)) \begin{cases} 0 & y_i = G_m(x_i) \\ 1 & y_i \neq G_m(x_i) \end{cases} \tag{15}$$

And the corresponding weak classifier weights α_m^* are calculated as

$$\alpha_m^* = \frac{1}{2} \ln \frac{1-e_m}{e_m} \tag{16}$$

$$e_m = \frac{\sum_{i=1}^N w_{mi} I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_{mi}} \tag{17}$$

where e_m denotes the error rate. Upon observing the Eqs (16) and (17) above, it can be noticed that the traditional Adaboost algorithm is a direct algorithm guided by the error rate. In each round, the best weak classifier is selected based on the lowest error rate, and the weight of the weak classifiers in the final ensemble is only related to the classification error rate. Adaboost also assigns larger weights to samples with larger loss values, which may be outliers. Assigning larger weights to these outlier samples may cause the subsequent weak classifiers to be biased towards these outliers and ignore the normal samples, leading to poor learning performance of the trained model on normal samples. To reduce the occurrence of this overfitting phenomenon and improve the robustness of the model, many researchers have made improvements from the perspective of the objective function, sample update method, weak learner selection strategy, etc. In this paper, we will optimize the traditional Adaboost objective function by incorporating AMWSPL. After introducing sample weights V and self-paced regularization term, the original objective (13) becomes:

$$\{\alpha_m^*, G_m^*\} = \arg\min_{\alpha,G,V} [\sum_{i=1^N} v_i w_{mi} \exp(-y_i \alpha_m G_m(x_i)) + \sum_{i=1}^N (\frac{\lambda_1 - \lambda_2}{t} \mathrm{v}_{i^t} - \lambda_{1v_i})] \tag{18}$$

From (9), V can be directly used as a constant when solving for. α, G

$$G_m^* = \arg\min_G \sum_{i=1^N} v_i w_{mi} \exp(-y_i \alpha_{mG_m(x_i)}) \tag{19}$$

Here we approximate the expansion to the second order using Taylor expansions [20], and since considering $y_i \in (-1, 1), G_m \in (-1, 1)$ the final weak classifier selection strategy for AMWSPLAdaboost becomes

$$\begin{aligned} G_m^* &\approx \arg\min_G \sum_{i=1}^N v_i w_{mi} (1 - y_i \alpha_m f_{m-1}(x_i) + \frac{\alpha_m^2 f_{m-1}(x_i)^2}{2}) \\ &= \arg\min_G \sum_{i=1}^N v_i w_{mi} (y_i - \alpha_m f_{m-1}(x_i))^2 \end{aligned} \tag{20}$$

The following solution is available at α_m^*.

$$\begin{aligned} \alpha_m^* &= \arg\min_\alpha \sum_{i=1}^N v_i w_{mi} \exp(-y_i \alpha_m G_m(x_i)) \\ &= \arg\min_\alpha \sum_{y_i=G_m(x_i)}^N v_i w_{mi} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)}^N v_i w_{mi} e^{\alpha} \\ &= \arg\min_\alpha [\sum_{y_i=G_m(x_i)}^N v_i w_{mi} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)}^N v_i w_{mi} e^{-\alpha} \\ &\quad - \sum_{y_i \neq G_m(x_i)}^N v_i w_{mi} e^{-\alpha} + \sum_{y_i \neq G_m(x_i)}^N v_i w_{mi} e^{\alpha}] \\ &= \arg\min_\alpha \sum_{i=1}^N v_i w_{mi} e^{-\alpha} + (e^\alpha - e^{-\alpha}) \\ &\quad \times \sum_{i=1}^N v_i w_{mi} I(y_i \neq G_m(x_i)) \end{aligned} \tag{21}$$

The derivative of α is given by making the derivative function 0 and dividing both sides by $\sum_{i=1}^N v_i w_{mi}$, and (21) is rewritten as

$$\begin{aligned} (e^\alpha + e^{-\alpha}) e_m^a &= e^{-\alpha} \\ (e^{2\alpha} + 1) e_m^a &= 1 \\ \alpha &= \frac{1}{2} \ln \frac{1 - e_m^a}{e_m^a} \end{aligned} \tag{22}$$

where at this point e_m^a is the error rate of AMWSPLAdaboost

$$e_m^a = \frac{\sum_{i=1}^N v_i w_{mi} I(y_i \neq G_m(x_i))}{\sum_{i=1}^N v_i w_{mi}} \tag{23}$$

Since (21) is a convex function, the local minimum found after taking the derivative of α is the global minimum, i.e., α obtained from (22) satisfies $\alpha = \alpha_m^*$. By comparing the traditional Adaboost algorithm, it is easy to see that AMWSPLAdaboost is no longer guided solely by the error rate.At the same time, considering the above SPL methods, the general core is to change the construction of V. Therefore, after using these SPLs to optimize the Adaboost, the weight calculation of the weak learner and the decision are the same.For this reason, the SPLAdaboost algorithm framework is unified in this paper, as shown in Algorithm 2

B. D-AMWSPLADABOOST ALGORITHM

In the credit card fraud detection problem, the number of sample categories is extremely imbalanced. As the model's precision is greatly affected by the ratio of positive and negative samples, we attach more importance to the model's generalization performance rather than its precision in the credit card fraud detection scenario [21]. In ensemble learning, the final model's generalization performance is not only related to the accuracy of weak classifiers but also to the diversity between the weak classifiers participating in the ensemble. Krogh et al. [22] proposed the following formula to illustrate the relationship between the diversity and accuracy of weak classifiers on the final ensemble accuracy.

$$E = \bar{E} - \bar{A} \tag{24}$$

where $\bar{E}$ denotes the weighted mean of the generalization error of the weak learners and $\bar{A}$ denotes the weighted mean of the diversity among the weak learners. The equation

```
Algorithm 2 SPLAdaboost Algorithm Framework
Input: training set $D = \{(x_1, y_1), (x_2, y_2) \ldots\ldots.(x_n, y_n)\}$, maximum number of cycles T.
Output: Strong classifier H(x)
 1. Sample weights initialization:
      $D = \{d(1), d(2), \ldots..d(N)\},$
      $V = \{v(1), v(2), \ldots.v(N)\},$
      $d_1(i) = \frac{1}{N}, v_1(i) = 1, i = 1, \ldots\ldots N$
 2. for m = 1 to T do:
 3.   Training on the training set to obtain the weak classifier $G_m$
 4.   Calculate the error rate $e_m^a$ of $G_m$ on the training set according to (23)
 5.   Calculate the weights $\alpha_m$ of $G_m$ according to (22)
 6.   The best base classifier $G_m^*$ for this round is selected according to (20)
 7.   Integration of weak classifiers according to weighted voting $G_m^*$
        $f_m = f_{m-1} + \alpha_m G_m^*$
 8.   Update the sample weight distribution
        $D_{m+1}(i) = \begin{cases} \frac{D_m(i)}{Z_m} e^{-\alpha_m} G_m(x_i) = y_i \\ \frac{D_m(i)}{Z_m} e^{\alpha_m} G_m(x_i) \neq y_i \end{cases} = \frac{D_m(i)\exp(-\alpha_m y_i G_m(x_i))}{Z_m}$
      (where $z_m$ is the normalization factor)
 9.   Integrate using $G_m^*$ to calculate loss values for each sample and generate sample loss sequences $L_m$
10.   Update the sample weights $V_{m+1}$ using the specified update method for the chosen SPL
11. end for
12. The final strong classifier is obtained. $H(x) = sign(f_T)$
```

states that the higher the accuracy of the weak learner and the greater the diversity, the better the final integration will be. For this reason, the final integration result may not be good if we pursue high accuracy of weak learners and ignore the effect of diversity. And there is no unified standard about the measure of learner diversity, the most common ones nowadays are:in pairwise metric:Q-statistic in pairwise diversity measures, correlation coefficient, Disagreement measure,Double-Fault measure (DF) etc. [23]

Among them, Jiang Zheng et al [24] showed experimentally that the Pearson correlation coefficient between the DF and the test error is the largest, indicating that the DF has the highest correlation with the test error, so the DF is chosen in this paper to calculate the diversity among classifiers.The prediction of two different classifiers is shown in Table 1.

TABLE 1. Classifier prediction matrix.

Classifier a Forecast	Classifier b forecast	
	Correct(1)	Incorrect(0)
Correct(1)	n_{11}	n_{10}
Incorrect(0)	n_{01}	n_{00}

And satisfies $n_{11} + n_{10} + n_{01} + n_{00} = m$, where the DF is given by (20), $DF \in [0, 1]$, and a smaller DF indicates greater diversity between the two base classifiers

$$DF = \frac{n_{00}}{m} \tag{25}$$

In this paper, in order to improve the final integration, the training process of AMWSPLAdaboost will be improved i.e. the optimization algorithm proposed in this paper: D-AMWSPLAdaboost, which is mainly reflected in the optimization of (15) as

$$\alpha = sign(\alpha)^*[|\alpha| + \lambda^*(1 - DF)] \tag{26}$$

In this paper, we hope that the composition of the weights of the new weak classifier will remain with the original weight formula as the main body, and only secondarily supplemented with diversity considerations. Because of the excessive change in the nature of (21), errors may occur in selecting the best weak classifier for each round, and the classifier that minimizes the generalization error cannot be selected, and the speed of model convergence may decrease, and may even lead to a decrease in the overall model performance.Therefore, in considering the new weak classifier weights, the sign() representation is first used to still take the sign of the original weights, ensuring that the prediction direction of the trained weak classifier remains unchanged. In other words, the diversity only changes the strength of the weak classifier's final participation in the integration, not the direction of its final decision. Secondly, considering the different magnitudes of α and DF. Therefore, a diversity influence coefficient is introduced, and the main role of this coefficient λ is to act as a trade-off term to balance the accuracy rate and diversity. Comparing (15) again, it can be found that the weight of the final weak learner involved in the integration is composed of both the error rate (which is directly related to the accuracy rate) and the diversity of that weak learner. Also for the same error rate, the classifier with greater diversity will have a greater weight and will eventually be chosen more favorably in each round.

However, since there is no quantitative formula for accuracy and diversity, if λ is taken too large, it will be over-corrected and will definitely affect the learning performance of the final model, while if it is taken too small,

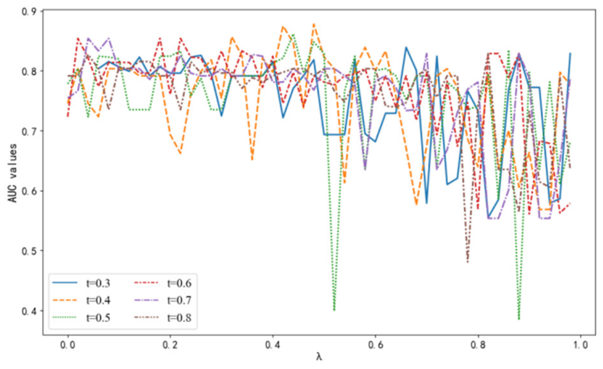

FIGURE 1. Curve of AUC value with λ at different learning rates.

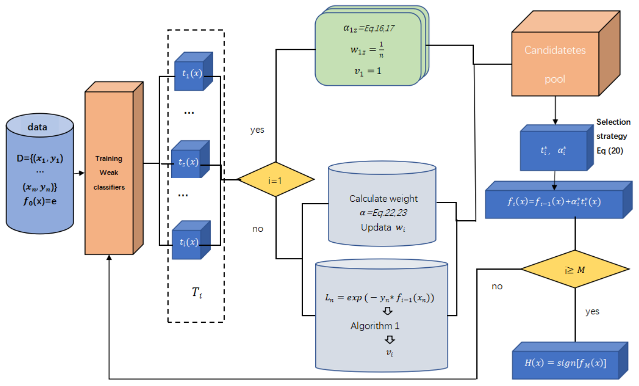

FIGURE 2. D-AMWSPLAdaboost rough flow chart.

it will not be corrected, so how to adjust λ is a critical issue, while λ equals to 0 is equivalent to not considering the influence of diversity, i.e., the original model is used. As shown in Fig. 1 through experiments, we found that λ in the interval of (0,0.6), the AUC value of the model is generally better than λ when it is equal to 0, while the curve fluctuates more when λ is greater than 0.6, indicating that the model is less robust under this interval, and the model performance decreases instead with the increment of λ. In this paper, the grid search method is used to determine the appropriate λ.Finally, we refer to the optimized AMWSPLAdaboost as D-AMWSPLAdaboost,The general flow of D-AMWSPLAdaboost is shown in Fig. 2.

IV. EXPERIMENT

To verify whether the proposed D-AMWSPLAdaboost algorithm is superior to other Adaboost variants in credit card fraud detection, this paper conducts experiments on different datasets to compare MWSPLAdaboost,PSWSPLAdaboost,AMWSPLAdaboost, PF_AWTAdaboost, and D-AMWSPLAdaboost. To avoid interference from other factors, decision stumps are used as weak learners in the model training process.

A. PERFORMANCE EVALUATION METRICS

The traditional dichotomous evaluation metrics include Precision rate, Recall rate, ROC curve and F-index. We divide the samples as follows: when the actual sample value is positive, the classifier predicts the positive case as positive TP; when the actual sample value is negative, the classifier predicts the positive case as false positive FP; when the actual sample value is negative, the classifier predicts the negative case as true negative TN; when the actual sample value is positive, the classifier predicts the negative case as false negative FN. Thus, we have.

$$Precision\ rate: P = \frac{TP}{TP + FP} \tag{27}$$

$$Recall\ rate: R = \frac{TP}{TP + FN} \tag{28}$$

They represent the prediction accuracy and recognition ability of the model for positive class samples, respectively, but obviously there is a mutual constraint between these two metrics, the best case would be high for both metrics, but the actual situation is usually high for one metric and low for the other. In contrast, the F1 value can be a good balance of both accuracy and recall.

$$F1 = \frac{2^* P^* R}{P + R} \tag{29}$$

In addition to this, ROC is a graph with FP/(FP+TN) (False positive rate) as the horizontal axis and TP/(TP+FN) (True case rate) as the vertical axis, which reflects the visualization of Precision rate and Recall rate constraints under different parameter thresholds, AUC value is the area enclosed by the ROC curve, which is a quantitative representation of the ROC curve. It is usually a performance indicator used to reflect the measure of learner's strengths and weaknesses. The larger the AUC value, the better the overall performance of the classifier.

In unbalanced classification, an increase in minority class accuracy is often accompanied by a decrease in majority class accuracy. Therefore, AUC and F1 are often used to evaluate the overall performance of unbalanced classifiers [25]. The credit card fraud detection problem is a typical class of unbalanced classification problem, so the AUC value as well as the F1 value are used as evaluation metrics in this paper in order to assess the comprehensive performance of the model.

B. CREDIT CARD FRAUD DATASET

To validate the effectiveness of our proposed approach, we conducted more experiments using a publicly available dataset of credit card transactions. In this chapter, we use a dataset containing transactions made by European cardholders using credit cards in September 2013 on the kaggle platform, which contains 24,357,143 legitimate credit card transactions and 29,757 fraudulent transactions. The attributes from V1 to V28 do not have specific names because they have been desensitized in advance to protect the sensitive information of users. The final sample label

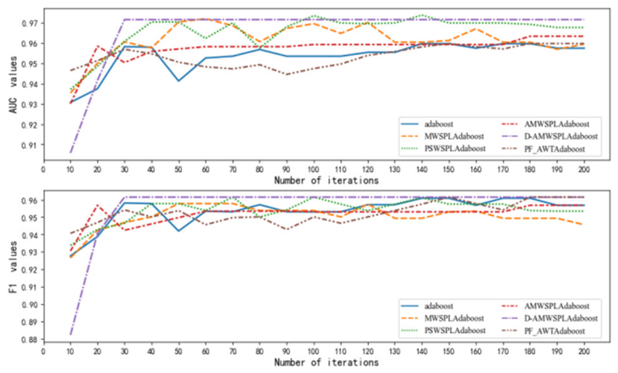
FIGURE 3. Performance of each algorithm on credit card fraud dataset.

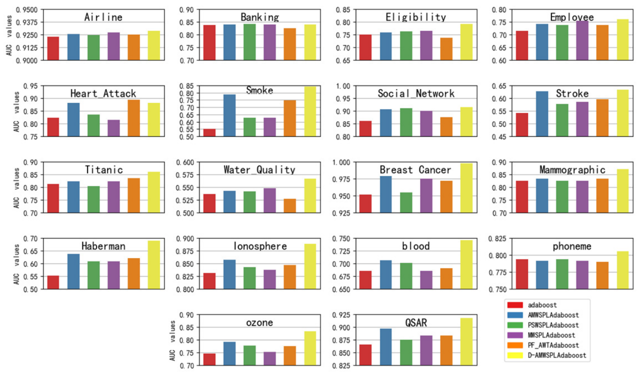
FIGURE 4. AUC values of each algorithm on different data sets.

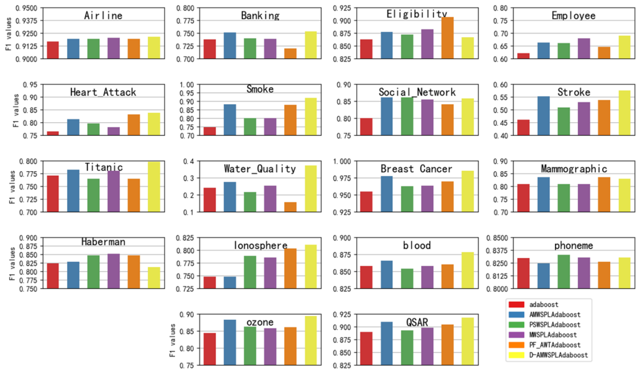
FIGURE 5. F1 values of each algorithm on different data sets.

value of 1 indicates a legitimate transaction and -1 indicates a fraudulent transaction. Before training the model we first processed the dataset for missing values and undersampling. The final sample size of the dataset is 1108, of which 70% is taken as the training set and 30% as the test. The change curves of AUC value and F1 value of each algorithm on this dataset with increasing number of iterations are obtained as shown in Fig. 3.

From the Fig.3, we can find that D-AMWSPLAdaboost outperforms the other algorithms on the credit card fraud dataset in aggregate. The AUC value of D-AMWSPLAdaboost is 1.41% higher than the traditional Adaboost algorithm and 0.85% higher than the AMWSPLAdaboost algorithm when all the algorithms converge. The F1 value of D-AMWSPLAdaboost is 0.47% higher than the conventional Adaboost algorithm, and 0.467% higher than the AMWSPLAdaboost algorithm. It also converges faster than other algorithms: D-AMWSPLAdaboost converges at 30 iterations, while the traditional Adaboost algorithm converges at 140 iterations, and the fastest AMWSPLAdaboost algorithm converges at about 60 iterations. AMWSPLAdaboost has better robustness than other Adaboost improvement algorithms. However, it is obvious from Figure 3 that D-AMWSPLAdaboost suffers more from underfitting than other algorithms when the number of iterations is small, so we should try to avoid setting fewer iterations when using D-AMWSPLAdaboost in practice. Collectively, the improved algorithm proposed in this paper is advantageous in credit card fraud detection compared to other Adaboost improved algorithms.

C. OTHER DATASET

In order to verify that the improved algorithm proposed in this paper is also applicable to other scenarios, we selected 18 binary classification datasets from three mainstream data platforms, Kaggle, UCI, and OpenML, and performed the same data preprocessing as in the previous section, the experimental results of each algorithm are shown in Fig. 4 and Fig. 5.

Observing Figs. 4 and 5, we can see that the AMWSPLAdaboost algorithm performs better than the other two SPLAdaboosts in terms of overall performance. The main reason is that the parameters of AWMSPLAdaboost are determined by an adaptive algorithm, while the other SPLAdaboosts rely more on human experience to figure out, and the reliance on grid search not only leads to a significant increase in model training time, but also makes it more difficult to set the parameter search range, so it is more reasonable to use the AMWSPLAdaboost algorithm to train the model in a practical scenario. AMWSPLAdaboost algorithm to train the model is more reasonable in practical scenarios. The D-AMWSPLAdaboost algorithm, which is optimized on the basis of AMWSPLAdaboost in this paper, has better comprehensive performance than AMWSPLAdaboost and PF_AWAdabooost, as shown in the 18 data sets:

In terms of AUC values, D-AMWSPLAdaboost achieved a total of 16 highest values, with a maximum improvement of 13.79% compared to the Adaboost algorithm, 5.346% compared to the AMWSPLAdaboost algorithm, and 9.253% compared to the PF_AWTAdaboost algorithm. For the F1 value, D-AMWSPLAdaboost achieved a total of 13 highest values, with a maximum improvement of 17.01% compared to the Adaboost algorithm, 9.51% compared to the AMWSPLAdaboost algorithm, and 21.58% compared to the PF_AWTAdaboost algorithm. However, in the two datasets of Stroke and Water_Quality, D-AMWSPLAdaboost also performed poorly, with AUC values of 0.634005 and 0.567059.

Finally, D-AMWSPLAdaboost outperforms all the compared Adaboost optimization algorithms in this paper, indicating that the improvement method proposed in this paper is effective in improving Adaboost, and D-AMWSPLAdaboost can be applied to other scenarios in addition to credit card fraud.

V. CONCLUSION

In this paper, the latest self-paced learning method with adaptive hybrid weights is embedded into the objective function of the Adaboost algorithm, thereby changing the traditional weak classifier selection strategy of Adaboost, which is solely based on the error rate. The SPL algorithm used in this paper has an adaptive threshold iterative parameter algorithm, which is more convenient for selecting parameters compared to other SPL algorithms. In addition, when calculating the weights of weak classifiers, the influence of diversity on model decision accuracy is considered, and DF is used to measure the diversity among weak classifiers. The strength of participation of weak classifiers in decision-making is changed, but the direction of decision-making is not changed, thereby reconstructing the weak classifier weight update method. Finally, D-AMWSPLAdaboost is compared with several Adaboost improvement algorithms in experiments on credit card fraud datasets and other datasets. In the credit card fraud dataset, compared to the traditional Adaboost algorithm, D-AMWSPLAdaboost increased the AUC value by 1.41% and the F1 score by 0.47%, and it also had faster convergence speed than other algorithms. In other datasets, compared to the Adaboost algorithm, D-AMWSPLAdaboost achieved the highest improvement of 13.79% in terms of AUC value and 17.01% in terms of F1 score. Therefore, the experiments show that the proposed D-AMWSPLAdaboost algorithm can enhance the robustness of the traditional Adaboost algorithm and be effectively applied to credit card fraud detection.

REFERENCES

[1] K. K. Mohbey, M. Z. Khan, and A. Indian, "Credit card fraud prediction using XGBoost: An ensemble learning approach," *Int. J. Inf. Retr. Res.*, vol. 12, no. 2, pp. 1–17, Jul. 2022.

[2] J. I.-Z. Chen and K.-L. Lai, "Deep convolution neural network model for credit-card fraud detection and alert," *J. Artif. Intell.*, vol. 3, no. 2, pp. 101–112, Jun. 2021, doi: 10.36548/jaicn.2021.2.003.

[3] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25579–25587, 2020, doi: 10.1109/ACCESS.2020.2971354.

[4] E. Ileberi, Y. Sun, and Z. Wang, "A machine learning based credit card fraud detection using the GA algorithm for feature selection," *J. Big Data*, vol. 9, no. 1, pp. 1–17, Dec. 2022, doi: 10.1186/s40537-022-00573-8.

[5] H. Ahmad, B. Kasasbeh, B. Aldabaybah, and E. Rawashdeh, "Class balancing framework for credit card fraud detection based on clustering and similarity-based selection (SBS)," *Int. J. Inf. Technol.*, vol. 15, no. 1, pp. 325–333, Jan. 2023, doi: 10.1007/s41870-022-00987-w.

[6] B. Gedela and P. R. Karthikeyan, "Credit card fraud detection using AdaBoost algorithm in comparison with various machine learning algorithms to measure accuracy, sensitivity, specificity, precision and F-score," in *Proc. Int. Conf. Bus. Anal. Technol. Secur. (ICBATS)*, Dubai, United Arab Emirates, Feb. 2022, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9759022

[7] H. Muhal, G. Khatri, G. K. Dhama, and D. Bansal, "Ensemble approach for credit card fraud detection using champion-challenger analysis," in *Proc. Int. Conf. Innov. Comput., Intell. Commun. Smart Electr. Syst. (ICSES)*, Chennai, India, Jul. 2022, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9914387

[8] E. F. Malik, K. W. Khaw, B. Belaton, W. P. Wong, and X. Chew, "Credit card fraud detection using a new hybrid machine learning architecture," *Mathematics*, vol. 10, no. 9, pp. 1480–1496, 2022, doi: 10.3390/math10091480.

[9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

[10] A. Petrovic, N. Bacanin, M. Zivkovic, M. Marjanovic, M. Antonijevic, and I. Strumberger, "The AdaBoost approach tuned by firefly metaheuristics for fraud detection," in *Proc. IEEE World Conf. Appl. Intell. Comput. (AIC)*, Sonbhadra, India, Jun. 2022, pp. 834–839. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9848902

[11] J. Inga and E. S. Cabrera, "Credit default risk analysis using machine learning algorithms with hyperparameter optimization," in *Intelligent Technologies: Design and Applications for Society*. Cham, Switzerland: Springer, 2023, pp. 81–95, doi: 10.1007/978-3-031-24327-1_8.

[12] W. Ning, S. Chen, F. Qiang, H. Tang, and S. Jie, "A credit card fraud model prediction method based on penalty factor optimization AWTAdaBoost," *Comput., Mater. Continua*, vol. 74, no. 3, pp. 5951–5965, 2023, doi: 10.32604/cmc.2023.035558.

[13] E. Ileberi, Y. Sun, and Z. Wang, "Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost," *IEEE Access*, vol. 9, pp. 165286–165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

[14] Z. Mo, Y. R. Gai, and G. L. Fan, "Credit card fraud classification based on GAN-AdaBoost-DT imbalanced classification algorithm," *J. Comput. Appl.*, vol. 39, no. 2, pp. 618–622, 2019, doi: 10.11772/j.issn.1001-9081.2018061382.

[15] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2010, pp. 1189–1197. [Online]. Available: https://dl.acm.org/doi/10.5555/2997189.2997322

[16] D. Meng, Z. Xu, and J. Shu, "Meta self-paced learning," *Scientia Sinica Informationis*, vol. 50, no. 6, pp. 781–793, Jun. 2020, doi: 10.1360/SSI-2020-0005.

[17] Y. J. Chen, L. W. Cao, and Y. Q. Du, "Improvement of fuzzy C-means clustering algorithm based on self-paced data reconstruction regularization," *Comput. Modernization*, pp. 120–126, Jun. 2020.

[18] M. Gong, H. Li, D. Meng, Q. Miao, and J. Liu, "Decomposition-based evolutionary multiobjective optimization to self-paced learning," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 288–302, Apr. 2019, doi: 10.1109/TEVC.2018.2850769.

[19] K. Wang, Y. Wang, Q. Zhao, D. Meng, X. Liao, and Z. Xu, "SPLBoost: An improved robust boosting algorithm based on self-paced learning," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1556–1570, Mar. 2021, doi: 10.1109/TCYB.2019.2957101.

[20] H. Li, Y. Zhao, G. G. Mao, Y. Wu, and J. Y. Liu, "Self-paced learning method with adaptive mixture weighting," *J. Softw.*, vol. 34, no. 5, pp. 1–13, 2023, doi: 10.13328/j.cnki.jos.006438.

[21] H. John and S. Naaz, "Credit card fraud detection using local outlier factor and isolation forest," *Int. J. Comput. Sci. Eng.*, vol. 7, no. 4, pp. 1060–1064, Apr. 2019, doi: 10.26438/ijcse/v7i4.10601064.

[22] A. Krogh and V. Jesper, "Neural network ensembles, cross validation, and active learning," in *Proc. 7th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 1994, pp. 231–238.

[23] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.

[24] Z. S. Jiang, H. Z. Liu, B. Fu, and Z. H. Wu, "Decomposition theories of generalization error and AUC in ensemble learning with application in weight optimization," *Chin. J. Comput.*, vol. 42, no. 1, pp. 1–15, 2019.

[25] X. Chao, G. Kou, Y. Peng, and A. Fernández, "An efficiency curve for evaluating imbalanced classifiers considering intrinsic data characteristics: Experimental analysis," *Inf. Sci.*, vol. 608, pp. 1131–1156, Aug. 2022, doi: 10.1016/j.ins.2022.06.045.

WANG NING was born in 1982. He received the master's degree from Xiangtan University, in 2017. Currently, he is the Deputy Secretary of the School of Computer and Communication, Hunan University of Engineering. His research interests include algorithm design and analysis and artificial intelligence.

SILIANG CHEN is currently pursuing the degree in data science and big data technology with the Hunan University of Engineering. He have published a SCI and an invention patent during his studies.

SONGYI LEI is currently pursuing the degree in data science and big data technology with the Hunan University of Engineering.

XIONGBIN LIAO is currently pursuing the degree in data science and big data technology with the Hunan University of Engineering.

• • •