Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Union Adaboost Based on Sample Weight Updating Mechanism

Yuting Zhang[1], Libin Xu[2,✉]

[1] *School of Humanities and Design, Chengdu Technological University, Chengdu, Sichuan, 610000, China*

[2] *School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610000, China*

ABSTRACT
Aiming at the limitations of the sample weight updating mechanism of the traditional Adaboost algorithm, the article proposes three improved algorithms based on the joint weight updating mechanism to solve the problems of sample distribution imbalance, etc. The MW_UA algorithm is centered on the updating of the proportion of the sample weight, the OW_UA algorithm realizes the updating of the weight of the sample set based on the classification effect of the initial samples, the MAR_UA algorithm employs sample The MAR_UA algorithm uses the sample Margin to quantify the degree of difficulty of sample classification and then obtain the corresponding sample weights. The performance test experiments and prediction simulation experiments of the improved algorithm are based on the MWSP and Caltech datasets. The experimental results show that the average accuracy and F1 score of MAR_UA algorithm in the two datasets are over 90%, which is the best performance among all the improved algorithms. The algorithm also shows optimal prediction error convergence performance in both datasets, and the training error can be converged to the minimum within 40 times of training. When the algorithm is applied to the simulation experiment of pedestrian recognition, it has the best recognition effect in the sunny environment, with a detection rate of 94.1%. In addition, the error between its predicted and real values of offshore wind speed is no more than 0.2 m/s, and the ERMS and EMA are reduced by 63.52% and 55.5%, respectively, compared with the traditional Adaboost. This study optimizes the weight updating mechanism of the joint Adaboost algorithm using various methods, which can provide new ideas for the optimization research of the weight updating mechanism.

Keywords: Sample weight update, Joint Adaboost algorithm, Margin, Algorithm improvement

1. Introduction

As the development of society enters the digital era, data mining becomes an important means to study big data, and integrated learning plays an important role in data mining [1-2]. In complex situations, it is very difficult to construct a single classifier and there may be unstable outputs, while integrated learning can effectively overcome this drawback [3-4]. AdaBoost (Adaptive boosting), as a

✉ Corresponding author.
E-mail address: XLBchengdu@yeah.net (L. Xu).
Received 01 February 2024; Revised 20 April 2024; Accepted 10 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-392

representative algorithm in integrated learning, only requires the base classifier to be slightly better than random guessing, and this property can well solve the classifier construction problem in complex cases [5]. Due to its excellent performance, the AdaBoost integrated learning algorithm has been widely used in real-world problems such as image acquisition, speech recognition, power demand prediction, seabed classification, indoor localization and face recognition [6-8].

The AdaBoost algorithm possesses the excellent property of integrated learning, i.e., it can overcome the limitation that individual classifiers have insufficient generalization ability on their own and improve the generalization ability of the system [9-10]. As a member of the Boosting family, the AdaBoost algorithm is also a meta-algorithmic framework with no specific algorithm on its own, which is used to obtain a classifier with better performance by combining multiple weak classifiers learned by a specific algorithm [11-13]. Due to the flexibility of the framework, the AdaBoost theory can not only combine with different algorithms, but also integrate other domain theories, so it has a strong compatibility and room for improvement [14-16]. Therefore, AdaBoost theory has become a popular research field in machine learning in the past decade or so, and the research in this field mainly focuses on the principle research and the transformation research. The principle research mainly focuses on the theoretical discussion of "why AdaBoost algorithm is not easy to overfitting" [17]. Adaptive research is mainly based on the problems of the AdaBoost framework itself, such as sensitivity to noise, as well as further improvement of classification accuracy and extension of multi-classification algorithms [18-19]. However, due to the singularity of its own framework, the traditional AdaBoost still needs to further improve its comprehensive classification ability and overcome its own problems, so how to design a more robust AdaBoost with higher classification accuracy and wider applicability is still a relevant topic [20-23].

In this paper, based on the brand-new updating mechanism of sample weights, three different perspectives of weight updating ideas are proposed. First, the MW_UA algorithm can adjust the distribution structure of the sample training, using segmented weighting method to determine the proportion of sample weights. Second, the OW_UA algorithm weight update originates from the original sample distribution, calculates the weight of its midbase classifier, and avoids excessive sample noise weight. Third, the training sample interval in the MAR_UA algorithm can map the correct degree of sample classification to the range of sample weight changes to realize the accurate division of training samples. In addition, in order to explore the prediction performance of the three joint Adaboost algorithms based on the sample weight updating mechanism, performance comparison experiments are conducted on the coastal wind farm dataset and pedestrian detection dataset collected in this paper. And the better improved algorithms are further validated by prediction simulation.

2. Improvement of joint Adaboost algorithm based on sample weighting mechanism

2.1. Key to Improvement

The core of the Adaboost algorithm [24] is to change the distribution of training samples by giving different weights to each sample in order to generate different base classifiers, and there are several key points of concern in the process of sample weight update:

1) During each iteration, the multiplicity of the sample weight update is directly related to the classification error of the previous round of base classifiers, which may result in the base classifiers generated in the future only focusing on the samples that are currently misclassified.

2) As the number of iterations increases, the sample distribution gradually "moves away" from the original sample distribution. This may result in the individual classifiers learned from this sample distribution only adapting to the current distribution.

3) The classification effect of the individual classifiers on the data is used as the basis for updating the

weights of the samples, which ignores the degree to which different data are correctly/incorrectly categorized, resulting in inaccurate weight updating results.

2.2. Improvement based on sample weight update ratio

This section improves the Adaboost algorithm based on the problems of sample weight updating in the Adaboost algorithm. It is determined by the following two conditions:

1) Adjusting the structure of the training sample distribution in the $T+1$-round. Let the classification error of the base classifiers in round T be ε_T, then the ratio of training samples distribution in round T is $x_i^{T,0}:x_i^{T,1}=\varepsilon_T:1-\varepsilon_T$, and the ratio of samples $x_i^{T,0}$ and $x_i^{T,1}$ in round $T+1$ is 1:1 through the derivation of the classical Adaboost algorithm, according to the relationship between the two, we think that the ratio of samples $x_i^{T,0}$ and $x_i^{T,1}$ in the improved model should be less than 1:1, and greater than the ratio of the two samples in the T th round of iteration, $x_i^{T,0}$, and $x_i^{T,1}$, which is $\varepsilon_T:1-\varepsilon_T$. The weights of the samples in the new round can not only "penalize" the misclassified samples, but also reduce the possibility that the correctly classified samples in the previous round will be misclassified again.

2) Adjust the proportion of weight expansion in the next round for samples with judgment errors in round T. The weights of the samples $x_i^{T,0}$ misclassified by the base classifiers in classical Adaboost will be enlarged by a factor of $1/(2\varepsilon_T)$ in the next round of iterations, and this weight amplification exponentially decreases with the training error ε_T and increases. Therefore, the sample weight enlargement multiple of the next round of $x_i^{T,0}$ in the improved sample updating method should be less than $1/(2\varepsilon_T)$, i.e., the corresponding curves should be in the lower region of the above curve.

This section proposes a model improvement scheme based on the proportion of sample weights, i.e., the MW_UA (Multiple Weight Union-Adaboost) algorithm.

The error rate θ is used as the threshold, combined with $1/\sqrt{\varepsilon_T}$ as the sample weight expansion ratio. The algorithm uses a segmented weighting method with θ as the cutoff point, where $\theta\le 0.25$, and the original weighting formula is used when the training error of the Tth iteration $\varepsilon_T>\theta$, i.e.:

$$w_{T+1,i}\leftarrow w_{T,i}\exp\left(-0.5\ln\left(\frac{1-\varepsilon_T}{\varepsilon_T}\right)y_i h_T(x_i)\right) \tag{1}$$

When the training error $\varepsilon_T<\theta$ of the T round of iteration: combined with the requirements in condition 2, it is necessary to find a sample weight expansion ratio $<1/(2\varepsilon_T)$, and based on the calculation of several analyses, $1/\sqrt{\varepsilon_T}$ for the sample weight expansion ratio, and get the relationship curve between $1/\sqrt{\varepsilon_T}$ and ε_T.

When the training error $\varepsilon_T<\theta$ in the first T round of iteration, $1/\sqrt{\varepsilon_T}$ is the proportion of the sample weight enlargement, and set the samples that have been classified wrongly in the last round $x_i^{T,0}$, and the samples that have been judged correctly as $x_i^{T,1}$, then the sample weight of $x_i^{T,0}$ in round $T+1$ is expanded by $1/\sqrt{\varepsilon_T}$ times to $\varepsilon_T/\sqrt{\varepsilon_T}$, and the sample weight of $x_i^{T,1}$ is $1-\varepsilon_T=\sqrt{\varepsilon_T}-\varepsilon_T/\sqrt{\varepsilon_T}$. The ratio of the two in the $T+1$ round of iterations is:

$$\begin{aligned} x_i^{T,0}:x_i^{T,1} &= \frac{\varepsilon_T}{\sqrt{\varepsilon_T}}:\frac{\sqrt{\varepsilon_T}-\varepsilon_T}{\sqrt{\varepsilon_T}} \\ &= \frac{\varepsilon_T}{\sqrt{\varepsilon_T}-\varepsilon_T} \end{aligned} \tag{2}$$

Let $e^{\alpha'}$ be the sample update mechanism of the improved model with $x_i^{T,0}:x_i^{T,1}=\varepsilon_T:1-\varepsilon_T$ in round T, then the updated ratio of sample weights for the two is:

$$\begin{aligned} x_i^{T,0} : x_i^{T,1} &= \varepsilon_T * e^{\alpha'} : (1-\varepsilon_T) * e^{-\alpha'} \\ &= \varepsilon_T * e^{2\alpha'} : (1-\varepsilon_T) \\ &= \frac{\varepsilon_T * e^{2\alpha'}}{1-\varepsilon_T} \end{aligned} \tag{3}$$

From the analysis, Eq. (2) is equal to Eq. (3), i.e:

$$\frac{\varepsilon_T}{\sqrt{\varepsilon_T} - \varepsilon_T} = \frac{\varepsilon_T * e^{2\alpha'}}{1-\varepsilon_T} \tag{4}$$

Calculations are obtained:

$$\alpha' = 0.5 \log \frac{1-\varepsilon_T}{\sqrt{\varepsilon_T - \varepsilon_T}} \tag{5}$$

Where, after analyzing, we get $1-\varepsilon_T / \sqrt{\varepsilon_T} - \varepsilon_T > 0$, when $\varepsilon_T = 0$, that is, the iteration is stopped, and no more new sample weights are generated. Therefore, when $\varepsilon_T \le \theta$ the improved Union-Adaboost model samples are updated in the following way:

$$\begin{aligned} w_{T+1,i} &\leftarrow w_{T,i} \exp\left(-\alpha' y_i h_T(x_i)\right) \\ w_{T+1,i} &\leftarrow w_{T,i} \exp\left(-0.5 \log \frac{1-\varepsilon_T}{\sqrt{\varepsilon T} - \varepsilon_T} y_i h_T(x_i)\right) \end{aligned} \tag{6}$$

Taken together, the weight update mechanism of the improved Union-Adaboost model is:

$$D_{T+1}(x) = \frac{D_T(x)}{Z_T} \times \begin{cases} \exp\left(-0.5 \log\left(\frac{1-\varepsilon_T}{\varepsilon_t}\right) y_i h_T(x_i)\right) & if\ \varepsilon_T > \theta \\ \exp\left(-0.5 \log\left(\frac{1-\varepsilon_T}{\sqrt{\varepsilon_T} - \varepsilon_T}\right) y_i h_T(x_i)\right) & if\ \varepsilon_T \le \theta \end{cases} \tag{7}$$

2.3. Sample weight update method based on initial distribution

In order to overcome the problem that the sample distribution gradually deviates from the original sample space during the iteration process of the classical Adaboost algorithm, this paper proposes the Origin Weight Union-Adaboost (OW_UA) algorithm by using the original sample distribution to generate a new sample distribution.

Let $h_t(x)$ be the base classifier generated by the t iteration, and ε_t be the classification error rate of $h_t(x)$ under the D_t distribution of the weighted training dataset, and ε_t' is $h_t(x)$ the classification error rate under the initial training data distribution D, defined by the following formula:

$$\begin{aligned} \varepsilon_t &= \sum_{i=1}^{N} P\left(h_t(x_{ti}) \ne y_{ti}\right) = \sum_{i=1}^{N} w_{ti} I\left(h_t(x_{ti}) \ne y_{ti}\right) \\ \varepsilon_t' &= \sum_{i=1}^{N} P\left(\left(h_t(x_i) \ne y_i\right)\right) = \sum_{i=1}^{N} \frac{1}{n} I\left(h_t(x_i) \ne y_i\right) \end{aligned} \tag{8}$$

Let α_t' be the base classifier weights in the improved Adaboost algorithm:

$$\alpha_t' = 0.5 \ln\left(\frac{1-\varepsilon_t'}{\varepsilon_t}\right) \tag{9}$$

$w_{t+1,i}'$ is the sample weight in the $t+1$ round of iteration, then the improved weight update method

is:

$$w'_{t+1,i} = w'_{1,i} \exp\left(-\alpha'_t y_i h_t\left(x_i\right)\right)$$
$$D_{t+1} = \left(w'_{t+1,1}, \cdots, w'_{t+1,i}, \cdots, w'_{t+1,N}\right) \tag{10}$$

The final integrated classifier is:

$$H'(x) = sign\left(\sum_{t=1}^{T} \alpha' h_t(x)_t\right) \tag{11}$$

This approach redefines the training error ε_t', which to some extent reflects the adaptation of the base classifier h_t to the data under the real sample space.

Figure 1 shows the structure of the weight update method based on the initial distribution. Each round of iterative distribution update is based on the original distribution, which not only avoids some difficult to delineate sample points or noisy sample weights overly squeezing the weights of other training data in the iterative process, but also reduces computational expenses and memory required for storing the data in the context of data-intensive real-world, where a single training step is more expensive to compute.

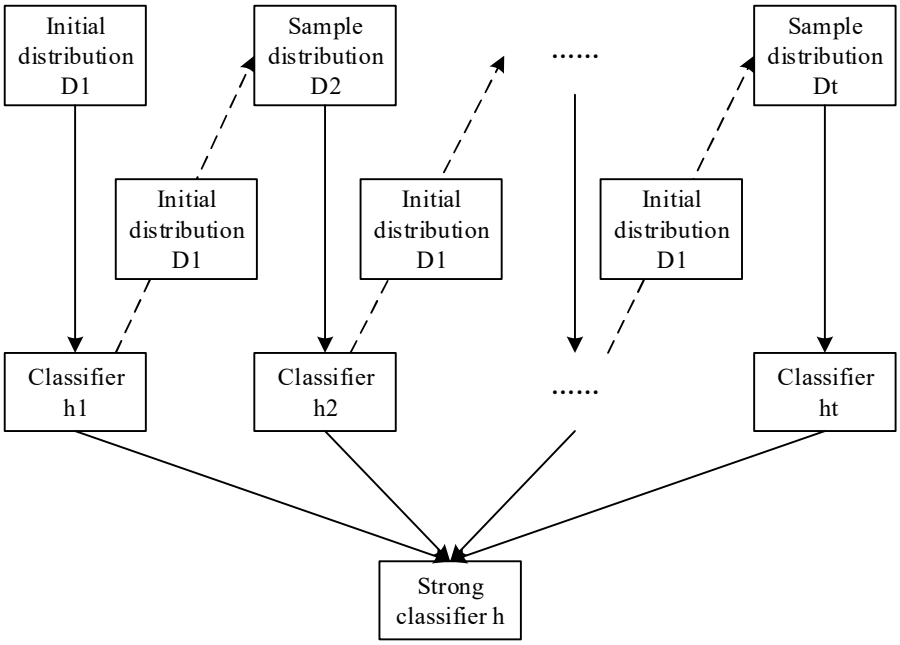

Fig. 1. Based on the original distribution of the weight update pattern diagram

2.4. Improvement in the degree of updating of sample weights under Margin theory

Margin [25] explains the antioverfitting phenomenon [26] by defining intervals, arguing that although the training error reaches zero, Adaboost's intervals will increase with the number of iterations.

The interval of the ith training sample $\left(x_i, y_i\right)$ for h_j can be defined as:

$$m\arg in\left(x_i\right) = \frac{y_i \sum_{j=1}^{M} a_j h_j\left(x_i\right)}{\sum_{j=1}^{M} \alpha_j} \tag{12}$$

Where α_j is the weight of the base classifier h_j.

The MarginWeight Union-Adaboost (MAR_UA) algorithm is proposed by taking the confidence level of the base classifier on the classification result of the samples, and the interval of the samples as an index of the weight updating multiplier.

Let the interval of the ith training sample (x_i, y_i) for the base classifier h_j be:

$$m\arg in(x_i) = \frac{y_i \sum_{j=1}^{M} \alpha_j h_j(x_i)}{\sum_{j=1}^{M} \alpha_j} = \frac{1}{3} y_i \sum_{j=1}^{M} h_j(x_i) \tag{13}$$

where α_j is all 1/3 here since the weights of the internal models of the joint classifiers in Union-Adaboost are the same. The original samples are updated in a manner determined by the following equation:

$$w_{t+1,i} \leftarrow w_{t,i} \exp\left(-0.5 \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) y_i h_t(x_i)\right) \tag{14}$$

The interval of samples is used in the improved model instead of $y_i h_t(x_i)$ in the original equation:

$$w'_{t+1,i} \leftarrow w'_{t,i} \exp\left(-0.5 \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) * m\arg in(x_i)\right) \tag{15}$$

$$w'_{t+1,i} \leftarrow w'_{t,i} \exp\left(-0.5 \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) * \frac{1}{3} y_i \sum_{j=1}^{M} h_j(x_i)\right) \tag{16}$$

Where M is the number of classifiers within the Union-Adaboost base classifier, and $w'_{t,i}$ is the weight of the x_i th t round of iteration. If the member classifiers (KNN, BT, NB) all classify this sample correctly (wrongly), then $m\arg in(x_i) = 1(-1)$, and it is considered that the member classifiers are very sure about the classification result of this sample. If only one of the member classifiers (KNN, BT, NB) classifies incorrectly, $m\arg in(x_i) = 1/3$, it is considered that the base classifiers classify the sample correctly for that sample, but the confidence that the classification result is correct is smaller, and vice versa.

The training sample interval $\frac{1}{3} y_i \sum_{j=1}^{M} h_j(x_i)$ maps the degree of correctness of the classification results of all samples between $[-1,1]$, and the corresponding range of variations in the weights of the samples is $\left[e^{-\alpha_t}, e^{\alpha_t}\right]$, whereas the traditional scaling varies as a binary $\left\{e^{-\alpha_t}, e^{\alpha_t}\right\}$, and so this approach more accurately reflects the degree of difficulty in classifying the training sample .

3. Modeling and experimental analysis

In this section, the improved Adaboost algorithm is utilized to construct a prediction model, brought into different datasets, and the performance is compared with different prediction algorithm models. The evaluation metrics and performance comparisons are carefully elucidated and the results of different experiments are analyzed in detail from which conclusions are drawn.

3.1. Experimental environment and evaluation indicators

3.1.1. Experimental environment. Operating system: 64-bit operating system for Winows 10.
Hardware configuration of the computer: 4G RAM, 120G hard disk.
Core processor: Intel(R) Core(TM) i7.
Graphics card: NVIDIA Ge Force RTX 2060.

3.1.2. Benchmarking model. In order to verify the effectiveness of the proposed improved Adaboost algorithm for prediction, this paper slices the real publicly available dataset, evaluates the performance using multiple datasets, and uses the traditional Adaboost algorithm, LSTM algorithm, GNN algorithm with CNN algorithm as the benchmark model for comparison.

1) Traditional Adaboost algorithm model: in the improved Adaboost algorithm, the core idea of the algorithm has not changed, it is to balance the performance and generalization ability of the classifiers by adjusting the weight of each weak classifier, which makes the final classifiers have a better generalization ability and stronger robustness.
2) LSTM algorithm model: the LSTM model is a commonly used recurrent neural network model, which can deal with sequential data and has better time series characteristics.
3) GNN algorithm model: GNN model is a commonly used graph neural network model, which can handle graph data and has better graph structure properties.
4) CNN algorithm model: CNN model is a commonly used convolutional neural network model, which can deal with grid-like data, and its core idea is to efficiently extract spatial features through local sensing and parameter sharing.

3.1.3. Data sets. In order to verify the prediction diversity of the above proposed joint Adaboost model based on the sample weight updating mechanism, this study collects the coastal wind field A site data (notated as MWSP dataset) provided by the Marine Science Data Center for offshore wind speed prediction example analysis. This dataset contains wind speed data from 1996 to 2025, covering wind speed in the latitude and longitude direction and site latitude and longitude information. When making predictions, 20 consecutive data are used to predict the next set of data. Pedestrian data recorded by vehicles were also collected from the official website of Pedestrian Detection Dataset Caltech, which covers about 250,000 frames labeled with pedestrians, bicyclists, and wheelchair users, etc. The data are in ASCII format and are used to predict the next set of data. The data is in ASCII format and is decoded, format checked, code converted, normalized, automatically quality controlled, visually checked and calibrated to form a standardized dataset.

3.1.4. Evaluation indicators. This experiment is a binary classification experiment, not a regression experiment. In order to evaluate the performance of the model, four classification experiment evaluation metrics were used in this study: accuracy, recall, precision, and F1 score. The evaluation metrics were determined by true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. These four metrics are commonly used to evaluate the performance of the model with the following attributes:
1) Accuracy rate: in traffic congestion prediction, the accuracy rate can indicate the ratio of the number of congested situations (or non-congested situations) correctly predicted by the classifier to the total number of samples:

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

2) Recall: Recall can be expressed as the ratio of the number of correct predictions made by the dispenser to the number of actual cases:

$$recall = \frac{TP}{TP + FN} \tag{18}$$

3) Accuracy: Accuracy indicates the number of classifier predictions that are true, as a proportion of the number of all predicted scenarios:

$$precision = \frac{TP}{TP + FP} \tag{19}$$

4) F1 score: The F1 score can combine the advantages and disadvantages of precision and recall:

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{20}$$

3.2. Performance Study of Improved Joint Adaboost Algorithm

3.2.1. Comparative analysis of prediction experiments. In this section of experiments, the prediction performance of the three joint Adaboost algorithms proposed in this paper, i.e., MW_UA algorithm, OW_UA algorithm, and MAR_UA algorithm models based on the sample weight updating mechanism, is mainly analyzed and compared with that of the traditional Adaboost model. The datasets selected for the prediction performance pairs are the Caltech and MWSP datasets collected above. The evaluation metrics are accuracy, recall, precision and F1 score.

The prediction performance evaluation results of different models on the two datasets are obtained as shown in Table 1. As can be seen from the table, the traditional Adaboost model has the worst prediction performance in both datasets, with an average prediction accuracy and F1 score of 54.305% and 60.255%, respectively, whereas for the three improved Adaboost algorithms proposed in this paper, their prediction performance is increased to different degrees. Among them, the MAR_UA model based on Margin theory shows the best prediction performance, with an average accuracy and F1 score of 88.035% and 90.035% in the two datasets, which are 33.73 and 29.78 percentage points higher than the traditional Adaboost algorithm, respectively. It means that the sample weight updating mechanism based on Margin theory can more accurately respond to the degree of differentiation of the training data.

Table 1. The predictive ability of different models in two data sets can be evaluated

Date set	Indicators	Models			
		Adaboost	MW_UA	OW_UA	MAR_UA
Caltech	Accuracy(%)	55.57	72.66	76.57	91.4
	Recall(%)	61.64	78.73	80.2	89.46
	Precision(%)	60.46	69.45	80.49	90.85
	F1(%)	61.04	73.80	80.34	90.15
MWSP	Accuracy(%)	53.04	69.08	73.56	84.67
	Recall(%)	62.92	76.64	80.27	90.26
	Precision(%)	56.37	78.23	82.31	89.58
	F1(%)	59.47	77.43	81.28	89.92

At the end of this section of the experiment, the performance capability of the improved model is explained from the other side by analyzing the ROC as well as AUC metrics of the different models in the two datasets. The results of ROC as well as AUC metrics for each model are shown in Figure 2. In the figure, from the ROC index, the four curves have a similar trend, but the area enclosed has a great difference. The above conclusion can also be drawn from the AUC metrics, the average AUC values of the traditional Adaboost algorithm, MW_UA algorithm, OW_UA algorithm, and MAR_UA algorithm in the two datasets are 0.599, 0.7035, 0.8025, and 0.949, respectively. From this comprehensive comparative analysis of this experiment, it is found that the overall performance of the MAR_UA algorithm is the best among the improved models, which is much better than the performance of the traditional Adaboost algorithm.

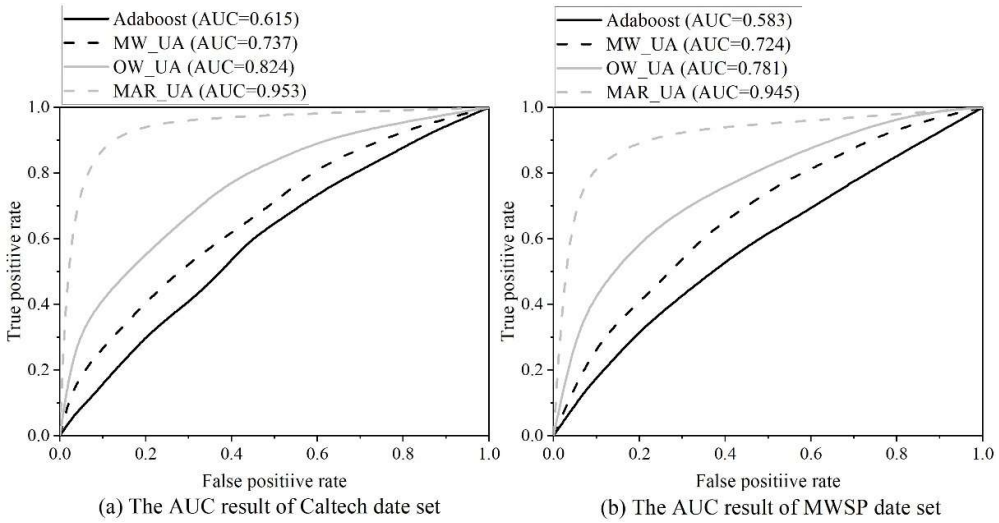

Fig. 2. ROC and AUC index results of each model

3.2.2. Convergence Stability Performance Analysis. In order to better validate the stability of the joint Adaboost algorithm based on the sample weight updating mechanism, the 10-fold cross-validation is used in Experiment 4 of this section. Under different number of iterations, the average of the errors of the traditional Adaboost algorithm and the three improved algorithms, MW_UA algorithm, OW_UA algorithm, and MAR_UA algorithm, are compared for 10 times of training respectively in two datasets.

The experimental results comparing the convergence speed of the four algorithms are shown in Figure 3. On the Caltech dataset, the traditional Adaboost algorithm error converges to 0.43 after about 140 iterations, while the MW_UA algorithm, the OW_UA algorithm, and the MAR_UA algorithm converge by 100 iterations, and the MAR_UA algorithm converges to the lowest error in about 37

iterations. In the MWSP dataset, the MW_UA algorithm, the OW_UA algorithm, and the MAR_UA algorithm were stabilized at 0.62, 0.49, and 0.21 for 86, 80, and 33 iterations, respectively. The traditional Adaboost algorithm requires 180 iterations and the error converges to 0.94. In summary, the traditional Adaboost algorithm converges much slower than the improved algorithm, and the convergence value of the prediction error is much higher than that of the improved algorithm. Among the improved algorithms, the MAR_UA algorithm shows the best convergence performance in both datasets, which will be utilized in the following for the offshore wind speed prediction as well as the pedestrian identification simulation experiments.

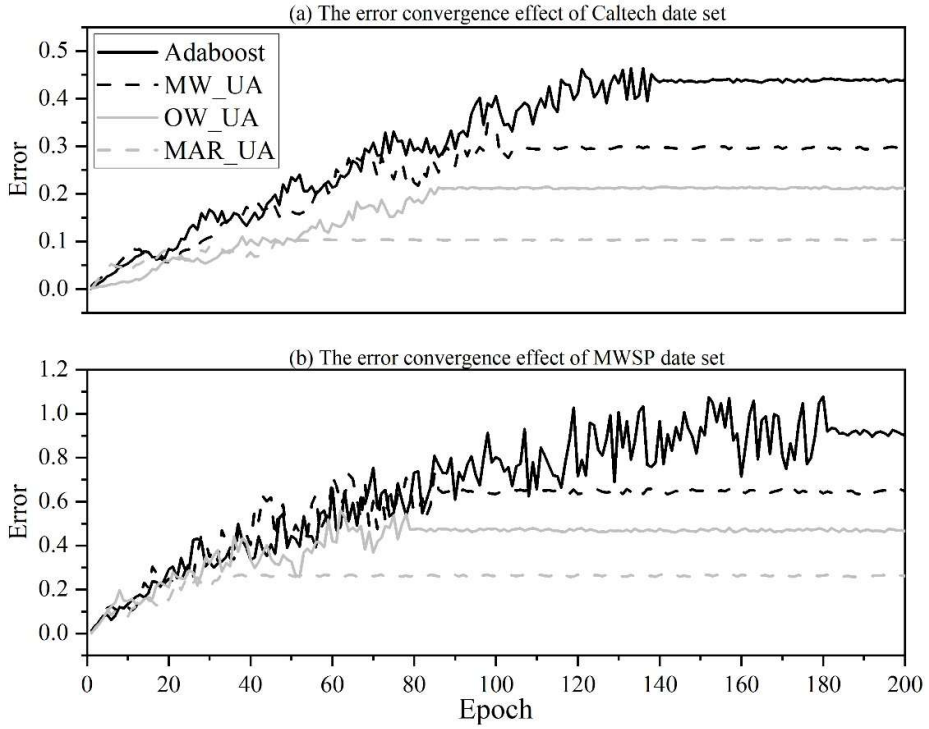

Fig. 3. The convergence velocity of the four algorithms is compared

3.3. MAT_UA algorithm prediction simulation validation

From the performance comparison experiments above, it can be seen that the improved MAT_UA algorithm proposed in this paper has the best prediction performance, so in this section, the improved algorithm is used with the commonly used prediction baseline model to carry out simulation experiments on pedestrian identification and offshore wind speed prediction, respectively, in order to further validate the improved effect of the algorithm.

3.3.1. Pedestrian recognition applications. In this section, the pedestrian recognition performance tests were conducted on multiple video image sequences from the Caltech test set, including real scenes with bright light, low light, sunny, rainy, snowy, and windy conditions. For the video sequence images, the following three performance metrics were used to measure the performance of the detection system.

False Alarm Rate: the ratio of the number of non-pedestrian windows with false detections in all frames to the sum of all detected windows.

Missed Detection Rate: the ratio of the number of pedestrians detected in all frames to the total number of pedestrians detected in all frames.

Detection rate: the ratio of the number of pedestrians detected in all frames to the sum of all pedestrians.

The system detection results obtained for different scenarios are shown in Fig. 4. Figures (a)~(b) represent the statistical results of false alarm rate, missed detection rate, detection rate and average processing speed in turn. It can be seen that the false alarm rate, omission rate, and detection rate under strong light conditions are 0.8%, 6.1%, and 93.2%, respectively. In strong light the illumination is better, the image obtained is clearer and it is easy to distinguish the background from the pedestrians. However, due to the strong light, the image increases the noise, which improves the detection rate of omission and false alarm and affects the detection effect. When the light is weak, it is difficult to separate the pedestrians from the background in the image due to insufficient exposure. However, this system is able to detect pedestrians as long as it can extract the pedestrian features matched by the trainer. From Fig. (c), the detection rate is still able to reach more than 90%, which is still able to achieve relatively good detection results. Conditions such as light and visibility in the sunny environment perform better, so the system shows the lowest false alarm rate as well as omission rate, while having the highest detection rates of 0.3%, 4.4%, and 94.1%, respectively. The detection results of rainy, snowy and windy days are similar to those of the low-light scenes, which are interfered by natural factors such as rain, which affects the system's performance on recognition, but the detection rate is still more than 90%. For the average processing time, the system has the fastest recognition speed for sunny day scenes, with an average processing time of 4 frames/second. Overall the Adaboost algorithm, which is improved based on the sample weight updating mechanism in this paper, has good detection results when applied to pedestrian recognition.

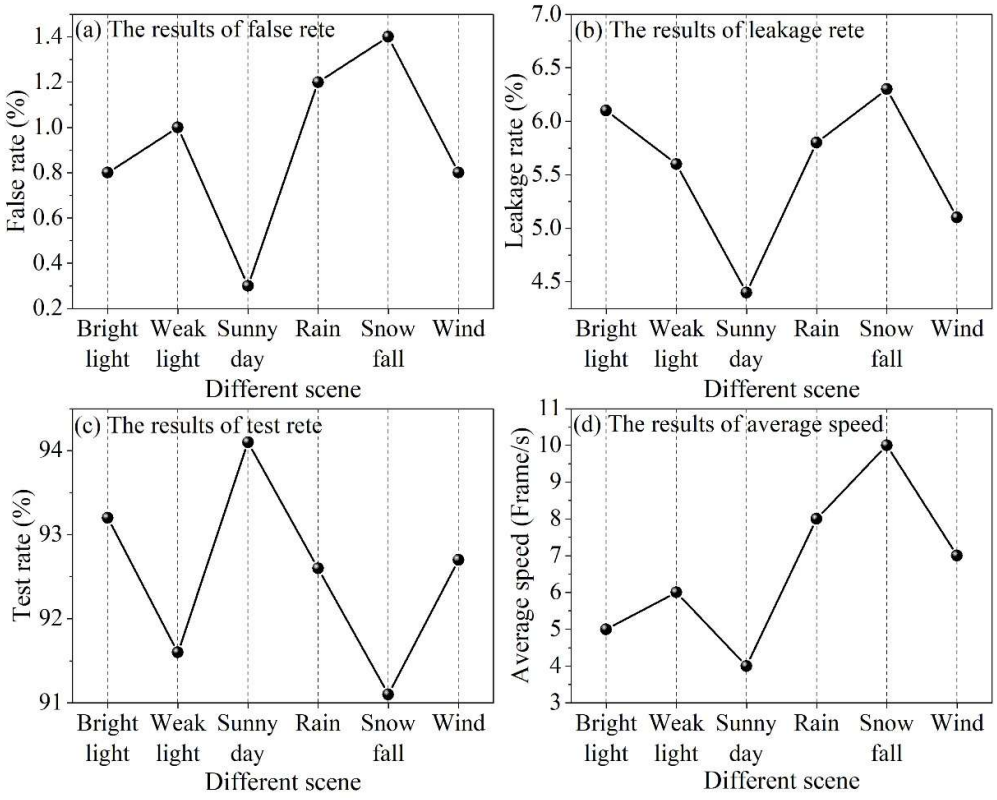

Fig. 4. System test results under different scenarios

In order to further highlight the recognition performance of this paper's algorithm, it is compared with the classical algorithm. Pedestrian recognition evaluation indexes are chosen as Reasonable, All, large, near, medium and far, which represent a statistical rule, and the smaller the indexes are, the better the detection performance of the system.

Figure 5 shows the comparison results between the improved Adaboost algorithm in this paper and the classical algorithm above. The data used in the comparison algorithms are from the Caltech

pedestrian detection dataset. The results show that the measured values of the improved algorithm in this paper on Reasonable, All, large, near, medium, and far metrics are 9.72%, 55.52%, 1.13%, 4.26%, 47.84%, and 72.49%, respectively, which are the lowest performances among all models. In addition, the improved algorithm in this paper has higher accuracy for pedestrian targets with larger size compared to other algorithms. And for weak and small pedestrian targets, the improved pedestrian detection system in this paper has significantly higher accuracy than several other algorithms in terms of detection accuracy. In addition, it should be noted that this paper improves the detection accuracy of the original algorithm through the sample weight updating mechanism. It is effective for a variety of pedestrian detection systems. When the detection accuracy of the base system is higher, the detection accuracy of the improved optimized system is higher.

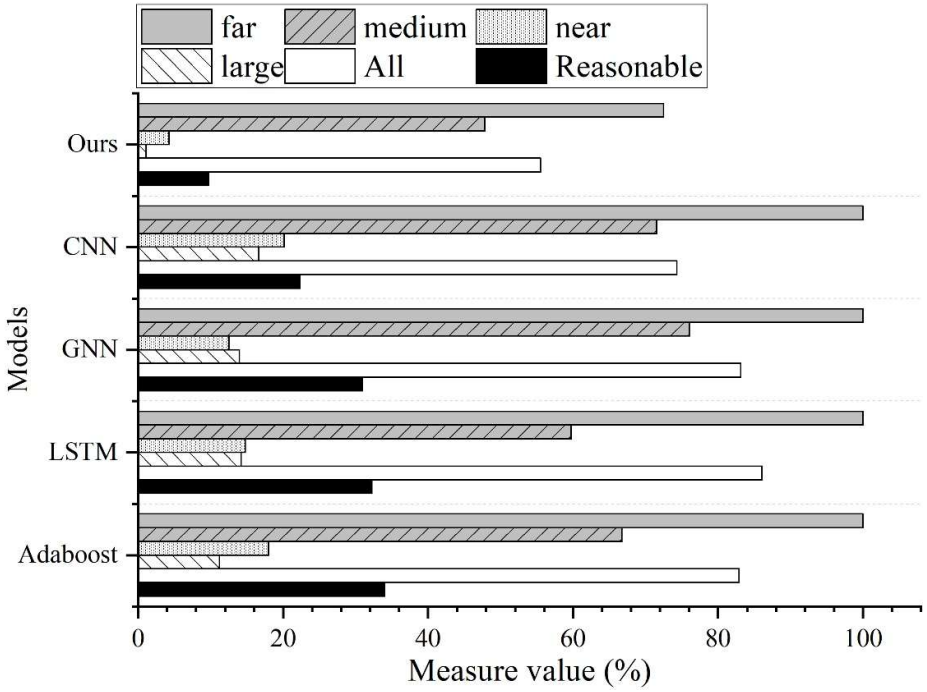

Fig. 5. The improved Adaboost algorithm is compared with the classical algorithm

3.3.2. Forecasts of offshore wind speeds. In this study, 9522 sets of data from the NMSP test dataset for Site A from 0:00 on October 1, 2023 to 0:00 on November 30, 2024, with a data interval of 1 h. Empirically, the first 70% of the sets of data are selected as the training set, and the last 30% of the sets of data are used as the test set, and 20 sets of data are randomly selected as the validation set.

Again, the commonly used prediction benchmark models mentioned above are the traditional Ababoost model, LSTM model, GNN model, and CNN model as a control. The wind speed prediction results of different models in longitude and latitude directions at sea are shown in Fig. 6 and Fig. 7, respectively. From the two figures, it can be seen that the predicted values using the joint Adaboost prediction model based on the sample weight updating mechanism in this paper are closer to the real values, and the prediction trend is smoother. By comparing with the real wind speed, the maximum error of this paper's prediction method in the longitude direction is about 0.2 m/s, and the maximum error in the latitude direction is about 0.15 m/s. In each group of predictions, the prediction value derived from the comparison model algorithm has a larger relative error, which reflects the superior performance of this paper's algorithm in the prediction process.

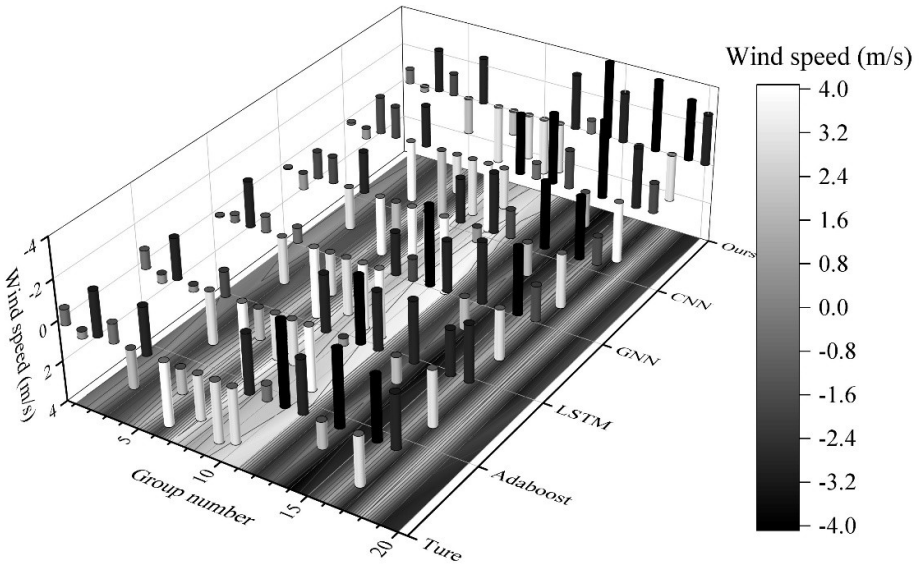

Fig. 6. The result of the wind velocity prediction in the direction of orb

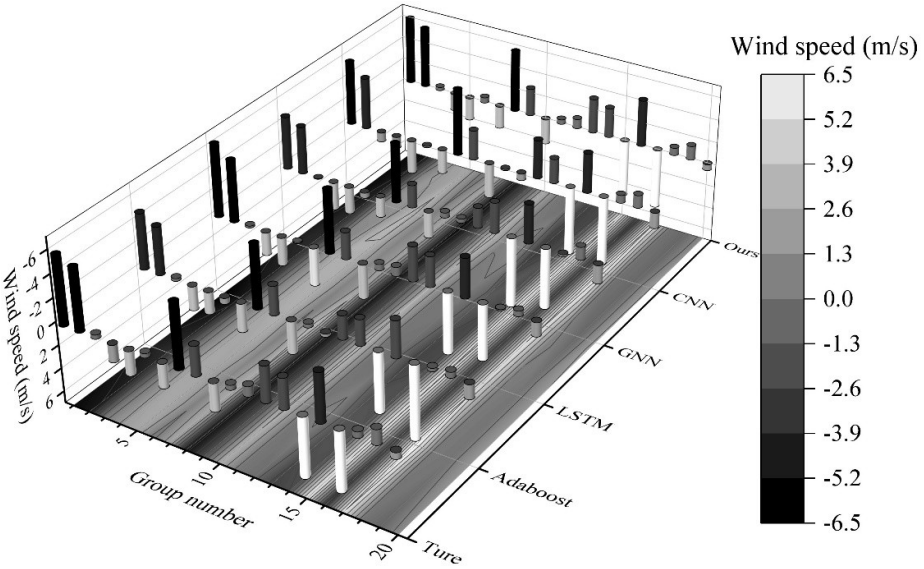

Fig. 7. The result of the wind velocity prediction in the direction of latitude

In this subsection of the example analysis, root mean square error (ERMS), absolute mean error (EMA) and R^2 are used as the evaluation indexes of the algorithm's effectiveness in order to get a better and more comprehensive evaluation of the effect. Table 2 shows the comprehensive model evaluation results of different models for offshore wind speed prediction. From the table, it can be seen that when analyzed in terms of wind speed prediction error and prediction uncertainty in longitude or latitude direction, the ERMS and EMA of the Adaboost improved algorithm in this paper have a greater reduction than that of the traditional Adaboost algorithm. The ERMS value is reduced by about 63.52% and the EMA is reduced by about 55.5%. For the indicator R^2, the R^2 value of this paper's method is improved by about 64.13% compared with the traditional Adaboost algorithm. The improved algorithm in this paper makes good use of the global optimality seeking property and avoids the problem of anomalous predicted values at the time of prediction.

Table 2. The results of each model evaluation index

Models	ERNS	ERNS	EMA	EMA	R^2	R^2

	X	Y	X	Y	X	Y
Adaboost	1.152	1.376	0.913	0.942	0.441	0.502
LSTM	0.625	0.781	0.664	0.681	0.553	0.528
GNN	0.979	0.956	0.751	0.693	0.584	0.608
CNN	0.614	0.663	0.784	0.806	0.642	0.617
Ours	0.443	0.475	0.406	0.418	0.763	0.779

4. Conclusion

In this study, according to the sample weight updating mechanism, the joint Adaboost algorithm improvement based on three aspects, namely, weight updating ratio, initial distribution, and degree of weight updating, is realized. Firstly, using the error rate as the threshold value and segmental weighting, the samples are trained with multiple rounds of iterative errors to obtain the optimal sample weights. The initial distribution is adopted as the basis for updating the sample distribution in each iteration, and the weights of the initial dataset are updated according to the classification effect of the base classifier on the initial sample set. Further, Margin theory is introduced to measure the difficulty of different samples classification and generate the appropriate weight size for different samples. The improved model with better performance is screened through comparative experiments, and then further prediction experiments are designed to verify its application effect.

1) The MAR_UA algorithm has the best prediction performance, and its average prediction accuracy and F1 score on the MWSP and Caltech datasets are increased by 36.525% and 29.78%, respectively, compared with the traditional Adaboost algorithm.

2) The algorithm has better convergence stability, with the prediction error converging to 0.21 in the MWSP dataset after 33 training sessions.

3) The algorithm is more effective in pedestrian recognition applications in different situations, and it has the lowest measured values of indicators such as far on the Caltech dataset, i.e., it shows high pedestrian detection accuracy.

4) The offshore wind speed prediction results show that the maximum error between the predicted and true wind speed values of the MAR_UA algorithm in the direction of longitude and latitude is 0.2m/s and 0.1m/s, respectively, which is significantly lower than the comparison method.

References

[1] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14, 241-258.

[2] Zhang, Y., Liu, J., & Shen, W. (2022). A review of ensemble learning algorithms used in remote sensing applications. Applied Sciences, 12(17), 8654.

[3] Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. Neural Computing and Applications, 32(12), 8675-8690.

[4] Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. Information Fusion, 37, 132-156.

[5] Wyner, A. J., Olson, M., Bleich, J., & Mease, D. (2017). Explaining the success of adaboost and random forests as interpolating classifiers. Journal of Machine Learning Research, 18(48), 1-33.

[6] Shahraki, A., Abbasi, M., & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. Engineering Applications of Artificial Intelligence, 94, 103770.

[7] Liu, H., Tian, H. Q., Li, Y. F., & Zhang, L. (2015). Comparison of four Adaboost algorithm based artificial

neural networks in wind speed predictions. Energy Conversion and Management, 92, 67-81.

[8] Ding, Y., Zhu, H., Chen, R., & Li, R. (2022). An efficient AdaBoost algorithm with the multiple thresholds classification. Applied sciences, 12(12), 5872.

[9] Wang, W., & Sun, D. (2021). The improved AdaBoost algorithms for imbalanced data classification. Information Sciences, 563, 358-374.

[10] Wang, F., Li, Z., He, F., Wang, R., Yu, W., & Nie, F. (2019). Feature learning viewpoint of AdaBoost and a new algorithm. IEEE Access, 7, 149890-149899.

[11] Hornyák, O., & Iantovics, L. B. (2023). AdaBoost algorithm could lead to weak results for data with certain characteristics. Mathematics, 11(8), 1801.

[12] Sevinç, E. (2022). An empowered AdaBoost algorithm implementation: A COVID-19 dataset study. Computers & Industrial Engineering, 165, 107912.

[13] Chengsheng, T., Huacheng, L., & Bing, X. (2017). AdaBoost typical Algorithm and its application research. In MATEC Web of Conferences (Vol. 139, p. 00222). EDP Sciences.

[14] Shan, W., Li, D., Liu, S., Song, M., Xiao, S., & Zhang, H. (2024). A random feature mapping method based on the AdaBoost algorithm and results fusion for enhancing classification performance. Expert Systems with Applications, 256, 124902.

[15] Sun, B., Chen, S., Wang, J., & Chen, H. (2016). A robust multi-class AdaBoost algorithm for mislabeled noisy data. Knowledge-Based Systems, 102, 87-102.

[16] Liu, H., Zhang, X., & Zhang, X. (2019). PwAdaBoost: Possible world based AdaBoost algorithm for classifying uncertain data. Knowledge-Based Systems, 186, 104930.

[17] Piao, C., Wang, N., & Yuan, C. (2023). Rebalance Weights AdaBoost - SVM Model for Imbalanced Data. Computational Intelligence and Neuroscience, 2023(1), 4860536.

[18] Wang, Y., & Feng, L. (2020). Improved Adaboost algorithm for classification based on noise confidence degree and weighted feature selection. IEEE Access, 8, 153011-153026.

[19] Huang, X., Li, Z., Jin, Y., & Zhang, W. (2022). Fair-AdaBoost: Extending AdaBoost method to achieve fair classification. Expert Systems with Applications, 202, 117240.

[20] Bahad, P., & Saxena, P. (2020). Study of adaboost and gradient boosting algorithms for predictive analytics. In International Conference on Intelligent Computing and Smart Communication 2019: Proceedings of ICSC 2019 (pp. 235-244). Springer Singapore.

[21] Lee, W., Jun, C. H., & Lee, J. S. (2017). Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. Information Sciences, 381, 92-103.

[22] Jiang, X., Xu, Y., Ke, W., Zhang, Y., Zhu, Q. X., & He, Y. L. (2022). An imbalanced multifault diagnosis method based on bias weights AdaBoost. IEEE Transactions on Instrumentation and Measurement, 71, 1-8.

[23] Li, X., & Li, K. (2022). Imbalanced data classification based on improved EIWAPSO-AdaBoost-C ensemble algorithm. Applied Intelligence, 1-26.

[24] Robert Cep,Muniyandy Elangovan,Janjhyam Venkata Naga Ramesh,Mandeep Kaur Chohan & Amit Verma. (2025). Convolutional Fine-Tuned Threshold Adaboost approach for effectual content-based image retrieval.. Scientific reports,15(1),9087.

[25] Yijun Sun,Sinisa Todorovic & Jian Li. (2006). Unifying multi-class AdaBoost algorithms with binary base learners under the margin framework. Pattern Recognition Letters,28(5),631-643.

[26] Weiming Shao,Xuemin Tian & Honglong Chen. (2013). Adaptive Anti-Over-Fitting Soft Sensing Method Based on Local Learning. IFAC Proceedings Volumes,46(32),415-420.