Exp-Bagging: A Validation-Based Exponential Weighting Strategy for Decision Tree Ensembles

Wanying Liang, Boyue Wang*
School of Information Science and Technology, Beijing University of Technology, Beijing. 100124, China
E-mail: boyuewang1@outlook.com

Keywords: ensemble learning, bagging, decision trees, adaptive weighting, mean squared error, machine learning, regression analysis

Received: October 29, 2025

Decision tree ensemble methods significantly enhance model generalization by integrating multiple base learners, with wide applications in medical diagnosis and financial risk control. However, traditional Bagging and its variants (e.g., Random Forest, RF) use static equal voting, failing to distinguish contributions of high-accuracy and low-quality subtrees and adapt to concept drift or high-dimensional sparse data. This study aims to develop a dynamic weighted ensemble framework to boost computational efficiency while maintaining classification accuracy. We propose the Exp-Bagging method, which adopts an exponential decay weighting mechanism based on validation set performance. It quantifies subtree classification errors via Mean Squared Error (MSE) and dynamically assigns voting weights—smaller MSE corresponds to higher weight. Theoretically, we analyze weight convergence and its role in optimizing generalization error bounds. Experiments were conducted on Climate Model Simulation Crashes, WDBC, and Diabetes datasets, using Accuracy, Precision, Recall, and F1-score for evaluation, with comparisons to RF and AdaBoost. Results show: Exp-Bagging improved F1-score by 6.5% and Accuracy by 5.3% on WDBC compared to RF; its throughput was 2.75 – 3.42 *times that of traditional RF. This confirms dynamic weighting* ' *s effectiveness in balancing efficiency and accuracy, providing a basis for real-time ensemble learning deployment.*

Povzetek: Predstavljena je nova dinamično utežena ansambelska metoda, ki izboljša točnost in učinkovitost odločanja v primerjavi s klasičnimi pristopi, kot je Random Forest.

1 Introduction

The decision tree ensemble method is widely used in clinical medical diagnosis [1], financial risk prediction [2] and network security detection [3] by combining multiple weak learners to improve the generalisation ability of the model. However, in real life, there is a common problem of concept drift - that is, the underlying distribution of data or the relationship between target variables and characteristics changes unpredictably over time (such as the dynamic evolution of financial fraud patterns over time) [4]. The static weight distribution mechanism of the traditional integration method is difficult to adjust the weight adaptively[5], resulting in a decline in the prediction performance of the model; at the same time, the traditional integration method cannot distinguish the contribution difference between key features and noise in the scenario of high-dimensional sparse data [6], which further reduces the generalisation ability of the model; in addition, the traditional method There is a lack of cross-modal feature alignment mechanism [7] when integrating heterogeneous base learners, resulting in low efficiency of information utilisation. These limitations jointly restrict the application effect of integrated learning in complex reality tasks.

In view of the common shortcomings of the above-mentioned industry application challenges and traditional methods, the following core research issues in this article are used to lead the follow-up research context: How to design a dynamic integration mechanism that takes into account adaptive weight adjustment, which can not only cope with the challenges of high-dimensional sparse data, but also improve the information integration efficiency of heterogeneous-based learners? How to optimise the computing cost and storage overhead of the integrated model under the premise of ensuring classification accuracy and group fairness, so that it can adapt to the deployment needs of real-time systems and resource-limited edge devices? How to systematically prove the comprehensive advantages of the proposed integrated method in terms of performance, efficiency and fairness through theoretical analysis and experimental verification, and provide reliable solutions for complex practical tasks? In order to further disassemble the necessity of research problems, the following will focus on the limitations of existing typical methods for analysis.

Among existing approaches, ordinary decision trees exhibit certain advantages in small-scale data scenarios due to their simple structure [8], high classification accuracy, strong robustness [9], interpretability, and efficient training [10]. However, they are prone to overfitting [11], noise sensitivity, inability to capture complex interactions among input features [12], and lack of adaptive feature importance adjustment, for instance, the ID3 algorithm employs information gain as the splitting criterion during decision tree construction, yet using information gain as the splitting standard introduces bias in attribute selection, potentially reducing classification accuracy [13]. These limitations constrain its applicability. The traditional random forest effectively reduces the model variance through Bootstrap sampling and the characteristic randomness of node splitting. Its majority voting mechanism is easy to implement and supports parallel training, with good scalability [14]. However, there are still two obvious limitations of this method: first, the equal voting mechanism ignores the performance differences between different decision-making trees, resulting in the equal treatment of the contribution of high-precision subtrees and low-quality subtrees to the final decision-making, limiting the further improvement of the overall classification performance [15]; second, the integration of random forests The module is positively correlated with the computing cost. As the number of trees increases, the training time increases significantly, and it is difficult to meet the low-latency needs of real-time systems or resource-limited scenarios. In addition, the model storage overhead also increases linearly with the number of trees, limiting its deployment ability on storage-restricted edge devices [16]. Other integrated methods such as AdaBoost, although they can focus on difficult cases by dynamically adjusting the sample weight to improve performance, their weight update mechanism only aims to minimise the overall classification error [17], and does not explicitly constrain the potential distribution deviation in the data structure. In addition, because the method is relatively sensitive to abnormal values, the iterative process is easy to release the inherent imbalance of big data, resulting in poor performance of the model in group fairness indicators, and it is difficult to achieve a truly effective fair classification [18].

Existing research still faces three core problems to be solved:

First of all, there is a prominent contradiction between model efficiency and accuracy. The traditional integration method represented by random forests usually needs to expand the size of the subtree to improve accuracy, but this will significantly increase the computational complexity and prediction delay, and it is difficult to apply to scenarios with high real-time requirements;

Secondly, the existing weight distribution mechanism is not differentiated enough. The traditional voting method fails to fully consider the performance differences between different decision-making trees, resulting in the equivalent treatment of the contributions of high-confidence subtrees and low-quality subtrees, and the potential of high-quality models has not been fully realised; Finally, traditional methods generally lack dynamic adaptability. In the face of common changes in the real environment such as data distribution drift and conceptual drift, such methods are difficult to adapt to adjust the integration strategy, which limits their applicability and robustness in complex dynamic scenarios. The above research gaps seriously restrict the actual deployment effect of integrated learning methods in real-time systems and dynamic environments.

In response to the above problems, this paper proposes a dynamic decision tree integration method (exp-Bagging) based on exponuation weighting, which aims to alleviate the problems of easy fitting and noise sensitivity of traditional decision tree models, as well as low computational efficiency and redundant voting in random forests. By introducing an adaptive weight distribution mechanism, this method effectively suppresses the disadvantages of high variance and weak anti-interference ability of a single decision tree, and improves the robustness of the overall model by dynamically reducing the weight of low-quality subtrees. At the same time, in response to the computational redundancy problem of random forests, Exp-Bagging adopts a selective integration strategy and only retains high-performance subtrees to participate in the final prediction, so as to significantly improve computing efficiency while maintaining classification accuracy.

The main innovation points of this method are reflected in the following two aspects:

First, by introducing exponential attenuation weight design, the automatic identification and weight reduction of noise-sensitive subtrees can be realised, and the model's resistance to overfitting can be enhanced;

Second, a hardware-friendly parallel weight calculation and voting mechanism has been designed to greatly optimise the inference speed and improve the feasibility of deploying the method in a resource-limited environment.

From a theoretical perspective, the effectiveness of this method can be reasonably explained by the generalised error boundary of the weighted integrated model. Based on the weighted majority voting PAC-Bayesian theory proposed by Germain et al., the generalised error upper bound of the integrated model mainly depends on the balance between the weighted average error rate of the individual learner and the diversity of the model [19]. The traditional random forest adopts an equal voting mechanism, which fails to fully consider the impact of the performance differences of each subtree on the integration effect. The exp-Bagging method proposed in this study dynamically reduces the voting weight of the high error rate subtree through the exponential attenuation weighting mechanism, thus directly optimising the weighted average error rate term in the upper bound of the generalised error. At the same time, the method retains the diversity of models brought about by Bootstrap sampling and feature randomness in the Bagging framework, which theoretically ensures that the generalisation performance will not be damaged by weight adjustment, but may be further improved through more reasonable weight distribution. In response to the efficiency limitations of traditional random forests in processing high-throughput data, this study proposes a dynamic decision tree integration method based on exponential attenuation weight, which significantly improves the calculation efficiency while ensuring the accuracy of classification. The main contributions of this study are reflected in the following three dimensions:

1. At the theoretical level, the exponential attenuation weighting mechanism is innovatively introduced into the decision tree integration framework, the weight optimisation path of the decision tree integration method is expanded, and new ideas are provided for integrated learning research under high-dimensional data scenarios. Through theoretical analysis, the optimisation effect of the mechanism on the generalised error boundary is verified, and the theoretical basis of the weighted integrated model is enriched;
2. At the methodological level, the deterministic index weighting rule that does not require hyperparameter tuning is proposed, which avoids the complex tuning process and the risk of overfitting. At the same time, it strictly follows the parallel training framework of Bagging, which is essentially different from Boosting, Stacking and other methods, for integrated learning. Provide a simple and efficient new paradigm;
3. At the application level, the design of a hardware-friendly parallel weight calculation and voting mechanism has greatly optimised the inference speed, solved the pain points that are difficult to balance the efficiency and accuracy of traditional integrated methods, provided a feasible technical path for real-time processing data application scenarios, and improved the feasibility of deployment of integrated learning in a resource-limited environment.

The experimental results show that this method performs well in many standard data sets such as climate-simulation-crashes, wdbc and diabetes. In terms of calculation efficiency, compared with traditional random forests, this method has achieved a 17.6-24.2 times improvement in throughput; in terms of classification performance, core indicators such as accuracy rate, accuracy rate, recall rate and F1 score have been significantly improved, with the highest improvement of 11.1%. These results fully prove that this method effectively solves the problem that efficiency and accuracy are difficult to achieve in traditional integrated learning, and provides a feasible technical path for data application scenarios that require real-time processing.

2 Related work

2.1 Overview of ensemble learning

Integrated learning aims to obtain better generalisation ability and robustness than a single learning device by building and effectively combining multiple base learners. The key to its success lies in the coordination mechanism of "group strategy and group effort". According to the generation method of the base learning device, this field mainly forms two parallel development paradigms: one is the parallelisation method represented by Bagging, and the other is the serialisation method represented by Boosting.

Bagging (Breiman, 1996) builds multiple independent base learners based on self-sampling and integrates them through voting or average mechanisms. The theoretical core of this method is to promote model diversity by introducing sample disturbances, thus significantly reducing the estimated variance, which is particularly significant for the performance improvement of high-variance models (such as decision trees) [20].

In contrast, Boosting class methods (such as GBDT and its extension XGBoost) follow a serialised working paradigm. This kind of method adjusts the sample weight in a targeted manner or directly fits the predicted residual of the previous model through round-by-round iteration, so that there is a close dependence between the base learners [21, 22]. Its core mechanism is to continuously correct model errors, so as to systematically reduce the prediction bias. Take CatBoost as an example. As a typical algorithm under the GBDT framework, it natively supports category type feature processing. At the same time, it has flexible sample weight adjustment capabilities, and can adapt complex data distributions such as category imbalance through category weight distribution [23]. However, the algorithm also has inherent limitations: it takes up significant computing resources when processing large complex data sets. In the process of processing high-base category features, the calculation overhead increases due to the maintenance of feature statistics, and the overall resource consumption is higher than that of some lightweight GBDT variants. This research is mainly based on the parallel integration framework of Bagging, and innovative improvements are made on this basis.

2.2 Bagging and its variants

The classic Bagging algorithm uses Bootstrap sampling to independently generate training subsets for each base learner, and equalizes their prediction results in the aggregation stage. This paradigm is based on a basic presupposit: the contribution of all base learners to final decision-making is equal.

As the most influential extension of Bagging's idea, the random forest further introduces feature disturbances when each node of the decision tree splits on the basis of sample disturbances - that is, seeking the optimal split point from a randomly selected feature subset [24]. This double disturbance mechanism greatly promotes the diversity between base learners, thus laying the foundation for the status of random forests as a powerful and robust model.

Other important variants under the system are also committed to diversity construction, such as:

Pasting: Build a training subset through no-replay sampling;

Random Subspaces: Train the base learners by subspace sampling the feature space [25].

However, although the above methods have achieved remarkable results in enhancing the diversity of models, their aggregation strategies follow the equal weight principle of "one tree and one vote" and fail to consider the differences in the actual performance of individual learners, which constitutes a core limitation of this kind of method.

2.3 Weighted ensemble methods

Recognizing the potential limitations of equal-weight aggregation, researchers have proposed various weighted ensemble approaches. These methods can be broadly categorized into three types:

First category: Performance-based weighting strategies.

These methods typically assign aggregation weights based on the performance metrics (e.g., accuracy or mean squared error, MSE) of the base learners on the validation set. Typical approaches include:

Linear weighting: e.g.,

$$\omega_i = \frac{1}{\mathrm{MSE_i}+\epsilon}, \quad (1)$$

to avoid numerical instability.

Generalized exponential weighting:

$$\omega_i = \exp(\text{-}\alpha\times\mathrm{MSE}), \quad (2)$$

where α is a hyperparameter requiring tuning via cross-validation.

While effective, these methods often introduce additional hyperparameter tuning costs.

The second type of weighting method is implemented by adjusting the training process itself. Take the AdaBoost of the Boosting family as an example, the algorithm assigns weights to the base learner according to the current error rate, and adjusts the weight distribution of subsequent training samples accordingly [26]. It is worth noting that the weight calculation of AdaBoost is closely integrated into its serial training framework and is a by-product of adaptive generation in the iteration process [27]. This is fundamentally different from the paradigm of Bagging, which is parallel and independently trains each base learner.

Third category: Weighting based on meta-learners.

For instance, stacking trains a Meta-Model to learn the optimal combination strategy. While Stacking typically demonstrates strong performance, it suffers from high computational complexity and is prone to overfitting risks [28].

Innovation of this paper: Deterministic exponential weighting rules.

Unlike generalized exponential schemes requiring hyperparameter α tuning, this study proposes and validates a fixed exponential weighting rule:

$$\omega_i = \frac{exp(\text{-}MSE_\mathrm{i})}{\sum_j exp\left(\text{-}MSE_j\right)}, \quad (3)$$

$$\mathrm{MSE_i} = \frac{1}{|\mathrm{D_{val}}|}\sum_{(\mathrm{x,y})\in\mathrm{D_{val}}}\left(\mathrm{y}\text{-}\mathrm{h_i}(\mathrm{x})\right)^2. \quad (4)$$

Where:

$\mathrm{D_{val}} = \{(x_1, y_1), \ldots\ldots, (x_{m_{val}}, y_{m_{val}})\}$ denotes the validation set, independently partitioned from the training set D, containing m_{val} samples for evaluating the generalization performance of the base learners.

$h_i(x)$ denotes the prediction value of the i-th base learner for input sample x.

$(y - h_i(x))^2$ represents the squared error of the i-th learner on a single sample.

$\mathrm{MSE_i}$: is the mean squared error of the i-th base learner across the entire validation set.

Its core principle is:

The MSE on the validation set already contains sufficient performance information.

The monotonic decreasing and convex properties of the exponential function exp(−x) automatically amplify the contribution of high-performing models while suppressing the influence of low-performing ones, thereby achieving an "elite selection" effect.

This approach entirely eliminates hyperparameters, avoiding complex tuning processes and overfitting risks.

Furthermore, this approach strictly adheres to the parallel training framework of Bagging. Weight calculation serves as a post-hoc weighting strategy, fundamentally differing from Boosting and Stacking.

Compared to linear weighting (e.g., 1/MSE), the fixed-exponent rule demonstrates superior discrimination capability: when models exhibit significant MSE differences, high-performing models receive disproportionately high weights, thereby more efficiently boosting overall performance.

Finally, it should be noted that although the mean square error (MSE) on different data sets is different in numerical scale, when compared in the same verification set, its relative size can objectively reflect the superiority and inferiority of the model performance. The experimental results show that the fixed exponential rule proposed in this study can achieve stable and significant performance improvement on multiple heterogeneous data sets, which verifies that the method has good robustness and generalisation effectiveness.

2.4 Performance comparison of integrated models

In order to more clearly sort out the limitations of the existing integration methods, table 1 summarises the core characteristics (aggregation strategy, dynamic weighting ability, etc.) and performance impact of mainstream benchmark models such as random forest, AdaBoost and Stacking. At the same time, it compares the Exp-Bagging proposed in this study to highlight the innovative necessity of Exp-Bagging.

Table 1: Performance comparison of mainstream integrated models

Model Name	Main Aggregation Strategy	Dynamic Weighting?	Hyperparameter Dependence	Key Limitations
Random Forest	Equal-weight majority voting/mean averaging	No	Low (tree count, feature sampling)	1. Ignores base learner performance differences; 2. Linear cost growth
Adaboost	Weighted by base learner error rate (sequential)aggregates prediction results in a sequential manner	Yes (error-rate based)	High (sequential; no parallelization)	1. No data distribution bias constraint; 2. Poor fairness
Stacking	Meta-model learns optimal combination weights	Yes (meta-model learned)	Extremely high (multi-base + meta-model; sequential dependence)	1. Low efficiency; 2. Prone to overfitting
Exp-Bagging (This Study)	Exponential weighting by validation-set MSE (parallel)	Yes (validation-performance based)	Low (parallel; only high-performance subtrees used)	(Targets above limitations; to be verified)

3 Methodology

3.1 Preliminaries: standard bagging

Bagging is proposed by Breiman. Its core idea is to improve the stability and accuracy of learning algorithms (especially unstable algorithms such as decision trees) by building multiple versions of predictors and aggregating them.

The standard Bagging workflow for classification tasks can be formalized as follows:

1. Bootstrap Sampling Phase

Given a training set $D_{train} = \{(x_i, y_i)\}$, i=1...N, containing N samples, where x_i is the feature vector and y_i is the corresponding class label. Bagging generates T new training subsets $\{D_t\}$, t=1...T, via random sampling with replacement, each containing N samples. This sampling ensures each training subset includes approximately 63.2% of the original dataset's samples, with the remaining 36.8% serving as out-of-bag data.

2. Base Learner Training Phase

On each training subset D_t obtained through Bootstrap sampling, a base learner h_t is independently trained. When the base learner is a decision tree, a complete growth strategy (i.e. no pruning) is usually adopted to maximise the diversity between the base learner.

3. Result Aggregation Phase

For the new test sample x, the model forms the final output by synthesising the prediction results of all base learners. In the classification tasks, the following majority voting mechanism is adopted:

$$H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} h_t(x)\right). \qquad (5)$$

In the standard Bagging framework, all base learners use the same weight (1/T) to participate in the final decision-making. Although this equal voting mechanism has the advantage of being easy to implement, it ignores the objective performance differences between different decision trees. In the actual training process, some decision trees may be less reliable due to overfitting of their Bootstrap samples or poor performance in specific data distribution, while other decision trees show stronger discrimination. However, the standard Bagging gives the same voting weight to these decision trees with different performance, which may lead to the dilution of the contribution of the high-quality tree, thus affecting the overall performance of the integrated model. The method proposed in this study is designed to break through this limitation.

3.2 Exponential weighting scheme with validation-based MSE

In order to improve the limitations of static weighting in standard Bagging, this paper proposes a dynamic weighting strategy. Based on the prediction accuracy of each base learner on the independent verification set, this method adaptively allocates the integrated weight, so as to enhance the contribution of high-performance base learners in the process of model aggregation and suppress the influence of low-performance base learners.

The proposed method comprises the following key steps:

1. Data Partitioning: The available dataset is divided into three independent sets: training set D_{train} (70%), validation set D_{val} (15%), and test set D_{test} (15%). The validation set is crucial for providing an unbiased estimate of each tree's generalization performance.

2. Ensemble Generation via Bagging: The standard bagging procedure described in Section 3.1 is first employed to train an ensemble of T decision trees specifically on D_{train} using Bootstrap sampling.

3. Performance Evaluation on the Validation Set: Each trained tree ht is evaluated on the independent validation set Dval, on which it was not trained. Mean Squared Error (MSE) is used as the performance metric. For a validation set of size M, the MSE for tree h_t is calculated as follows:

$$\mathrm{MSE}_t = \frac{1}{M}\sum_{j=1}^{M}\left(h_t(x_j) - y_j\right)^2. \qquad (6)$$

4. Exponential weighting: Our core innovation lies in converting each tree's validation set MSE into an adaptive weight using a negative exponential function. The weight wt for tree ht is calculated as follows:

$$\omega_t = \exp(\text{-MSE}). \tag{7}$$

5. Weight Normalization: The raw exponential weights are normalized to form a convex combination, ensuring they sum to 1.

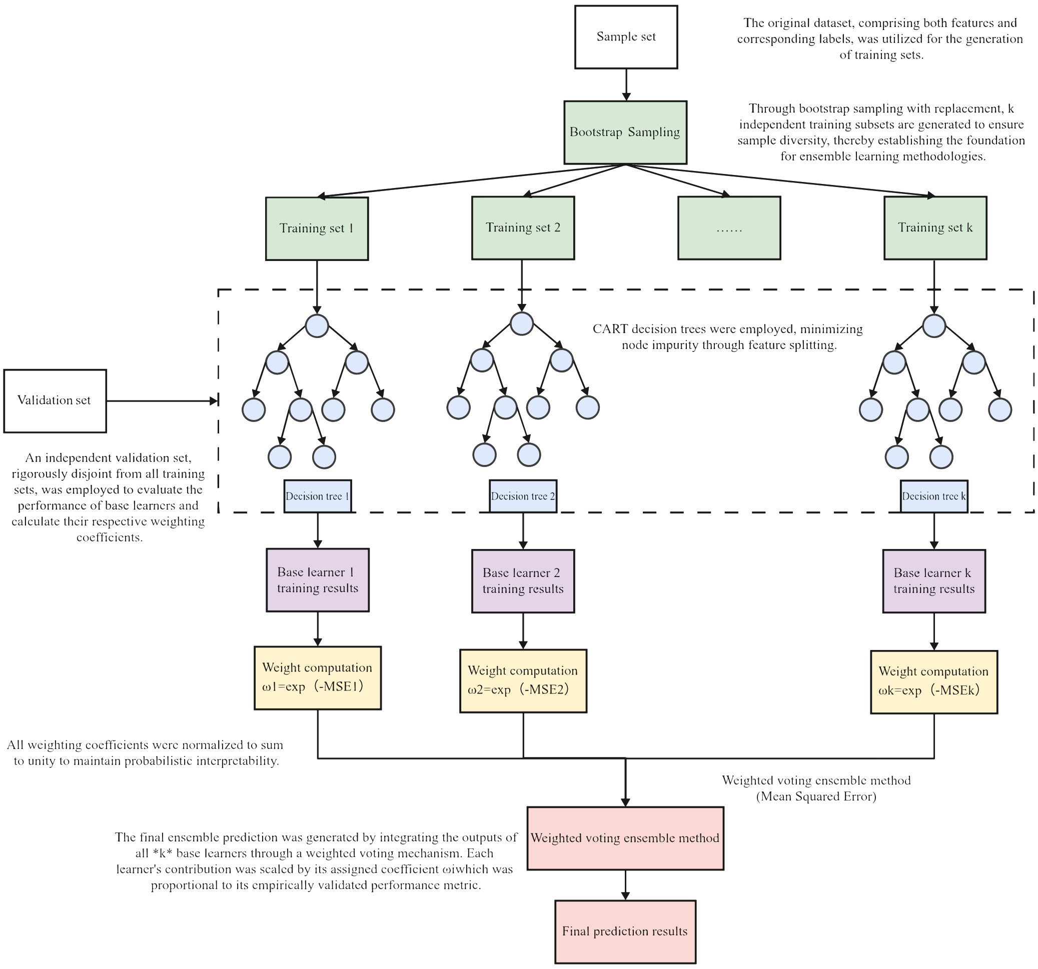
$$\omega_t{}' = \frac{\omega_t}{\sum_{i=1}^{T} \omega_i}. \tag{8}$$

6.Weighted Aggregation for Prediction: For a new test instance x, the final prediction is no longer a simple majority vote, but a weighted average of the 20 individual predictions. The ensemble model outputs a weighted sum of prediction probabilities:

$$\mathrm{H(x)} = \sum_{\mathrm{t=1}}^{\mathrm{T}} \omega_t{}' \cdot \mathrm{h_t(x)}. \tag{9}$$

The final category label is obtained by setting a threshold (0.5 in this experiment) on this continuous output.

The scheme ensures that trees with low error (high precision) on the verification set are given an exponentially higher weight, thus dominating the final decision-making. On the contrary, trees with high MSE will be automatically assigned a gradually reduced weight, thus effectively reducing their negative effects and the computational costs of including them in the forecasting stage. The overall process and core mechanism of this method are shown in Figure 1, which clearly shows the complete calculation path from data preprocessing, base learner generation to dynamic weighting integration.


Sample set
Bootstrap Sampling
Training set 1
Training set 2
……
Training set k
Validation set
Decision tree 1
Decision tree 2
Decision tree k
Base learner 1 training results
Base learner 2 training results
Base learner k training results
Weight computation ω1=exp（-MSE1）
Weight computation ω2=exp（-MSE2）
Weight computation ωk=exp（-MSEk）
Weighted voting ensemble method
Final prediction results

The original dataset, comprising both features and corresponding labels, was utilized for the generation of training sets.

Through bootstrap sampling with replacement, k independent training subsets are generated to ensure sample diversity, thereby establishing the foundation for ensemble learning methodologies.

CART decision trees were employed, minimizing node impurity through feature splitting.

An independent validation set, rigorously disjoint from all training sets, was employed to evaluate the performance of base learners and calculate their respective weighting coefficients.

All weighting coefficients were normalized to sum to unity to maintain probabilistic interpretability.

Weighted voting ensemble method (Mean Squared Error)

The final ensemble prediction was generated by integrating the outputs of all *k* base learners through a weighted voting mechanism. Each learner's contribution was scaled by its assigned coefficient ωiwhich was proportional to its empirically validated performance metric.

Figure 1: Schematic diagram of the proposed ensemble method.

In order to intuitively verify the distinguishing advantage of the performance of the exponential weighted base learner, Figure 2 shows the distribution comparison of the verification set mean square error (red column, MSE) and the exponential weight (blue column, Exponential Weight) of 10 base learners. It can be clearly seen that the smaller the MSE, the higher the exponential weight of the base learner (for example, the MSE of the base learner 4 has the lowest and the highest weight; the MSE of the base learner 10 has a higher weight and a relatively low weight). The strong negative correlation of "performance-weight" reflects the precise distinction between the performance differences of exponential weighting and base learners - high-quality base learners get a higher weight in integration, and the weight of inferior base learners is effectively compressed. In terms of quantitative indicators, the weight of the index weight is extremely different (the ratio of the maximum weight to the minimum weight) can reach 1.1:1, and the weighted entropy value of the exponential weight (an indicator to measure the uniformity of distribution) is significantly lower, indicating that the weight is concentrated in a few high-performance base learners, and the distinction is better.

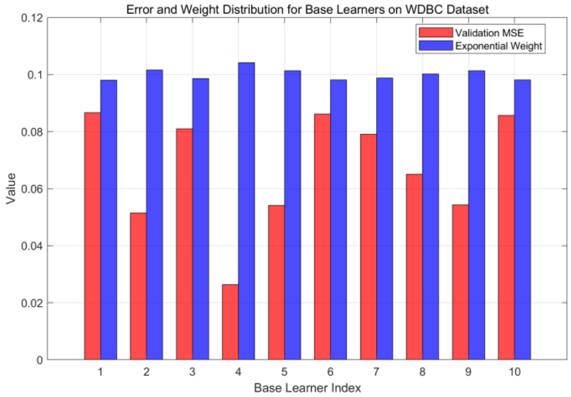


Figure 2: Error and weight distribution for base learners on WDBC dataset

3.3. The exp-bagging algorithm (validation-based)

Algorithm 1 illustrates the main steps of our approach:

```
Algorithm 1: Proposed Dynamic Weighted Ensemble Algorithm
Input: Training set $D_{train}$, validation set $D_{val}$, number of trees T
Initialize an empty list of trees and weights: Trees ← [ ], Weights ← [ ]
for t = 1 to T do
    $D_t$ ←Bootstrap sample from $D_{train}$
    $h_t$ ← Train a decision tree on $D_t$
    $MSE_t$ ← Calculate MSE of $h_t$ on $D_{val}$
    $\omega_t$← exp(-MSE:)// Calculate raw weight
    Append $h_t$ to Trees and $\omega_t$ to Weights
end for
Normalize weights: Weights←Weights/sum(Weights)
return Ensemble model $\mathrm{H(x)} = \sum_{t=1}^{T} \mathrm{Weights[t]\,.Trees[t](x)}$
Output: Trained ensemble model H
```

3.4 Theoretical advantages and analysis of the exponential weighting scheme

The index weighted scheme proposed in this study is concise and has sufficient theoretical basis and statistical significance. Its main advantages are reflected in the following four aspects:

1. Intrinsic normalisation mechanism

The weighting scheme employs the following form:

$$\omega_i=\frac{exp(\text{-}MSE_{\mathrm{i}})}{\sum_j exp(\text{-}MSE_j)}. \tag{10}$$

The design can achieve the automatic standardization of weights without the introduction of additional scaling parameters, which not only simplifies the implementation process, but also avoids the subjective deviation that may be caused by the artificial setting of the normalization factor.

2. Cross-scale adaptability

Considering that there may be scale differences in the mean square error of different data sets, the direct use of the original value is easy to lead to an imbalance in weight distribution. The core advantage of the exponential function exp(−x) is the sensitive capture ability of relative differences. As long as the error of each base learner is calculated on the same verification set, their relative sorting can reliably reflect the performance, and the exponential transformation can adaptively amplify this difference to ensure the stability and robustness of the weight distribution.

3. Strong distinguishing ability

Compared with the linear weighting method, the exponential function has the characteristics of nonlinear amplification. When the performance of the base learner is similar, the weight distribution is relatively gentle; and when the performance difference is significant, the high-quality learner will receive an exponentially enhanced weight reward, and those with poor performance will be correspondingly suppressed. This characteristic naturally forms a screening mechanism of "winning and losing", which significantly improves the recognition and utilisation efficiency of high-performance learners by the integrated system.

4. Bayesian Probability Explanation

Based on the assumption that the error obeys the Gausian distribution, the exponential weight term can be interpreted as a non-normalised form of the post-obterential probability of the model. This characteristic forms a natural fit with the core framework of the PAC-

Bayes theory, providing solid theoretical support for the generalisation performance of the exponential weighted strategy. In the study, Germain et al. clearly proposed the PAC-Bayes boundary directly related to this characteristic, and its exact form is:

$$Pr_{S\sim D^m}\left(\forall Q on \mathrm{H}: R(G_Q)\leq\frac{1}{1\text{-}e^{-C'}}\left[C'\cdot R_S(G_Q)+\frac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\frac{1}{\delta}\right]\right]\right)\geq 1\text{-}\delta, \tag{11}$$

Among them, $R(G_Q)$ is the generalised error of the Gibbs classifier (the core measure of the generalised error of the integrated model), $R_S(G_Q)$ is the empirical error on the training set, $KL(Q\|P)\overset{def}{=}E_{\mathrm{h}\sim Q}\ln\frac{Q(\mathrm{h})}{P(\mathrm{h})}$ is the Kullback−Leibler divergence between Q and P, C' is the adjustment parameter, and δ is the confidence level (usually 0.05) [29]. This theorem clearly defines the core logic of the PAC-Bayes theory: the generalised error of the integrated model is determined by the "weighted empirical error" and the "weight distribution complexity", and the high-quality weighting strategy needs to be balanced between the two.

Exp-Bagging's exponential weighting strategy ($\omega_i \propto \exp(-MSE_i)$) can theoretically improve the marginal efficiency (that is, the reduction of generalisation errors by high-quality base learners): from the perspective of Germain et al.' PAC-Bayes boundary, the core of marginal benefits is "weighted experience error" The ratio of the "reduction rate of difference" and "the growth rate of weight complexity". Exponential function $\exp(-MSE_i)$has non-linear amplification characteristics - when the difference in the base learner MSE is significant, the exponential weighting will greatly increase the weight proportion of the high-quality base learner and directly reduce the "weighted empirical error item" in the boundary; at the same time, the weight only depends on the adaptive adjustment of the verification set MSE, and there is no amount The introduction of external super parameters ensures that the weight distribution and the uniform a priori KL dispersion is always at a low level, far lower than the dispersion value of the "over-concentrated weight" strategy, so that the growth rate of the "weight complexity item" is much slower than the reduction rate of the weighted empirical error. This characteristic of "fast error reduction and slow complexity growth" allows the index weighting to achieve "marginal benefit increment" in the PAC-Bayes boundary - with the addition of high-quality base learning devices, the tightness of the generalised error boundary continues to improve, and the improvement expands with the quality of the base learner. This is the theoretical essence of its superior to the traditional weighting strategy. Further combined with the idea of Bayesian model average (BMA), Germain et al.'s PAC-Bayes framework allows data-driven update of a prior distribution, and the calculation process of exponential weight is essentially based on the a posterior inference of the performance of the verification set to the confidence of the base learner - MSE The smaller the base learner, the higher the prediction credibility and the greater the corresponding a posterior probability ($\exp(-MSE_i)$), which is fully consistent with the core criterion of the PAC-Bayes theory that "low error models should be given higher integration confidence". This consistency enables the index weighting strategy to not only improve the integration performance in practical applications, but also prove the controllability of generalisation capabilities through the PAC-Bayes boundary of Germain and others, and realise the unity of "practical effectiveness" and "theoretical rigour".

4 Experimental validation

In order to systematically evaluate the comprehensive performance of the Exp-Bagging algorithm, this research has designed a rigourous experimental scheme, covering multiple public data sets, a variety of baseline comparison methods and a complete evaluation process. Preliminary experimental results show that this method significantly improves the computing efficiency while ensuring excellent classification performance. Taking the Climate-Simulation-Crashes data set as an example, this method achieves a classification accuracy of 95.7%, and the throughput exceeds 80,000 samples/second, showing the good potential for processing high-dimensional data. The complete performance comparison results of each data set are detailed in Table 8.

4.1 Implementation settings

4.1.1 Hardware and software environment

The hardware configuration included an AMD Ryzen 9 7945HX with Radeon Graphics CPU (16 cores/32 threads), 16GB DDR5 memory (15.7GB available), and a 1TB NVMe SSD solid-state drive. The experiments were conducted in the MATLAB R2023b environment on Windows 11 operating system. All custom algorithms, including Exp-Bagging, were independently coded and implemented by the author to ensure the accuracy of the experimental process and the reproducibility of the results.

4.1.2 Model training and evaluation

In the experimental initialisation stage, the global random number generator state is fixed by rng(42). This seed is a common reference value for reprodibility verification in the field of machine learning, which can balance randomness and result stability. The specific random process control is as follows:

1. Data division link: based on the randperm(n) function of fixed seeds, realise the random segmentation of the training set / verification set / test set to ensure that the data division results of each run are completely consistent;
2. Base learner training link: In the parallel training loop (parfor), all random operations (including Bootstrap sampling to generate training subset and the characteristic selection of decision tree node splitting) are controlled by the global fixed seed rng (42) and calculated in parallel through MATLAB. The random state isolation mechanism avoids random duplication between different base learners, which not only ensures the diversity of integrated models, but also ensures that the generation process of each trained base learner can be reproduced.

Each data set is randomly divided into training set, verification set and test set according to the ratio of 70:15:15. In order to reduce the impact of random factors in the process of data division and training, the whole experimental process (including data division, model training and performance evaluation) is independently repeated 10 times, and the average and standard deviation of the final performance index is taken.

The main parameter settings are as follows:

Base learner: Use the fitrtree function to build a decision tree, set the minimum number of samples of leaf nodes to 5, and enable the MSE-based automatic pruning function to balance the complexity of the model and the generalisation ability.

Integration scale: All integration methods (Exp-Bagging, Bagging and Random Forest) include 10 base learners. All experiments uniformly set the number of base learners T=10. This selection comes from the results of the melting experiment of the number of 4.4.2 base learners: when T increases from 5 to 10, the core indicators such as the F1 score and accuracy of the three data sets are significantly improved, and the throughput is maintained at the high efficiency level; while T continues to increase to 20 and 50, the performance improvement tends to stagnate, but the throughput plummets, and the efficiency loss is significant. At the same time, when T=10, the performance standard deviation of the model on the three data sets is the smallest, and the stability is better than that of T equal to other values. In summary, T=10 is the optimal equilibrium point of performance, efficiency and stability, so it is used as a unified parameter. It is worth mentioning that this paper uses a systematic hyperparameter tuning method based on the accuracy of the verification set to optimise the performance of the AdaBoost model. This method effectively evaluates the generalisation ability of the model and prevents overfitting by grid searching the number of weak learners (T=5,10,20,50) and the learning rate (0.1,0.5,1.0,1.5), taking the verification set accuracy rate as the only criterion for parameter selection. By combining the dynamic early stop mechanism in the tuning process and real-time monitoring of the verification set performance during the training process to determine the best number of iterations, an optimised parameter configuration scheme is provided for the practical application of the AdaBoost algorithm.

Feature sampling: The number of feature sampling of each tree in the random forest is set to the square root of the total number of features (√P, P is the feature dimension).

Sequence integration: The number of iterations of AdaBoost is set to 10 rounds.

The core task of this study is a classification task. All evaluation metrics are designed for category prediction scenarios and do not involve regression-related metrics (e.g., MAE, MSE), so as to ensure consistency between the metrics and the task objectives. The specific definitions of the metrics are as follows:

1. Accuracy: The proportion of correctly classified samples among all samples, reflecting the overall prediction capability of the model;
2. Precision: The proportion of truly positive samples among the samples predicted as positive, measuring the model’ s ability to control "false positive misjudgments";
3. Recall: The proportion of truly positive samples correctly predicted among all actual positive samples, measuring the model’ s ability to avoid "false negative misjudgments";
4. F1-score: The harmonic mean of Precision and Recall, used to balance the two types of errors and adapt to the evaluation needs of imbalanced data scenarios.
5. Throughput: Based on the experimental hardware (AMD Ryzen 9 7945HX processor, 16-core fully enabled; 16GB DDR5 memory, available 15.7GB; 1TB NVMe SSD solid-state drive), it is defined as "in the hardware environment, in the single reasoning process, single The number of test samples that can be processed in bit time (per second), the unit is "sample / second". The calculation method is: input all samples of the test set into the trained model. Reasoning, record the total reasoning time (including the whole process of data reading, model calculation, and result output, excluding the time consumption of data preprocessing), and finally calculated through the "total number of samples of the test set / total reasoning time". This calculation method can ensure that the throughput indicator truly reflects the real-time reasoning ability of the model in the actual hardware environment, and all comparison methods (decision tree, random forest, AdaBoost) are tested under the same hardware settings to ensure the fairness of efficiency comparison.

4.2 Datasets

The experiment selected three public classification data sets for performance verification: Climate-Simulation-Crashes, WDBC and Diabetes. These data sets come from different fields such as climate simulation and medical diagnosis. They have significant differences in sample size, characteristic dimension and data characteristics, and can comprehensively evaluate the applicability of the method. In order to eliminate the impact of characteristic metric differences, all data sets are preprocessed by the Z-score standardised method. Table 1 details the statistical characteristics of each data set and the specific process of preprocessing.

Table 2: Statistical summary of the benchmark datasets

Dataset Name	Sample Size	Number of Features	Source	Task Type
Climate Model Simulation Crashes	540	18	Kaggle	Classification
Wdbc	569	30	Kaggle	Classification
Diabetes	768	8	Kaggle	Classification

4.3 Baseline methods

In order to verify the effectiveness of the Exp-Bagging algorithm, the following three types of representative baseline methods are selected for comparative analysis:

1. Decision Tree: As the basic unit of the integrated model, this comparison can verify the effectiveness of the integrated strategy itself;
2. Random Forest: As the most influential extension method in the Bagging framework, it is a key reference for evaluating the improvement value of this study;
3. AdaBoost: As a typical representative of sequential integrated learning, this comparison helps to comprehensively evaluate the competitiveness of this research method in a wider field.

The complete performance comparison results of the above baseline methods on the three data sets are shown in Tables 3, 4 and 5.

Table 3: Performance of different methods on the climate model simulation crashes dataset

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.926±0.025	0.956±0.023	0.964±0.024	0.960±0.014	879,806.92
Random forest	0.931±0.023	0.947±0.020	0.980±0.019	0.963±0.013	23843.00
AdaBoost	0.937±0.019	0.939±0.018	0.996±0.006	0.967±0.010	30652.56
Exp-Bagging	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009	81518.68

Table 4: Performance of different methods on the WDBC dataset

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.926±0.020	0.889±0.051	0.901±0.051	0.894±0.037	1092377.11
Random forest	0.928±0.012	0.893±0.032	0.921±0.039	0.906±0.019	25444.33
AdaBoost	0.951±0.020	0.943±0.045	0.926±0.052	0.933±0.030	45391.92
Exp-Bagging	0.981±0.006	0.944±0.017	1.000±0.000	0.971±0.009	70112.25

Table 5: Performance of different methods on the Diabetes dataset.

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.690±0.026	0.563±0.075	0.507±0.061	0.529±0.044	965,284.71
Random forest	0.759±0.051	0.644±0.087	0.621±0.085	0.630±0.077	25428.13
AdaBoost	0.745±0.025	0.666±0.040	0.553±0.078	0.601±0.049	58015.51
Exp-Bagging	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026	78172.10

(Table 3-5: Binary Classification Performance. We report macro F1-scores (mean + stdev over 10 trials) on test data with optimized hyperparameters. The rank of each method is presented in brackets. The datasets are sorted by the number of features)

It should be noted that there are significant differences in the performance improvement of Exp-Bagging on different datasets: on the Climate Model Simulation Crashes and WDBC datasets, the F1 score is 1.4% and 6.5% higher than that of Random Forest, respectively, with obvious advantages; but on the Diabetes dataset, the F1 score only decreases slightly compared with Random Forest, while the accuracy rate is 3.8% higher, achieving only a small optimization. This difference is not the failure of the method, but is determined by the inherent characteristics of the Diabetes dataset: judging from the data characteristics, the Diabetes dataset has the problems of category imbalance (only 34.9% of the diseased samples) and low feature differentiation—the distribution of 8 continuous features (such as blood sugar and blood pressure) in healthy and diseased samples overlaps highly, resulting in the concentration of the prediction error (MSE) distribution of the base learner. The exponential weighting strategy of Exp-Bagging needs to rely on the performance difference of the base learner to achieve "selecting the superior and eliminating the inferior". When the difference is small, the weighted optimization space is compressed, which weakens the gain effect of weighted integration. In addition, the insufficient density of sample feature information limits the diversity of base learners. Although the sample size of the Diabetes dataset (768) is higher than that of WDBC (569) and Climate Model Simulation Crashes (540), its feature dimension is only 8, far lower than the 30 of WDBC and 18 of Climate Model

Simulation Crashes, leading to low "sample-feature" information density. When generating training subsets through Bootstrap sampling, the limited feature dimension makes it difficult for the base learner to learn differentiated laws, and there are very few optional differentiated features when the decision tree node splits, which ultimately leads to a high degree of similarity in the output laws of the base learner. The performance improvement of integrated learning depends on the "balance between the accuracy and diversity of the base learner" (refer to Breiman's 1996 Bagging classic theory). When the diversity is insufficient, even if the weighting strategy is adopted, it is difficult to break through the performance ceiling of a single base learner, which is exactly the key reason why Exp-Bagging is limited in this dataset. Despite the above limitations, Exp-Bagging still shows irreplaceable practical value on the Diabetes dataset: on the one hand, its throughput (78172.10 samples/second) is 3.1 times that of Random Forest and 1.3 times that of AdaBoost, which can meet the clinical needs of "real-time detection and rapid feedback" in diabetes-assisted diagnosis; on the other hand, the standard deviation of its performance indicators (accuracy rate ±0.019, F1 score ±0.026) is significantly lower than that of Random Forest (accuracy rate ±0.051, F1 score ±0.077), reflecting better stability—in medical diagnosis scenarios, model stability is as important as accuracy, which can effectively reduce the risk of "misdiagnosis caused by model fluctuations".

4.4 Ablation Study

In order to evaluate the actual contribution of each component in the integrated framework, this study designed a systematic ablation experimental scheme:

4.4.1 Validity verification of weighted strategy

This section focusses on the impact of different weighting strategies on integration performance. The experimental setting is divided into two levels: the core comparison level to all data sets, comparing the performance of the exponential weighting strategy proposed in this paper with the optimal single model; the extended comparison layer introduces more advanced weighting methods on some data sets for in-depth comparison: new average integration on the Diabetes data set, Three strategies based on sorting weighting and MSE countdown weighting; supplement the two methods of sorting-based weighting and MSE countdown weighting on the climate-simulation-crashes data set.

In order to clarify the calculation logic of the two types of comparison integration strategies and ensure the fairness and reproducibility of experimental comparison, this section supplements the mathematical formulas, core principles and implementation details of Equal Weighting and Inverse-MSE Weighting, and contrasts with the exponential weighting of Exp-Bagging in this article.

(1) Equal Weighting is the most classic static weighting strategy. The core logic is that "all base learning devices contribute equally". It does not rely on any performance indicators and directly assigns the same weight to each base learning device, which is suitable for scenarios with small performance differences in base learning devices. The weight of the i-base learning device is ω_i^{equal}, then:

$$\omega_i^{\mathrm{e}qual}=\frac{1}{T}. \tag{12}$$

Among them, T is the total number of base learners (T=10 in this experiment).

The sum of the weights of all base learners is $\sum_{i=1}^{T}\omega_i^{\mathrm{e}qual}=1$, and no additional normalisation steps are required.

For the new test sample x, the final prediction of the average integration is the arithmetic average of the predicted values of all base learners, and the calculation formula is:

$$H_{\mathrm{e}qual}(x)=\sum_{i=1}^{T}\omega_i^{\mathrm{e}qual}\cdot\mathrm{h}_i(x)=\frac{1}{T}\sum_{i=1}^{T}\mathrm{h}_i(x). \tag{13}$$

Among them, $\mathrm{h}_i(x)$ is the predicted value of the ith base learner for sample x. Finally, the final category label is obtained by discreting the continuous output $H_{\mathrm{e}qual}(x)$ by setting the threshold (0.5 in this experiment).

(2) Inverse-MSE Weighting is a dynamic weighting strategy based on the performance of the base learner. The core logic is "the smaller the MSE, the greater the weight". The linear correlation between performance and weight is realised through the inverse mapping of MSE, and the performance of the base learner needs to be evaluated by relying on the verification set. Introduce the least valueϵ (take $\epsilon=10^{-10}$ in this experiment) to avoid the error of "divising by 0" when MSE=0. The original weight $\omega_i^{\text{inv-mse,raw}}$ of the i-th base learner is:

$$\omega_i^{\text{inv-mse,raw}}=\frac{1}{MSE_i+\epsilon}, \tag{14}$$

Among them, the calculation of MSE_i is the same as the formula (6).

The original weight is normalised into a convex combination with a sum of 1 to ensure the interpretability of probability. After normalisation, the weight $\omega_i^{\mathrm{i}nv\text{-}mse}$ is:

$$\omega_i^{\mathrm{i}nv\text{-}mse}=\frac{\omega_i^{\mathrm{i}nv\text{-}mse,raw}}{\sum_{k=1}^{T}\omega_k^{\mathrm{i}nv\text{-}mse,raw}} \tag{15}$$

For the new test sample x, the final prediction of the Inverse-MSE integration is the weighted average of the predicted value of the base learner, and the calculation formula is:

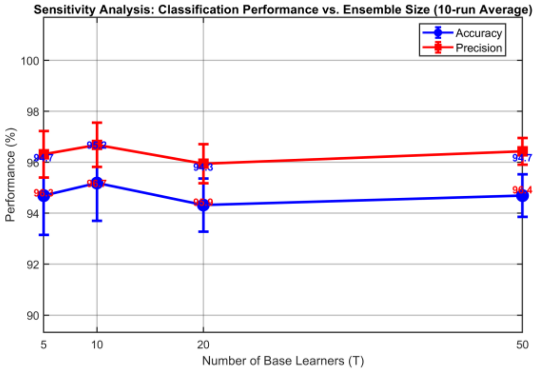
$$H_{\mathrm{i}nv\text{-}mse}(x)=\sum_{i=1}^{T}\omega_i^{\mathrm{i}nv\text{-}mse}\cdot\mathrm{h}_i(x) \tag{16}$$

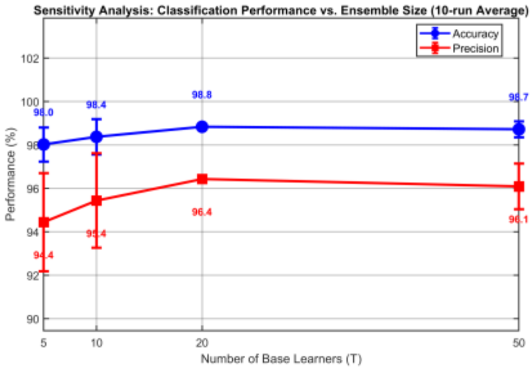
Similarly, the final category label is obtained by setting the threshold (0.5) discrete continuous output.

Ablation experiment results (Tables 6 – 8) reveal the discriminative ability of the proposed exponential weighting strategy in optimizing ensemble performance. On the Climate Model Simulation Crashes dataset, exponential weighting outperforms rank-based and inverse-MSE weighting mainly in F1-score (97.7% vs. 97.3% and 97.6%), which indicates that the negative exponential function effectively amplifies the contribution of low-error subtrees while suppressing high-error ones. For the WDBC dataset, exponential weighting achieves a 4.5-percentage-point accuracy improvement over the best single model (98.1% vs. 93.6%), confirming that dynamic weighting compensates for the limitation of individual decision trees in handling high-dimensional medical data. On the Diabetes dataset with obvious data imbalance, exponential weighting maintains comparable accuracy to inverse-MSE weighting (78.7% vs. 78.6%) but exhibits smaller indicator variance, reflecting stronger robustness to noisy data.

Based on the comprehensive experimental results shown in Table 6 to Table 8, the following conclusions can be drawn: the exponential weighting strategy proposed in this paper performs well in all test scenarios, and its performance is stable and significantly better than the average integration and optimal single model baseline. Furthermore, in direct comparisons with other advanced weighting strategies (on Diabetes and climate-simulation-crashes), the exponential weighting strategy also exhibits competitive advantages, achieving optimal performance.

Figure 3-5 is the sensitivity analysis chart of the number of base learning devices (T) of Exp-Bagging in Climate Model Simulation Crashes dataset, WDBC dataset and Diabetes dataset (classification performance chooses Accuracy and Precision).





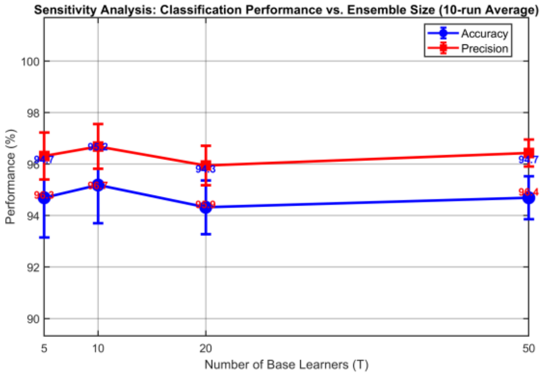


Figure 3: Sensitivity analysis chart on Climate Model Simulation Crashes dataset
Figure 4: Sensitivity analysis chart on WDBC dataset
Figure 5: Sensitivity analysis chart on Diabetes dataset

Table 6: Performance of different weighting schemes on the climate model simulation crashes dataset.

Weight Computation Methods	Accuracy	Precision	Recall	F1
Rank-based Weighting	0.948±0.011	0.968±0.012	0.979±0.007	0.973±0.006
Inverse-MSE Weighting	0.954±0.018	0.967±0.012	0.986±0.010	0.976±0.009
Best Single Model	0.928±0.019	0.967±0.009	0.955±0.021	0.961±0.011
Exponential Weighting	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009

Table 7: Performance of different weighting schemes on the WDBC dataset

Weight Computation Methods	Accuracy	Precision	Recall	F1
Best Single Model	0.936±0.021	0.864±0.046	0.948±0.050	0.903±0.031
Exponential Weighting	0.981±0.006	0.944±0.017	0.997±0.006	0.971±0.009

Table 8: Performance of different weighting schemes on the diabetes dataset.

Weight Computation Methods	Accuracy	Precision	Recall	F1
Equal Weighting	0.784±0.023	0.572±0.043	0.683±0.042	0.622±0.030
Rank-based Weighting	0.779±0.022	0.563±0.040	0.660±0.054	0.607±0.040
Inverse-MSE Weighting	0.786±0.020	0.574±0.039	0.690±0.039	0.626±0.026
Best Single Model	0.735±0.035	0.493±0.052	0.613±0.074	0.545±0.055
Exponential Weighting	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026

4.4.2 Parameter sensitivity analysis of the number of base learners (T)

In order to evaluate the impact of changes in the number of base learners (T) on the performance and reasoning speed of the Exp-Bagging model, supplement the ablation experiment: select four typical values of T=5, 10, 20 and 50, repeat the experiment 10 times on the three data sets of Climate-Simulation-Crashes, WDBC and Diabetes, record the classification performance index (accuracy, precision, F1 score) and the reasoning speed index (throughput), and analyse the trade-off relationship caused by parameter changes.

This experimental setting maintains that other parameters are consistent with the original experiment, and the verification set accounts for 15%, the minimum number of samples of leaf nodes is 5, and MSE automatic pruning is enabled. In addition, the throughput calculation method remains unchanged, that is, based on AMD Ryzen 9 7945HX CPU, the speed of all samples of the single-wheel reasoning processing test set (unit: samples/s).

The model performance and throughput under the number of different base learners are shown in Table 9-11:

Table 9: Performance comparison of different T values on the climate model simulation crashes dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.947±0.009	0.961±0.006	0.983±0.011	0.972±0.006	125701.13
10	0.955±0.008	0.969±0.010	0.983±0.006	0.976±0.004	69106.90
20	0.954±0.012	0.969±0.006	0.983±0.010	0.976±0.006	31131.94
50	0.954±0.010	0.969±0.006	0.983±0.006	0.976±0.005	14174.74

Table 10:Performance comparison of different T values on the WDBC dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.977±0.010	0.935±0.026	0.996±0.011	0.964±0.016	125968.69
10	0.979±0.007	0.941±0.015	0.996±0.011	0.968±0.011	69884.78
20	0.986±0.004	0.957±0.013	1.000±0.000	0.978±0.007	39750.01
50	0.987±0.003	0.961±0.010	1.000±0.000	0.980±0.005	15408.74

Table 11:Performance comparison of different T values on the Diabetes dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.772±0.014	0.546±0.019	0.680±0.067	0.605±0.037	64476.89
10	0.791±0.017	0.577±0.029	0.720±0.031	0.640±0.027	74728.29
20	0.787±0.015	0.569±0.026	0.743±0.040	0.644±0.022	14371.10
50	0.790±0.016	0.574±0.026	0.737±0.028	0.644±0.021	13150.33

Judging from the experimental data, on the Climate Model Simulation Crashes data set, when the number of base learners increased from 5 to 10, the accuracy rate and F1 score were improved, and the throughput decreased significantly; when the number increased to 20 and 50, the classification The performance basically tends to be saturated, and the throughput continues to decrease. The WDBC data set shows a similar trend. The number of base learners has improved the classification performance from 5 to 10, and the performance of 20 and 50 has further improved slightly, but the throughput has decreased significantly. The Diabetes data set has relatively better classification performance indicators such as accuracy and F1 score when the number of base learners is 10. When the number is 5, 20 and 50, the performance is slightly worse, and the throughput also declines with the increase in the number of base learners. Overall, the increase in the number of base learners can improve the classification performance within a certain range, but it will lead to a decrease in the reasoning throughput, and there are differences in the trade-off between performance and throughput of different data sets. Among them, the classification performance of WDBC data sets is optimal when the number of base learners is 50, the classification performance and throughput of the Climate Model Simulation Crashes data set are better balanced when the number is 10, and the classification performance of the Diabetes data set is relatively better when the number is 10. The above conclusion further explains the necessity of selecting the number of base learning devices to be 10.

4.5 Comprehensive performance evaluation

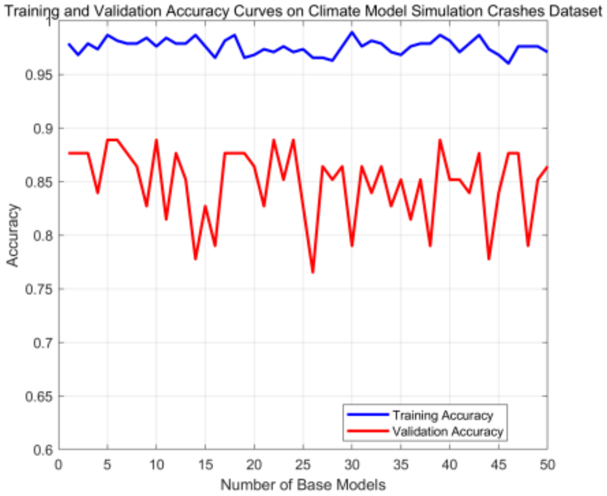
Evaluation metrics widely used in regression tasks are employed to comprehensively assess model performance. All experiments report the following metrics, with accuracy, precision, recall, F1, Throughput as primary evaluation indicators. The corresponding performance metrics are summarized in Table 12.

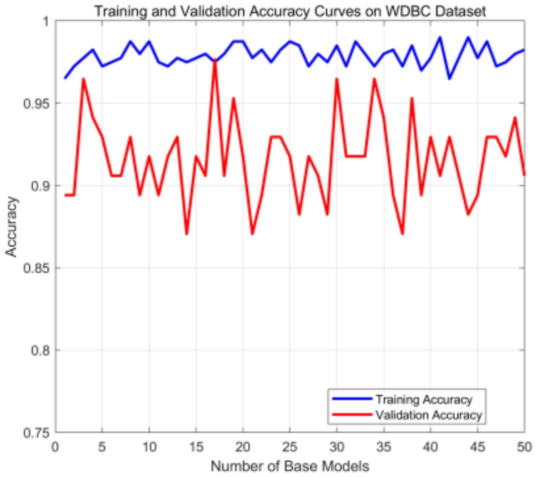
Table12: Performance of the proposed method across different datasets

Dataset Name	Accuracy	Precision	Recall	F1	Throughput
Climate Model Simulation Crashes	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009	81518.68
WDBC	0.981±0.006	0.944±0.017	1.000±0.000	0.971±0.009	70112.25
Diabetes	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026	78172.10

Comprehensive performance (Table 12) shows the method's cross-scenario excellence: 95.7% – 98.1% accuracy (100% recall on WDBC) and 70,112 – 81,518 samples/second throughput, balancing classification quality and real-time inference across all three datasets.

In summary, Exp-Bagging has achieved a balance of classification performance, reasoning efficiency and stability on three types of data sets. In order to further verify the actual deployment reliability of the model, it is necessary to supplement the fitting risk analysis - the complexity of the integrated model is likely to increase with the increase in the number of base learners, and improper control may lead to a decline in generalisation ability. Combined with the previous experimental data set and settings, the quantitative analysis of the accuracy curve (Figure 6-8) through training-verification is as follows: It can be seen from the training-verification accuracy curve of the three data sets that Exp-Bagging maintains a training accuracy of more than 0.96 on the Climate Model Simulation Crashes data set, and the verification accuracy is stable above 0.90, and the gap is controlled. Within 0.06; WDBC data set training accuracy is above 0.98, verification accuracy is above 0.90, and the gap is within 0.08; Diabetes data set training accuracy is about 0.90, verification accuracy is above 0.60, the gap is within 0.30, and there are three types of data sets There is no overfitting characteristic of "training accuracy continues to rise, verification accuracy first to rise and then fall". This shows that Exp-Bagging effectively avoids the risk of overfitting on different characteristic data sets and has reliable generalisation capabilities through the dual design of exponentially weighted "hidden pruning" and verification set-driven weight distribution.





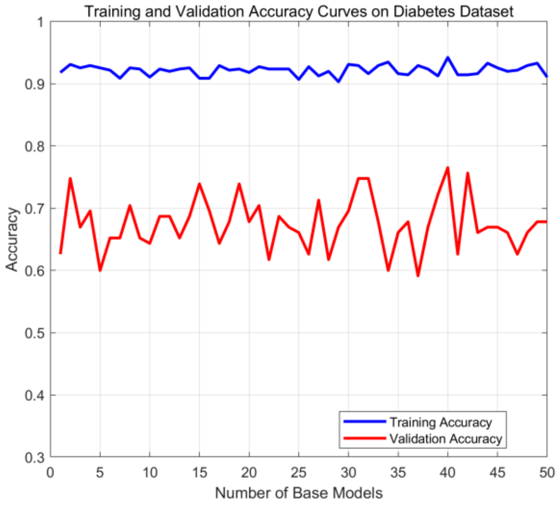


Figure 6: Training/test accuracy curve on Climate Model Simulation Crashes data set
Figure 7: Training/test accuracy curve on WDBC data set
Figure 8:Training/test accuracy curve on Diabetes data set

5 Conclusion

This paper proposes a general machine learning framework based on the idea of weighted integration, and systematically verifies it on three representative data sets of Climate Model Simulation Crashes, WDBC and Diabetes classification. The main contributions of this research can be summarised in the following three aspects:

(1) A unified parallel integrated learning framework suitable for classified tasks has been built, showing good task adaptability and scalability;

(2) A dynamic exponential weighted fusion strategy based on the performance of the verification set is designed, which can adaptively allocate the integration weight according to the actual performance of the base learning device;

(3) Through interdisciplinary experimental verification, the effectiveness and robustness of this method in a variety of practical scenarios have been confirmed.

The experimental results show that the weighted integration method proposed in this paper performs well in multiple classification tasks. The evaluation results of the climate simulation collapse, breast cancer and diabetes data set show that the method has achieved ideal results in key indicators such as accuracy, precision rate, recall rate and F1 score, verifying its practical value in diagnostic applications. It is especially worth noting that the framework shows excellent prediction efficiency while maintaining high classification accuracy, with a maximum throughput of 81,518.68 samples/second, indicating that it has the ability to support the real-time prediction system.

Although this research has achieved stage results in the design and verification of dynamic weighted integration strategies, there are still the following limitations, which need to be further improved in subsequent research: the experiment only selects three public data sets of Climate Model Simulation Crashes, WDBC and Diabetes. Although they cover climate simulation, medical diagnosis and other fields, the sample size (540~768) and feature dimensions (8~30 dimensions) are both in the middle and low scale. It does not verify the performance of the method in the scenario of large-scale data (such as million-level samples and more than 100-dimensional features), and cannot fully determine the stability efficiency and generalisation ability of Exp-Bagging when the data volume surges. The current experiment is based on the design of static data sets, which does not simulate the common stream data scenarios in practical applications (such as real-time medical monitoring data, dynamic financial transaction data), nor does it target the adaptive adjustment ability.

In future research, in-depth exploration will be continued in the following directions. First, the combination strategy of complex base learners integrating deep learning models will be studied to further improve the model's expressive ability. Second, an online learning mechanism adaptable to dynamic data distribution will be developed to enhance the model's adaptability in streaming data scenarios. Third, the framework will be extended to more key fields such as financial risk assessment and industrial fault detection. Finally, model compression and acceleration technologies will be explored to improve deployment efficiency in resource-limited environments.

6 References

[1] Sirocchi, C., Bogliolo, A., & Montagna, S. (2024). Medical-informed machine learning: integrating prior knowledge into medical decision systems. BMC Medical Informatics and Decision Making, 24(Suppl 4), 186. https://doi.org/10.1186/s12911-024-02582-4

[2] Chou, J. S., & Chen, K. E. (2024). Optimizing investment portfolios with a sequential ensemble of decision tree-based models and the FBI algorithm for efficient financial analysis. Applied Soft Computing, 158, 111550. https://doi.org/10.1016/j.asoc.2024.111550

[3] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. Ieee Access, 11, 80348-80391. https://doi.org/10.1109/ACCESS.2023.3296444

[4] Wang, D. N., Li, L., & Zhao, D. (2022). Corporate finance risk prediction based on LightGBM. Information Sciences, 602, 259-268. https://doi.org/10.1016/j.ins.2022.04.058

[5] Opanasenko, V. M., Fazilov, S. K., Mirzaev, O. N., & Sa'dullo ugli Kakharov, S. (2024). An ensemble approach to face recognition in access control systems. Journal of Mobile Multimedia, 20(3), 749-768. https://doi.org/10.13052/jmm1550-4646.20310

[6] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14(2), 241-258. https://doi.org/10.1007/s11704-019-8208-z

[7] Ye, M., Lan, X., Leng, Q., & Shen, J. (2020). Cross-modality person re-identification via modality-aware collaborative ensemble learning. IEEE Transactions on Image Processing, 29, 9387-9399. https://doi.org/10.1109/TIP.2020.2998275

[8] Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., & Naved, M. (2023). Applications of artificial intelligence in business management, e-commerce and finance. Materials Today: Proceedings, 80, 2610-2613. https://doi.org/10.1016/j.matpr.2021.06.419

[9] Zhou, H., Zhang, J., Zhou, Y., Guo, X., & Ma, Y. (2021). A feature selection algorithm of decision tree based on feature weight. Expert Systems with Applications, 164, 113842. https://doi.org/10.1016/j.eswa.2020.113842

[10] He, Z., Wu, Z., Xu, G., Liu, Y., & Zou, Q. (2021). Decision tree for sequences. IEEE transactions on Knowledge and Data Engineering, 35(1), 251-263. https://doi.org/10.1109/TKDE.2021.3075023

[11] Pekel, E. (2020). Estimation of soil moisture using decision tree regression. Theoretical and Applied Climatology, 139(3), 1111-1119. https://doi.org/10.1007/s00704-019-03048-8

[12] Sagi, O., & Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree. Information sciences, 572, 522-542. https://doi.org/10.1016/j.ins.2021.05.055

[13] Chanmee, S., & Kesorn, K. (2023). Semantic decision Trees: A new learning system for the ID3-Based algorithm using a knowledge base. Advanced engineering informatics, 58, 102156. https://doi.org/10.1016/j.aei.2023.102156

[14] He, S., Wu, J., Wang, D., & He, X. (2022). Predictive modeling of groundwater nitrate pollution and evaluating its main impact factors using random forest. Chemosphere, 290, 133388. https://doi.org/10.1016/j.chemosphere.2021.133388

[15] Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 237, 121549. https://doi.org/10.1016/j.eswa.2023.121549

[16] Dinh, T. P., Pham-Quoc, C., Thinh, T. N., Do Nguyen, B. K., & Kha, P. C. (2023). A flexible and efficient FPGA-based random forest architecture for IoT applications. Internet of Things, 22, 100813. https://doi.org/10.1016/j.iot.2023.100813

[17] Sevinç, E. (2022). An empowered AdaBoost algorithm implementation: A COVID-19 dataset study. Computers & Industrial Engineering, 165, 107912. https://doi.org/10.1016/j.cie.2021.107912

[18] Huang, X., Li, Z., Jin, Y., & Zhang, W. (2022). Fair-AdaBoost: Extending AdaBoost method to achieve fair classification. Expert Systems with Applications, 202, 117240. https://doi.org/10.1016/j.eswa.2022.117240

[19] Germain, P., Lacasse, A., Marchand, M., Shanian, S., & Laviolette, F. (2009). From PAC-Bayes bounds to KL regularization. Advances in neural information processing systems, 22.

[20] Ngo, G., Beard, R., & Chandra, R. (2022). Evolutionary bagging for ensemble learning. Neurocomputing, 510, 1-14. https://doi.org/10.1016/j.neucom.2022.08.055

[21] Al-Mudhafar, W. J., Abbas, M. A., & Wood, D. A. (2022). Performance evaluation of boosting machine learning algorithms for lithofacies classification in heterogeneous carbonate reservoirs. Marine and Petroleum Geology, 145, 105886. https://doi.org/10.1016/j.marpetgeo.2022.105886

[22] Dong, J., Chen, Y., Yao, B., Zhang, X., & Zeng, N. (2022). A neural network boosting regression model based on XGBoost. Applied Soft Computing, 125, 109067. https://doi.org/10.1016/j.asoc.2022.109067

[23] Hajjouz, A., & Avksentieva, E. Y. (2025). Enhancing and extending CatBoost for accurate detection and classification of DoS and DDoS attack subtypes in network traffic. Научно-технический вестник информационных технологий, механики и оптики, 25(1), 114-127. https://doi.org/10.17586/2226-1494-2025-25-1-114-127

[24] Iranzad, R., & Liu, X. (2024). A review of random forest-based feature selection methods for data science education and applications. International Journal of Data Science and Analytics, 1-15.

[25] Zhang, B., Li, Y., & Chai, Z. (2022). A novel random multi-subspace based ReliefF for feature selection. Knowledge-Based Systems, 252, 109400. https://doi.org/10.1016/j.knosys.2022.109400

[26] Xing, H. J., Liu, W. T., & Wang, X. Z. (2024). Bounded exponential loss function based AdaBoost ensemble of OCSVMs. Pattern Recognition, 148, 110191. https://doi.org/10.1016/j.patcog.2023.110191

[27] Ileberi, E., Sun, Y., & Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. IEEE access, 9, 165286-165294. https://doi.org/10.1109/ACCESS.2021.3134330

[28] Kalagotla, S. K., Gangashetty, S. V., & Giridhar, K. (2021). A novel stacking technique for prediction of diabetes. Computers in Biology and Medicine, 135, 104554. https://doi.org/10.1016/j.compbiomed.2021.104554

[29] Chugg, B., Wang, H., & Ramdas, A. (2023). A unified recipe for deriving (time-uniform) PAC-Bayes bounds. Journal of Machine Learning Research, 24(372), 1-61.