# A novel multi-stage ensemble method for noisy labels using sample selection and label correction

*Abstract*—**Machine learning has made significant progress in data classification. However, due to their excellent feature learning capabilities, they may remember all the label information in the training dataset, leading to a deterioration in generalization performance when trained on noisy labeled datasets commonly found in real-world problems. In this paper, we propose a multi-stage ensemble method with sample selection and label correction to build an effective classification model under noisy labels. The proposed method uses silhouette coefficient thresholds optimized by the optimization method to screen for clean and noisy sample sets. Then, by re-labeling noisy samples at the end of each stage and iteratively updating the clean dataset, it allows models trained at each stage to learn different features, making more comprehensive use of the dataset compared to traditional sample selection methods. By integrating these models, the proposed multi-stage ensemble method demonstrates strong generalization performance. Experimental results on benchmark and real-world datasets show that the proposed method outperforms the comparison methods in classifying datasets with noisy labels.**

*Index Terms*—**Noisy label, sample selection, label correction, ensemble model**

## I. Introduction

Machine learning models rely on annotations in the training dataset, but traditional annotation requires a significant amount of time and cost to build. As an alternative and cost-effective approach, methods such as manual labeling by a single annotator [1], using web crawlers [2], and employing machine-generated labels [3] have been widely used for efficient labeling. However, these labeling processes are prone to introducing incorrect labels (referred to as noisy labels) into the training dataset. In the real world, it is inevitable that a large amount of labeled data contains noisy labels for various reasons (such as misjudgment by the annotator). Directly training machine learning models on noisy datasets may lead to overfitting and reduced robustness, as their strong learning ability allows them to fit both clean and noisy labels simultaneously. Noisy label samples can impair the performance of machine learning models, resulting in decreased generalization performance [4], [5]. Therefore, introducing learning methods that are robust to noisy labels is crucial for training on datasets with noisy labels.

Noisy label learning (NLL) has garnered widespread attention as a method to mitigate the negative impacts of noisy labels [6]. Various types of NLL methods have been proposed. The methods for achieving robust machine learning in the presence of noisy labels can be broadly categorized into four types: 1) Robust loss functions [7]–[11], which aim to design loss functions that can withstand noisy labels. 2) Loss reweighting [12], which reduces the influence of incorrect labels during training by lowering their weights. 3) Sample selection [13], [14], which typically relies on model predictions to filter clean samples and uses semi-supervised methods to train models on newly partially labeled datasets. 4) Label correction [15], [16], which learns the relationship between clean and noisy labels and replaces the latter with highly reliable pseudo-labels.

Designing robust loss functions and loss reweighting often performs poorly when the noise ratio is high or the number of classes is large, as they either overlook certain information about noisy labels or require strong statistical assumptions. Sample selection methods filter clean samples based on specific criteria (such as loss values) and train only on these samples [17], [18]. Sample selection methods are very effective in reducing the negative impact of noisy labels, but they do not fully utilize all samples in the dataset since noisy samples are not used for training. Consequently, many sample selection methods treat the remaining noisy samples as unlabeled and assign pseudo-labels to them through semi-supervised learning, achieving the state-of-the-art performance [19], [20]. Although these methods show promising performance, they only utilize a subset of training samples. Therefore, when the dataset is highly contaminated, performance may significantly degrade due to the insufficient number of clean samples available for training the model. However, most methods require knowledge of the specific ratio of noisy labels when dealing with datasets containing noisy labels, which is often difficult to ascertain in practical applications [17]. Additionally, some methods focus on improving the reliability of labels to enhance the quality of the dataset. These methods primarily rely on constructing complex noise models to identify and replace erroneous labels. These methods primarily rely on constructing complex noise models to identify and replace erroneous labels [21].

Recent research trends have leaned towards utilizing deep neural networks (DNNs) to achieve this goal [22]. For instance, Tanaka et al. [23] developed a comprehensive framework aimed at simultaneously optimizing the parameters of the classifier and the labels. Yuan et al. [24] trained multiple networks on different subsets of the dataset and corrected the labels of samples with mismatched predictions. Furthermore, there are label correction methods based on the Co-teaching strategy. Mandal et al. [25] extracted samples with both small and large losses from each mini-batch and constructed average features for each class based on the Co-teaching network, then performed label correction based on the distance between sample features and class average features. However, these methods are specifically tailored for the Co-teaching network and require knowledge of the noise ratio in the dataset.

Additionally, these methods face a significant challenge in selecting clean samples, as the chosen samples must have sufficiently high reliability to allow for the modification of training labels. To address this issue, this study proposes an integrated method named Threshold-Optimized FWAdaBoost enhanced by HPSOBOA optimization (HBT-FWAdaBoost) for noisy label learning based on sample selection and label correction. This method distinguishes between clean and noisy datasets using silhouette coefficients, making the selection of the silhouette coefficient threshold crucial. Therefore, this paper adaptively optimizes this threshold using the HPSOBOA (Hybrid BOA with PSO) method [26]. In each training round, label correction based on prototypes is performed on noisy labels, and the corrected noisy labels will be used for the next round of model learning tasks. During the training process, the noise ratio of the dataset gradually decreases due to repeated label corrections. This approach allows for the complete utilization of the dataset, enabling each model to learn different features through iterative datasets while maintaining robustness against noisy labels. The proposed method does not require any additional information that is difficult to obtain in practice, such as validated clean samples and the noise ratio of the dataset, to effectively correct noisy label samples. Therefore, the proposed method can be directly applied to real-world problems.

The main contributions of this paper can be summarized as follows:

- We propose an effective method to construct an ensemble model under noisy label datasets.
- We introduce a novel dataset label correction method that utilizes prototype-based label correction to obtain a high-quality dataset, allowing models to capture diverse features based on different weights and corrected datasets.
- Our re-labeling method does not require any additional information about the contaminated dataset, such as validated clean samples and the noise ratio of the dataset. The proposed method demonstrates state-of-the-art performance on noise-injected benchmark datasets.

## II. Related Work

In the research of noisy label learning (NLL), there are mainly two approaches to mitigate the impact of incorrect labels on model training: loss-driven methods and data-driven methods. Loss-driven methods primarily focus on constructing robust loss functions to reduce the strength of the supervisory signals generated by incorrect labels [27]. However, loss-driven methods have certain limitations, especially in the estimation of the transition matrix, which often relies on the accurate estimation of noisy class posteriors, a task that is often challenging to achieve. Moreover, loss-driven methods may require additional information, such as the noise ratio of the dataset, which are difficult to obtain in practical applications. In contrast, data-driven methods offer a more direct and effective solution. The core of these methods lies in preprocessing the data to reduce the impact of incorrect labels during training, thereby fostering more robust models. Data-driven methods reduce the negative impact of noisy data through the following approaches: 1) Selecting Reliable Samples. 2) Correcting Incorrect Labels. 3) Reweighting Samples. 4) Smoothing Label Distributions. Although these data-driven methods have shown certain effects in practice, they also face challenges, especially in accurately identifying and handling noisy data. A key challenge is how to effectively identify noisy data without relying too much on heuristic rules to avoid overfitting the model to noisy data. Compared to loss-driven methods, data-driven methods provide a more direct way to reduce the impact of noisy labels by directly manipulating the data. A main research direction for these methods is to develop more sophisticated data preprocessing techniques to better identify and utilize clean information in the data, thereby improving model performance in noisy label learning tasks. With further research, data-driven methods are expected to play a greater role in the field of noisy label learning, providing a solid data foundation for building more robust models.

## III. Method

### A. Problem Setting

In many machine learning tasks, the labels in the training set are assumed to be accurate. However, in practice, labels may contain noise due to annotation errors, data collection limitations, or other factors. These noisy labels can significantly impact the training and generalization performance of classifiers, especially in datasets with limited data or class imbalance.

Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ represents the features of the $i$-th sample and $y_i \in \mathcal{Y}$ is its corresponding label, asymmetric noise is introduced by randomly modifying some of the labels from their true values $\hat{y}_i$ to other categories $y_i \neq \hat{y}_i$. The noise distribution is expressed as:

$$P(y_i \neq \hat{y}_i \mid x_i) = \eta(x_i), \tag{1}$$

where $\eta(x_i)$ denotes the asymmetric noise rate, reflecting the probability that a label is misclassified as another category for a given sample $x_i$. This method of introducing noisy labels simulates real-world uneven annotation errors.

Next, we briefly introduce the basic models of Zeroth-Order Evolutionary Fuzzy Systems (EFS) [28] and Fuzzy Weighted Adaptive Boosting (FWAdaBoost) [29].

### B. Zeroth-Order EFS

Taking the AnYa type of fuzzy rules as an example, a standard zeroth-order EFS consists of C IF-THEN rules, one rule for each class, in the following form:

$$R_c : \text{If } (x \sim a_{c,1}) \text{ or} \ldots \text{or } (x \sim a_{c,P_c}) \text{ then } (y = c) \tag{2}$$

Here, $a_{c,i}$ represents the $i$-th premise (prototype) of rule $R_c$, and $P_c$ is the total number of prototypes associated with that rule.

The prototypes of a zeroth-order EFS are determined based on the integrative characteristics and mutual distances of the

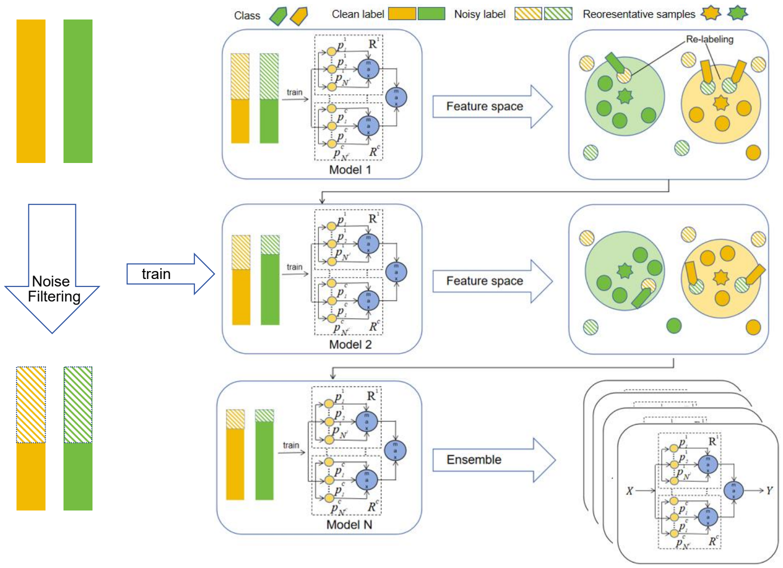


Fig. 1. Architecture of the proposed method (HBT-FWAdaBoost)

data samples through a one-time, non-iterative process. These prototypes constitute the knowledge base of the system for reasoning and decision-making.

In the decision-making process, each IF-THEN rule generates a confidence score $\lambda_c(x)$ for a given sample $x$. For example, in SOFIS, this score is calculated based on the similarity between $x$ and the nearest prototype $a_{c,n^*}$:

$$\lambda_c(x) = e^{-\|x-a_{c,n^*}\|^2} \tag{3}$$

where $n^*$ indicates the index of the prototype most similar to $x$.

Ultimately, the predicted class $\hat{y}$ for sample $x$ is determined by the rule with the highest confidence score:

$$\hat{y} = c^*; \quad \text{where} \quad c^* = \arg\max_{c=1,2,\ldots,C}(\lambda_j(x)). \tag{4}$$

### C. FWAdaBoost

FWAdaBoost is a novel boosting algorithm used to construct powerful fuzzy ensemble classifiers based on zeroth-order FIS. FWAdaBoost utilizes the confidence scores produced by the ensemble components in the sample weight update, as shown below, which gives more weight to difficult-to-classify samples, thus forming a more precise ensemble classification boundary:

$$w_{t,k} = \frac{w_{t-1,k}e^{-\alpha_t\phi_{t,k}}}{W_t} \tag{5}$$

Here, $w_{t,k}$ represents the weight of sample $x_k$ in the $t$-th iteration, $\alpha_t$ is the weight of the base classifier $h_t(x)$, $W_t$ is the normalization factor, and $\phi_{t,k}$ is the difference between the confidence score of the true class and the highest confidence score (excluding the true class).

Another feature of FWAdaBoost is the use of confidence scores in the generation of ensemble outputs, as shown below, which allows more confident predictions to play a greater role in the final output:

$$F(x) = \arg\max_{c=1,2,\ldots,C}\left(\sum_{i=1}^{T}\alpha_i\hat{\phi}_i\hat{Y}_{i,c}\right) \tag{6}$$

where $\hat{\phi}_t$ is the difference between the highest and second-highest confidence scores, and $\hat{Y}_{t,c}$ is the encoded vector based on the predicted label $\hat{y}_t$.

### D. Silhouette Coefficient

In data analysis, especially when dealing with datasets that come with labels, evaluating the accuracy of sample labels is an important task. The Silhouette Coefficient can serve as a quantitative tool to help us identify samples that may be incorrectly labeled, i.e., noise labels.

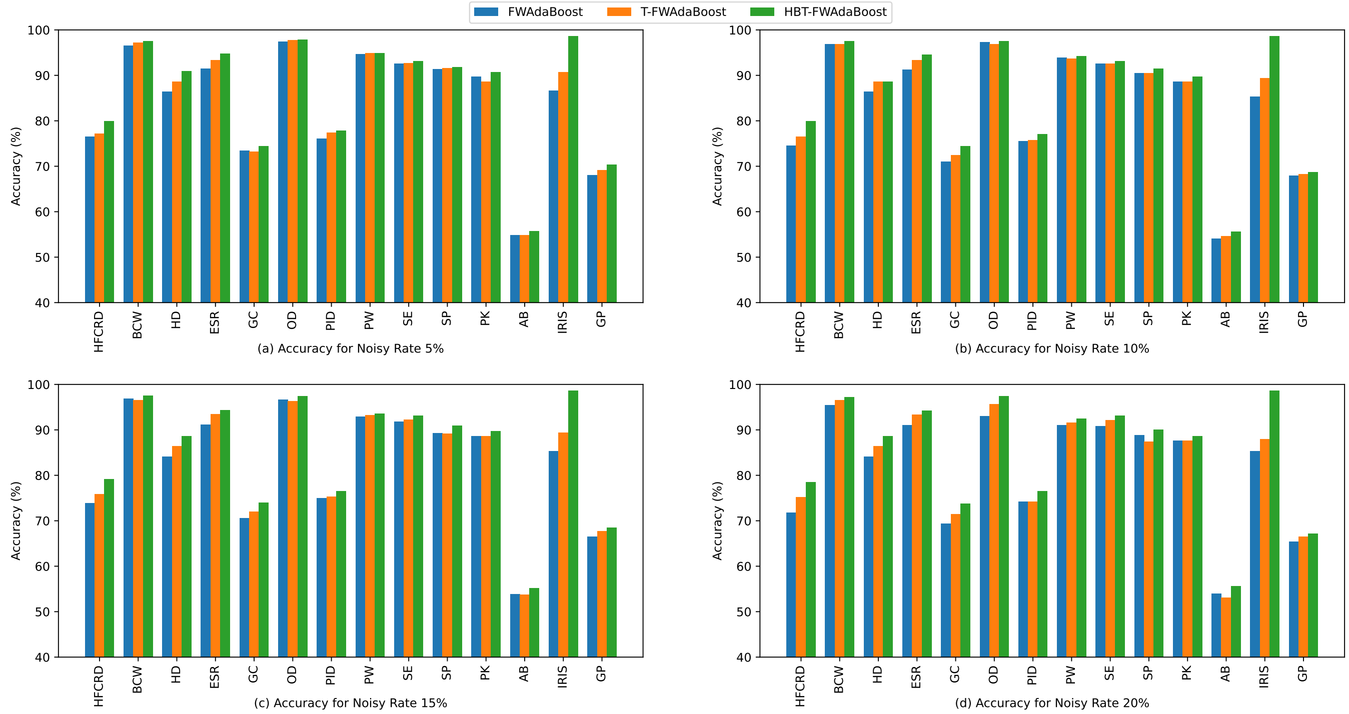
It is calculated using the following formula for sample $i$:

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{7}$$

In this formula, $a(i)$ represents the average distance between sample $i$ and all other samples in the same category, while $b(i)$ represents the average distance between sample $i$ and samples in the nearest different category. The Silhouette Coefficient, $s(i)$, thus provides a measure of how well sample $i$ is matched to its own category in relation to the nearest other category.

- **Close to 1**: Sample $i$ is highly similar to samples in its category and dissimilar to samples in other categories, indicating that the sample's label is likely accurate.
- **Close to -1**: Sample $i$ is dissimilar to samples in its category and similar to samples in other categories, which is a strong indicator of a noise label.

Therefore, the Silhouette Coefficient values calculated serve as a soft indicator of the trust in the class labels for $x_k$. By using the Silhouette Coefficient of the samples, we can



(a) Accuracy for Noisy Rate 5%

(b) Accuracy for Noisy Rate 10%

(c) Accuracy for Noisy Rate 15%

(d) Accuracy for Noisy Rate 20%

Fig. 2. Comparison of ablation study results

identify those samples that are inconsistent with their labeled categories, thereby improving the quality of the dataset and the accuracy of subsequent analyses.

### E. HPSOBOA

HPSOBOA is a hybrid optimization algorithm that combines the advantages of Particle Swarm Optimization (PSO) and Butterfly Optimization Algorithm (BOA). The algorithm enhances global search capabilities and convergence speed through chaotic initialization and dynamic parameter adjustment. This study applies the HPSOBOA method to optimize the silhouette coefficient threshold to adapt to the characteristics of different datasets, thereby achieving accurate discrimination between non-noise label samples and noise label samples in the dataset. HPSOBOA initializes the swarm using the Cubic chaotic map to enhance the diversity of the initial population. The Cubic map is given by:

$$z_{n+1} = \alpha z_n^3 - \beta z_n \tag{8}$$

where $\alpha$ and $\beta$ are parameters controlling the chaotic behavior. During the iterative process, HPSOBOA updates the butterfly positions by combining the velocity update mechanism of PSO and the search strategy of BOA:

$$X_i^{t+1} = w \cdot X_i^t + c_1 \cdot r_1 \cdot (p_i^t - X_i^t) + c_2 \cdot r_2 \cdot (g^t - X_i^t) \tag{9}$$

where $X_i^t$ represents the position of the $i$-th butterfly at iteration $t$, $p_i^t$ is its personal best position, $g^t$ is the global best position, $w$, $c_1$, and $c_2$ are algorithm parameters, and $r_1$ and $r_2$ are random numbers. To balance global and local searches, HPSOBOA dynamically adjusts the parameter $a$:

$$a(t) = a_{initial} - \frac{(a_{initial} - a_{final})}{T} \cdot t \tag{10}$$

where $a_{initial}$ and $a_{final}$ are the initial and final values of parameter $a$, respectively, $T$ is the total number of iterations, and $t$ is the current iteration number. The algorithm terminates when a preset number of iterations is reached or when the solution quality meets a specific threshold.

HPSOBOA achieves efficient problem-solving for complex optimization problems through the aforementioned mechanisms.

### F. The proposed HBT-FWAdaBoost

The proposed method uses a silhouette coefficient threshold to differentiate between clean data and noisy data within a dataset. During each training iteration, noise samples are corrected based on a prototype approach: the distance from each sample to every prototype is calculated, and voting is conducted based on the nearest n prototypes. If the number of votes exceeds half, the label of the sample is changed to the majority label from the votes. The corrected noise samples

TABLE I
KEY INFORMATION OF DATASETS

| Abbreviation | Dataset | Samples | Features | Classes |
|---|---|---|---|---|
| HFCRD | Heart Failure Clinical Records | 299 | 12 | 2 |
| BCW | Breast Cancer Wisconsin | 569 | 30 | 2 |
| HD | Heart Disease | 303 | 13 | 2 |
| ESR | Epileptic seizure recognition | 11500 | 178 | 2 |
| GC | German credit | 1000 | 24 | 2 |
| OD | Occupancy detection | 17895 | 5 | 2 |
| PID | Pima Indians diabetes | 768 | 8 | 2 |
| PW | Phishing websites | 11055 | 30 | 2 |
| SE | Seismic | 2584 | 18 | 2 |
| SP | Spambase | 4601 | 57 | 2 |
| PK | PARKINSONS | 197 | 22 | 2 |
| AB | Abalone | 4177 | 8 | 3 |
| IRIS | Iris | 150 | 4 | 3 |
| GP | Gesture phase segmentation | 9901 | 17 | 5 |

are then used for the next round of model learning tasks (in other words, incorporating the corrected noise samples into the clean dataset for training). Throughout the training process, the noise ratio in the dataset is gradually reduced through repeated label corrections. This approach allows for the complete utilization of the dataset, enabling each model to learn various features through iterative processing of the dataset and providing robustness against noisy labels. The method does not require any additional information that is difficult to obtain in practice, such as valid clean samples and the noise ratio of the dataset, to effectively deal with noisy label samples. The architecture of the proposed method is shown in Fig. 1.

Here are the mathematical representations for the key steps of the method:

1) Noise Classification Based on Silhouette Coefficient Threshold: If a sample's silhouette coefficient $r$ is less than the threshold $\theta$, the sample is classified as noise ($f(r) = 0$); otherwise, it is classified as clean data ($f(r) = 1$).

$$f(r) = \begin{cases} \text{clean}, & \text{if } r \geq \theta \\ \text{noisy}, & \text{otherwise} \end{cases} \tag{11}$$

2) Prototype-Based Noise Sample Correction: For each sample $\mathbf{x}_k$ and prototype $\mathbf{p}_i$, the similarity $\lambda^i(\mathbf{x}_k)$ is calculated using the following formula:

$$\lambda^i(\mathbf{x}_k) = e^{-\gamma \|\mathbf{x}_k - \mathbf{p}_i\|^2} \tag{12}$$

For each sample, whether to correct is determined by the voting results from the nearest $n$ prototypes,Record the votes as the label with the most votes:

$$f(\text{votes}) = \begin{cases} \text{Original label}, & \text{if votes} < \frac{n}{2}, \\ \text{Majority label}, & \text{if votes} \geq \frac{n}{2}. \end{cases} \tag{13}$$

3) Integration of $f(\text{votes})$: Integrate the Majority label samples into the clean dataset for further training:

$$\mathcal{D}' = \mathcal{D} \cup \{\mathbf{x}_k | \text{Majority label}(\mathbf{x}_k)\} \tag{14}$$

This method ensures that the model can learn different features from the complete dataset and maintain robustness against noisy labels without additional information that is hard to obtain.

## IV. EXPERIMENTAL INVESTIGATIONS

The model proposed in this study was developed on the MATLAB 2024 platform, and performance evaluations were conducted on a laptop equipped with a dual-core i7 processor at 4.20 GHz×2 and 16-GB of RAM. All numerical experiments were carried out in an offline scenario, and the reported results are based on the average of 20 Monte Carlo experiments to introduce a certain degree of randomness. This approach ensures the robustness and reliability of our experimental outcomes and also reflects the model's performance under different noise percentage (5%, 10%, 15%, 20%). The method of adding noise is asymmetric random injection. To ensure a fair comparison, all comparison methods used default parameters on all datasets. The goal of this study is to demonstrate that the HBT-FWAdaBoost method can exhibit strong robustness when dealing with noisy data.

Since HBT-FWAdaBoost is applicable to both binary and multi-class classification problems, experiments utilized benchmark datasets of these two types. In this study, eleven numerical binary classification problems, three numerical multi-class classification problems were considered. Key information about the datasets is presented in Table I.

### A. Performance Demonstration

To compare the accuracy of the HBT-FWAdaBoost method in handling noisy label data, we contrasts it with three traditional algorithms: SVM, KNN, and Decision Tree. Additionally, we included the advanced methods in our comparison: 1. FWAdaBoost [29], a fuzzy-weighted adaptive ensemble learning algorithm that combines multiple weak classifiers to improve classification performance and has some robustness to noise. 2. PEER [10], a noise-resilient learning method that evaluates classifier predictions by constructing ”peer” samples,

TABLE II
Comparison of Accuracy Rates of Different Algorithms under Various Noise Conditions

| DATESET | Noisy Rate | SVM | KNN | Decision Tree | FWAdaBoost | PEER | DIM | ours |
|---|---|---|---|---|---|---|---|---|
| HFCRD | 5% | 79.20% | 75.84% | 78.52% | 76.51% | 70.20% | 78.10% | **79.87%** |
|  | 10% | 78.52% | 73.15% | 75.84% | 74.48% | 70.20% | 78.10% | **79.87%** |
|  | 15% | 77.18% | 72.48% | 73.15% | 73.83% | 70.50% | 76.30% | **79.19%** |
|  | 20% | 71.14% | 71.81% | 71.14% | 71.81% | 70.20% | 75.40% | **78.52%** |
| BCW | 5% | 96.13% | 97.18% | 83.80% | 96.48% | 93.00% | 96.00% | **97.54%** |
|  | 10% | 95.78% | 96.13% | 85.92% | 96.83% | 93.00% | 96.00% | **97.54%** |
|  | 15% | 95.07% | 95.42% | 78.17% | 96.83% | 92.90% | 96.20% | **97.54%** |
|  | 20% | 94.01% | 94.72% | 76.76% | 95.42% | 92.60% | 95.80% | **97.18%** |
| HD | 5% | 86.36% | 81.82% | 72.73% | 86.36% | 79.80% | 79.00% | **90.91%** |
|  | 10% | 84.09% | 79.55% | 72.73% | 86.36% | 79.80% | 79.00% | **88.64%** |
|  | 15% | 81.82% | 77.27% | 68.18% | 84.09% | 83.10% | 80.60% | **88.64%** |
|  | 20% | 77.27% | 75.00% | 63.64% | 84.09% | 82.30% | 79.40% | **88.64%** |
| ESR | 5% | 83.32% | 92.57% | 87.27% | 91.43% | 26.60% | 94.10% | **94.78%** |
|  | 10% | 83.27% | 92.10% | 82.97% | 91.20% | 26.60% | 94.10% | **94.56%** |
|  | 15% | 83.53% | 92.17% | 78.71% | 91.17% | 26.70% | 93.90% | **94.33%** |
|  | 20% | 83.22% | 88.59% | 74.97% | 91.03% | 25.80% | **94.30%** | 94.19% |
| GC | 5% | 73.40% | 70.20% | 69.20% | 73.40% | 64.90% | 68.00% | **74.40%** |
|  | 10% | 72.40% | 68.40% | 66.40% | 71.00% | 64.90% | 68.00% | **74.40%** |
|  | 15% | 69.60% | 67.40% | 64.00% | 70.60% | 64.90% | 66.40% | **74.00%** |
|  | 20% | 68.00% | 65.20% | 61.60% | 69.40% | 64.50% | 65.40% | **73.80%** |
| OD | 5% | **98.62%** | 96.99% | 90.57% | 97.37% | 84.90% | 94.10% | 97.75% |
|  | 10% | 97.28% | 96.69% | 85.16% | **97.34%** | 84.90% | 94.10% | 96.81% |
|  | 15% | 95.37% | **96.62%** | 80.89% | 96.60% | 86.80% | 93.80% | 96.27% |
|  | 20% | 95.41% | 93.63% | 80.64% | 92.96% | 86.70% | 93.30% | **95.67%** |
| PID | 5% | 76.04% | 75.78% | 74.74% | 76.04% | 72.50% | 71.00% | **77.34%** |
|  | 10% | 75.00% | 72.66% | 71.62% | 75.78% | 72.50% | 71.00% | **77.08%** |
|  | 15% | 74.74% | 71.35% | 71.09% | 75.26% | 72.10% | 72.10% | **76.56%** |
|  | 20% | 74.74% | 69.53% | 68.23% | 74.22% | 72.40% | 71.50% | **76.56%** |
| PW | 5% | 92.02% | 92.65% | 93.83% | 94.63% | 87.60% | 89.90% | **94.84%** |
|  | 10% | 91.32% | 91.61% | 93.11% | **93.94%** | 87.60% | 89.90% | 93.67% |
|  | 15% | 91.10% | 91.48% | 89.92% | 92.85% | 87.60% | 89.20% | **93.54%** |
|  | 20% | 88.52% | 89.78% | 88.53% | 91.08% | 87.50% | 87.50% | **92.47%** |
| SE | 5% | 93.11% | 92.26% | 89.55% | 92.57% | 63.30% | 68.10% | **93.11%** |
|  | 10% | 93.11% | 92.18% | 88.85% | 92.57% | 63.30% | 68.10% | **93.11%** |
|  | 15% | 93.11% | 92.65% | 86.07% | 91.80% | 63.90% | 66.00% | **93.11%** |
|  | 20% | 93.11% | 91.18% | 84.68% | 90.79% | 59.90% | 63.40% | **93.11%** |
| SP | 5% | 91.70% | 89.17% | 89.04% | 91.39% | 89.50% | 90.70% | **91.83%** |
|  | 10% | 89.70% | 88.35% | 86.87% | 90.48% | 89.50% | 90.70% | **91.43%** |
|  | 15% | 88.09% | 86.87% | 83.44% | 89.22% | 89.30% | 90.20% | **90.96%** |
|  | 20% | 86.04% | 86.35% | 80.30% | 88.87% | 88.80% | **90.40%** | 90.04% |
| PK | 5% | 86.60% | 88.66% | 87.63% | 89.69% | 68.10% | 74.60% | **90.72%** |
|  | 10% | 85.57% | 87.63% | 86.60% | 88.66% | 68.10% | 74.60% | **89.69%** |
|  | 15% | 84.54% | 86.60% | 84.54% | 88.66% | 67.80% | 75.70% | **89.69%** |
|  | 20% | 81.44% | 84.54% | 82.47% | 87.63% | 67.80% | 75.00% | **88.66%** |
| AB | 5% | 55.70% | 52.06% | 50.34% | 54.89% | N/A | N/A | **55.75%** |
|  | 10% | 54.93% | 50.81% | 48.80% | 54.02% | N/A | N/A | **55.65%** |
|  | 15% | 54.93% | 50.62% | 47.37% | 53.88% | N/A | N/A | **55.12%** |
|  | 20% | 54.89% | 50.14% | 47.27% | 53.93% | N/A | N/A | **55.56%** |
| IRIS | 5% | 94.67% | 93.33% | 93.33% | 86.67% | N/A | N/A | **98.67%** |
|  | 10% | 93.33% | 92.22% | 90.67% | 85.33% | N/A | N/A | **98.67%** |
|  | 15% | 93.33% | 90.67% | 89.33% | 85.33% | N/A | N/A | **98.67%** |
|  | 20% | 92.00% | 81.33% | 85.33% | 85.33% | N/A | N/A | **98.67%** |
| GP | 5% | 67.05% | 67.14% | 58.09% | 69.12% | N/A | N/A | **70.38%** |
|  | 10% | 66.48% | 66.06% | 57.64% | 68.31% | N/A | N/A | **68.66%** |
|  | 15% | 66.02% | 65.79% | 56.26% | 67.74% | N/A | N/A | **68.43%** |
|  | 20% | 65.79% | 64.87% | 55.57% | 65.44% | N/A | N/A | **67.16%** |

implicitly encoding information about noise and true labels to counteract label noise. This method is designed to enhance the robustness of classifiers against label noise, although it is primarily intended for binary classification tasks. 3. DMI [11], a new information-theoretic loss function based on a generalized version of mutual information, defining loss by calculating the correlation between classifier output and noisy labels, ensuring information monotonicity and relative invariance. Due to the limitations of PEER and DMI in handling multi-class classification, our comparative analysis focuses on binary classification tasks. For multi-class noisy label data, we only consider HBT-FWAdaBoost, SVM, KNN, Decision Tree, and FWAdaBoost in our study.

Table II shows that HBT-FWAdaBoost achieved optimal performance across all datasets and noise ratios (where the best experimental results have been shown in bold and ”N/A” in Table II indicates that the provided source code does not include functionality for multi-class classification, hence no relevant information or data is provided for the multi-class sectio). On the HFCRD dataset, our method outperformed the second-best method by 1.37% at a 5% noise ratio. Similarly, at a 5% noise ratio on the BCW dataset, our method outperformed the second-best method by 0.36%. On the ESR and GC datasets, our method outperformed other methods by 4.58% and 0.24%, respectively. Across all datasets, our method not only leads in accuracy but also demonstrates better robustness and adaptability to noise. At each noise ratio, our method shows a clear advantage over traditional algorithms (SVM, KNN, Decision Tree) and robust loss function improvement algorithms (PEER, DMI). For example, at a 5% noise ratio, our method outperforms FWAdaBoost by 1.00% on the GC dataset and DMI by 0.68% on the ESR dataset. Furthermore, at a 20% noise ratio, our method still maintained good performance while the accuracy of other methods significantly declined, indicating that our algorithm is more robust in the face of higher noise. Compared to other baseline methods, HBT-FWAdaBoost demonstrates superior classification accuracy. We conducted 20 experiments on noisy datasets with different random seeds, introducing a higher degree of label randomness. Despite different data generation, our method shows robust performance. Our method outperforms the comparison methods in all metrics. The results indicate that HBT-FWAdaBoost can enhance the model’s robustness to labeling noise.

The reasons for our method’s excellent results are as follows. First, our method screens noisy label data based on silhouette coefficients. By selecting clean samples, the reliability of label correction is further enhanced. Second, we optimize the silhouette coefficient threshold through the HPSOBOA method, avoiding the filtering of a large number of clean samples due to the characteristics of the dataset, which would prevent the classifier from learning sufficiently. Finally, by voting with representative samples, we only consider the corrected labels as reliable if the vote passes by more than half, further enhancing the model’s robustness under noisy conditions, and unreliable samples will not be used for classifier learning.

Overall, the experimental results prove the effectiveness of our method. Whether in low-noise or high-noise environments, our method has shown leading performance, especially in high-noise conditions, it is more robust compared to other methods, demonstrating the strong advantages of the HPSOBOA-FWAdaBoost method in handling noisy data.

### *B. Ablation Analysis*

In this subsection, we conducted an ablation analysis to demonstrate the effectiveness of the proposed HBT-FWAdaBoost method. In the following example, a variant of HBT-FWAdaBoost, namely T-FWAdaBoost (with a fixed threshold), is considered. T-FWAdaBoost uses a fixed threshold (set to 0 in the experimental setup) to determine noise-labeled samples, while HBT-FWAdaBoost employs the HPSOBOA method for adaptively optimizing the threshold. Fig. 2 (a)-(d) shows the comparison of classification accuracy for different models on corresponding datasets under different noise rates. From Fig. 2, it can be observed that under noise-label conditions, T-FWAdaBoost is able to construct more accurate ensemble systems than FWAdaBoost. However, the performance of T-FWAdaBoost declines in some datasets. This is because T-FWAdaBoost classifies too many samples as noise-label data due to the fixed threshold during the training process, leading to insufficient training of the classifiers. The corresponding threshold values for noise labels in each dataset are different; a fixed threshold results in the loss of many non-noise-label samples, leading to an insufficient number of training samples. Therefore, by using the HPSOBOA method to optimize the threshold for determining noise labels, the issue of non-noise label sample loss caused by a fixed threshold can be successfully addressed, allowing for more precise screening of non-noise label samples, as the characteristics of each dataset are considered in the process of distinguishing noise-label samples through the threshold. Consequently, it can be concluded from Fig. 2 that HBT-FWAdaBoost can effectively enhance the overall ensemble system’s ability to combat noise labels.

## V. Conclusion

This research explores the HBT-FWAdaBoost method, which combines sample selection and label correction to handle noisy label data. It demonstrates the superior performance and stability of HBT-FWAdaBoost across a range of benchmark problems through numerical examples. However, the loss function used in the current study has not been improved for noisy data, leaving room for further enhancement. To improve the accuracy and stability of our method, there are several directions for future work. Firstly, we will consider improving the loss function specifically for noisy data to further enhance the model’s robustness against noise. Secondly, introducing HBT-FWAdaBoost into the field of image processing will be an interesting avenue for future research. Such improvements could potentially extend the capabilities of HBT-FWAdaBoost in handling graphical data.

# References

[1] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11244–11253, 2019.

[2] Jufeng Yang, Xiaoxiao Sun, Yu-Kun Lai, Liang Zheng, and Ming-Ming Cheng. Recognition from web data: A progressive filtering approach. *IEEE Transactions on Image Processing*, 27(11):5303–5315, 2018.

[3] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[4] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[5] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[6] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153, 2022.

[7] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[9] Taoyong Cui, Jianze Li, Yuhan Dong, and Li Liu. Taotf: A two-stage approximately orthogonal training framework in deep neural networks. In *ECAI 2023*, pages 509–516. IOS Press, 2023.

[10] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020.

[11] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. *Advances in Neural Information Processing Systems*, 32, 2019.

[12] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017.

[13] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022.

[14] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. *Advances in Neural Information Processing Systems*, 33:21382–21393, 2020.

[15] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11053–11061, 2021.

[16] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020.

[17] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

[19] Ruixuan Xiao, Yiwen Dong, Haobo Wang, Lei Feng, Runze Wu, Gang Chen, and Junbo Zhao. Promix: Combating label noise via maximizing clean sample utility. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4442–4450. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[20] Filipe R Cordeiro, Ragav Sachdeva, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Longremix: Robust learning with high confidence samples in a noisy label environment. *Pattern Recognition*, 133:109013, 2023.

[21] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017.

[22] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[23] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[24] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. Iterative cross learning on noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 757–765. IEEE, 2018.

[25] Devraj Mandal, Shrisha Bharadwaj, and Soma Biswas. A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390, 2020.

[26] Mengjian Zhang, Daoyin Long, Tao Qin, and Jing Yang. A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems. *Symmetry*, 12(11):1800, 2020.

[27] Chihyeon Choi, Woojin Lee, and Youngdoo Son. Multi-stage ensemble with refinement for noisy labeled data classification. *Expert Systems with Applications*, 255:124672, 2024.

[28] Xiaowei Gu, Plamen Angelov, and Hai-Jun Rong. Local optimality of self-organising neuro-fuzzy inference systems. *Information Sciences*, 503:351–380, 2019.

[29] Xiaowei Gu and Plamen P Angelov. Multiclass fuzzily weighted adaptive-boosting-based self-organizing fuzzy inference ensemble systems for classification. *IEEE Transactions on Fuzzy Systems*, 30(9):3722–3735, 2021.