CONTINUOUS-AWARE MULTI-VIEW ATTRIBUTE WEIGHTED NAIVE BAYES

Anonymous authors
Paper under double-blind review

ABSTRACT

Attribute weighting provides a flexible way to alleviate the conditional independence assumption in naive Bayes. Within this widely studied area, multi-view attribute weighting has emerged as a new paradigm and achieved state-of-the-art performance by exploiting complementary information from multiple data representations. However, existing multi-view attribute weighting methods are primarily designed for nominal attributes, whereas continuous attributes are more prevalent in real-world applications. Although discretization enables these methods to handle continuous attributes, it removes differences between values within the same interval, limiting their ability to capture fine-grained continuous information. To address this issue, we propose a novel method called Continuous-aware Multi-view Attribute Weighted Naive Bayes (CMAWNB), which directly models continuous attributes by constructing three additional views from the perspective of complex attribute dependencies, continuous space partitions, and local group structures, respectively. Subsequently, attribute weights are independently optimized within each view, and the view-specific predictions are fused for classification. Experiments on 55 benchmark continuous datasets demonstrate the effectiveness of CMAWNB.

1 INTRODUCTION

Naive Bayes (NB) is a widely used probabilistic classifier owing to its conceptual simplicity, computational efficiency, and competitive performance. However, its attribute conditional independence assumption rarely holds in real-world applications, motivating various improvements that can be broadly grouped into four categories: structure-oriented improvements (Friedman et al., 1997; Webb et al., 2005; Meng et al., 2026), probability-oriented improvements (El Hindi, 2014; El Hindi et al., 2020; Zhang & Jiang, 2022), instance-oriented improvements (Kohavi, 1996; Frank et al., 2003; Zhang et al., 2026; Wang et al., 2026), and attribute-oriented improvements (Langley & Sage, 1994; Zaidi et al., 2013; Zhang et al., 2023; Li et al., 2025; Wu et al., 2026).

Among attribute-oriented improvements, attribute weighting provides a flexible way to alleviate the unrealistic independence assumption of NB by learning attribute-specific weights. For instance, Hall (2007) derived attribute weights from their depths in decision trees, while Lee et al. (2011) used information-theoretic criteria to determine attribute weights. Moving beyond these criteria-based methods, Zaidi et al. (2013) proposed weighting to alleviate the naive Bayes independence assumption (WANBIA), which learns attribute weights by minimizing the negative conditional log-likelihood or the mean squared error. In addition, Jiang et al. (2019) proposed class-specific attribute weighted naive Bayes (CAWNB), which learns a separate weight for each attribute within each class. Zhang et al. (2020) further introduced class-specific attribute value weighted naive Bayes (CAVWNB), assigning a separate weight to each attribute value within each class. In recent years, multi-view attribute weighting has emerged as a new paradigm, and a few multi-view learning-based attribute weighting methods have been proposed (Zhang et al., 2023; Liu et al., 2025; Wu et al., 2026) by exploiting complementary information from multiple data representation, which has achieved the state-of-the-art classification performance of attribute weighting.

Despite these advances, existing multi-view attribute weighted NB methods are predominantly designed for nominal attributes. For continuous data, they typically rely on discretization, which converts continuous attributes into nominal ones at the cost of losing distinctions among values within

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the same interval. The loss of fine-grained information substantially limits their ability to capture the intrinsic characteristics of continuous attributes. Moreover, distances computed from continuous attributes can reveal local groups of similar instances. The class information of these groups offers complementary predictive information that remains insufficiently exploited in existing multi-view attribute weighting frameworks.

To address these limitations, we propose Continuous-Aware Multi-view Attribute Weighted Naive Bayes (CMAWNB). CMAWNB directly exploits continuous attributes in the raw view to construct the label view, while incorporating class information from local instance groups. Specifically, CMAWNB employs class-specific attribute weighted Gaussian naive Bayes for the raw view, and constructs three label views using Gaussian super-parent one-dependence estimators (GSPODEs), continuous random trees (RTs), and granular ball k-nearest neighbor classifiers (GBKNNs), which exploit class-conditional linear dependencies, continuous space partitions, and distance-based local groups, respectively. Subsequently, CMAWNB learns class-specific attribute weights in the raw view and class-specific attribute value weights in the three label views through conditional log-likelihood optimization. The four views are fused to produce the prediction. Experiments on 55 benchmark continuous datasets demonstrate the effectiveness of CMAWNB.

The main contributions of this study are summarized as follows:

- We propose Continuous-aware Multi-view Attribute Weighted Naive Bayes (CMAWNB) that directly exploits continuous attributes in both raw view learning and label view construction, avoiding the loss of continuous distinctions caused by preliminary discretization.
- We design a granular-ball label view that generates attributes from the class information of local instances formed by distances between continuous attributes, which provides group-level information to complement other representations within the multi-view framework.
- We evaluate CMAWNB on 55 benchmark continuous datasets in classification accuracy, and conduct ablation study as well as case study to examine the effects of each component.

2 Related Work

Multi-view weighted naive Bayes explores underlying data representations to construct additional views and enhance classification performance. An early exploration of representation enrichment is Attribute Augmented and Weighted Naive Bayes (A^2WNB), proposed by Zhang et al. (2022), which augments the original attributes with labels predicted by random one-dependence estimators. Zhang et al. (2023) introduced Multi-view Attribute Weighted Naive Bayes (MAWNB), which uses multiple base classifiers to construct additional label views from the raw view. Enhanced Multi-view Attribute Weighted Naive Bayes (EMAWNB) uses cross-validation-based view construction and instance-specific weights to improve view reliability and adapt prediction fusion (Liu et al., 2025). Multi-view Collaborative Optimization for Weighted Naive Bayes (MCOWNB) adopts collaborative optimization to coordinate view-specific learning and prediction fusion (Zhou et al., 2025). Perturbation-driven Matrix-view Weighted Naive Bayes (PMWNB) combines input perturbations with hard label and soft probability outputs to construct diverse representations (Wu et al., 2026). Related studies have further explored instance-level information and weighting mechanisms. Instance Correlation Graph-based Naive Bayes (ICGNB) learns continuous attributes from an instance correlation graph and combines them with the original attributes for weighted Gaussian NB classification (Li et al., 2025). Dual-view Instance Weighted Naive Bayes (DIWNB) extends multi-view learning to instance weighting, using selected k-nearest neighbor classifiers to construct an additional view (Zhang et al., 2026).

Granular ball computing represents data through local groups rather than individual instances. Xia et al. (2019) introduced this framework and developed granular ball support vector machines (GB-SVM) and granular ball k-nearest neighbor classifiers. A granular ball represents local instances, with its center and radius describing the group's location and spatial extent. Its label is determined by the most frequent class within the group. GBKNN predicts class labels using sample-to-ball distances and ball labels, extending nearest-neighbor classification from individual instances to local groups. Subsequent studies have improved granular ball generation and prediction: Xia et al. (2024) investigated efficient splitting and adaptive generation strategies, while Xie et al. (2024) proposed GBG++ for stable generation and GBKNN++ to incorporate ball quality into classification.

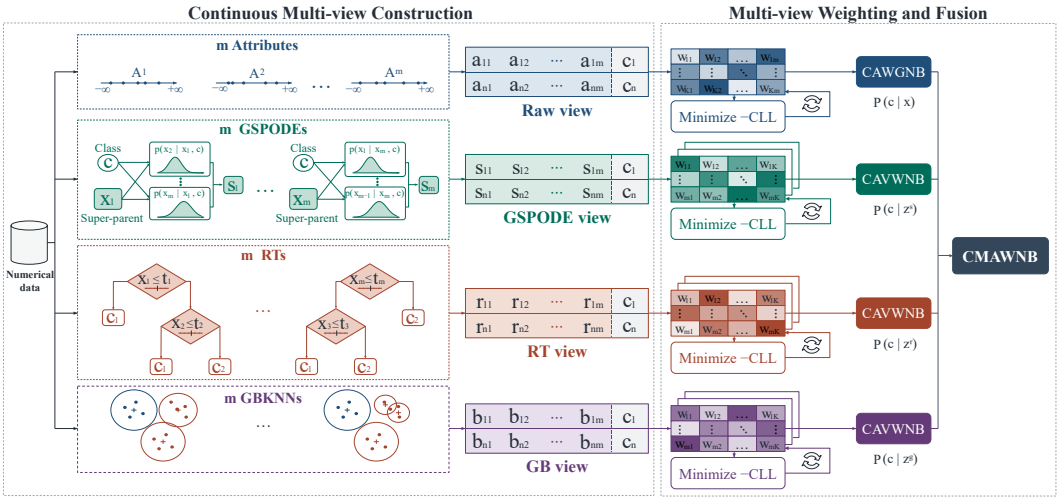


Figure 1: Overall framework of CMAWNB. (1) Continuous Multi-view Construction: Constructing the raw view directly from continuous attributes and three label views from the predictions of GSPODE, RT, and GBKNN base classifiers. (2) Multi-view Weighting and Fusion: Learning view-specific weights and fusing the estimated class-membership probabilities for classification.

3 METHOD

3.1 MOTIVATION AND FRAMEWORK

Discretization maps distinct continuous values within the same interval to a common nominal value, removing their within-interval differences. When multiple views are constructed from discretized data, these differences are unavailable to all resulting views. Attribute weighting can adjust attribute weights, but cannot recover continuous distinctions discarded during preprocessing. Retaining continuous values only in the raw view therefore leaves the inputs to label view construction subject to the same information loss. This motivates the direct use of continuous attributes in both raw view learning and label view construction. Beyond individual attribute values, distances computed from continuous attributes describe similarities between instances and can identify local groups. Representing these groups through their centers and labels summarizes their shared characteristics and local class structure. Such group-level information offers a complementary perspective to representations based on attribute dependencies and space partitions. CMAWNB therefore uses continuous attributes directly in both raw view learning and label view construction, and builds a granular ball label view to capture local group structure.

As illustrated in Figure 1, CMAWNB mainly consists of two modules: continuous multi-view construction and multi-view weighting and fusion. In the first module, it retains the raw continuous view and constructs three label views from the predictions of GSPODE, RT, and GBKNN, respectively. In the second module, class-specific attribute weights and class-specific attribute value weights are independently learned in the raw view and in each label view, respectively. Finally, the resulting class-membership probabilities are averaged for prediction.

3.2 CONTINUOUS MULTI-VIEW CONSTRUCTION

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ denote the training instances, where $\mathbf{x}_i = \langle a_{i1}, \ldots, a_{im} \rangle$ contains m continuous attribute values and y_i is the true class label. Here, n is the number of training instances and $K = |\mathcal{C}|$ is the number of classes. CMAWNB retains the raw continuous view and constructs three complementary label views using GSPODEs, RTs, and GBKNN base classifiers. Each type contains m base classifiers, whose predicted class labels serve as the attributes of the corresponding label view. For an instance $\mathbf{x}$, its representation in label view v is defined as

$$\mathbf{z}^v(\mathbf{x}) = \langle z_1^v(\mathbf{x}), \ldots, z_m^v(\mathbf{x}) \rangle, \qquad v \in \{s, r, g\}, \tag{1}$$

where $z_t^v(\mathbf{x})$ is the class label predicted by the tth base classifier associated with view v. Each label representation contains m categorical attributes. The superscripts s, r, and g denote the GSPODE, random tree, and granular ball views, respectively.

3.2.1 RAW CONTINUOUS VIEW

The raw view retains the original continuous attribute values without preliminary discretization. For subsequent raw view classification, CMAWNB estimates class-conditional densities using Gaussian distributions. The class prior $P(c)$ and the conditional density $P(a_j \mid c)$ are estimated by Eqs. (2) and (3), respectively:

$$P(c) = \frac{\sum_{i=1}^{n} \mathbb{I}(y_i = c) + 1/K}{n+1}, \tag{2}$$

$$P(a_j \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,j}} \exp\left(-\frac{(a_j - \mu_{c,j})^2}{2\sigma_{c,j}^2}\right), \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function, equal to 1 when its argument is true and 0 otherwise. The parameters $\mu_{c,j}$ and $\sigma_{c,j}^2$ are the class-conditional mean and variance of attribute j, estimated from the training instances belonging to class c.

3.2.2 GSPODE LABEL VIEW

To obtain a representation complementary to the raw view, CMAWNB uses GSPODEs to model class-conditional linear dependencies between continuous attributes. Each of the m attributes is selected in turn as the super-parent, yielding m GSPODE base classifiers.

For the tth GSPODE, the remaining attributes are assumed to be conditionally independent given the super-parent attribute A_t and the class. Under this assumption, its predicted class label is

$$z_t^s(\mathbf{x}) = \arg\max_{c \in \mathcal{C}} P(c)\, P(a_t \mid c) \prod_{\substack{j=1 \\ j \neq t}}^{m} P(a_j \mid a_t, c), \tag{4}$$

where $P(c)$ is the class prior estimated using m-estimation, and $P(a_t \mid c)$ is the super-parent's class-conditional density, estimated using the Gaussian form in Eq. (3). To account for the dependence of each child attribute on the super-parent, its conditional density is modeled as

$$P(a_j \mid a_t, c) = \frac{1}{\sqrt{2\pi\tau_{c,j,t}^2}} \exp\left[-\frac{[a_j - \mu_{c,j} - \beta_{c,j,t}(a_t - \mu_{c,t})]^2}{2\tau_{c,j,t}^2}\right], \tag{5}$$

with

$$\beta_{c,j,t} = \frac{\mathrm{Cov}(A_j, A_t \mid c)}{\sigma_{c,t}^2}, \qquad \tau_{c,j,t}^2 = \sigma_{c,j}^2 - \frac{\mathrm{Cov}(A_j, A_t \mid c)^2}{\sigma_{c,t}^2}. \tag{6}$$

Here, A_j and A_t denote the child and super-parent attribute variables, respectively, and $\mathrm{Cov}(A_j, A_t \mid c)$ is their class-conditional covariance. The coefficient $\beta_{c,j,t}$ determines how the child attribute's conditional mean changes with the super-parent value, while $\tau_{c,j,t}^2$ is the conditional variance. All means, variances, and covariances are estimated from training instances belonging to class c.

Once trained, the m GSPODEs each predict a class label for an instance $\mathbf{x}$. These predictions serve as label attributes and together form the GSPODE label view, with $\mathbf{z}^s(\mathbf{x})$ defined in Eq. (1).

3.2.3 RANDOM TREE LABEL VIEW

Random trees recursively partition the continuous attribute space using threshold-based splits and predict classes from the class distributions in the resulting regions. To construct this view, m random trees are trained on bootstrap samples of the continuous training data, and their predicted class labels are used as label attributes.

To construct the tth random tree, $\lfloor \log_2 m \rfloor$ candidate attributes are randomly selected at each node. For each candidate attribute, the midpoints between consecutive distinct values in sorted order at

the current node are considered as candidate thresholds. The data are recursively partitioned by selecting the attribute–threshold pair that maximizes the information gain:

$$\mathrm{Gain}(\mathcal{D}, j, \theta) = H(\mathcal{D}) - \sum_{b \in \{L,R\}} \frac{|\mathcal{D}_b|}{|\mathcal{D}|} H(\mathcal{D}_b), \tag{7}$$

where $\mathcal{D}$ denotes the training samples at the current node, $\mathcal{D}_b$ denotes the subset assigned to child node $b \in \{L, R\}$ by splitting on attribute j at threshold θ, and $H(\cdot)$ denotes class entropy. The two child subsets are $\mathcal{D}_L = \{\mathbf{x}_i \in \mathcal{D} : a_{ij} < \theta\}$ and $\mathcal{D}_R = \{\mathbf{x}_i \in \mathcal{D} : a_{ij} \geq \theta\}$.

Each tree predicts the majority class in the leaf reached by the instance, counting each bootstrap occurrence separately. The predictions of all m trees form $\mathbf{z}^r(\mathbf{x})$, as defined in Eq. (1).

3.2.4 Granular Ball Label View

The granular ball label view uses distances between continuous instances to identify local groups and derives label attributes from their class labels. This provides a group-based representation that complements the label views constructed by GSPODEs and random trees. To construct this view, m GBKNN base classifiers (Xia et al., 2019) are built using granular ball partitions obtained with different random initializations.

To account for differences in attribute scales, both partition construction and prediction use range-normalized Euclidean distance:

$$d(\mathbf{x}, \mathbf{u}) = \sqrt{\sum_{\substack{j=1 \\ R_j > 0}}^{m} \left(\frac{a_j - u_j}{R_j} \right)^2}, \qquad R_j = \max_{1 \leq i \leq n} a_{ij} - \min_{1 \leq i \leq n} a_{ij}, \tag{8}$$

where $\mathbf{x}$ and $\mathbf{u}$ are continuous attribute vectors, a_j and u_j are their values on attribute j, and R_j denotes the range of attribute j in the training data.

Each partition starts with a single granular ball containing all training instances. For a ball B, its center $\boldsymbol{\mu}_B$ is the mean attribute vector of the instances it contains. Its label l_B is their most frequent class, and its purity $\mathrm{Purity}(B)$ is the proportion of these instances whose class label is l_B.

For each ball B, its purity is compared with a prescribed threshold ρ, which is set to 0.9 in this study. If $\mathrm{Purity}(B) \geq \rho$, the ball is retained without further splitting. Otherwise, its instances are partitioned into k clusters using k-means, where k is the number of distinct classes in B. Specifically, k instances are randomly selected as initial centers. Each instance is assigned to its nearest center according to Eq. (8), and each center is then updated as the mean of its assigned instances. These two steps are repeated until convergence, and each nonempty cluster forms a child ball. The center, label, and purity of each child ball are then computed as described above. This purity-based splitting procedure is repeated until the total number of balls no longer increases.

Let $\mathcal{B}_t = \{B_{t1}, \ldots, B_{tM}\}$ denote the tth partition, where $\boldsymbol{\mu}_{tb}$ is the center of ball B_{tb}. For an instance $\mathbf{x}$, the tth GBKNN uses the label of the nearest-center ball as its prediction. The predictions of all m GBKNNs form the label representation $\mathbf{z}^g(\mathbf{x})$, where the selected ball index $b_t(\mathbf{x})$ is determined by

$$b_t(\mathbf{x}) = \arg \min_{1 \leq b \leq M} d(\mathbf{x}, \boldsymbol{\mu}_{tb}). \tag{9}$$

3.3 Multi-view Weighting and Fusion

After constructing the four views, CMAWNB learns view-specific weights to mitigate the effects of attribute redundancy. Class-specific attribute weighting is applied to the continuous raw view, while class-specific attribute value weighting is applied to the three categorical label views. The weights are optimized independently within each view, and the resulting class-membership probabilities are averaged for prediction.

3.3.1 RAW VIEW WEIGHTING

For the raw view, a weight $w_{c,j}$ is introduced as the exponent of the class-conditional density of attribute j given class c. The resulting class-membership probability is

$$P(c \mid \mathbf{x}; \mathbf{W}) = \frac{P(c) \prod_{j=1}^{m} P(a_j \mid c)^{w_{c,j}}}{\sum_{c' \in \mathcal{C}} P(c') \prod_{j=1}^{m} P(a_j \mid c')^{w_{c',j}}}, \tag{10}$$

where $\mathbf{W} = [w_{c,j}]$ is the class-specific attribute weight matrix and $w_{c,j} \in [0, 1]$. The class priors and Gaussian distribution parameters estimated in Section 3.2.1 remain fixed during weight optimization.

All weights are initialized to 1.0 and learned by minimizing the negative conditional log-likelihood, defined as

$$-\mathrm{CLL}(\mathbf{W}) = -\sum_{i=1}^{n} \log P(y_i \mid \mathbf{x}_i; \mathbf{W}) = -\sum_{i=1}^{n} \log \frac{P(y_i, \mathbf{x}_i; \mathbf{W})}{\sum_{c' \in \mathcal{C}} P(c', \mathbf{x}_i; \mathbf{W})}, \tag{11}$$

where $P(c, \mathbf{x}_i; \mathbf{W})$ denotes the weighted joint probability:

$$P(c, \mathbf{x}_i; \mathbf{W}) = P(c) \prod_{j=1}^{m} P(a_{ij} \mid c)^{w_{c,j}}. \tag{12}$$

Since $w_{c,j}$ appears only in $P(c, \mathbf{x}_i; \mathbf{W})$ among the class-specific terms, the gradient of the negative conditional log-likelihood is derived as follows:

$$\begin{aligned} \frac{\partial[-\mathrm{CLL}(\mathbf{W})]}{\partial w_{c,j}} &= \frac{\partial}{\partial w_{c,j}} \sum_{i=1}^{n} \left[\log \left(\sum_{c' \in \mathcal{C}} P(c', \mathbf{x}_i; \mathbf{W}) \right) - \log P(y_i, \mathbf{x}_i; \mathbf{W}) \right] \\ &= \sum_{i=1}^{n} \left[\frac{P(c, \mathbf{x}_i; \mathbf{W}) \log P(a_{ij} \mid c)}{\sum_{c' \in \mathcal{C}} P(c', \mathbf{x}_i; \mathbf{W})} - \mathbb{I}(y_i = c) \log P(a_{ij} \mid c) \right] \\ &= \sum_{i=1}^{n} [P(c \mid \mathbf{x}_i; \mathbf{W}) - \mathbb{I}(y_i = c)] \log P(a_{ij} \mid c). \end{aligned} \tag{13}$$

Using this gradient, L-BFGS-B optimizes the weights subject to $0 \leq w_{c,j} \leq 1$. The learned weights are then substituted into Eq. (10) to estimate the raw view class membership probabilities for subsequent multi-view fusion.

3.3.2 LABEL VIEW WEIGHTING

Within each label view, different values of the same label attribute may require different weights for a given target class. CMAWNB therefore extends class-specific attribute weighting to class-specific attribute value weighting in the three label views.

Let $\mathbf{z} = \langle z_1, \ldots, z_m \rangle$ denote a label representation and $\mathbf{W} = [w_{c,j,k}]$ the weight array, where $k \in \mathcal{C}$ and $w_{c,j,k} \in [0, 1]$. The posterior $P(c \mid \mathbf{z}; \mathbf{W})$ follows Eq. (10), with $P(a_j \mid c)$ replaced by $P(z_j \mid c)$ and $w_{c,j}$ by w_{c,j,z_j}. The class priors and conditional probabilities are estimated from the current training view.

Attribute value weighting introduces more parameters than raw view attribute weighting. To reduce the risk of overfitting, we augment the negative conditional log-likelihood with an L_2 penalty centered at unit weights:

$$\mathcal{L}(\mathbf{W}) = -\sum_{i=1}^{n} \log P(y_i \mid \mathbf{z}_i; \mathbf{W}) + \lambda \|\mathbf{W} - \mathbf{W}_{\mathrm{one}}\|^2, \tag{14}$$

where $\mathbf{z}_i$ is the label representation of training instance $\mathbf{x}_i$, $\mathbf{W}_{\mathrm{one}}$ is an all-one array with the same dimensions as $\mathbf{W}$, and $\lambda \geq 0$ controls the regularization strength. The squared norm sums the squared differences over all entries of the weight array.

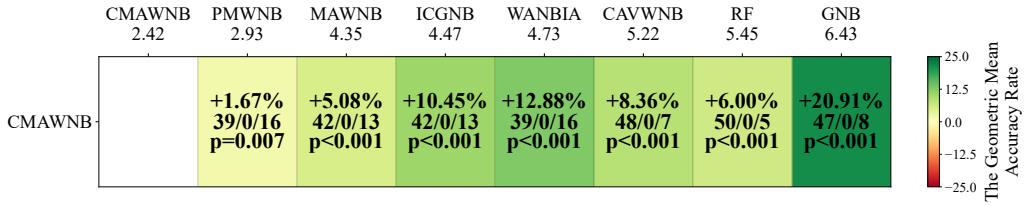


Figure 2: Classification accuracy comparisons on 55 benchmark datasets. Each cell reports, from top to bottom, CMAWNB's geometric mean accuracy improvement rate, Win/Draw/Loss (W/D/L) counts, and the two-sided Wilcoxon signed-rank test p-value. Bold entries indicate significant differences ($p < 0.05$). Numbers below method names denote average ranks (lower is better).

Following the raw view gradient derivation, the gradient for $w_{c,j,k}$ includes only instances whose value on label attribute j equals k, together with the derivative of the regularization term:

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\partial w_{c,j,k}} = \sum_{\substack{i=1 \\ z_{ij}=k}}^{n} [P(c \mid \mathbf{z}_i; \mathbf{W}) - \mathbb{I}(y_i = c)] \log P(z_{ij} \mid c) + 2\lambda(w_{c,j,k} - 1), \tag{15}$$

where z_{ij} is the value of label attribute j for training instance $\mathbf{x}_i$. As in raw view weighting, all weights are initialized to 1.0 and optimized using L-BFGS-B within $[0, 1]$, while the probability estimates remain fixed. Through independent weight optimization, the three label views each obtain a class-specific attribute value weighted classifier for estimating class-membership probabilities in subsequent multi-view fusion.

3.3.3 Prediction Fusion

For a test instance $\mathbf{x}$, the trained GSPODE, RT, and GBKNN base classifiers generate its three label representations. The four weighted classifiers then estimate their respective class-membership probabilities. CMAWNB averages these probabilities and selects the class with the highest average:

$$c(\mathbf{x}) = \arg\max_{c \in \mathcal{C}} \frac{1}{4} \left[P(c \mid \mathbf{x}; \mathbf{W}^0) + \sum_{v \in \{s,r,g\}} P(c \mid \mathbf{z}^v(\mathbf{x}); \mathbf{W}^v) \right]. \tag{16}$$

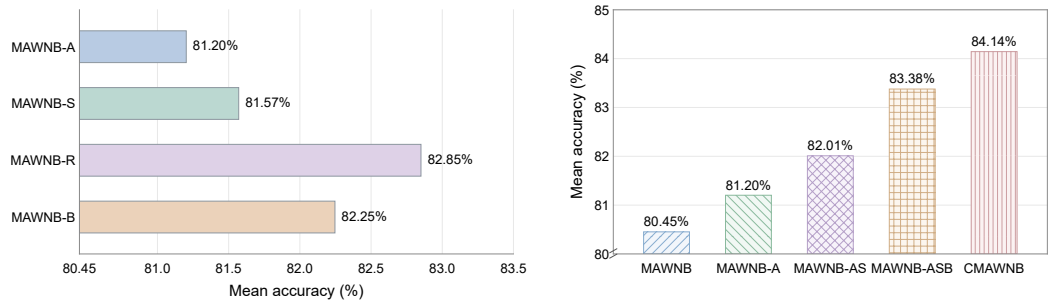
Here, $\mathbf{W}^0$ and $\mathbf{W}^v$ denote the learned weights of the raw view and label view v, respectively.

4 Experiments

4.1 Experiments on Benchmark Datasets

We first evaluate CMAWNB on 55 benchmark datasets from the UCI Machine Learning Repository. All datasets contain only continuous attributes and vary in instance size, attribute dimensionality, and class number. Detailed dataset descriptions are provided in Appendix A.1. For data-preprocessing, missing values are imputed using the observed mean of the corresponding attributes. For competing methods that only support nominal inputs, we employ equal-width discretization with 10 bins. For other methods that are capable of handling continuous inputs, the original continuous data is directly used for evaluation without discretization. We compare the classification accuracy (%) of CMAWNB with its seven competitors: PMWNB (Wu et al., 2026), ICGNB (Li et al., 2025), MAWNB (Zhang et al., 2023), CAVWNB (Zhang et al., 2020), WANBIA (Zaidi et al., 2013), random forest (RF) (Breiman, 2001), and Gaussian naive Bayes (GNB) (John & Langley, 1995), using 10 independent runs of 10-fold cross-validation, with a separately randomized partition in each run. In our experiments, ICGNB is implemented in Python, while CMAWNB and all other competitors are implemented on the Waikato Environment for Knowledge Analysis (WEKA) platform (Witten et al., 2011).

Classification Performance. The detailed accuracy results on each dataset are provided in Appendix A.2. For simplicity, Figure 2 summarizes the overall classification accuracy comparisons



(a) Individual Component Effects

(b) Incremental Component Integration

Figure 3: Ablation results on 55 benchmark datasets: (a) mean classification accuracy after individual modifications to MAWNB; (b) mean classification accuracy during their progressive integration.

between CMAWNB and its seven competitors across 55 benchmark datasets. The main findings are summarized as follows:

(1) Compared with other competitors, CMAWNB achieves the lowest average rank of 2.42, followed by PMWNB (2.93) and MAWNB (4.35). Additionally, CMAWNB obtains positive geometric mean relative accuracy improvements over all seven competitors, and higher accuracy on at least 39 datasets in each comparison. The Wilcoxon signed-rank tests yield $p = 0.007$ against PMWNB and $p < 0.001$ against all remaining competitors, indicating statistically significant differences at the 0.05 level. These results support the overall performance advantage of CMAWNB.

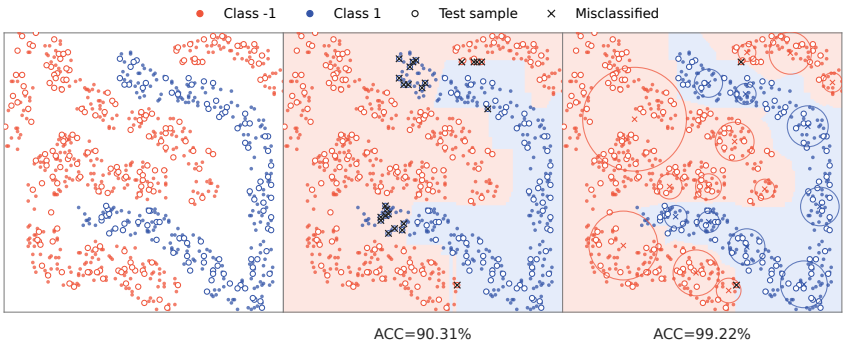
(2) Compared with PMWNB, MAWNB, WANBIA, and CAVWNB, CMAWNB achieves higher accuracy on 39, 42, 39, and 48 datasets, respectively, demonstrating its advantage over these attribute-weighted NB methods. These results support the effectiveness of preserving fine-grained continuous information and incorporating complementary label representations within a multi-view attribute weighting framework.

(3) Compared with ICGNB, RF, and GNB, CMAWNB achieves higher accuracy on 42, 50, and 47 datasets, respectively, with statistically significant advantages over all three. These results demonstrate the overall performance advantage of combining attribute dependencies, space partitions, and local group structure through view-specific weighting and fusion on the evaluated datasets.

Ablation Study. We use MAWNB as the baseline to examine the individual and combined effects of each component in CMAWNB across 55 benchmark datasets. Figure 3 reports the mean accuracy of the variants, and their detailed results are provided in Appendix A.2.

Among them, Figure 3(a) examines each component individually while keeping the remaining components of MAWNB unchanged. The MAWNB-A variant introduces continuous raw view modeling; the MAWNB-S and MAWNB-R variants replace the corresponding base classifiers with GSPODEs and RTs using continuous splits, respectively; and the MAWNB-B variant adds a label view constructed using GBKNN. All four variants achieve higher mean accuracy than MAWNB, with continuous RTs yielding the largest improvement. The gains from the first three variants support direct continuous modeling in both raw view learning and label view construction, suggesting that retaining fine-grained information otherwise lost through discretization benefits classification. The improvement from the MAWNB-B variant further suggests that local group representations provide useful information complementary to the existing views.

Figure 3(b) further examines the progressive integration of these modifications. MAWNB-AS introduces continuous raw view modeling and additionally incorporates GSPODEs, while MAWNB-ASB further adds the granular ball view. Finally, replacing the remaining discretized RTs with RTs using continuous splits yields CMAWNB. Mean accuracy increases at every stage, showing that each component provides an additional benefit in the evaluated integration sequence. The gain from adding the granular ball view supports its complementarity with the raw and GSPODE views, while the final improvement further supports direct continuous modeling in the RT view. The complete CMAWNB achieves the highest mean accuracy, exceeding MAWNB by 3.69 percentage points. To-



(a) Original Data (b) CMAWNB w/o GB (c) CMAWNB

Figure 4: A visual comparison on the fourclass dataset. (a) Original data. (b) Decision regions of CMAWNB without the GB view. (c) Decision regions of CMAWNB with the 19 granular balls from its first granular ball partition. Point colors indicate true classes, while background colors indicate predicted classes. Filled and hollow points represent training and test instances, respectively; ACC denotes test accuracy. Black crosses mark misclassified test instances, and colored crosses mark ball centers. Circle radii indicate mean within-ball distances in the normalized attribute space, not decision boundaries.

gether, these results support combining continuous modeling with local group representations within the multi-view framework.

4.2 Case Study

To examine how the granular ball view complements the other representations, we conduct a visual case study on a synthetic two-dimensional four-class dataset (Ho & Kleinberg, 1996), which contains 862 instances from two classes. A stratified split assigns 604 instances to training and 258 to testing. We compare CMAWNB with CMAWNB w/o GB, which retains the continuous raw, GSPODE, and RT views but excludes the granular ball view. Both models use the same data split, and granular balls are constructed exclusively from training instances.

Figure 4(a) shows two classes arranged in curved structures that require a nonlinear decision boundary to separate them. Without the granular ball view, several test instances near the upper and lower curved structures are misclassified (Figure 4(b)). After incorporating this view, the decision regions better follow the local class structure (Figure 4(c)). Test accuracy increases from 90.31% to 99.22%, with 23 of the 25 previously misclassified instances corrected and no additional errors.

The 19 granular balls displayed in Figure 4(c) belong to the first partition and summarize local instance groups. Their labels are used by GBKNN to construct additional label attributes for subsequent weighting and fusion. The reduction in local classification errors suggests that these group-level representations provide useful information beyond that captured by the raw, GSPODE, and RT views. This case therefore supports the complementary role of the granular ball view.

5 Conclusions

In this paper, we propose Continuous-Aware Multi-view Attribute Weighted Naive Bayes (CMAWNB) to directly exploit continuous attributes without preliminary discretization. CMAWNB combines continuous raw view modeling with three complementary label views constructed using GSPODEs, RTs, and GBKNN classifiers. View-specific weighting and prediction fusion integrate attribute dependencies, continuous space partitions, and local group structure for classification. Experiments on 55 benchmark datasets demonstrate the effectiveness of CMAWNB. The ablation study supports the individual and combined benefits of the proposed components, while the visual case study illustrates the complementary role of the granular ball view.

AI USE STATEMENT

In this work, we used generative AI tools only for minor text editing and formatting assistance. We have not used generative AI tools for research ideation, method development, experiment implementation, data analysis, or result interpretation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We submit the code and datasets as supplementary materials, and the details of dataset preprocessing and algorithm implementation are provided in our manuscript. Once our paper is accepted, we will make the code and datasets publicly available on GitHub.

REFERENCES

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Khalil El Hindi. Fine tuning the naive Bayesian learning algorithm. *AI Communications*, 27(2):133–141, 2014. doi: 10.3233/AIC-130588.

Khalil M. El Hindi, Reem R. Aljulaidan, and Hussien AlSalman. Lazy fine-tuning algorithms for naive Bayesian text classification. *Applied Soft Computing*, 96:106652, 2020. doi: 10.1016/j.asoc.2020.106652.

Eibe Frank, Mark Hall, and Bernhard Pfahringer. Locally weighted naive Bayes. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pp. 249–256. Morgan Kaufmann, 2003.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997. doi: 10.1023/A:1007465528199.

Mark Hall. A decision tree-based attribute weighting filter for naive Bayes. *Knowledge-Based Systems*, 20(2):120–126, 2007. doi: 10.1016/j.knosys.2006.11.008.

Tin Kam Ho and Eugene M. Kleinberg. Building projectable classifiers of arbitrary complexity. In *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 880–885, 1996.

Liangxiao Jiang, Lungan Zhang, Liangjun Yu, and Dianhong Wang. Class-specific attribute weighted naive Bayes. *Pattern Recognition*, 88:321–330, 2019. doi: 10.1016/j.patcog.2018.11.032. URL https://doi.org/10.1016/j.patcog.2018.11.032.

George H. John and Pat Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345. Morgan Kaufmann, 1995. URL https://arxiv.org/abs/1302.4964.

Ron Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 202–207. AAAI Press, 1996.

Pat Langley and Stephanie Sage. Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 399–406. Morgan Kaufmann, 1994. doi: 10.1016/B978-1-55860-332-5.50055-9.

Chang-Hwan Lee, Fernando Gutierrez, and Dejing Dou. Calculating feature weights in naive Bayes with Kullback-Leibler measure. In *2011 IEEE 11th International Conference on Data Mining*, pp. 1146–1151. IEEE, 2011. doi: 10.1109/ICDM.2011.29. URL https://doi.org/10.1109/ICDM.2011.29.

Chengyuan Li, Liangxiao Jiang, Wenjun Zhang, Liangjun Yu, and Huan Zhang. Instance correlation graph-based naive Bayes. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35021–35033. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25ar.html.

Guanzhi Liu, Kexin Meng, Pei Lv, and Huan Zhang. EMAWNB: Enhanced multi-view attribute weighted naive Bayes. In *Advances in Knowledge Discovery and Data Mining*, volume 15872 of *Lecture Notes in Computer Science*, pp. 66–78. Springer, 2025. doi: 10.1007/978-981-96-8180-8_6. URL https://doi.org/10.1007/978-981-96-8180-8_6.

Kexin Meng, Huan Zhang, Liangxiao Jiang, Pei Lv, Shuo He, and Mingliang Xu. K-free dependence Bayesian classifiers. *IEEE Transactions on Neural Networks and Learning Systems*, 2026. doi: 10.1109/TNNLS.2026.3664196. Early access.

Baozheng Wang, Huan Zhang, Kexin Meng, and Yafei Zhang. Frequency-based lazy naive Bayes. *Applied Soft Computing*, 198:115249, 2026. doi: 10.1016/j.asoc.2026.115249.

Geoffrey I. Webb, Janice R. Boughton, and Zhihai Wang. Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005. doi: 10.1007/s10994-005-4258-6.

Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3 edition, 2011. ISBN 9780123748560.

Siyao Wu, Huan Zhang, Kexin Meng, Zhipeng Ding, and Pei Lv. Learning representations from perturbation: A novel matrix-view weighting framework for naive Bayes. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 7416–7432. PMLR, 2026. URL https://proceedings.mlr.press/v337/wu26b.html.

Shuyin Xia, Yunsheng Liu, Xin Ding, Guoyin Wang, Hong Yu, and Yuoguo Luo. Granular ball computing classifiers for efficient, scalable and robust learning. *Information Sciences*, 483:136–152, 2019. doi: 10.1016/j.ins.2019.01.010. URL https://doi.org/10.1016/j.ins.2019.01.010.

Shuyin Xia, Xiaochuan Dai, Guoyin Wang, Xinbo Gao, and Elisabeth Giem. An efficient and adaptive granular-ball generation method in classification problem. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5319–5331, 2024. doi: 10.1109/TNNLS.2022.3203381. URL https://doi.org/10.1109/TNNLS.2022.3203381.

Qin Xie, Qinghua Zhang, Shuyin Xia, Fan Zhao, Chengying Wu, Guoyin Wang, and Weiping Ding. GBG++: A fast and stable granular ball generation method for classification. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(2):2022–2036, 2024. doi: 10.1109/TETCI.2024.3359091. URL https://doi.org/10.1109/TETCI.2024.3359091.

Nayyar A. Zaidi, Jesús Cerquides, Mark J. Carman, and Geoffrey I. Webb. Alleviating naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, 14(60):1947–1988, 2013. URL https://www.jmlr.org/papers/v14/zaidi13a.html.

Huan Zhang and Liangxiao Jiang. Fine tuning attribute weighted naive Bayes. *Neurocomputing*, 488:402–411, 2022. doi: 10.1016/j.neucom.2022.03.020.

Huan Zhang, Liangxiao Jiang, and Liangjun Yu. Class-specific attribute value weighting for naive Bayes. *Information Sciences*, 508:260–274, 2020. doi: 10.1016/j.ins.2019.08.071. URL https://doi.org/10.1016/j.ins.2019.08.071.

Huan Zhang, Liangxiao Jiang, and Chaoqun Li. Attribute augmented and weighted naive Bayes. *Science China Information Sciences*, 65(12):222101, 2022. doi: 10.1007/s11432-020-3277-0. URL https://doi.org/10.1007/s11432-020-3277-0.

Huan Zhang, Liangxiao Jiang, Wenjun Zhang, and Chaoqun Li. Multi-view attribute weighted naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7291–7302, 2023. doi: 10.1109/TKDE.2022.3177634. URL https://doi.org/10.1109/TKDE.2022.3177634.

Huan Zhang, Kexin Meng, Pei Lv, Shuo He, and Mingliang Xu. A general dual-view framework for instance weighted naive Bayes. *Pattern Recognition*, 171:112181, 2026. doi: 10.1016/j.patcog.2025.112181. URL https://doi.org/10.1016/j.patcog.2025.112181.

Xiaoliang Zhou, Yongli Wang, Li Zhang, Anqi Huang, and Xiaoli Wang. An innovative multi-view collaborative optimization framework for weighted naive Bayes. *Knowledge-Based Systems*, 317: 113378, 2025. doi: 10.1016/j.knosys.2025.113378. URL https://doi.org/10.1016/j.knosys.2025.113378.

A More Experimental Details

A.1 Dataset Information

Our experiments are conducted on 55 benchmark classification datasets from the UCI Machine Learning Repository, as summarized in Table 1. All datasets contain only continuous attributes and vary in sample size, dimensionality, and number of classes. Missing attribute values are replaced with the means of the observed values of the corresponding attributes. Attribute counts in the table exclude the class label.

A.2 Detailed Experimental Results

Table 2 presents the detailed classification accuracy comparisons between CMAWNB and its seven competitors: PMWNB, MAWNB, ICGNB, WANBIA, CAVWNB, RF, and GNB.

Table 3 reports the results of the individual-component ablation study, comparing MAWNB with MAWNB-A, MAWNB-S, MAWNB-R, and MAWNB-B. Each variant introduces a single modification while keeping the remaining components unchanged. MAWNB-A introduces continuous raw view modeling; MAWNB-S replaces SPODEs with GSPODEs; MAWNB-R replaces discretized RTs with RTs using continuous splits; and MAWNB-B adds a granular ball label view constructed using GBKNN.

Table 4 further presents the results of progressively integrating these modifications. Starting from MAWNB, MAWNB-A introduces continuous raw view modeling, MAWNB-AS additionally incorporates GSPODEs, and MAWNB-ASB further adds the granular ball view. Replacing the remaining discretized RTs with RTs using continuous splits yields CMAWNB. Tables 3 and 4 provide the dataset-level results underlying Figure 3(a) and Figure 3(b), respectively.

All results are obtained from 10 independent runs of 10-fold cross-validation and reported as mean classification accuracy (%) $\pm$ standard deviation. The highest mean accuracy on each dataset is highlighted in bold. The Average row reports the arithmetic mean of the dataset-level accuracies. Significant W/D/L summarizes the numbers of statistically significant wins, non-significant differences, and statistically significant losses. In contrast, W/D/L counts strict wins, equalities, and strict losses based on direct comparisons of mean accuracy. CMAWNB is the reference method in Table 2, while MAWNB is the reference method in Tables 3 and 4. The symbols $\bullet$ and $\circ$ indicate that the reference method achieves significantly higher and lower accuracy than the method in the corresponding column, respectively. Both Significant W/D/L and W/D/L are reported from the perspective of the reference method.

Table 1: Descriptions of the 55 benchmark datasets.

Dataset	Instance	Total Attribute	Continuous Attribute	Class
abalone	4177	8	8	3
appendicitis	106	7	7	2
artificial-characters	10218	7	7	10
australian	12546	8	8	3
balance-scale	625	4	4	3
breast-c	699	9	9	2
breast-t	106	9	9	6
cardioto	2126	35	35	3
cardiotocography-2	2126	21	21	3
climate	540	20	20	2
connectionist-vowel	528	10	10	11
diabetes	768	8	8	2
ecoli	336	7	7	8
eeg-eye	14980	14	14	2
energy-efficiency-y1	768	8	8	37
energy-efficiency-y2	768	8	8	38
fertility	100	9	9	2
glass	214	9	9	7
hayes-roth	160	4	4	3
heart-statlog	270	13	13	2
ionosphere	351	34	34	2
iris	150	4	4	3
leaf	340	15	15	30
libras	360	90	90	15
liver	345	6	6	2
magic	19020	10	10	2
mammo	961	5	5	2
mfeat-fourier	2000	76	76	10
monks-1	556	6	6	2
new-thyroid	215	5	5	3
optdigits	5620	64	64	10
page-blocks	5473	10	10	5
parkinsons	195	22	22	2
planning	182	12	12	2
qsar-biodegradation	1055	41	41	2
robot24	5456	24	24	4
seeds	210	7	7	3
segment	2310	19	19	7
sonar	208	60	60	2
spectf-heart	349	44	44	2
statlog-g	1000	24	24	2
statlog-h	270	13	13	2
steel-plates-faults	1941	33	33	2
texture	5500	40	40	11
turkiye	5820	32	32	13
user	403	5	5	5
vehicle	846	18	18	4
waveform-5000	5000	40	40	3
wdbc	569	30	30	2
wholesale-c	440	7	7	2
wholesale-r	440	7	7	3
wilt	4839	5	5	2
wine-rec	178	13	13	3
wine-red	1599	11	11	6
wine-white	4898	11	11	7

Table 2: Classification accuracy comparisons between CMAWNB and its competitors.

Dataset	CMAWNB	PMWNB	MAWNB	ICGNB	WANBIA	CAVWNB	RF	GNB
abalone	53.91 ± 2.23	54.50 ± 1.96	54.52 ± 2.07	**54.80** ± 2.09	54.15 ± 1.99	54.43 ± 2.10	51.33 ± 2.19 •	51.84 ± 1.94 •
appendicitis	87.45 ± 8.00	85.09 ± 8.72	81.40 ± 8.56 •	87.59 ± 8.99	**88.01** ± 9.22	80.43 ± 9.94 •	84.46 ± 9.99	85.21 ± 10.07
artificial-characters	94.37 ± 0.99	**94.55** ± 0.91	67.44 ± 1.47 •	40.52 ± 1.37 •	32.34 ± 1.28 •	46.44 ± 1.45 •	87.39 ± 1.13 •	30.03 ± 1.18 •
australian	**87.04** ± 0.95	81.74 ± 1.14 •	77.82 ± 1.22 •	62.03 ± 1.20 •	58.69 ± 1.31 •	64.69 ± 1.50 •	85.28 ± 1.12 •	50.24 ± 1.49 •
balance-scale	85.71 ± 3.17	85.50 ± 2.77	87.01 ± 2.41	88.85 ± 2.48 ○	90.58 ± 1.65 ○	**91.94** ± 0.74 ○	79.97 ± 3.69 •	90.53 ± 1.67 ○
breast-c	96.62 ± 2.10	**96.92** ± 1.86	96.81 ± 2.01	95.82 ± 2.35	96.25 ± 2.25	96.00 ± 2.06	95.24 ± 2.52	96.12 ± 2.16
breast-t	44.94 ± 13.91	40.15 ± 11.56	37.16 ± 11.53	39.10 ± 12.70	**47.05** ± 12.74	40.35 ± 12.67	38.06 ± 12.54	46.44 ± 11.18
cardioto	**99.07** ± 0.73	98.47 ± 0.90 •	98.18 ± 0.96 •	98.66 ± 0.76	98.59 ± 0.87 •	98.33 ± 0.94 •	97.93 ± 1.13 •	90.67 ± 2.09 •
cardiotocography-2	**93.95** ± 1.41	93.47 ± 1.85	92.14 ± 1.59 •	88.78 ± 1.96 •	87.76 ± 1.80 •	91.90 ± 1.47 •	92.99 ± 2.12	81.79 ± 2.39 •
climate	90.67 ± 2.29	**90.91** ± 1.65	89.69 ± 1.90	90.24 ± 2.86	90.67 ± 2.25	87.11 ± 2.89 •	89.85 ± 2.96	88.85 ± 3.78
connectionist-vowel	**96.80** ± 2.70	90.11 ± 4.00 •	90.13 ± 4.25 •	71.04 ± 5.95 •	64.01 ± 7.16 •	83.84 ± 5.84 •	81.28 ± 5.32 •	64.92 ± 7.12 •
diabetes	76.25 ± 4.76	75.64 ± 4.64	74.95 ± 4.88	76.22 ± 4.37	**76.63** ± 5.05	75.17 ± 4.10	72.17 ± 4.99 •	75.75 ± 5.32
ecoli	**85.69** ± 5.65	84.07 ± 5.51	82.00 ± 6.21	85.58 ± 4.95	84.40 ± 5.31	83.40 ± 5.55	80.62 ± 5.90 •	85.50 ± 5.46
eeg-eye	**89.70** ± 1.94	79.05 ± 1.09 •	57.19 ± 1.17 •	73.35 ± 1.28 •	45.70 ± 2.31 •	57.19 ± 1.16 •	86.60 ± 0.85 •	47.19 ± 3.11 •
energy-efficiency-y1	64.92 ± 4.41	**67.95** ± 3.73	67.78 ± 3.51	38.41 ± 4.09 •	36.35 ± 3.93 •	53.25 ± 4.79 •	62.56 ± 4.92	36.81 ± 4.43 •
energy-efficiency-y2	**55.98** ± 4.43	54.31 ± 4.47	54.48 ± 5.03	42.29 ± 4.87 •	45.06 ± 4.62 •	51.75 ± 4.73 •	53.26 ± 5.08	44.39 ± 5.04 •
fertility	86.30 ± 6.61	**88.00** ± 4.92	85.00 ± 8.35	84.80 ± 6.59	87.30 ± 4.68	84.40 ± 8.57	83.80 ± 9.93	86.50 ± 6.26
glass	**75.33** ± 8.27	73.18 ± 9.57	63.72 ± 8.90 •	64.78 ± 8.84 •	57.12 ± 8.70 •	60.26 ± 9.90 •	70.29 ± 8.64	49.45 ± 9.50 •
hayes-roth	79.56 ± 9.10	82.38 ± 8.01	79.81 ± 8.88	63.75 ± 10.95 •	68.62 ± 12.15 •	**83.81** ± 7.85	82.50 ± 8.61	70.06 ± 13.16 •
heart-statlog	82.48 ± 6.84	**83.89** ± 6.06	82.89 ± 6.31	82.81 ± 7.58	83.81 ± 6.40	81.93 ± 5.70	77.37 ± 6.64	83.59 ± 5.98
ionosphere	**94.53** ± 3.66	92.20 ± 4.43 •	91.89 ± 4.52 •	92.05 ± 4.70	92.11 ± 4.41 •	90.97 ± 4.49 •	90.71 ± 5.17 •	82.17 ± 6.14 •
iris	**96.00** ± 4.34	94.27 ± 5.77	95.53 ± 4.84	95.93 ± 5.09	95.93 ± 4.63	95.40 ± 5.25	94.27 ± 5.36	95.53 ± 5.02
leaf	**78.35** ± 6.18	66.18 ± 8.12 •	56.74 ± 8.79 •	72.15 ± 6.98	70.76 ± 6.90 •	57.15 ± 9.12 •	60.00 ± 7.25 •	73.06 ± 7.36 •
libras	**87.42** ± 5.18	79.47 ± 6.03 •	75.25 ± 6.88 •	57.89 ± 9.10 •	65.92 ± 7.44 •	68.50 ± 7.84 •	66.14 ± 7.32 •	64.14 ± 7.90 •
liver	**68.45** ± 7.83	64.24 ± 7.33	65.23 ± 7.70	62.05 ± 7.16	60.42 ± 6.30 •	67.03 ± 7.39	65.94 ± 8.34	54.89 ± 8.83 •
magic	**85.58** ± 0.66	85.25 ± 0.71	83.29 ± 0.69 •	78.53 ± 0.84 •	77.17 ± 0.81 •	82.72 ± 0.83 •	84.81 ± 0.78 •	72.68 ± 0.81 •
mammo	79.75 ± 3.37	**82.55** ± 3.57 ○	80.75 ± 3.36	80.28 ± 3.84	80.11 ± 4.00	80.87 ± 3.73	77.66 ± 3.86	77.64 ± 4.22
mfeat-fourier	**84.05** ± 2.29	81.55 ± 2.52 •	80.00 ± 2.55 •	80.14 ± 2.58 •	80.22 ± 2.60 •	77.95 ± 2.52 •	70.48 ± 3.10 •	76.41 ± 3.04 •
monks-1	94.46 ± 2.81	99.14 ± 1.66 ○	**99.17** ± 1.58 ○	68.02 ± 5.74 •	66.53 ± 5.55 •	74.66 ± 4.54 •	96.62 ± 3.29	66.49 ± 5.52 •
new-thyroid	96.27 ± 3.75	95.81 ± 4.06	94.80 ± 4.61	96.56 ± 3.61	**97.24** ± 3.65	94.51 ± 4.39	94.46 ± 4.54	96.92 ± 3.46
optdigits	**98.28** ± 0.51	96.28 ± 0.81 •	95.55 ± 0.93 •	94.51 ± 0.85 •	90.28 ± 1.15 •	95.40 ± 0.86 •	90.69 ± 1.28 •	91.39 ± 1.04 •
page-blocks	96.49 ± 0.72	96.43 ± 0.61	94.18 ± 0.73 •	94.29 ± 0.68 •	93.29 ± 0.96 •	93.93 ± 0.73 •	**96.88** ± 0.74	90.01 ± 2.08 •
parkinsons	**93.24** ± 5.89	90.39 ± 7.10	90.44 ± 6.83	86.36 ± 7.13 •	83.25 ± 6.85 •	89.70 ± 6.22	88.54 ± 7.22 •	70.14 ± 9.30 •
planning	66.18 ± 7.49	65.62 ± 8.62	56.73 ± 11.40 •	67.16 ± 6.57	**69.93** ± 4.28	52.78 ± 10.84 •	61.22 ± 10.49	66.20 ± 7.73
qsar-biodegradation	**87.69** ± 2.99	86.38 ± 3.32	84.87 ± 3.39 •	85.08 ± 3.39	81.61 ± 6.42 •	83.68 ± 3.58 •	83.59 ± 3.76 •	75.89 ± 4.36 •
robot24	94.62 ± 1.00	97.38 ± 0.73 ○	94.68 ± 1.00	72.21 ± 1.85 •	62.69 ± 1.72 •	92.53 ± 1.10 •	**98.88** ± 0.50 ○	52.50 ± 1.96 •
seeds	**94.76** ± 4.71	91.29 ± 5.58 •	91.95 ± 5.93	94.43 ± 4.69	93.24 ± 5.25	92.38 ± 5.82	91.43 ± 5.62	90.52 ± 6.15 •
segment	**97.67** ± 0.98	96.61 ± 1.14 •	95.43 ± 1.46 •	89.98 ± 1.87 •	88.03 ± 1.99 •	93.97 ± 1.54 •	96.33 ± 1.41 •	80.13 ± 2.14 •
sonar	**87.70** ± 7.56	79.78 ± 9.14 •	79.36 ± 9.73 •	76.80 ± 8.28 •	79.78 ± 8.78 •	77.39 ± 10.17 •	73.94 ± 9.94 •	67.71 ± 8.66 •
spectf-heart	88.89 ± 5.09	**91.38** ± 4.53 ○	91.30 ± 4.61	79.34 ± 6.55 •	79.87 ± 6.20 •	90.52 ± 4.61	86.77 ± 5.51	71.75 ± 6.52 •
statlog-g	75.54 ± 3.94	76.16 ± 3.24	76.23 ± 3.18	75.52 ± 3.29	**76.29** ± 3.72	75.05 ± 3.37	71.44 ± 3.47 •	75.59 ± 3.73
statlog-h	82.33 ± 6.44	83.74 ± 5.95	82.59 ± 6.35	82.81 ± 7.58	**83.81** ± 6.40	81.93 ± 5.70	77.41 ± 7.64	83.59 ± 5.98
steel-plates-faults	99.89 ± 0.23	99.22 ± 0.62 •	98.74 ± 0.93 •	**99.99** ± 0.05	99.62 ± 0.43	99.73 ± 0.35	97.57 ± 1.56 •	58.25 ± 6.47 •
texture	**98.62** ± 0.50	97.25 ± 0.76 •	97.13 ± 0.72 •	96.84 ± 0.62 •	84.64 ± 1.29 •	96.31 ± 0.76 •	94.19 ± 1.07 •	77.44 ± 1.53 •
turkiye	34.31 ± 1.70	36.81 ± 1.68 ○	34.11 ± 1.79	35.86 ± 1.43	35.20 ± 1.27	**37.95** ± 1.74 ○	31.50 ± 1.78 •	25.77 ± 1.81 •
user	**91.06** ± 4.09	89.50 ± 4.94	87.78 ± 5.17 •	85.55 ± 4.96 •	85.38 ± 5.98 •	83.93 ± 5.10 •	88.53 ± 4.68	85.03 ± 5.72 •
vehicle	**74.85** ± 4.14	73.18 ± 4.05	72.53 ± 4.26	65.87 ± 4.30 •	55.68 ± 4.28 •	71.06 ± 3.82 •	72.37 ± 4.15	44.68 ± 4.59 •
waveform-5000	**85.76** ± 1.66	85.48 ± 1.56	84.53 ± 1.62 •	83.97 ± 1.61 •	83.76 ± 1.69 •	82.98 ± 1.63 •	76.58 ± 2.06 •	80.01 ± 1.45 •
wdbc	96.33 ± 2.48	95.96 ± 2.84	95.64 ± 2.69	96.41 ± 2.34	**97.21** ± 1.98	95.41 ± 2.76	94.24 ± 2.81 •	93.31 ± 3.30 •
wholesale-c	90.66 ± 4.31	90.52 ± 4.16	86.93 ± 4.72 •	**91.09** ± 3.89	90.45 ± 3.67	87.61 ± 4.30 •	89.77 ± 4.44	89.55 ± 4.11
wholesale-r	68.05 ± 3.42	70.95 ± 1.50 ○	69.91 ± 2.53	71.36 ± 1.55 ○	**71.45** ± 1.30 ○	69.80 ± 2.45	59.18 ± 6.56 •	47.00 ± 11.41 •
wilt	96.96 ± 0.52	94.67 ± 0.20 •	94.62 ± 0.18 •	94.61 ± 0.06 •	94.61 ± 0.06 •	94.60 ± 0.16 •	**98.15** ± 0.64 ○	89.26 ± 1.31 •
wine-rec	**98.20** ± 2.88	97.97 ± 3.04	96.28 ± 3.93	97.74 ± 3.32	97.25 ± 4.11	97.02 ± 3.86	94.09 ± 6.19	97.46 ± 3.70
wine-red	**68.95** ± 3.07	66.89 ± 3.55 •	66.49 ± 3.79 •	58.70 ± 3.29 •	57.31 ± 3.75 •	60.04 ± 3.63 •	61.86 ± 3.30 •	54.84 ± 3.87 •
wine-white	**69.28** ± 1.92	64.69 ± 2.24 •	64.10 ± 1.98 •	52.59 ± 1.97 •	52.02 ± 1.92 •	52.41 ± 1.88 •	60.19 ± 2.13 •	44.45 ± 2.00 •
Average	**84.14**	82.82	80.45	77.13	75.75	78.26	79.81	71.24
Significant W/D/L	–	17/32/6	27/27/1	27/26/2	32/21/2	32/21/2	28/25/2	38/16/1
W/D/L	–	39/0/16	42/0/13	42/0/13	39/0/16	48/0/7	50/0/5	47/0/8

Table 3: Classification accuracy of MAWNB and its single-component variants.

Dataset	MAWNB	MAWNB-A	MAWNB-S	MAWNB-R	MAWNB-B
abalone	54.52 ± 2.07	54.65 ± 2.10	**54.67** ± 2.10	54.26 ± 2.36	54.11 ± 2.01
appendicitis	81.40 ± 8.56	**84.62** ± 8.13	83.67 ± 8.86	83.33 ± 9.19	83.50 ± 8.49
artificial-characters	67.44 ± 1.47	67.76 ± 1.40	68.78 ± 1.33 ○	**91.01** ± 1.01 ○	87.51 ± 1.19 ○
australian	77.82 ± 1.22	79.37 ± 1.26 ○	78.67 ± 1.27 ○	**83.45** ± 1.02 ○	83.19 ± 1.00 ○
balance-scale	87.01 ± 2.41	86.19 ± 2.55 ●	**89.92** ± 2.00 ○	86.37 ± 2.93	84.51 ± 3.09 ●
breast-c	96.81 ± 2.01	**96.90** ± 1.90	96.81 ± 2.02	96.70 ± 1.97	96.71 ± 2.06
breast-t	37.16 ± 11.53	40.53 ± 11.32	**42.68** ± 12.58	40.16 ± 13.29	40.62 ± 13.14
cardioto	98.18 ± 0.96	98.38 ± 0.90	98.70 ± 0.86 ○	**98.76** ± 0.81 ○	98.61 ± 0.88 ○
cardiotocography-2	92.14 ± 1.59	92.03 ± 1.72	91.85 ± 1.67	**94.12** ± 1.51 ○	92.69 ± 1.45
climate	89.69 ± 1.90	**90.52** ± 1.69	90.04 ± 2.28	89.61 ± 2.54	90.39 ± 1.64
connectionist-vowel	90.13 ± 4.25	90.74 ± 4.21	91.80 ± 3.75	92.45 ± 3.77 ○	**94.30** ± 2.90 ○
diabetes	74.95 ± 4.88	75.29 ± 4.95	74.80 ± 4.99	**76.50** ± 4.31	75.49 ± 4.83
ecoli	82.00 ± 6.21	83.21 ± 5.45	83.82 ± 4.92	**84.50** ± 6.51	83.24 ± 5.60
eeg-eye	57.19 ± 1.17	58.34 ± 1.10 ○	57.84 ± 1.08 ○	**91.75** ± 0.66 ○	83.92 ± 4.11 ○
energy-efficiency-y1	67.78 ± 3.51	67.69 ± 3.74	67.13 ± 4.23	**68.33** ± 4.39	66.49 ± 4.19
energy-efficiency-y2	54.48 ± 5.03	54.24 ± 5.08	55.08 ± 4.90	**57.21** ± 5.17	54.95 ± 5.21
fertility	85.00 ± 8.35	85.70 ± 7.00	84.70 ± 8.22	85.80 ± 7.94	**86.00** ± 6.67
glass	63.72 ± 8.90	66.27 ± 8.56	64.94 ± 8.70	**72.09** ± 9.35 ○	68.03 ± 9.37
hayes-roth	79.81 ± 8.88	79.75 ± 8.84	82.06 ± 9.04	**82.56** ± 7.61	78.44 ± 8.58
heart-statlog	82.89 ± 6.31	82.74 ± 6.37	**83.37** ± 5.92	83.22 ± 5.92	81.52 ± 6.65
ionosphere	91.89 ± 4.52	92.80 ± 4.07	91.54 ± 4.25	**92.83** ± 4.27	92.82 ± 3.98
iris	95.53 ± 4.84	95.80 ± 4.61	**96.47** ± 4.18	95.27 ± 5.22	95.73 ± 4.79
leaf	56.74 ± 8.79	64.18 ± 8.76 ○	**70.00** ± 7.69 ○	60.85 ± 7.88 ○	61.38 ± 8.89 ○
libras	75.25 ± 6.88	77.97 ± 6.71	**79.81** ± 6.73 ○	77.58 ± 6.57	79.69 ± 6.07 ○
liver	65.23 ± 7.70	64.73 ± 7.52	65.74 ± 7.95	**69.26** ± 7.66	66.47 ± 7.50
magic	83.29 ± 0.69	83.11 ± 0.70	82.93 ± 0.74 ●	**86.17** ± 0.71 ○	83.94 ± 0.71 ○
mammo	80.75 ± 3.36	81.06 ± 3.54	81.07 ± 3.76	**81.15** ± 3.71	79.92 ± 3.77
mfeat-fourier	80.00 ± 2.55	80.92 ± 2.63	**81.56** ± 2.45 ○	81.01 ± 2.46 ○	81.31 ± 2.44 ○
monks-1	99.17 ± 1.58	99.25 ± 1.47	95.27 ± 3.57 ●	99.46 ± 1.21	**99.95** ± 0.31
new-thyroid	94.80 ± 4.61	95.82 ± 4.24	**96.28** ± 3.85	95.45 ± 4.17	95.25 ± 4.17
optdigits	95.55 ± 0.93	95.97 ± 0.92 ○	95.68 ± 0.93	96.68 ± 0.78 ○	**96.79** ± 0.83 ○
page-blocks	94.18 ± 0.73	94.72 ± 0.83 ○	95.06 ± 0.76 ○	**95.51** ± 0.72 ○	95.09 ± 0.76 ○
parkinsons	90.44 ± 6.83	90.74 ± 6.62	90.75 ± 6.80	90.76 ± 6.55	**92.33** ± 6.46
planning	56.73 ± 11.40	64.74 ± 8.44 ○	**65.13** ± 7.88 ○	54.85 ± 11.03	60.52 ± 9.46
qsar-biodegradation	84.87 ± 3.39	85.06 ± 3.75	85.06 ± 3.32	**87.11** ± 3.24 ○	85.64 ± 3.40
robot24	94.68 ± 1.00	94.51 ± 1.05	94.42 ± 1.05	**96.74** ± 0.81 ○	95.01 ± 1.00
seeds	91.95 ± 5.93	92.43 ± 5.75	**93.90** ± 5.25	93.24 ± 5.29	92.81 ± 5.22
segment	95.43 ± 1.46	95.41 ± 1.47	95.87 ± 1.46	**96.47** ± 1.25 ○	96.33 ± 1.24 ○
sonar	79.36 ± 9.73	81.75 ± 7.90	78.40 ± 9.46	80.88 ± 8.94	**83.77** ± 8.66 ○
spectf-heart	91.30 ± 4.61	91.07 ± 4.65	**91.38** ± 4.57	91.21 ± 4.32	90.41 ± 4.77
statlog-g	76.23 ± 3.18	76.40 ± 3.22	76.02 ± 3.53	**76.60** ± 3.52	75.13 ± 3.29
statlog-h	82.59 ± 6.35	82.67 ± 5.69	83.11 ± 6.19	**83.33** ± 6.24	81.78 ± 6.20
steel-plates-faults	98.74 ± 0.93	98.89 ± 0.87	**99.80** ± 0.32 ○	99.65 ± 0.39 ○	99.18 ± 0.70 ○
texture	97.13 ± 0.72	96.60 ± 0.77 ●	97.92 ± 0.61 ○	97.53 ± 0.68 ○	**97.98** ± 0.57 ○
turkiye	34.11 ± 1.79	33.03 ± 1.70 ●	33.97 ± 1.69	**35.16** ± 1.89 ○	34.33 ± 1.74
user	87.78 ± 5.17	88.08 ± 5.45	**90.67** ± 4.45 ○	89.77 ± 4.44	88.40 ± 4.58
vehicle	72.53 ± 4.26	72.11 ± 4.09	73.86 ± 3.68	**73.92** ± 4.26	73.45 ± 4.05
waveform-5000	84.53 ± 1.62	84.89 ± 1.64	85.06 ± 1.68 ○	**85.17** ± 1.58	84.74 ± 1.68
wdbc	95.64 ± 2.69	95.77 ± 2.62	96.03 ± 2.73	**96.10** ± 2.63	95.26 ± 2.84
wholesale-c	86.93 ± 4.72	87.84 ± 4.47	88.20 ± 4.34	**89.70** ± 4.49 ○	87.86 ± 4.47
wholesale-r	69.91 ± 2.53	70.18 ± 2.17	**70.27** ± 2.11	69.95 ± 2.43	68.09 ± 3.21
wilt	94.62 ± 0.18	94.63 ± 0.16	94.63 ± 0.14	**94.63** ± 0.17	94.63 ± 0.16
wine-rec	96.28 ± 3.93	96.45 ± 3.72	**97.30** ± 3.86	96.79 ± 3.78	96.24 ± 4.07
wine-red	66.49 ± 3.79	67.04 ± 3.81	66.84 ± 3.68	67.11 ± 3.69	**69.08** ± 3.03 ○
wine-white	64.10 ± 1.98	64.60 ± 1.95	65.02 ± 2.04 ○	66.67 ± 1.96 ○	**68.01** ± 1.92 ○
Average	80.45	81.20	81.57	**82.85**	82.25
Significant W/D/L	–	3/46/6	2/38/15	0/35/20	1/37/17
W/D/L	–	14/0/41	12/0/43	7/0/48	13/0/42

Table 4: Classification accuracy along the incremental component integration path.

Dataset	MAWNB	MAWNB-A	MAWNB-AS	MAWNB-ASB	CMAWNB
abalone	54.52 ± 2.07	**54.65** ± 2.10	54.59 ± 2.06	53.62 ± 1.97	53.91 ± 2.23
appendicitis	81.40 ± 8.56	84.62 ± 8.13	87.25 ± 8.55 ○	87.16 ± 7.90	**87.45** ± 8.00 ○
artificial-characters	67.44 ± 1.47	67.76 ± 1.40	68.39 ± 1.39 ○	92.34 ± 1.08 ○	**94.37** ± 0.99 ○
australian	77.82 ± 1.22	79.37 ± 1.26 ○	77.21 ± 1.31	84.52 ± 1.01 ○	**87.04** ± 0.95 ○
balance-scale	87.01 ± 2.41	86.19 ± 2.55 ●	**88.72** ± 2.31	86.72 ± 2.59	85.71 ± 3.17
breast-c	96.81 ± 2.01	**96.90** ± 1.90	96.77 ± 1.96	96.57 ± 2.02	96.62 ± 2.10
breast-t	37.16 ± 11.53	40.53 ± 11.32	**47.76** ± 11.60 ○	44.84 ± 14.30	44.94 ± 13.91
cardioto	98.18 ± 0.96	98.38 ± 0.90	99.03 ± 0.73 ○	99.06 ± 0.75 ○	**99.07** ± 0.73 ○
cardiotocography-2	92.14 ± 1.59	92.03 ± 1.72	90.82 ± 1.78 ●	92.39 ± 1.62	**93.95** ± 1.41 ○
climate	89.69 ± 1.90	90.52 ± 1.69	91.15 ± 1.61 ○	**91.28** ± 1.40 ○	90.67 ± 2.29
connectionist-vowel	90.13 ± 4.25	90.74 ± 4.21	90.51 ± 3.94	96.12 ± 2.99 ○	**96.80** ± 2.70 ○
diabetes	74.95 ± 4.88	75.29 ± 4.95	75.32 ± 5.06	75.52 ± 4.37	**76.25** ± 4.76
ecoli	82.00 ± 6.21	83.21 ± 5.45	84.73 ± 5.24	84.31 ± 5.32	**85.69** ± 5.65
eeg-eye	57.19 ± 1.17	58.34 ± 1.10 ○	57.94 ± 1.76	83.78 ± 3.95 ○	**89.70** ± 1.94 ○
energy-efficiency-y1	**67.78** ± 3.51	67.69 ± 3.74	66.96 ± 4.20	64.96 ± 4.41 ●	64.92 ± 4.41
energy-efficiency-y2	54.48 ± 5.03	54.24 ± 5.08	53.88 ± 4.87	54.78 ± 5.04	**55.98** ± 4.43
fertility	85.00 ± 8.35	85.70 ± 7.00	85.80 ± 6.06	**86.40** ± 6.12	86.30 ± 6.61
glass	63.72 ± 8.90	66.27 ± 8.56	67.54 ± 8.44	72.11 ± 9.08 ○	**75.33** ± 8.27 ○
hayes-roth	79.81 ± 8.88	79.75 ± 8.84	**80.31** ± 9.42	77.62 ± 10.10	79.56 ± 9.10
heart-statlog	82.89 ± 6.31	82.74 ± 6.37	**83.96** ± 5.74	82.56 ± 6.71	82.48 ± 6.84
ionosphere	91.89 ± 4.52	92.80 ± 4.07	92.77 ± 3.99	93.65 ± 3.71	**94.53** ± 3.66 ○
iris	95.53 ± 4.84	95.80 ± 4.61	**96.47** ± 4.39	96.20 ± 4.57	96.00 ± 4.34
leaf	56.74 ± 8.79	64.18 ± 8.76 ○	75.82 ± 6.90 ○	74.94 ± 7.08 ○	**78.35** ± 6.18 ○
libras	75.25 ± 6.88	77.97 ± 6.71	84.36 ± 6.02 ○	86.75 ± 5.10 ○	**87.42** ± 5.18 ○
liver	65.23 ± 7.70	64.73 ± 7.52	64.78 ± 7.44	66.31 ± 8.44	**68.45** ± 7.83
magic	83.29 ± 0.69	83.11 ± 0.70	82.28 ± 0.69 ●	83.26 ± 0.66	**85.58** ± 0.66 ○
mammo	80.75 ± 3.36	81.06 ± 3.54	**81.23** ± 3.55	80.24 ± 3.58	79.75 ± 3.37
mfeat-fourier	80.00 ± 2.55	80.92 ± 2.63	82.09 ± 2.45 ○	82.97 ± 2.35 ○	**84.05** ± 2.29 ○
monks-1	99.17 ± 1.58	**99.25** ± 1.47	95.49 ± 3.06 ●	94.39 ± 2.80 ●	94.46 ± 2.81 ●
new-thyroid	94.80 ± 4.61	95.82 ± 4.24	**96.83** ± 3.70	96.27 ± 4.24	96.27 ± 3.75
optdigits	95.55 ± 0.93	95.97 ± 0.92 ○	95.96 ± 0.84	97.44 ± 0.57 ○	**98.28** ± 0.51 ○
page-blocks	94.18 ± 0.73	94.72 ± 0.83 ○	95.19 ± 0.82 ○	95.81 ± 0.69 ○	**96.49** ± 0.72 ○
parkinsons	90.44 ± 6.83	90.74 ± 6.62	91.26 ± 7.15	93.03 ± 6.13	**93.24** ± 5.89
planning	56.73 ± 11.40	64.74 ± 8.44 ○	67.55 ± 6.03 ○	**67.76** ± 6.64 ○	66.18 ± 7.49 ○
qsar-biodegradation	84.87 ± 3.39	85.06 ± 3.75	85.70 ± 3.41	86.58 ± 3.42 ○	**87.69** ± 2.99 ○
robot24	**94.68** ± 1.00	94.51 ± 1.05	91.79 ± 1.19 ●	92.36 ± 1.11 ●	94.62 ± 1.00
seeds	91.95 ± 5.93	92.43 ± 5.75	94.24 ± 4.75	94.38 ± 4.85	**94.76** ± 4.71
segment	95.43 ± 1.46	95.41 ± 1.47	95.40 ± 1.38	96.78 ± 1.20 ○	**97.67** ± 0.98 ○
sonar	79.36 ± 9.73	81.75 ± 7.90	82.00 ± 8.85	86.26 ± 8.03 ○	**87.70** ± 7.56 ○
spectf-heart	**91.30** ± 4.61	91.07 ± 4.65	90.63 ± 4.69	89.17 ± 5.10	88.89 ± 5.09
statlog-g	76.23 ± 3.18	**76.40** ± 3.22	76.24 ± 3.67	75.32 ± 3.65	75.54 ± 3.94
statlog-h	82.59 ± 6.35	82.67 ± 5.69	**83.89** ± 5.75	82.63 ± 6.20	82.33 ± 6.44
steel-plates-faults	98.74 ± 0.93	98.89 ± 0.87	99.88 ± 0.27 ○	99.85 ± 0.27 ○	**99.89** ± 0.23 ○
texture	97.13 ± 0.72	96.60 ± 0.77 ●	97.29 ± 0.69	**98.77** ± 0.44 ○	98.62 ± 0.50 ○
turkiye	34.11 ± 1.79	33.03 ± 1.70 ●	32.81 ± 1.69 ●	33.36 ± 1.65	**34.31** ± 1.70
user	87.78 ± 5.17	88.08 ± 5.45	90.69 ± 4.63	90.44 ± 4.70	**91.06** ± 4.09 ○
vehicle	72.53 ± 4.26	72.11 ± 4.09	72.58 ± 4.03	74.34 ± 4.06	**74.85** ± 4.14
waveform-5000	84.53 ± 1.62	84.89 ± 1.64	85.40 ± 1.59 ○	85.46 ± 1.56 ○	**85.76** ± 1.66 ○
wdbc	95.64 ± 2.69	95.77 ± 2.62	96.26 ± 2.41	95.68 ± 2.76	**96.33** ± 2.48
wholesale-c	86.93 ± 4.72	87.84 ± 4.47	89.66 ± 3.79 ○	89.77 ± 3.90 ○	**90.66** ± 4.31 ○
wholesale-r	69.91 ± 2.53	70.18 ± 2.17	**71.16** ± 1.50	69.34 ± 2.90	68.05 ± 3.42
wilt	94.62 ± 0.18	94.63 ± 0.16	94.62 ± 0.13	94.62 ± 0.09	**96.96** ± 0.52 ○
wine-rec	96.28 ± 3.93	96.45 ± 3.72	97.64 ± 3.01	97.87 ± 3.27	**98.20** ± 2.88
wine-red	66.49 ± 3.79	67.04 ± 3.81	66.85 ± 3.79	68.85 ± 3.13 ○	**68.95** ± 3.07 ○
wine-white	64.10 ± 1.98	64.60 ± 1.95	65.08 ± 1.82 ○	68.25 ± 1.79 ○	**69.28** ± 1.92 ○
Average	80.45	81.20	82.01	83.38	**84.14**
Significant W/D/L	–	3/46/6	5/36/14	3/30/22	1/27/27
W/D/L	–	14/0/41	12/0/43	14/0/41	13/0/42