Iranian Journal of Fuzzy Systems

Volume 22, Number 5, (2025), pp. 49-68

Original paper

University of Sistan and Baluchestan

Building classification algorithm based on combining the improved principal component analysis and fuzzy relationship of Bayesian approach

H. Huynh-Van [1] and T. Vo-Van [2]

[1] *Faculty of Applied Sciences, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam*
[1] *Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam*
[1] *Faculty of Fundamental Science, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam*
[2] *College of Natural Science, Can Tho University, Can Tho City, Vietnam.*

huynhvanhieu@iuh.edu.vn, vvtai@ctu.edu.vn

Abstract

This article develops a novel classification algorithm that integrates the improved Bayesian approach with Principal Component Analysis (PCA). First, PCA significantly reduces the dimensionality of the data while enhancing inter-class separability, thereby improving classification performance. Next, the prior probabilities are refined based on newly derived variables established for both the training and test sets using a fuzzy clustering analysis technique. Finally, the Bayesian classification rule is constructed based on the probability density functions of the classes, using the transformed variables obtained from PCA and the estimated prior probabilities. The algorithm is presented in detail, including procedural steps, illustrative example, and implementation on both numerical and image data through a the establised RStudio procedure. A key contribution of this study lies in the theoretical enhancement of Bayes error analysis and the proof of convergence of the proposed algorithm. Empirical applications demonstrate that the proposed algorithm yields stable results and outperforms several existing approaches when applied to certain numerical and image datasets using statistical parameters.

Keywords: Bayesian classifier, fuzzy relationship, image analysis, posterior probability, principal component analysis.

1 Introduction

Classification is to assign an element to a suitable group based on its variables. This field is fundamental in data science and has widespread applications across various domains. For example, in agriculture, classification categorizes crops, goods, or land [10, 37]. In economics, it plays a crucial role in customer segmentation and credit scoring [19, 20]. Medicine is currently one of the most prominent fields for classification applications, particularly in disease diagnosis [22]. Despite significant interest from practitioners and researchers, classification remains an ongoing challenge, especially in optimizing real-world applications [21, 25, 27, 28, 34].

Classification algorithms can be broadly categorized into two main approaches: statistical and machine learning. Key statistical classification approaches include Fisher's discriminant analysis [11], logistic regression [12, 13], and Naive Bayes for discrete variables [25]. The Fisher approach has a simple classification principle but is constrained by the assumption that covariance matrices among groups are equal, an often unrealistic condition in practical applications [31, 35]. While widely used, the logistic regression approach is only applicable for two groups when they are linearly separable [38]. Naive Bayes, a foundational machine learning approach, is often impractical due to its assumption that real-world events are conditionally independent [8].

Corresponding Author: T. Vo-Van
Received: July 2025; Revised: August 2025; Accepted: September 2025.
https://doi.org/10.22111/ijfs.2025.52578.9280

From a machine learning perspective, several commonly used classification approaches include Quadratic Discriminant Analysis (QDA), XGBoost, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Random Trees, K-Nearest Neighbors (kNN), and Subspace-kNN. QDA extends Fisher's approach because it does not require equal covariance matrices across groups [6]. XGBoost sequentially builds decision trees to provide accurate and scalable results while reducing overfitting [3]. However, its performance declines when dealing with data that has significant overlap between groups. SVM is effective for high-dimensional data but can be unstable and prone to high error rates when the data contains uneven factors [2, 14]. ANN is a popular approach for detecting complex nonlinear relationships between variables but is computationally expensive and sensitive to initial parameter settings, which depend on the researcher's experience [23]. The Random Tree algorithm, a supervised classification approach, generates an ensemble of decision trees and classifies them based on the most frequent output. However, training time becomes a significant limitation as the number of trees increases. Adaptive Boosting, a statistical classification meta-algorithm, enhances the performance of other learning algorithms. While it is robust, it struggles with handling noise and outliers and requires expertise in selecting the number of iterations and other parameters [39]. K-Nearest Neighbors (kNN) is a simple yet powerful approach that can achieve high classification accuracy. However, its performance heavily depends on parameter selection and the choice of distance metric. Additionally, it requires computing distances between the target element and all training samples, making it computationally expensive [1]. Subspace-kNN, which combines predictions from multiple decision trees trained on different subsets, shares similar computational drawbacks as kNN [38].

Machine learning classification algorithms are inherently linked to Bayesian statistics. Bayes-based approaches for continuous variables require fewer restrictive assumptions and can handle multiple groups [24, 32, 33, 35], which makes them theoretically appealing. However, several practical limitations remain [9, 15, 16]. First, prior probabilities are often determined subjectively and may not generalize well to diverse datasets. To address this, statistical alternatives such as uniform distribution, sample proportion, and the Laplace approach have been proposed. Second, estimating the probability density function is challenging, as it frequently assumes variable independence and struggles when correlations exist in the data. Finally, computational complexity poses a significant obstacle, particularly in calculating the maximum likelihood function and performing multidimensional integration [32, 36], challenges that are amplified in high-dimensional and multicollinear datasets. Recently, numerous studies have introduced hybrid approaches to address the limitations of traditional models, including Adaptive Proximity Neighbors (APN) for arrhythmia classification [7], enhanced K-means updating rules to improve clustering accuracy in engineering applications [4], and neural network–based models deployed on edge computing platforms [26]. Although these studies are highly meaningful in terms of applications, they do not provide any theoretical breakthroughs in classification. Following this line of research, the present study proposes an integrated framework that combines Principal Component Analysis (PCA), Fuzzy C-means clustering, and Kernel Density Estimation to enhance classification performance.

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique in data science and artificial intelligence. It identifies the most important principal components by analyzing the variance-covariance structure of a dataset through linear combinations. This transformation converts the original variables into new, uncorrelated principal components while preserving as much information as possible. Given these advantages, PCA is a logical approach for addressing challenges related to high-dimensional data and multicollinearity, potentially enhancing classification performance in practical applications. In carrying out classification problem, we often have to deal with datasets containing a large number of variables. Using many variables increases the computational cost of classification. Moreover, when irrelevant or noisy variables are included, the classification performance may decline. In practice, selecting and using important variables has been shown to improve classification results. This is considered the main motivation for applying the PCA technique in classification problems. To the best of our knowledge, the simultaneous integration of PCA and improved Bayesian prior probabilities within a unified framework for classification problems has not received much attention so far.

This study focuses on classification problem involving datasets with a large number of variables and aims to improve practical efficiency through the following key contributions:

(i) Optimizing the number of variables in classification by incorporating improved PCA techniques. This approach not only reduces computational complexity but also enhances classification accuracy.

(ii) Enhancing the Bayesian classification algorithm by making reasonable assumptions for prior probability and PDF estimation based on the training set while leveraging the optimized input data from (i).

(iii) Developing a statistical software implementation of the proposed algorithm, enabling practical application to numerical and image data.

Compared to many existing statistical and machine learning classification approaches, the proposed algorithm has demonstrated stability and competitive performance across diverse datasets, including numerical and image data with varying characteristics.

The remainder of the paper is structured as follows. Section 2 discusses key issues related to the proposed algorithm,

which is introduced in Section 3. Sections 4 and 5 apply the algorithm to numerical and image data. Finally, the paper concludes with a summary.

2 Bayesian classier

2.1 Classification principle and Bayes error

* Let $f_i(x)$ be the PDFs of class w_i, for $i = 1, 2, ..., k$. An element with variable x_0 will be classified into w_i if

$$i = \arg\max_{1\le i\le k} P(w_i|x_0) = \arg\max_{1\le i\le k} \left\{ \frac{q_i f_i(x_0)}{f(x_0)} \right\}, \tag{1}$$

where $f(x) = q_1 f_1(x) + q_2 f_2(x) + ... + q_k f_k(x)$, q_i is the prior probability of classifying x_0 into w_i, and satisfies $q_1 + q_2 + \ldots + q_k = 1$.

Since $f(x)$ is identical for all w_i, $i = 1, 2, \ldots, k$, the classification rule (1) becomes: An element with variable x_0 will be classified into w_i if

$$i = \arg\max_{1\le i\le k} \{q_i f_i(x_0)\}. \tag{2}$$

* The theoretical error when classifying according to the above rule is called the Bayes error, which is defined as:

$$Pe = 1 - \int_{R^n} g_{\max}(x) \mathrm{d}\, x, \tag{3}$$

where $g_{\max}(x) = \max_i\{g_i(x)\} = \max_i\{q_i f_i(x)\}$, $i = \overline{1, k}$.

2.2 Bound of Bayes error

Theorem 2.1. *Let $f_i(x), i = 1, 2, ..., k$ be k PDFs defined on $\mathbb{R}^n, n \ge 1, q_i \in (0; 1)$. We have the following results about the lower and upper bounds for Bayes error:*

1. *Bayes error and prior probabilities:*
$$0 \le Pe \le 1 - \max_i \{q_i\}, k \ge 2. \tag{4}$$

2. *Bayes error and L_1 distance:*
$$LB \le Pe \le UB, \tag{5}$$
where
$g_i(x) = q_i f_i(x), i = 1, 2, ..., k$, $LB = \frac{1}{k}\left(k - 1 - \sum\limits_{i<j}\int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})| \,\mathrm{d}\mathbf{x}\right)$,
$UB = \frac{1}{2}\max_{i<j}\left\{\int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})| \,\mathrm{d}\mathbf{x}\right\} + \min_i \{q_i\}$.

Proof. 1. For each $i = 1, 2, \ldots, k$, we have

$$q_i f_i(\mathbf{x}) \le \max_i \{q_i f_i(\mathbf{x})\} \le \sum_{i=1}^{k} q_i f_i(\mathbf{x}), \quad \text{and} \quad \int\limits_{\mathbb{R}^n} f_i(\mathbf{x}) \,\mathrm{d}\mathbf{x} = 1.$$

Integrating the inequality above, we obtain

$$q_i \le \int\limits_{\mathbb{R}^n} g_{\max}(\mathbf{x}) \,\mathrm{d}\mathbf{x} \le 1.$$

Since the inequality holds for every q_i, we conclude

$$\max_i \{q_i\} \le \int\limits_{\mathbb{R}^n} g_{\max}(\mathbf{x}) \,\mathrm{d}\mathbf{x} \le 1,$$

or equivalently,

$$0 \leq 1 - \int\limits_{\mathbb{R}^n} g_{\max}(\mathbf{x})\,\mathrm{d}\mathbf{x} \leq 1 - \max_i \{q_i\}\,.$$

Thus, the first result is proven.

2. We have

$$\begin{aligned}
\int\limits_{\mathbb{R}^n} g_{\max}(\mathbf{x})\,\mathrm{d}\mathbf{x} &\geq \max_{i<j} \int\limits_{\mathbb{R}^n} \max\{g_i(\mathbf{x}), g_j(\mathbf{x})\}\,\mathrm{d}\mathbf{x} \\
&= \max_{i<j} \left\{ \frac{1}{2} \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} + \frac{1}{2}(q_i + q_j) \right\} \\
&\geq \max_{i<j} \left\{ \frac{1}{2} \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} \right\} + \min_{i<j} \left\{ \frac{1}{2}(q_i + q_j) \right\} \\
&\geq \max_{i<j} \left\{ \frac{1}{2} \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} \right\} + \min_i \{q_i\}\,.
\end{aligned}$$

Therefore,

$$Pe^{(q)}_{1,2,\ldots,k} \leq \frac{1}{2} \max_{i<j} \left\{ \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} \right\} + \min_i \{q_i\}\,. \tag{6}$$

Moreover, since

$$\begin{aligned}
\sum_{i<j} \sum_j |g_i(\mathbf{x}) - g_j(\mathbf{x})| &\geq \sum_{j=1}^{k} \left[\max_i \{g_i(\mathbf{x})\} - g_j(\mathbf{x}) \right] \\
&= k\left[\max_i \{g_i(\mathbf{x})\}\right] - \sum_{j=1}^{k} g_j(\mathbf{x}),
\end{aligned}$$

we have

$$\max_i \{g_i(\mathbf{x})\} \leq \frac{1}{k} \sum_{i<j} \sum_j |g_i(\mathbf{x}) - g_j(\mathbf{x})| + \frac{1}{k} \sum_{j=1}^{k} g_j(\mathbf{x}).$$

Since

$$\sum_{j=1}^{k} \int\limits_{\mathbb{R}^n} g_j(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1,$$

we obtain the inequality:

$$\int\limits_{\mathbb{R}^n} g_{\max}(\mathbf{x})\,\mathrm{d}\mathbf{x} \leq \frac{1}{k} \sum_{i<j} \sum_j \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} + \frac{1}{k}. \tag{7}$$

From the result in (7), we obtain:

$$Pe^{(q)}_{1,2,\ldots,k} \geq \frac{1}{k} \left(k - 1 - \sum_{i<j} \int\limits_{\mathbb{R}^n} |g_i(\mathbf{x}) - g_j(\mathbf{x})|\,\mathrm{d}\mathbf{x} \right). \tag{8}$$

Combining (6) and (8), we obtain inequality (5).

□

2.3 Evaluation parameters

In classification problem, four important metrics commonly used to evaluate performance of algorithms are Precision, Recall, F1-score, and accuracy (ACC). These are calculated based on the confusion matrix, with the following concepts:

* TP (True Positive): The number of positive samples correctly predicted.
* FP (False Positive): The number of negative samples incorrectly predicted as positive.
* FN (False Negative): The number of positive samples incorrectly predicted as negative.
* TN (True Negative): The number of negative samples correctly predicted.

Based on that, we have the following metrics:

- Precision = $\frac{TP}{TP+FP}$.
- Recall = $\frac{\text{TP}}{\text{TP+FN}}$.
- F1-score = $\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$.
- ACC = $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

For these metrics, if their values are larger, the classification algorithm will perform better.

3 The proposed algorithm

3.1 The algorithm

Given an initial dataset consisting of p variables $\{x_1, x_2, \ldots, x_p\}$ and n observations, which are divided into k groups $\{w_1, w_2, \ldots, w_k\}$. The dataset is represented in matrix form as follows:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix}.$$

Let $x_0 = \{x_{01}\ x_{02}\ ...\ x_{0p}\}$ be the elements to be classified into groups. The proposed classification algorithm includes the following steps:

Step 1: Standardize the data X and x_0 by (9).

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}, i = 0, 1, ..., n; j = 1, 2, ..., p, \tag{9}$$

where

$\bar{x}_j$ and σ_j are the mean and standard deviation of the variable x_j from the original data, respectively.

x_{ij} is the value of the jth variable for the ith element.

After this step, we obtain a new dataset set Z normalized from X and the classified element $z_0 = (z_{01}\ z_{02}\ ...\ z_{0p})$ normalized from x_0.

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} & \ldots & z_{1p} \\ z_{21} & z_{22} & \ldots & z_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \ldots & z_{np} \end{bmatrix}.$$

Step 2: Perform Principal Component Analysis (PCA) on the training data Z and determine the orthogonal matrix P. Suppose the result of this process has q principal components ($q < p$) selected from the p columns of the matrix $Y = P^T Z^T$.

Thus, each initial element will be considered across q variables after this step. $(y_1, y_2, ..., y_q)$, $q < p$.

Step 3: Using the orthogonal matrix P, compute $y_0 = P^T z_0^T$.

This step also changes the p observed variables of the classified element $z_0 = (z_{01}\ z_{02}\ ...\ z_{0p})$ to q variables $y_0 = (y_{01}\ y_{02}\ ...\ y_{0q})$.

Step 4: Estimate the joint density function for each group $f_i(y_1, y_2, \ldots, y_q),\ i = 1, 2, \ldots, k$ based on the q variables corresponding to the first q columns of the matrix Y from Step 2.

Step 5: Compute the prior probabilities $u_i, i = 1, 2, ..., k$ for classifying y_{n+1} to the groups by the following steps:
Step 5.1: Set up an initial partition matrix $U^{(0)} = [\mu_{ij}^{(0)}]_{k\times(n+1)}$, where $\mu_{ij}^{(0)}$ satisfies the following conditions:

$$\mu_{ij}^{(0)} \in [0,1], \sum_{i=1}^{k} \mu_{ij}^{(0)} = 1, 0 < \sum_{j=0}^{n} \mu_{ij}^{(0)} < n+1, \tag{10}$$

$\mu_{ij}^{(0)}$ is the probability to assign the jth element to w_i, $i = 1, 2, ..., k$. Therefore, $\mu_{i0}^{(0)}$ is the initial probability of classifying y_0 to w_i.

In the implementation of this study, we choose $\mu_{ij}^{(0)}, j = 1, 2, ..., k$ in the matrix $U^{(0)}$ satisfying condition (10) as follows:

- For the first column (Column $\mu_{i0}^{(0)}$), $\mu_{ij}^{(0)}$ will be chosen according to the uniform distribution, i.e., $\mu_{i0}^{(0)} = 1/k$.
- For the remainder n columns, $\mu_{ij}^{(0)} = 1$ if the jth element belongs to w_i, and $\mu_{ij}^{(0)} = 0$ if the jth element does not belong to w_i.

Step 5.2: Find the prototype element for each group by formula (11):

$$v_i^{(0)} = \frac{1}{\sum\limits_{j=0}^{n} \left(\mu_{ij}^{(0)}\right)^2} \sum_{j=0}^{n} \left(\mu_{ij}^{(0)}\right)^2 y_j,\ i = 1, 2, ..., k, \tag{11}$$

Step 5.3: Update the new partition matrix $U^{(m)}$ by the following principle:

$$\mu_{ij}{}^{(m)} = \frac{1}{\sum\limits_{l=1}^{k} \left(d(v_i^{(m-1)}, y_j)/d(v_l^{(m-1)}, y_j)\right)^2}, j = 0, 1, ..., n, \tag{12}$$

where $d(.)$ is the Euclidean distance.
Step 5.4: Compute $V^{(m)}$ from $U^{(m)}$:

$$v_i^{(m)} = \frac{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2 y_j}{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2}, \quad i = 1, 2, \ldots, k. \tag{13}$$

Compute $\text{WMPDC}_m = \frac{1}{n+1} \sum_{j=0}^{n} \sum_{i=1}^{k} \mu_{ij}^{(m)} \cdot \left| d(v_i^{(m)}, y_j) - d(v_i^{(m-1)}, y_j) \right|$.
Repeat Step 5.3 and Step 5.4 m times until $\text{WMPDC}_m < \gamma$, where γ is an arbitrarily small positive constant, chosen as 0.001 in this study.

When Step 5 ends, we obtain the matrix with the first column as $u = \mu_{i0}^{(m)}$. Because u is the probability of assigning y_0 to the groups before applying the classification approach, it is considered the prior probability.

Step 6: Compute the posterior indices $u_i f_i(y_0), i = 1, 2, ..., k$ and classify y_0 into group $w_c, c = 1, 2, ..., k$ if

$$u_c.f_c(y_0) = \max\{u_i.f_i(y_{n0})\}, i = 1, 2, ..., k. \tag{14}$$

The flowchart and pseudocode of the proposed algorithm are presented in Figure 1 and Algorithm 1, respectively.

3.2 Detailed description of the proposed algorithm

For the proposed algorithm, we need to pay attention to the following problems:

- In Step 1, the study normalizes the data to ensure fairness among variables with different scales in their contribution to the classification. Normalization using the formula (9) is called the Z-score method. It is easy to see that the covariance matrix Σ of the original data X becomes the correlation matrix of the normalized data Z. Therefore, performing PCA on the normalized data essentially analyzes the original data's correlation matrix. PCA does not retain the measurement units of the original data but relies solely on the covariance matrix. As a result, variables with different measurement units will have an unbalanced influence on the analysis results. Using the data after normalization helps PCA produce reasonable principal components that accurately reflect the nature of the data without being affected by the different variables' dispersion levels.

Algorithm 1 Classification Algorithm

Input: y, training set X, classified element x_0 with variables $x_1, x_2, \ldots, x_p$
▷ Step 1: Standardize the data X and x_0
for $j = 1$ to p **do**
  for $i = 0$ to n **do**
    $z_{ij} \leftarrow \frac{x_{ij} - \bar{x}_j}{\sigma_j}$
$Z \leftarrow$ matrix$[z_{ij}]$ (normalized dataset from X)
$z_0 \leftarrow [z_{01}, z_{02}, \ldots, z_{0p}]$ (normalized from x_0)
▷ Step 2: Perform Principal Component Analysis (PCA)
Compute P (orthogonal matrix with q principal components, $q < p$)
$Y \leftarrow P^T \cdot Z^T$
$y_0 \leftarrow P^T \cdot z_0^T$ (transform x_0 to q variables $[y_{01}, y_{02}, \ldots, y_{0q}]$)
▷ Step 3: Estimate joint density function for each group
for $i = 1$ to k **do**
  Estimate $f_i(y_1, y_2, \ldots, y_q)$ (based on q variables from Y)
▷ Step 4: Compute prior probabilities u_i
$U^{(0)} \leftarrow [\mu_{ij}^{(0)}]$ (k x (n+1) initial partition matrix)
for $i = 1$ to k **do**
  $\mu_{i0}^{(0)} \leftarrow 1/k$ (uniform distribution for first column)
  for $j = 1$ to n **do**
    if jth element belongs to w_i **then**
      $\mu_{ij}^{(0)} \leftarrow 1$
    else
      $\mu_{ij}^{(0)} \leftarrow 0$
▷ Step 5: Iterative update until convergence
$m \leftarrow 0$
while $WMPDC_m \geq \gamma$ **do**
▷ Step 5.1: Find prototype element for each group
  for $i = 1$ to k **do**
    $v_i^{(m)} \leftarrow \frac{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2 \cdot y_j}{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2}$
▷ Step 5.2: Update partition matrix
  for $j = 0$ to n **do**
    for $i = 1$ to k **do**
      $\mu_{ij}^{(m)} \leftarrow \frac{1}{\sum_{l=1}^{k} \left(\frac{d(v_i^{(m-1)}, y_j)}{d(v_l^{(m-1)}, y_j)} \right)^2}$
▷ Step 5.3: Compute new prototypes
  for $i = 1$ to k **do**
    $v_i^{(m)} \leftarrow \frac{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2 \cdot y_j}{\sum_{j=0}^{n} (\mu_{ij}^{(m)})^2}$
▷ Step 5.4: Compute $WMPDC_m$
  $WMPDC_m \leftarrow \frac{1}{n+1} \cdot \sum_{j=0}^{n} \sum_{i=1}^{k} \mu_{ij}^{(m)} \cdot |d(v_i^{(m)}, y_j) - d(v_i^{(m-1)}, y_j)|$
  $m \leftarrow m + 1$
$u \leftarrow [\mu_{i0}^{(m)}]$ (prior probabilities for y_0)
▷ Step 6: Classify y_0
$u_c \leftarrow \arg\max(u_i \cdot f_i(y_0))$ for $i = 1$ to k
Classify y_0 into group w_c
Output: classified y_0

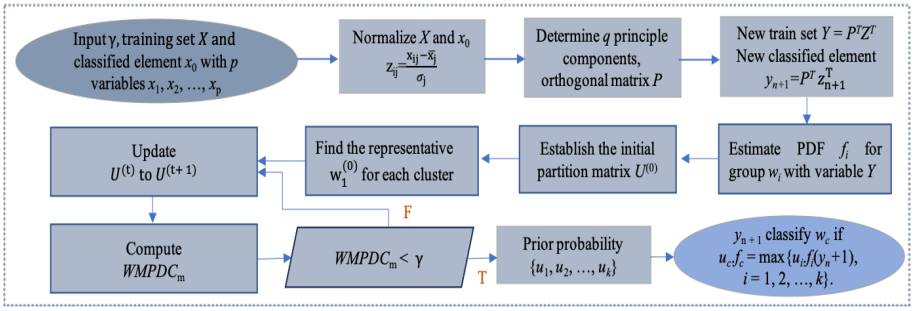


Figure 1: The flowchart of the proposed algorithm

- The purpose of Step 2 is to reduce the dimensionality of the variables to improve the efficiency of classification using the PCA method. In PCA, selecting k principal components from the original p variables in the dataset is crucial. Several methods have been used to address this issue, such as the Scree Plot Method [5] and Kaiser's Criterion [18]. This study employs the Cumulative Explained Variance Ratio (CEVR) approach to provide a reasonable criterion for determining the number of principal components. According to [17], this approach is based on the criterion that the first q principal components should be chosen such that they retain between 70% and 90% of the total variance in the data. The formula defines this ratio:

$$p^* = \frac{\sum_{j=1}^{q} \lambda_j}{\sum_{j=1}^{p} \lambda_j}, \quad q < p. \tag{15}$$

  This approach can preserve most of the original data information while eliminating noise and reducing the complexity of statistical models built from the q principal components. This approach is widely used due to its objectivity and strong generalizability in real-world applications.

- Step 4 involves estimating the probability density function (PDF) for the data from the q variables $\{y_1, y_2, \ldots, y_q\}$ obtained in Step 3. Various parametric and non-parametric approachs for PDF estimation have been considered, with kernel density estimation (KDE) identified as the most suitable for the proposed algorithm. The kernel density estimate is given by:

$$\widehat{f}(y) = \frac{1}{nh_1h_2\ldots h_q} \sum_{i=1}^{n} \prod_{j=1}^{q} K_j \left(\frac{y_j - y_{ij}}{h_j} \right), \tag{16}$$

  where:

  – The bandwidth parameter h_j, corresponding to the j-th variable, plays a critical role in probability density estimation. A small h_j results in an under-smoothed density estimate, while a large h_j reduces the accuracy of the estimated density (over-smoothing). No single bandwidth selection method is universally optimal for all datasets [24]. This study adopts the bandwidth formula proposed by [30]:

$$h_j = \left(\frac{4}{n(d+2)} \right)^{\frac{1}{d+4}} \times \sigma_j, \tag{17}$$

  where $\sigma_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}{n-1}}$ is the sample standard deviation of the j-th variable.
  The formula (17) is selected because it optimizes the mean integrated squared error (MISE) and is well-suited for multidimensional data, as it adjusts based on the number of observations n, the number of dimensions d, and the dispersion σ_j. Compared to rule-of-thumb or cross-validation methods, this formula is simpler, computationally efficient, and empirically effective [30].

  – $K_j(\cdot)$ is the kernel function for the j-th variable, typically a symmetric, non-negative function integrating to one. While numerous kernel functions and bandwidth selection methods exist for equation (16), research indicates no universally optimal combination for all datasets [24]. This study employs the Gaussian kernel function:

$$K(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2} \right),$$

  as proposed by [30], with bandwidth parameters selected based on established practices [24, 34, 35].

- Step 5 aims to find the prior probability for y_0. Without information, the prior probability is often chosen according to (i) the uniform distribution ($q_1 = q_2 = ... = q_k = 1/k$). If based on the train set, the prior probability is usually chosen according to (ii) the contribution ratio of the elements of each population $q_i = n_i/N$ or (iii) by the Laplace approach: $q_i = (n_i + n/k)/(N + n)$, where n_i is the number of elements in w_i, n is the number of dimensions, k is the number of populations, and N is the number of all objects in training data. We find that the approach of finding the prior probability by (i) is based on the performer's subjectivity. In contrast, approaches (ii) and (iii) are only based on the elements of the training set without considering the relationship between the classified element y_0 to two populations w_1 and w_2. The approach for finding the prior probability in Step 5 of the proposed algorithm is based on the fuzzy relationship between y_0 and the groups, using cluster analysis algorithm [29]. Therefore, it is considered more suitable than other approaches.

3.3 The convergence of the proposed algorithm.

The proposed algorithm will converge when Step 5 converges. In this section, we will prove that Step 5 of the proposed algorithm converges with the WMPDC stopping criterion after finitely many iterations.

$$J_m(U,V) = \sum_{i=1}^{k}\sum_{j=0}^{n}(\mu_{ij})^m d(v_i,y_j)^2, \tag{18}$$

subject to (10). Here, $d(v_i, y_j)$ is the Euclidean distance, μ_{ij} is the membership probability of data point y_j to cluster i, v_i is the prototype of cluster i, k is the number of clusters, $n+1$ is the number of data points (including a new point y_0), and $m = 2$ is the fuzziness parameter.

The Weighted Mean Prototype Distance Change (WMPDC) stopping criterion is defined as:

$$\text{WMPDC}_m = \frac{1}{n+1}\sum_{j=0}^{n}\sum_{i=1}^{k}\mu_{ij}^{(m)} \cdot \left| d(v_i^{(m)}, y_j) - d(v_i^{(m-1)}, y_j) \right|. \tag{19}$$

The algorithm iterates until $\text{WMPDC}_m < \gamma$, where $\gamma = 0.001$.

Theorem 3.1. *The prior probability estimation step in the proposed algorithm with the WMPDC stopping criterion converges to a critical point of J_m, and $\text{WMPDC}_m \to 0$ indicates this convergence.*

Proof. To prove that the prior probability estimation step in the proposed algorithm converges to a critical point of J_m with $\text{WMPDC}_m \to 0$ as an indicator, we begin by examining the objective function $J_m(U, V)$. Since $d(v_i, y_j)^2 \geq 0$, $(\mu_{ij})^m \geq 0$, and the constraints (10) ensure a finite solution space, $J_m \geq 0$ and is bounded below. Each iteration updates the partition matrix $U^{(m)} = [\mu_{ij}^{(m)}]_{k\times(n+1)}$ and prototypes $V^{(m)} = \{v_i^{(m)}\}_{i=1}^k$ following (12) and (13), which minimize J_m for fixed $V^{(m-1)}$ and $U^{(m)}$, respectively. Consequently, $J_m(U^{(m)}, V^{(m)}) \leq J_m(U^{(m-1)}, V^{(m-1)})$, so the sequence $\{J_m(U^{(m)}, V^{(m)})\}$ is non-increasing and converges to a finite limit L.

Given the convergence of $\{J_m(U^{(m)}, V^{(m)})\}$ to L and the finiteness of the data points $\{y_j\}_{j=0}^n$, the sequence $\{V^{(m)}\}$ is bounded in a compact set. The alternating minimization ensures that the differences $d(v_i^{(m)}, v_i^{(m-1)})$ diminish, leading $\{V^{(m)}\}$ to converge to some $V^* = \{v_i^*\}_{i=1}^k$. Next, we consider the WMPDC criterion defined in (19). By the triangle inequality and continuity of the Euclidean distance, we have:

$$\left| d(v_i^{(m)}, y_j) - d(v_i^{(m-1)}, y_j) \right| \leq d(v_i^{(m)}, v_i^{(m-1)}). \tag{20}$$

As $V^{(m)} \to V^*$, $d(v_i^{(m)}, v_i^{(m-1)}) \to 0$, implying that $\left| d(v_i^{(m)}, y_j) - d(v_i^{(m-1)}, y_j) \right| \to 0$ for all $i = 1, \ldots, k$; $j = 0, \ldots, n$.

Since $\mu_{ij}^{(m)} \in [0, 1]$ and $\sum_{i=1}^k \mu_{ij}^{(m)} = 1$, the WMPDC term is bounded:

$$\text{WMPDC}_m \leq \frac{1}{n+1}\sum_{j=0}^{n}\sum_{i=1}^{k}\mu_{ij}^{(m)} \cdot d(v_i^{(m)}, v_i^{(m-1)}) = \frac{1}{n+1}\sum_{j=0}^{n}\sum_{i=1}^{k} d(v_i^{(m)}, v_i^{(m-1)}).$$

With $\sum_{i=1}^k \mu_{ij}^{(m)} = 1$ and $d(v_i^{(m)}, v_i^{(m-1)}) \to 0$, it follows that $\text{WMPDC}_m \to 0$, confirming that $\text{WMPDC}_m \to 0$ as the prototypes stabilize at V^*.

From the convergence of $V^{(m)} \to V^*$, the update rule in (12) yields:

$$\mu_{ij}^{(m)} \to \mu_{ij}^* = \frac{1}{\sum_{l=1}^{k} \left(\frac{d(v_i^*, y_j)}{d(v_l^*, y_j)} \right)^2}. \tag{21}$$

Similarly, the update rule in (13) ensures $v_i^{(m)} \to v_i^*$, satisfying:

$$v_i^* = \frac{\sum_{j=0}^{n} (\mu_{ij}^*)^2 y_j}{\sum_{j=0}^{n} (\mu_{ij}^*)^2}. \tag{22}$$

These equations, (21) and (22), represent the first-order necessary conditions for a critical point of J_m, derived from the Lagrangian:

$$L(U, V, \lambda) = J_m(U, V) + \sum_{j=0}^{n} \lambda_j \left(1 - \sum_{i=1}^{k} \mu_{ij} \right).$$

Thus, (U^*, V^*) is a critical point of J_m, which may be a local minimum or a saddle point.

Finally, we address singular cases where $d(v_l^{(m-1)}, y_j) = 0$ for some l. In such cases, we set $\mu_{lj}^{(m)} = 1$ and $\mu_{ij}^{(m)} = 0$ for $i \neq l$ in (12), ensuring continuity of the update rules. The constraint $0 < \sum_{j=0}^{n} \mu_{ij} < n+1$ prevents empty clusters, and these singular cases do not affect convergence. Therefore, the sequence $\{J_m(U^{(m)}, V^{(m)})\}$ converges to a limit L, with $V^{(m)} \to V^*$ implying $\text{WMPDC}_m \to 0$, which detects prototype stability. The resulting (U^*, V^*) satisfies the critical point conditions of J_m, proving that the FCM algorithm with the WMPDC criterion converges to a critical point, and $\text{WMPDC}_m \to 0$ effectively indicates this convergence. □

4 Experimental validation

4.1 Data and approach

This section applies the proposed classification approach to two datasets.

* Data 1: The first dataset pertains to wines from the Piedmont region of Italy. Data includes three groups: Barolo (W_1), Barbera (W_2), and Grignolino (W_3). The dataset comprises 178 observations, each with 14 variables. It is freely available on the website: https://www.kaggle.com/datasets/tawfikelmetwally/wine-dataset. Table 1 provides a brief description of the variables.

Table 1: The brief description of variables for Wine dataset

x_i	Variable name	Description
x_1	Alcohol	Alcohol concentration in wine
x_2	Malic acid	Malic acid content
x_3	Ash	Ash content
x_4	Alkalinity of ash	Alkalinity of ash
x_5	Magnesium	Magnesium content
x_6	Total phenols	Total phenol content
x_7	Flavanoids	Flavonoid content
x_8	Nonflavanoid phenols	Non-flavonoid phenol content
x_9	Proanthocyanins	Proanthocyanidin content
x_{10}	Color intensity	Color intensity
x_{11}	Hue	Hue
x_{12}	OD280/OD315 of diluted wines	Absorbance ratio at 280nm and 315nm of diluted wine
x_{13}	Proline	Proline content
x_{14}	Class	Classification variable

For this dataset, 80% (142 sample elements) are randomly selected as the training set, and the remaining 20% (36 sample elements) are used as the test set.

* Data 2: The second dataset is the Breast Cancer Wisconsin dataset, a widely used dataset in machine learning and statistics for classifying breast tumours as benign or malignant. Breast cancer is the most common cancer

among women, accounting for approximately 30% of all cancer cases worldwide. This dataset was collected by Dr. Wolberg at the University of Wisconsin Hospitals between 1989 and 1991. It is freely available for download from https://www.kaggle.com/datasets/abdelazizsami/breast-cancer-wisconsin-diagnostic. The dataset includes 569 breast cell samples, each described by 30 variables, as detailed in Table 2. To evaluate the proposed algorithm's effectiveness compared to other approaches, we divided the data into a 70% training set and a 30% testing set.

Table 2: The brief description of variables for Breast Cancer Wisconsin (Diagnostic) dataset

x_i	Variable name	Description
x_1	ID	Patient identifier
x_2	Diagnosis	Diagnosis (M = malignant, B = benign)
x_3	Radius_mean	Mean radius of the nucleus
x_4	Texture_mean	Mean texture of the nucleus
x_5	Perimeter_mean	Mean perimeter of the nucleus
x_6	Area_mean	Mean area of the nucleus
x_7	Smoothness_mean	Mean smoothness of the nucleus
x_8	Compactness_mean	Mean compactness of the nucleus
x_9	Concavity_mean	Mean concavity of the nucleus
x_{10}	Concave_points_mean	Mean number of concave points of the nucleus
x_{11}	Symmetry_mean	Mean symmetry of the nucleus
x_{12}	Fractal_dimension_mean	Mean fractal dimension of the nucleus
x_{13}	Radius_se	Radius standard error
x_{14}	Texture_se	Texture standard error
x_{15}	Perimeter_se	Perimeter standard error
x_{16}	Area_se	Area standard error
x_{17}	Smoothness_se	Smoothness standard error
x_{18}	Compactness_se	Compactness standard error
x_{19}	Concavity_se	Concavity standard error
x_{20}	Concave_points_se	Standard error of concave points
x_{21}	Symmetry_se	Symmetry standard error
x_{22}	Fractal_dimension_se	Fractal dimension standard error
x_{23}	Radius_worst	Worst radius of the nucleus
x_{24}	Texture_worst	Worst texture of the nucleus
x_{25}	Perimeter_worst	Worst perimeter of the nucleus
x_{26}	Area_worst	Worst area of the nucleus
x_{27}	Smoothness_worst	Worst smoothness of the nucleus
x_{28}	Compactness_worst	Worst compactness of the nucleus
x_{29}	Concavity_worst	Worst concavity of the nucleus
x_{30}	Concave_points_worst	Worst number of concave points of the nucleus
x_{31}	Symmetry_worst	Worst symmetry of the nucleus
x_{32}	Fractal_dimension_worst	Worst fractal dimension of the nucleus

For the first dataset, we provide a detailed implementation to illustrate the steps of the proposed approach. In contrast, the second dataset compares the implementation results with other techniques to show their effectiveness. The approaches compared include both statistical and machine learning algorithms such as Naive Bayes, Support Vector Machine Radial Basis Function (SVM-RBF), Support Vector Machine with Linear Kernel (SVM-Linear), Random Forest, k-nearest Neighbors (k-NN), Decision Tree (DT), Artificial Neural Network (ANN), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGBoost), and Linear Discriminant Analysis (LDA).

4.2 Results

a) *For Data 1:*

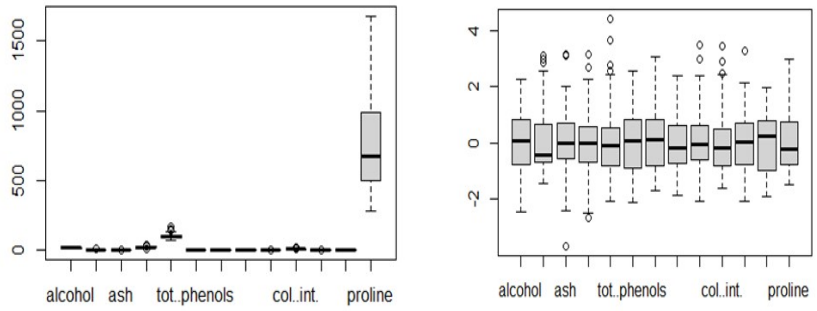
- **Step 1:** Normalizing the data using the proposed approach, we obtain a new dataset for the training and testing sets, which are respectively represented by Figures 4.2.

  Figure 4.2 illustrates a substantial discrepancy in the scale of the variables, which introduces bias when jointly assessing element similarity. Utilizing raw variables under these conditions may yield suboptimal classification

Table 3: Matrix of eigenvector columns

Variable	y_1	y_2	...	y_{12}	y_{13}
Alcohol	-0.1664	-0.4501	...	-0.1864	0.0937
Malic.acid	0.2481	-0.2476	...	0.0624	-0.0022
Ash	-0.0185	-0.3023	...	-0.0781	-0.0805
Alcalinity.of.ash	0.2293	0.0175	...	-0.0532	0.0808
Magnesium	-0.1625	-0.3303	...	0.0966	0.0335
Total.phenols	-0.3929	-0.0541	...	-0.3629	-0.4762
Flavanoids	-0.4319	0.0332	...	0.0039	0.8201
Nonflavanoid.phenols	0.3052	-0.0208	...	0.1030	0.0882
Proanthocyanins	-0.2883	-0.0304	...	-0.1153	-0.1063
Color.intensity	0.0870	-0.5349	...	0.5615	-0.0501
Hue	-0.2944	0.2782	...	0.2497	-0.0956
OD280.OD315.of.diluted.wines	-0.3686	0.1749	...	0.6308	-0.2154
Proline	-0.2899	-0.3685	...	-0.1090	-0.0272

outcomes. In contrast, Figure 4.2 demonstrates that the variables are standardized mainly in scale, ensuring equitable contributions to the classification process.



(a) Boxplot for the original data (b) Boxplot for the standardized data

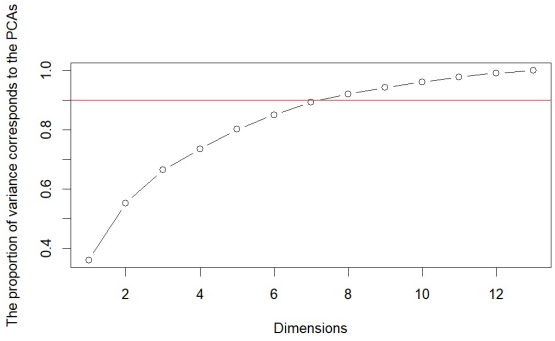
- **Step 2:** Perform PCA on the training data and find the orthogonal matrix P, the matrix of eigenvectors corresponding to the eigenvalues. The results of the PCA on the training data, including the matrix of eigenvectors and the principal component analysis summary, are presented in Tables 3 and 4.

Table 4: Descriptive analysis of principal component information

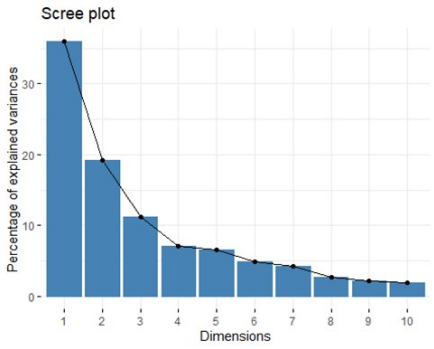
y_i	Standard deviation	Proportion of variance	Cumulative proportion
y_1	2.2017	0.3707	0.3707
y_2	1.6074	0.1976	0.5683
y_3	1.2192	0.1137	0.6819
y_4	0.9159	0.0642	0.7461
y_5	0.8532	0.0557	0.8017
y_6	0.8149	0.0508	0.8525
y_7	0.7263	0.0404	0.8928
y_8	0.5996	0.0275	0.9203
y_9	0.5484	0.0230	0.9433
y_{10}	0.5066	0.0196	0.9630
y_{11}	0.4652	0.0166	0.9795
y_{12}	0.4165	0.0133	0.9928
y_{13}	0.3077	0.0072	1.0000

- **Step 3:** Select eight principal components to retain for the training data, corresponding to 90% of the total variance of the original variables. Compute the principal component matrix for the test set, ensuring that the

same number of principal components is retained as in the training set. The results are illustrated in Figure 2.



(c) Cumulative variance ratio



(d) Proportion of variance

Figure 2: Explanation of the number of PCs retained corresponding to the percentage of variance of the original variables.

- **Step 4:** Construct the PDFs for each group using the kernel approach based on the eight new variables, i.e., the principal components (PCs) retained in the training set. After this step, we obtain three 8-dimensional PDFs, f_1, f_2, f_3, corresponding to each wine group: Grignolino, Barolo and Barbera.
- **Step 5:** The prior probabilities of each element in the test set are determined using the fuzzy clustering approach outlined in Step 5 of the proposed algorithm. This approach requires approximately 6 iterations to converge, substantially reducing computational complexity compared to traditional fuzzy clustering approaches. Upon convergence, the resulting prior probabilities assign each element in the test set to a specific wine category. Detailed results are presented in Figure 3.



Figure 3: The prior probability for the elements of the test set.

- **Step 6:** Compute the posterior probabilities f_{C1}, f_{C2}, f_{C3} for the elements of the test set using Equation (14) in Step 6 of the proposed algorithm.

  Suppose we need to classify a wine sample from the test set with the variable:

$$y_0 = (-2.187, 0.349, -1.988, -0.378, 0.056, -0.992, 0.029, -1.059)\,.$$

  Then, we have

$$f_{C1}(y_0) = u_1 f_1(y_0) = 8.358802 \times 10^{-6},$$
$$f_{C2}(y_0) = u_2 f_2(y_0) = 4.794124 \times 10^{-7},$$

$$f_{C3}(y_0) = u_3 f_3(y_0) = 3.074575 \times 10^{-24}.$$

Since $\max\{f_{C1}(y_0);\ f_{C2}(y_0);\ f_{C3}(y_0)\} = 8.358802 \times 10^{-6}$, the observation y_0 is assigned to the Grignolino class, yielding a correct classification outcome.

Continuing with the classification for the remaining 35 samples, we have Figure 4.

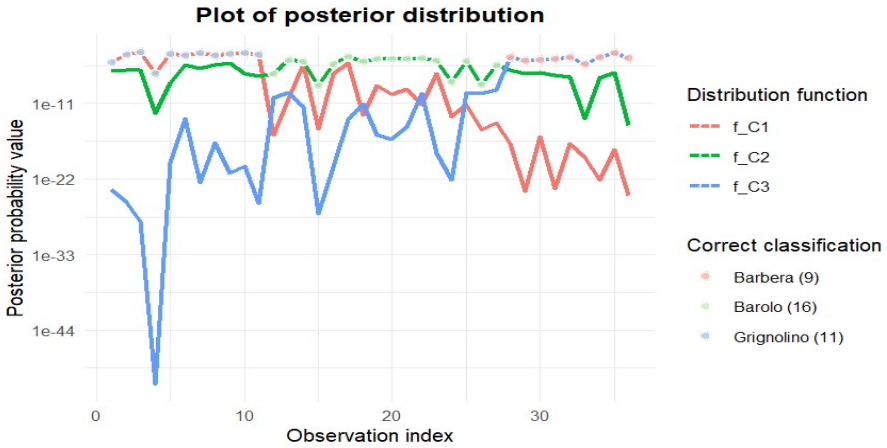


Figure 4: The posterior distribution for the elements in the test set.

From Figure 4 we find that all are correctly classified. Thus, the accuracy of the proposed approach for this dataset is 100%.

Compared with other classification algorithms, the results are provided in Table 5.

Table 5: The performance metrics (%) of approaches for Data 1

Approaches	ACC	F1-score	Precision	Recall	Time cost (s)
Naive Bayes	97	97	98	97	0.68
SVM-RBF	92	91	95	90	1.83
SVM-Linear	97	97	98	96	0.95
Random Forest	97	97	97	97	1.11
k-NN	72	72	80	79	0.42
Decision Tree	86	86	86	88	0.50
ANN	94	95	95	96	2.16
GBM	97	97	97	98	2.17
XGBoost	94	95	94	96	54.27
LDA	94	95	94	96	0.37
Proposed approach	100	100	100	100	3.30

Table 5 presents the performance metrics, including ACC, F1-score, Precision, and Recall for Daa 1 of algorithms. The proposed approach achieves the highest scores across all metrics, while methods such as Naive Bayes and SVM-Linear exhibit closely competitive performance. Because the proposed algorithm involves extra steps compared to traditional approaches, its time cost is indeed higher. However, these measurements were recorded on a personal laptop, and the time cost could be significantly reduced on more powerful machines, making the approach entirely feasible in practice.

b) *For Data 2:*

For the second dataset, Breast Cancer Wisconsin, after applying PCA to the training set to retain ten principal components $(y_1, \ldots, y_{10})$, we perform the proposed algorithm with seven iterations. The resulting performance metrics for the test set are presented in Table 6.

Table 6 indicates that the proposed algorithm achieves high performance across ACC, F1-score, Precision, and Recall for Data 2; therefore, it is considered the most effective in classifying this dataset. In this dataset, the time cost of the proposed algorithm is not the smallest; however, as previously explained, it is still entirely feasible for practical application in the current era of information technology.

Table 6: The performance metrics (%) of approaches for Data 2

Approaches	ACC	F1-score	Precision	Recall	Time cost (s)
Naive Bayes	94	94	93	96	0.86
SVM-RBF	74	74	69	89	7.73
Random Forest	94	95	92	97	1.07
k-NN	92	94	90	96	0.45
Decision Tree	91	92	90	95	0.53
ANN	94	95	97	93	3.97
GBM	94	95	91	93	2.63
XGBoost	94	95	94	95	38.36
LDA	94	95	92	96	0.36
Proposed approach	95	96	95	97	40.0

5 Applying to image data

5.1 Extracting of images

For each image, the Gray-Level Co-occurrence Matrix (GLCM) provides information about the intensity of pixels and their neighbours with fixed parameters for distance and angle, denoted as d and θ, respectively. An in-depth analysis of the image captures information about the shading present between two pixels, allowing for more accurate analysis and interpretation of differences between images. Given an image $f(x, y)$ of dimensions $Nx \times Ny$ (Nx pixels along the X-axis and Ny pixels along the Y-axis), and assuming there are G gray levels, the GLCM is defined as a matrix P of size $G \times G$. Each entry $p(i, j)$ indicates the probability that intensity i co-occurs with intensity j at a specific distance d and orientation θ. The value $p_{d,\theta}(i, j)$ for a given d and θ is calculated according to the following expression:

$$p_{d,\theta}(i,j) = \#\left\{(x,y),(x',y') \in N_x \times N_y | d = \|(x,y),(x',y')\|, \theta = \Theta((x,y),(x',y'), f(x,y) = i, f(x',y')) = j\right\},$$

where

# denotes the number of elements in the set as a sample,

$x = 1, 2, ..., N_x$ and $y = 1, 2, ..., N_y$ are the coordinates of the pixel elements of an image with size $N_x \times N_y$,

$i, j = 0, 1, ..., L - 1$ are the gray level values with a grayscale range of L,

$I(x, y)$ is the value of GLCM at coordinates (x, y),

d_x and d_y are position displacements calculated by the distance d and angle θ between pixel matrix pairs.

After extracting the GLCM for an image, we can compute representative features. Haralick [11] identified 14 texture features that can be extracted from each GLCM. However, many studies have shown that 3 to 6 features, such as contrast, homogeneity, correlation, and entropy, have good discriminatory power between images. Therefore, this study uses four image features when extracting its GLCM matrix. The image features are presented in Table 7.

Table 7: Four variables representing an image derived from GLCM

Feature	Expression	Description
Contrast	$\sum_{i,j} p_{i,j} (i-j)^2$	Measures the intensity difference between neighboring pixels.
Homogeneity	$\sum_{i,j} \frac{p_{i,j}}{1+(i-j)^2}$	Measures the closeness of the distribution of elements in the GLCM to the diagonal
Entropy	$\sum_{i,j} p_{i,j} \log(p_{i,j})$	Measures the uniformity of the texture.
Correlation	$\sum_{i,j} p_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}}\right]$	Measures how correlated a pixel is with its neighbors.

In which $\mu_i, \mu_j, \sigma_i, \sigma_j$ are respectively the mean values and standard deviations of the sum of rows and columns in the GLCM.

The use of four variables from Table 7 with the orientation angle $\theta = 0°$ for image recognition has been applied in many studies, such as Vo-Van et al. [33, 35] and Huynh-Van et al. [15]. However, the combination of optimal features extracted from multiple angles has not yet received much attention. In this study, an image is extracted with features at

four different angles $(0°, 45°, 90°, 135°)$. For each angle, four variables (Contrast, Homogeneity, Entropy, Correlation), as presented in Table 7, are established. Thus, an image is represented by $4 \times 4 = 16$ variables in this study.

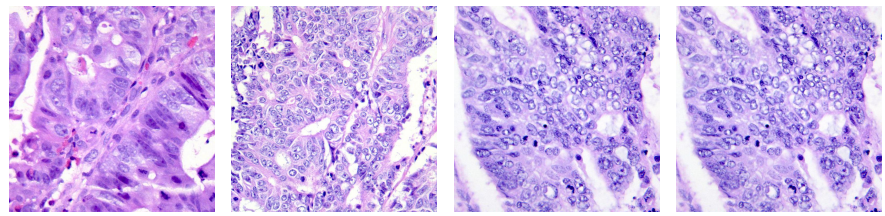
5.1.1 Data and approach

This section examines two specific image datasets for applying the proposed approach.

*** Data 1:** The image dataset under consideration is named `colon_image_data` (CID), which is freely available from the website www.kaggle.com/datasets with the following specific information:

- **Colon_aca**: images of tissue samples from patients with colorectal cancer (adenocarcinoma). These images may contain specific characteristics of cancer cells, such as abnormal cell division, altered cell structure, and transformed tissue.
- **Colon_n**: images of tissue samples from normal colorectal tissues (non-cancerous). These images display healthy tissue structures without cancerous alterations.

The dataset consists of 2,000 images, with 1,000 images in each group. Eight images for two groups are shown in Figure 5.



(a) colon_aca (b) colon_aca (c) colon_aca (d) colon_aca

(e) colon_n (f) colon_n (g) colon_n (h) colon_n

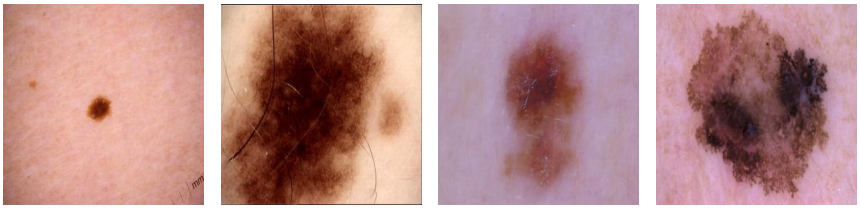
Figure 5: Some sample images of CID data

For this dataset, we randomly selected 1,600 images (800 images from each group) for the training set and 400 images (200 images from each group) for the test set.

*** Data 2:** The Melanoma Skin Cancer (MSC) dataset contains 10,000 images, which were downloaded for free from the page https://www.kaggle.com/datasets/ hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images. MSC is a deadly form of cancer, and early detection and treatment can save many lives. This dataset will help develop deep-learning models to classify melanoma accurately. The dataset is illustrated with the following information:

- **benign**: These are images of moles or skin lesions that are not cancerous. These moles or lesions do not pose a risk of developing into cancer and do not harm the patient's health.
- **malignant**: These are images of moles or skin lesions that have developed into melanoma cancer or are at high risk of developing cancer. These moles or lesions may indicate skin cancer and require timely medical intervention for treatment.

For this dataset, we randomly selected 400 images (200 images from each group) for the training set and 100 images (50 images from each group) independent from the training images for the test set. Some sample images for the two groups are shown in Figure 6.



(a) benign (b) benign (c) malignant (d) malignant

Figure 6: Some sample images of MSC data

Table 8: The ACC (%) of approaches with CID and MSC data sets

Approach	CID data	MSC data
RBFSVM	50	50
LiSVM	50	41
k-NN	43	81
RandomForest	50	57
Naive.Bayes	45.5	65
LDA	52	49
Logistic Regression	51	41
Decision Tree	47	51
ANN	49	52
GBM	50	48
XGBoost	50	59
Proposed approach	85.75	85

5.1.2 The result

For Data 1 and Data 2, we perform 10 and 9 iterations, respectively to estimate the prior probabilities. The ACC (%) of the proposed algorithm and others are presented in Table 8.

Tables 8 shows that the proposed algorithm provides consistently stable and significantly higher ACC values compared to other algorithms for both datasets.

6 Conclusion

This study has developed a classification algorithm based on a novel idea combining Principal Component Analysis (PCA) and an improved Bayesian approach. PCA is used to reduce data dimensionality, after which Fuzzy C-Means, based on fuzzy logic, estimates membership degrees to improve the determination of prior probabilities. On the reduced-dimensional data space, KDE enables more stable density estimation, thereby enhancing the accuracy of posterior probabilities under the Bayesian approach. The issues related to the PCA approach used in the proposed algorithm have been introduced, providing a clear and detailed presentation of the proposed algorithm. The proposed approach can quickly handle practical datasets with a RStudio program setup. Through applications on real-world datasets, the proposed algorithm reduces input variables for classification tasks and significantly improves performance. The proposed algorithm can compete with many popular classification approaches today, including statistical and machine-learning algorithms. A significant contribution of this study is applying the proposed algorithm to image data, where its features are extracted into 64 variables. When applied to specific image datasets, the proposed algorithm consistently demonstrated stable and effective classification results, further validating its reliability compared to other algorithms.

In this study, image classification was based solely on gray-level features. However, the proposed model's effectiveness could be further enhanced by extracting features from different aspects of the images. This promising avenue for future research will be a key focus in image classification.

Data availability statement

The datasets analyzed during this study are available from the public data on the website www.kaggle.com/datasets.

Conflict of interest

The authors have declared no conflicts of interest for this article.

References

[1] H. A. Alfeilat, V. B. Surya Prasath, A. B. A. Hassanat, O. Lasassmeh, A. Abu Alfeilat, A. S. Tarawneh, M. Bashir, *Distance and similarity measures effect on the performance of k-nearest neighbor classifier: A review*, Big Data, **7**(2) (2019), 123-144. https://doi.org/10.1089/big.2018.0175

[2] M. Awad, R. Khanna, *Support vector machines for classification*, in Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers (Springer, 2015), 39-66. https://doi.org/10.1007/978-1-4302-5990-9_3

[3] D. K. Behera, M. Das, S. Swetanisha, J. Nayak, S. Vimal, B. Naik, *Follower link prediction using the XGBoost classification model with multiple graph features*, Wireless Personal Communications, **127** (2022), 695-714. https://doi.org/10.1007/s11277-021-08399-y

[4] I. D. Borlea, R. E. Precup, F. Drăgan, A. B. Borlea, *Centroid update approach to k-means clustering*, Advances in Electrical and Computer Engineering, **17** (2017), 3-10. https://doi.org/10.4316/AECE.2017.04001

[5] R. B. Cattell, *The scree test for the number of factors*, Multivariate Behavioral Research, **1** (1966), 245-276. https://doi.org/10.1207/s15327906mbr0102_10

[6] Y. P. Chen, C. H. Liu, K. Y. Chou, S. Y. Wang, *Real-time and low-memory multi-face detection system design based on naive Bayes classifier using FPGA*, 2016 International Automatic Control Conference (CACS), IEEE, (2016), 7-12. https://doi.org/10.1109/CACS.2016.7973875

[7] H. S. Chiang, D. H. Shih, B. Lin, M. H. Shih, *An APN model for arrhythmic beat classification*, Bioinformatics, **30** (2014), 1739-1746. https://doi.org/10.1093/bioinformatics/btu101

[8] P. Domingos, M. Pazzani, *On the optimality of the simple Bayesian classifier under zero-one loss*, Machine Learning, **29**(2) (1997), 103-130. https://doi.org/10.1023/A:1007413511361

[9] J. C. Dunn, *A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters*, Journal of Cybernetics, **3** (1973), 32-57. https://doi.org/10.1080/01969727308546046

[10] Z. Feng, G. Huang, D. Chi, *Classification of the complex agricultural planting structure with a semi-supervised extreme learning machine framework*, Remote Sensing, **12**(22) (2020), 3708. https://doi.org/10.3390/rs12223708

[11] R. A. Fisher, *The statistical utilization of multiple measurements*, Annals of Eugenics, **8** (1938), 376-386. https://doi.org/10.1111/j.1469-1809.1938.tb02189.x

[12] T. Hastie, R. Tibshirani, J. H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed., Springer, 2009. https://doi.org/10.1007/978-0-387-84858-7

[13] D. W. Hosmer Jr., S. Lemeshow, R. X. Sturdivant, *Applied logistic regression*, 3rd ed., Wiley, 2013. https://doi.org/10.1002/9781118548387

[14] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, W. Xu, *Applications of support vector machine (SVM) learning in cancer genomics*, Cancer Genomics and Proteomics, **15** (2018), 41-51. https://doi.org/10.21873/cgp.20063

[15] H. Huynh-Van, T. Le-Hoang, T. Thai-Minh, H. Nguyen-Dinh, T. Vo-Van, *Classifying the lung images for people infected with COVID-19 based on the extracted feature interval*, Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, **11** (2023), 856-865. https://doi.org/10.1080/21681163.2022.2117645

[16] H. Huynh-Van, T. Le-Hoang, T. Vo-Van, *Classifying for images based on the extracted probability density function and the quasi Bayesian method*, Computational Statistics, **39** (2024), 2677-2701. https://doi.org/10.1007/s00180-023-01400-1

[17] I. T. Jolliffe, *Principal component analysis*, 2nd ed., Springer, 2002. https://doi.org/10.1007/b98835

[18] H. F. Kaiser, *The application of electronic computers to factor analysis*, Educational and Psychological Measurement, **20** (1960), 141-151. https://doi.org/10.1177/001316446002000116

[19] D. Lazo, R. Calabrese, C. Bravo, *The effects of customer segmentation, borrower behaviors and analytical methods on the performance of credit scoring models in the agribusiness sector*, The Journal of Credit Risk, **16**(4) (2021), 119-156. https://doi.org/10.21314/JCR.2020.272

[20] F. Louzada, A. Ara, G. B. Fernandes, *Classification methods applied to credit scoring: Systematic review and overall comparison*, Surveys in Operations Research and Management Science, **21** (2016), 117-134. https://doi.org/10.1016/j.sorms.2016.10.001

[21] E. G. Mansoori, M. J. Zolghadri, S. D. Katebi, *Using distribution of data to enhance performance of fuzzy classification systems*, Iranian Journal of Fuzzy Systems, **4**(1) (2007), 21-36. https://doi.org/10.22111/ijfs.2007.355

[22] M. Mirbabaie, S. Stieglitz, N. R. J. Frick, *Artificial intelligence in disease diagnostics: A critical review and classification on the current state of research guiding future direction*, Health and Technology, **11** (2021), 1-23. https://doi.org/10.1007/s12553-021-00555-5

[23] J. G. Neto, L. V. Ozorio, T. C. C. de Abreu, B. F. Dos Santos, F. Pradelle, *Modeling of biogas production from food, fruits and vegetables wastes using artificial neural network (ANN)*, Fuel, **285** (2021), 119081. https://doi.org/10.1016/j.fuel.2020.119081

[24] T. Nguyen-Trang, T. Vo-Van, *A new approach for determining the prior probabilities in the classification problem by Bayesian method*, Advances in Data Analysis and Classification, **11** (2017), 629-643. https://doi.org/10.1007/s11634-016-0253-y

[25] V. H. Nhu, D. Zandi, H. Shahabi, K. Chapi, A. Shirzadi, N. Al-Ansari, S. K. Singh, J. Dou, H. Nguyen, *Comparison of support vector machine, bayesian logistic regression, and alternating decision tree algorithms for shallow landslide susceptibility mapping along a mountainous road in the west of Iran*, Applied Sciences, **10**(15) (2020), 5047. https://doi.org/10.3390/app10155047

[26] J. Ning, *Neural network-based pattern recognition in the framework of edge computing*, Romanian Journal of Information Science and Technology, **27**(1) (2024), 106-119. https://doi.org/10.59277/ROMJIST.2024.1.08

[27] T. Pham-Gia, N. Turkkan, A. Bekker, *Bounds for the Bayes error in classification: A Bayesian approach using discriminant analysis*, Statistical Methods and Applications, **16** (2007), 7-26. https://doi.org/10.1007/s10260-006-0012-x

[28] T. Pham-Gia, N. Turkkan, T. Vovan, *Statistical discrimination analysis using the maximum function*, Communications in Statistics–Simulation and Computation, **37**(2) (2008), 320-336. https://doi.org/10.1080/03610910701790475

[29] D. Pham-Toan, T. Vo-Van, *Improving the genetic algorithm in fuzzy cluster analysis for numerical data and its applications*, Iranian Journal of Fuzzy Systems, **20**(5) (2023), 171-187. https://doi.org/10.22111/ijfs.2023.7834

[30] G. R. Terrell, D. W. Scott, *Variable kernel density estimation*, The Annals of Statistics, (1992), 1236-1265. http://www.jstor.org/stable/2242011

[31] T. Vo-Van, *L1-distance and classification problem by Bayesian method*, Journal of Applied Statistics, **44**(3) (2017), 385-401. https://doi.org/10.1080/02664763.2016.1174194

[32] T. Vo-Van, *Some results of classification problem by Bayesian method and application in credit operation*, Statistical Theory and Related Fields, **2**(2) (2018), 150-157. https://doi.org/10.1080/24754269.2018.1528420

[33] T. Vo-Van, *An improved fuzzy time series forecasting model using variations of data*, Fuzzy Optimization and Decision Making, **18** (2019), 151-173. https://doi.org/10.1007/s10700-018-9290-7

[34] T. Vo-Van, H. Che-Ngoc, N. Le-Dai, T. Nguyen-Trang, *A new strategy for short-term stock investment using Bayesian approach*, Computational Economics, **59**(2) (2022), 887-911. https://doi.org/10.1007/s10614-021-10115-8

[35] T. Vo-Van, H. Che-Ngoc, T. Nguyen-Trang, *Textural features selection for image classification by Bayesian method*, 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), (2017), 733-139. https://doi.org/10.1109/FSKD.2017.8393365

[36] T. Vo-Van, T. Pham-Gia, *Clustering probability distributions*, Journal of Applied Statistics, **37**(11) (2010), 1891-1910. https://doi.org/10.1080/02664760903186049

[37] J. Wagner, T. O. J. Klippel, *Using machine learning techniques for crop classification in agriculture*, Journal of Precision Agriculture, **19**(4) (2018), 200-215. https://doi.org/10.1007/s11119-017-9524-9

[38] Y. Wu, X. Jiang, J. Kim, L. Ohno-Machado, *Grid binary logistic regression (GLORE): Building shared models without sharing data*, Journal of the American Medical Informatics Association, **19**(5) (2012), 758-764. https://doi.org/10.1136/amiajnl-2012-000862

[39] A. J. Wyner, M. Olson, J. Bleich, D. Mease, *Explaining the success of AdaBoost and random forests as interpolating classifiers*, Journal of Machine Learning Research, **18**(48) (2017), 1-33. http://jmlr.org/papers/v18/15-240.html