

Adaptive machine learning approaches utilizing soft decision-making *via* intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices

Samet Memiş¹, Ferhan Şola Erduran² and Hivda Aydoğan³

¹ Department of Marine Engineering, Faculty of Maritime, Bandırma Onyedi Eylül University, Balıkesir, Türkiye

² Department of Mathematics, Faculty of Science, Gazi University, Ankara, Türkiye

³ Department of Mathematics, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Türkiye

ABSTRACT

The exponential data growth generated by technological advancements presents significant challenges in analysis and decision-making, necessitating innovative and robust methodologies. Machine learning has emerged as a transformative tool to address these challenges, especially in scenarios requiring precision and adaptability. This study introduces two novel adaptive machine learning approaches, *i.e.*, AIFPIFSC1 and AIFPIFSC2. These methods leverage the modeling ability of intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices (*ifpifs*-matrices). This state-of-the-art framework enhances the classification task in machine learning by employing soft decision-making through *ifpifs*-matrices. The proposed approaches are rigorously evaluated against leading fuzzy/soft-based classifiers using 15 widely recognized University of California, Irvine datasets, including accuracy and robustness, across six performance metrics. Statistical analyses conducted using Friedman and Nemenyi tests further substantiate the reliability and superiority of the proposed approaches. The results consistently demonstrate that these approaches outperform their counterparts, highlighting their potential for solving complex classification problems. This study contributes to the field by offering adaptable and effective solutions for modern data analysis challenges, paving the way for future advancements in machine learning and decision-making systems.

Submitted 7 October 2024
Accepted 23 January 2025
Published 28 February 2025

Corresponding author
Samet Memiş,
samettmemis@gmail.com

Academic editor
Dragan Pamucar

Additional Information and
Declarations can be found on
page 26

DOI 10.7717/peerj-cs.2703

© Copyright
2025 Memiş et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Algorithms and Analysis of Algorithms, Data Mining and Machine Learning, Optimization Theory and Computation

Keywords Intuitionistic fuzzy sets, Soft sets, *ifpifs*-matrices, Soft decision-making, Machine learning

INTRODUCTION

The amount of data made possible by technological advancement is constantly growing. This increasing amount of data may be analyzed and interpreted using machine learning, a technical improvement. Numerous industries frequently use this technology, including defense, finance, psychology, medicine, meteorology, astronomy, and space sciences. Such fields are encountered with many uncertainties. Several mathematical tools, such as fuzzy sets (*Zadeh, 1965*) and intuitionistic fuzzy sets (*Atanassov, 1986*), have been propounded

to model these uncertainties. Furthermore, modeling such uncertainties is a crucial process to enhance the performance of machine learning algorithms. To this end, many classical machine learning algorithms, such as k-nearest neighbor (kNN) ([Fix & Hodges, 1951](#); [Keller, Gray & Givens, 1985](#)), have been successfully modified as fuzzy kNN ([Keller, Gray & Givens, 1985](#)) using fuzzy sets.

Although fuzzy sets provide a mathematical framework for dealing with uncertainties where classical sets are inadequate, soft sets ([Molodtsov, 1999](#)) further this approach by offering additional flexibility in modeling uncertainties. In the last two decades, these concepts have evolved into various hybrid forms such as fuzzy soft sets ([Maji, Biswas & Roy, 2001a](#)), fuzzy parameterized soft sets ([Çağman, Çitak & Enginoğlu, 2011](#)), and fuzzy parameterized fuzzy soft sets (*fpfs*-sets) ([Çağman, Çitak & Enginoğlu, 2010](#)), which have manifested their utility in modeling scenarios where parameters or objects possess fuzzy values.

Developing fuzzy parameterized fuzzy soft matrices (*fpfs*-matrices) ([Enginoğlu & Çağman, 2020](#)) ensures a significant advancement in the field, particularly in computerizing decision-making scenarios involving a large amount of data. It has been applied to performance-based value assignment problems in image denoising by employing generalized fuzzy soft max-min decision-making method ([Enginoğlu, Memiş & Çağman, 2019](#)), and operability-configured soft decision-making methods ([Enginoğlu et al., 2021](#); [Enginoğlu & Öngel, 2020](#)) and classification problems in machine learning ([Memiş & Enginoğlu, 2019](#)). Moreover, classification algorithms based on soft decision-making methods constructed by *fpfs*-matrices have been suggested, namely FPFS-CMC ([Memiş, Enginoğlu & Erkan, 2022a](#)) and FPFS-AC ([Memiş, Enginoğlu & Erkan, 2022c](#)). However, *fpfs*-matrices lack modeling ability for intuitionistic fuzzy uncertainties. Therefore, the concepts of intuitionistic fuzzy soft sets (*ifs*-sets) ([Maji, Biswas & Roy, 2001b](#)), intuitionistic fuzzy parameterized soft sets (*ifps*-sets) ([Deli & Çağman, 2015](#)), and intuitionistic fuzzy parameterized fuzzy soft sets (*ifpfs*-sets) ([El-Yagubi & Salleh, 2013](#)) have been proposed. In addition, intuitionistic fuzzy parameterized intuitionistic fuzzy soft sets (*ifpifs*-sets) ([Karaaslan, 2016](#)) and matrices (*ifpifs*-matrices) ([Enginoğlu & Arslan, 2020](#)) have enabled the modeling of problems with both parameters and objects containing intuitionistic fuzzy uncertainties. Besides, various mathematical tools, such as bipolar soft sets ([Mahmood, 2020](#)), intuitionistic fuzzy mean operators ([Hussain et al., 2023](#)), and picture fuzzy soft matrices ([Memiş, 2023b](#)), have been proposed to deal with uncertainty and are still being studied. However, when the related literature is investigated, it is noteworthy that there are almost no applications, especially in real-world problems/data ([Bustince Sola et al., 2016](#); [Karakoç, Memiş & Sennaroglu, 2024](#)). Although several applications have recently been conducted to the classification problem in machine learning, which is a real problem modeled by *fpfs*-matrices, only a few studies employ *ifpifs*-matrices in machine learning ([Memiş et al., 2021](#)). Therefore, applying *ifpifs*-matrices, a pioneer mathematical tool to model uncertainty, to machine learning is a topic worthy of study.

Recently, [Memiş et al. \(2021\)](#) have defined metrics, quasi-metrics, semi-metrics, and pseudo-metrics, as well as similarities including quasi-, semi-, and pseudo-similarities over

ifpifs-matrices. Besides, [Memiş et al. \(2021\)](#) has proposed an Intuitionistic Fuzzy Parameterized Intuitionistic Fuzzy Soft Classifier (IFPIFSC), which utilizes six pseudo-similarities of *ifpifs*-matrices. This classifier has been simulated using 20 datasets from the University of California, Irvine Machine Learning Repository (UCI-MLR) ([Kelly, Longjohn & Nottingham, 2024](#)). Its performance has been evaluated using six metrics: accuracy (Acc), precision (Pre), recall (Rec), specificity (Spe), macro F-score (MacF), and micro F-score (MicF). The results have manifested that IFPIFSC has outperformed fuzzy and non-fuzzy-based classifiers in these metrics. Although machine learning algorithms have been suggested *via ifpifs*-matrices for real problems, machine learning datasets in UCI-MLR and other databases usually consist of real-valued raw data. In these datasets, no serious uncertainties, such as intuitionistic uncertainty, can be modeled with mathematical tools. However, our aim in this study is to propose a new machine learning algorithm that can work with a dataset that contains intuitionistic fuzzy uncertainties by first converting the raw data to fuzzy values and then to intuitionistic fuzzy values. On the other hand, the most significant disadvantage of the two algorithms described above based on *ifpifs*-matrices, *i.e.*, IFPIFSC and IFPIFS-HC, is that working by fixed λ_1 and λ_2 values. In this article, instead of fixed λ_1 and λ_2 values, we aim to develop an equation that allows the algorithm to determine these values based on the dataset and a machine learning algorithm that can work adaptively using this equation. Moreover, the soft decision-making methods constructed by *ifpifs*-matrices, one of the efficacious decision-making techniques, can be integrated into the machine learning process. As a result, in this study, we focus on proposing a machine learning algorithm based on pseudo-similarities of *ifpifs*-matrices, adaptive λ_1 and λ_2 values for intuitionistic fuzzification of the data, and soft decision-making methods constructed by *ifpifs*-matrices. The significant contributions of the article can be summed up as follows:

- Improving two adaptive machine learning algorithms employing *ifpifs*-matrices,
- Utilizing two soft decision-making methods constructed by *ifpifs*-matrices in machine learning.
- The fact that this article is one of the pioneer studies combining soft sets, intuitionistic fuzzy sets, and machine learning.
- Applying the similarity measures of *ifpifs*-matrices to classification problems in machine learning, in contrast to many soft set-based studies on hypothetical problems.

A key innovation in our approach is the adaptive modification of lambda values in the classification algorithms, which significantly enhances the adaptability and Acc of the classifiers. By integrating *ifpifs*-matrices with two classification algorithms and dynamically adjusting lambda values, our method represents a novel contribution to machine learning. This adaptive approach not only addresses the limitations of previous classifiers but also demonstrates superior performance in handling uncertainties and dynamic natures of real-world data, setting a new benchmark for future research in the domain.

The rest of the present study is organized as follows: The second section provides some fundamental definitions, notations, and algorithms to be needed for the following sections. The third section presents the basic definitions for the two proposed algorithms and their algorithmic designs. The fourth section details the utilized datasets and performance metrics. Secondly, it simulates the proposed algorithms with the well-known and state-of-the-art fuzzy and soft sets/matrices-based classification algorithms. Finally, it statistically analyzes the simulation results using the Friedman and Nemenyi tests. The final section discusses the proposed approaches, their performance results, and the need for further research.

PRELIMINARIES

This section first introduces the notion of *ifpifs*-matrices (Enginoğlu & Arslan, 2020) and several of its fundamental characteristics. During this research, let E and U represent a parameter set and an alternative (object) set, respectively.

Definition 1 (Atanassov, 1986) Let μ and ν represent two functions from E to $[0, 1]$ such that $\mu(x) + \nu(x) \leq 1$, for all $x \in E$. The set $\{(x, \mu(x), \nu(x)) : x \in E\}$ is referred to as an intuitionistic fuzzy set (if-set) over E .

Here, for all $x \in E$, $\mu(x)$ and $\nu(x)$ are called the membership and non-membership degrees, respectively. In addition, the indeterminacy degree of x is defined by $\pi(x) = 1 - (\mu(x) + \nu(x))$.

Across the study, the set of all the if-sets over E is denoted by $IF(E)$ and $f \in IF(E)$. Briefly, the notation $\begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix} x$ can be employed instead of $(x, \mu(x), \nu(x))$. Thus, an if-set over E can be denoted by $f = \left\{ \begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix} x : x \in E \right\}$.

Definition 2 (Karaaslan, 2016) Let $f \in IF(E)$ and α be a function from f to $IF(U)$. Then, the set $\left\{ \left(\begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix} x, \alpha \left(\begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix} x \right) \right) : x \in E \right\}$ is called an *ifpifs*-set parametrized via E over U (or over U for brevity) and denoted by α .

In the present study, the set of all the *ifpifs*-sets over U is denoted by $IFPIFS_E(U)$.

Definition 3 (Enginoğlu & Arslan, 2020) Let $\alpha \in IFPIFS_E(U)$. Then, $[a_{ij}]$ is called *ifpifs*-matrix of α and is defined by

$$[a_{ij}] := \begin{bmatrix} a_{01} & a_{02} & a_{03} & \cdots & a_{0n} \\ a_{11} & a_{02} & a_{03} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

$$\text{such that } i \in \{0, 1, 2, \dots\} \text{ and } j \in \{1, 2, \dots\}, a_{ij} := \begin{cases} \begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix}, & i = 0 \\ \alpha \left(\begin{smallmatrix} \mu(x) \\ \nu(x) \end{smallmatrix} x_j \right) (u_i), & i \neq 0 \end{cases} \text{ or}$$

briefly $a_{ij} := \begin{smallmatrix} \mu_{ij} \\ \nu_{ij} \end{smallmatrix}$. Here, if $|E| = n$ and $|U| = m - 1$, then $[a_{ij}]$ is an $m \times n$ *ifpifs*-matrix.

Hereinafter, the set of all the *ifpifs*-matrices over U is denoted by $IFPIFS_E[U]$.

Secondly, we provide iCCE10 (Arslan et al., 2021) and isMBR01 (Arslan et al., 2021) in Algorithms 1 and 2, respectively, by considering the notations used across this study.

Algorithm 1 Pseudocode of iCCE10.

Input: *ipifs*-matrix $[a_{ij}]_{m \times n}$
Output: Score matrix $[s_{i1}]_{(m-1) \times 1}$, Optimum alternatives $[op_{i1}]$

- 1: $[\mu] \leftarrow [0]_{(m-1) \times 1}$
- 2: $[v] \leftarrow [0]_{(m-1) \times 1}$
- 3: **for** i from 2 to m **do**
- 4: **for** j from 1 to n **do**
- 5: $\mu_{i1} \leftarrow \mu_{i1} + a_{01j}a_{ij1}$
- 6: $v_{i1} \leftarrow v_{i1} + a_{01j}a_{ij2}$
- 7: **end for**
- 8: $s_{i1} \leftarrow \frac{1}{n} \mu_{i1}$
- 9: $s_{i2} \leftarrow \frac{1}{n} v_{i1}$
- 10: **end for**
- 11: $[sv] \leftarrow [s_{i1} - s_{i2}]_{(m-1) \times 1}$
- 12: $[av] \leftarrow [s_{i1} + s_{i2}]_{(m-1) \times 1}$
- 13: **for** i from 1 to $m - 1$ **do**
- 14: **if** $\max_{k \in I_{m-1}} \{sv_{k1}\} = 0$ AND $\min_{k \in I_{m-1}} \{sv_{k1}\} = 0$ **then**
- 15: $dm_{i1} \leftarrow 1$
- 16: **else**
- 17: $dm_{i1} \leftarrow \frac{sv_{i1} + \left| \min_{k \in I_{m-1}} \{sv_{k1}\} \right|}{\max_{k \in I_{m-1}} \{sv_{k1}\} + \left| \min_{k \in I_{m-1}} \{sv_{k1}\} \right|}$
- 18: **end if**
- 19: **end for**
- 20: $[op] \leftarrow \operatorname{argmax}_{k \in I_{m-1}} \{sv_{k1}\}$

Algorithm 2 Pseudocode of is MBR01.

Input: *ipifs*-matrix $[a_{ij}]_{m \times n}$
Output: Score matrix $[s_{i1}]_{(m-1) \times 1}$, Optimum alternatives $[op_{i1}]$

- 1: $[b] \leftarrow [0]_{(m-1) \times 1}$ and $[c] \leftarrow [0]_{(m-1) \times 1}$
- 2: **for** i from 1 to $m - 1$ **do**
- 3: **for** k from 1 to $m - 1$ **do**
- 4: **for** j from 1 to n **do**
- 5: $b_{i1} \leftarrow b_{i1} + a_{1j}^1 \operatorname{sgn}(a_{ij}^1 - a_{kj}^1)$
- 6: $c_{i1} \leftarrow c_{i1} + a_{1j}^2 \operatorname{sgn}(a_{ij}^2 - a_{kj}^2)$
- 7: **end for**
- 8: **end for**
- 9: **end for**
- 10: $[s] \leftarrow [0]_{(m-1) \times 1 \times 2}$

(Continued)

Algorithm 2 (continued)

```

11: for  $i$  from 1 to  $m - 1$  do
12:   if  $\max_{k \in I_{m-1}} \{b_{k1}\} + \left| \max_{k \in I_{m-1}} \{c_{k1}\} \right| + \left| \min_{k \in I_{m-1}} \{b_{k1}\} \right| \neq 0$  AND  $\min_{k \in I_{m-1}} \{b_{k1}\} \neq 0$  then
13:      $s_{i1}^1 \leftarrow \frac{b_{i1} + \left| \min_{k \in I_{m-1}} \{b_{k1}\} \right|}{\max_{k \in I_{m-1}} \{b_{k1}\} + \left| \max_{k \in I_{m-1}} \{c_{k1}\} \right| + \left| \min_{k \in I_{m-1}} \{b_{k1}\} \right|}$ 
14:      $s_{i1}^2 \leftarrow 1 - \frac{b_{i1} + |c_{i1}| + \left| \min_{k \in I_{m-1}} \{b_{k1}\} \right|}{\max_{k \in I_{m-1}} \{b_{k1}\} + \left| \max_{k \in I_{m-1}} \{c_{k1}\} \right| + \left| \min_{k \in I_{m-1}} \{b_{k1}\} \right|}$ 
15:   else
16:      $s_{i1}^1 \leftarrow 1$ 
17:      $s_{i1}^2 \leftarrow 0$ 
18:   end if
19: end for
20:  $[sv] \leftarrow [0]_{(m-1) \times 1}$ 
21: for  $i$  from 1 to  $m - 1$  do
22:    $sv_{i1} \leftarrow s_{i1}^1 - s_{i1}^2$ 
23: end for
24:  $[av] \leftarrow [0]_{(m-1) \times 1}$ 
25: for  $i$  from 1 to  $m - 1$  do
26:    $av_{i1} \leftarrow s_{i1}^1 + s_{i1}^2$ 
27: end for
28:  $[op] \leftarrow \operatorname{argmax}_{k \in I_{m-1}} \{sv_{k1}\}$ 

```

PROPOSED ADAPTIVE MACHINE LEARNING APPROACHES

This section overviews the fundamental mathematical notations essential for the proposed classifier based on *ifpifs*-matrices. In this article, we represent the data with a matrix $D = [d_{ij}]_{m \times (n+1)}$, where m represents the number of samples, n is the number of parameters, and the last column of D contains the labels for the data. The training data matrix, denoted as $(D_{train})_{m_1 \times n}$, along with the corresponding class matrix $C_{m_1 \times 1}$, is used to generate a testing matrix $(D_{test})_{m_2 \times n}$ derived from the original data matrix D , where $m_1 + m_2 = m$. Additionally, we employ the matrix $U_{k \times 1}$ to stand for the unique class labels extracted from $C_{m_1 \times 1}$. Notably, $D_{i-train}$ and D_{i-test} refer to the i -th rows of D_{train} and D_{test} , respectively. Similarly, $D_{train-j}$ and D_{test-j} denote the j -th columns of D_{train} and D_{test} . Furthermore, $T'_{m_2 \times 1}$ represents the predicted class labels for the testing samples.

Definition 4 Let $x, y \in \mathbb{R}^n$. Then, $P : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [-1, 1]$ is a function defined by

$$P(x, y) := \frac{n \sum_{j=1}^n x_j y_j - \left(\sum_{j=1}^n x_j \right) \left(\sum_{j=1}^n y_j \right)}{\sqrt{\left[n \sum_{j=1}^n x_j^2 - \left(\sum_{j=1}^n x_j \right)^2 \right] \left[n \sum_{j=1}^n y_j^2 - \left(\sum_{j=1}^n y_j \right)^2 \right]}}.$$

is denoted as the Pearson correlation coefficient for the variables x and y .

Definition 5 Let $x \in \mathbb{R}^n$ and $j \in I_n$. The normalizing vector of x is defined as $\hat{x} \in \mathbb{R}^n$, where

$$\hat{x}_j := \begin{cases} \frac{x_j - \min_{k \in I_n} \{x_k\}}{\max_{k \in I_n} \{x_k\} - \min_{k \in I_n} \{x_k\}}, & \max_{k \in I_n} \{x_k\} \neq \min_{k \in I_n} \{x_k\} \\ 1, & \max_{k \in I_n} \{x_k\} = \min_{k \in I_n} \{x_k\} \end{cases}.$$

Definition 6 Let $D = [d_{ij}]_{m \times (n+1)}$ be a data matrix, $i \in I_m$, and $j \in I_n$. A column normalized matrix of D is defined by $\tilde{D} = [\tilde{d}_{ij}]_{m \times n}$, where

$$\tilde{d}_{ij} := \begin{cases} \frac{d_{ij} - \min_{k \in I_m} \{d_{kj}\}}{\max_{k \in I_m} \{d_{kj}\} - \min_{k \in I_m} \{d_{kj}\}}, & \max_{k \in I_m} \{d_{kj}\} \neq \min_{k \in I_m} \{d_{kj}\} \\ 1, & \max_{k \in I_m} \{d_{kj}\} = \min_{k \in I_m} \{d_{kj}\} \end{cases}.$$

Definition 7 Let $(D_{\text{train}})_{m_1 \times n}$ be a training matrix obtained from $D = [d_{ij}]_{m \times (n+1)}$, $i \in I_{m_1}$, and $j \in I_n$. A column normalized matrix of D_{train} is defined by $\tilde{D}_{\text{train}} = [\tilde{d}_{ij-\text{train}}]_{m_1 \times n}$, where

$$\tilde{d}_{ij-\text{train}} := \begin{cases} \frac{d_{ij-\text{train}} - \min_{k \in I_{m_1}} \{d_{kj}\}}{\max_{k \in I_{m_1}} \{d_{kj}\} - \min_{k \in I_{m_1}} \{d_{kj}\}}, & \max_{k \in I_{m_1}} \{d_{kj}\} \neq \min_{k \in I_{m_1}} \{d_{kj}\} \\ 1, & \max_{k \in I_{m_1}} \{d_{kj}\} = \min_{k \in I_{m_1}} \{d_{kj}\} \end{cases}.$$

Definition 8 Let $(D_{\text{test}})_{m_2 \times n}$ be a testing matrix obtained from $D = [d_{ij}]_{m \times (n+1)}$, $i \in I_{m_2}$, and $j \in I_n$. A column normalized matrix of D_{test} is defined by $\tilde{D}_{\text{test}} = [\tilde{d}_{ij-\text{test}}]_{m_2 \times n}$, where

$$\tilde{d}_{ij-\text{test}} := \begin{cases} \frac{d_{ij-\text{test}} - \min_{k \in I_{m_2}} \{d_{kj}\}}{\max_{k \in I_{m_2}} \{d_{kj}\} - \min_{k \in I_{m_2}} \{d_{kj}\}}, & \max_{k \in I_{m_2}} \{d_{kj}\} \neq \min_{k \in I_{m_2}} \{d_{kj}\} \\ 1, & \max_{k \in I_{m_2}} \{d_{kj}\} = \min_{k \in I_{m_2}} \{d_{kj}\} \end{cases}.$$

Definition 9 (Memiş et al., 2021) Let $D_{\text{train}} = [d_{ij-\text{train}}]_{m_1 \times n}$ and $C_{m_1 \times n}$ be a training matrix and its class matrix obtained from a data matrix $D = [d_{ij}]_{m \times (n+1)}$, respectively. Then, the

matrix if $w_{D_{\text{train}}}^{\lambda P} = \begin{bmatrix} \mu_{1j}^{\lambda P} \\ v_{1j}^{\lambda P} \end{bmatrix}_{1 \times n}$ is called intuitionistic fuzzification weight (ifw) matrix

based on Pearson correlation coefficient of D_{train} and defined by

$$\mu_{1j}^{\lambda P} := 1 - (1 - |P(D_{\text{train}-j}, C)|)^\lambda$$

and

$$v_{lj}^{\lambda P} := (1 - |P(D_{train-j}, C)|)^{\lambda(\lambda+1)}$$

such that $j \in I_n$ and $\lambda \in [0, \infty)$.

Definition 10 (Memiş et al., 2021) Let $\tilde{D}_{train} = [\tilde{d}_{ij-train}]_{m_1 \times n}$ be a column normalized matrix of a matrix $(D_{train})_{m_1 \times n}$. Then, the matrix $\tilde{\tilde{D}}_{train}^{\lambda} = [\tilde{\tilde{d}}_{train-ij}^{\lambda}] = \begin{bmatrix} \mu_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} \\ v_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} \end{bmatrix}_{m_1 \times n}$ is called intuitionistic fuzzification of $\tilde{D}_{train}$ and defined by

$$\mu_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} := 1 - (1 - \tilde{d}_{ij-train})^{\lambda}$$

and

$$v_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} := (1 - \tilde{d}_{ij-train})^{\lambda(\lambda+1)}$$

such that $i \in I_{m_1}$, $j \in I_n$, and $\lambda \in [0, \infty)$.

Definition 11 (Memiş et al., 2021) Let $\tilde{D}_{test} = [\tilde{d}_{ij-test}]_{m_2 \times n}$ be a column normalized matrix of a matrix $(D_{test})_{m_2 \times n}$. Then, the matrix $\tilde{\tilde{D}}_{test}^{\lambda} = [\tilde{\tilde{d}}_{test-ij}^{\lambda}] = \begin{bmatrix} \mu_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} \\ v_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} \end{bmatrix}_{m_2 \times n}$ is called intuitionistic fuzzification of $\tilde{D}_{test}$ and defined by

$$\mu_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} := 1 - (1 - \tilde{d}_{ij-test})^{\lambda}$$

and

$$v_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} := (1 - \tilde{d}_{ij-test})^{\lambda(\lambda+1)}$$

such that $i \in I_{m_2}$, $j \in I_n$, and $\lambda \in [0, \infty)$.

Definition 12 (Memiş et al., 2021) Let $(\tilde{D}_{train})_{m_1 \times n}$ be a column normalized matrix of a matrix $(D_{train})_{m_1 \times n}$ and $\tilde{\tilde{D}}_{train}^{\lambda} = [\tilde{\tilde{d}}_{train-ij}^{\lambda}] = \begin{bmatrix} \mu_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} \\ v_{ij-train}^{\tilde{\tilde{D}}^{\lambda}} \end{bmatrix}_{m_1 \times n}$ be the intuitionistic fuzzification of $\tilde{D}_{train}$. Then, the ifpifs-matrix $\begin{bmatrix} \tilde{\tilde{D}}_{k-train}^{\lambda} \\ b_{ij} \end{bmatrix}_{2 \times n}$ is called the training ifpifs-matrix obtained by k -th row of $\tilde{\tilde{D}}_{train}^{\lambda}$ and if $w_{D_{train}}^{\lambda P}$ and defined by

$$b_{0j}^{\tilde{\tilde{D}}_{k-train}^{\lambda}} := \frac{\mu_{lj}^{\lambda P}}{v_{lj}^{\lambda P}} \text{ and } b_{1j}^{\tilde{\tilde{D}}_{k-train}^{\lambda}} := \frac{\mu_{kj-train}^{\tilde{\tilde{D}}^{\lambda}}}{v_{kj-train}^{\tilde{\tilde{D}}^{\lambda}}}$$

such that $k \in I_{m_1}$ and $j \in I_n$.

Definition 13 (Memiş et al., 2021) Let $(\tilde{D}_{test})_{m_2 \times n}$ be a column normalized matrix of a matrix $(D_{test})_{m_2 \times n}$ and $\tilde{\tilde{D}}_{test}^{\lambda} = [\tilde{\tilde{d}}_{test-ij}^{\lambda}] = \begin{bmatrix} \mu_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} \\ v_{ij-test}^{\tilde{\tilde{D}}^{\lambda}} \end{bmatrix}_{m_2 \times n}$ be the intuitionistic fuzzification of

$\tilde{D}_{test}$. Then, the *ifpifs*-matrix $\left[a_{ij}^{\tilde{D}_{k-test}^\lambda} \right]_{2 \times n}$ is called the testing *ifpifs*-matrix obtained by k -th row of $\tilde{D}_{test}^\lambda$ and if $w_{D_{train}}^{\lambda P}$ and defined by

$$a_{0j}^{\tilde{D}_{k-test}^\lambda} := \frac{\mu_{1j}^{\lambda P}}{v_{1j}^{\lambda P}} \text{ and } a_{1j}^{\tilde{D}_{k-test}^\lambda} := \frac{\mu_{kj-test}^{\tilde{D}^\lambda}}{v_{kj-test}^{\tilde{D}^\lambda}}$$

such that $k \in I_{m_1}$ and $j \in I_n$.

Secondly, it presents the concept of pseudo-similarities over $IFPIFS_E[U]$ and seven pseudo-similarities over $IFPIFS_E[U]$.

Definition 14 (Memiş et al., 2021) Let $\tilde{s} : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ be a mapping. Then, $\tilde{s}$ is a pseudo-similarity over $IFPIFS_E[U]$ if and only if $\tilde{s}$ satisfies the following properties for all $[a_{ij}], [b_{ij}] \in IFPIFS_E[U]$:

- (i) $\tilde{s}([a_{ij}], [a_{ij}]) = 1$,
- (ii) $\tilde{s}([a_{ij}], [b_{ij}]) = \tilde{s}([b_{ij}], [a_{ij}])$,
- (iii) $0 \leq \tilde{s}([a_{ij}], [b_{ij}]) \leq 1$.

Thirdly, this part provides the Minkowski, Hamming, Euclidean, Hamming-Hausdorff, Chebyshev, Jaccard, and Cosine pseudo-similarities over $IFPIFS_E[U]$ using the normalized pseudo-metrics of *ifpifs*-matrices (For more details see (Memiş et al., 2023)).

Proposition 15 (Memiş et al., 2023) Let $p \in \mathbb{Z}^+$. Then, the mapping $\tilde{s}_M^p : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ defined by

$$\tilde{s}_M^p([a_{ij}], [b_{ij}]) := 1 - \left(\frac{1}{2} \sum_{i=1}^{m-1} \sum_{j=1}^n \left(|\mu_{0j}^a \mu_{ij}^a - \mu_{0j}^b \mu_{ij}^b|^p + |v_{0j}^a v_{ij}^a - v_{0j}^b v_{ij}^b|^p + |\pi_{0j}^a \pi_{ij}^a - \pi_{0j}^b \pi_{ij}^b|^p \right) \right)^{\frac{1}{p}}$$

is a pseudo-similarity and referred to as Minkowski pseudo-similarity.

In this case, $\tilde{s}_M^1$ and $\tilde{s}_M^2$ are represented by $\tilde{s}_H$ and $\tilde{s}_E$, respectively, and are referred to as Hamming pseudo-similarity (Memiş et al., 2021) and Euclidean pseudo-similarity.

Proposition 16 (Memiş et al., 2023) Let $p \in \mathbb{Z}^+$. Then, the mapping $\tilde{s}_{Hs} : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ defined by

$$\tilde{s}_{Hs}([a_{ij}], [b_{ij}]) := 1 - \frac{1}{2(m-1)} \sum_{i=1}^{m-1} \max_{j \in I_n} \left\{ |\mu_{0j}^a \mu_{ij}^a - \mu_{0j}^b \mu_{ij}^b| |v_{0j}^a v_{ij}^a - v_{0j}^b v_{ij}^b| + |\pi_{0j}^a \pi_{ij}^a - \pi_{0j}^b \pi_{ij}^b| \right\}$$

is a pseudo-similarity and referred to as Hamming-Hausdorff pseudo-similarity.

Proposition 17 (Memiş et al., 2023) Let $p \in \mathbb{Z}^+$. Then, the mapping $\tilde{s}_{Ch} : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ defined by

$$\tilde{s}_{Ch}([a_{ij}], [b_{ij}]) := 1 - \max_{i \in I_{m-1}} \left\{ \max_{j \in I_n} \left\{ |\mu_{0j}^a \mu_{ij}^a - \mu_{0j}^b \mu_{ij}^b| + |v_{0j}^a v_{ij}^a - v_{0j}^b v_{ij}^b| + |\pi_{0j}^a \pi_{ij}^a - \pi_{0j}^b \pi_{ij}^b| \right\} \right\}$$

is a pseudo-similarity and is referred to as Chebyshev pseudo-similarity.

Proposition 18 (Memiş et al., 2023) The mapping $\tilde{s}_j : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ defined by

$$\tilde{s}_j([a_{ij}], [b_{ij}]) := \frac{1}{m-1} \sum_{i=1}^{m-1} \frac{\varepsilon + x_i}{\varepsilon + y_i + z_i - x_i}$$

such that

$$x_i = \sum_{j=1}^n \mu_{0j}^a \mu_{ij}^a \mu_{0j}^b \mu_{ij}^b + v_{0j}^a v_{ij}^a v_{0j}^b v_{ij}^b + \pi_{0j}^a \pi_{ij}^a \pi_{0j}^b \pi_{ij}^b$$

$$y_i = \sum_{j=1}^n \left(\mu_{0j}^a \mu_{ij}^a \right)^2 + \left(v_{0j}^a v_{ij}^a \right)^2 + \left(\pi_{0j}^a \pi_{ij}^a \right)^2$$

and

$$z_i = \sum_{j=1}^n \left(\mu_{0j}^b \mu_{ij}^b \right)^2 + \left(v_{0j}^b v_{ij}^b \right)^2 + \left(\pi_{0j}^b \pi_{ij}^b \right)^2$$

is known as the Jaccard pseudo-similarity, and it is a pseudo-similarity. Here, $0 \ll \varepsilon < 1$, for example, $\varepsilon = 0.000001$.

Proposition 19 (Memiş et al., 2023) The mapping $\tilde{s}_c : IFPIFS_E[U] \times IFPIFS_E[U] \rightarrow \mathbb{R}$ defined by

$$\tilde{s}_c([a_{ij}], [b_{ij}]) := \frac{1}{m-1} \sum_{i=1}^{m-1} \frac{\varepsilon + x_i}{\varepsilon + \sqrt{y_i} \sqrt{z_i}}$$

such that

$$x_i = \sum_{j=1}^n \mu_{0j}^a \mu_{ij}^a \mu_{0j}^b \mu_{ij}^b + v_{0j}^a v_{ij}^a v_{0j}^b v_{ij}^b + \pi_{0j}^a \pi_{ij}^a \pi_{0j}^b \pi_{ij}^b$$

$$y_i = \sum_{j=1}^n \left(\mu_{0j}^a \mu_{ij}^a \right)^2 + \left(v_{0j}^a v_{ij}^a \right)^2 + \left(\pi_{0j}^a \pi_{ij}^a \right)^2$$

and

$$z_i = \sum_{j=1}^n \left(\mu_{0j}^b \mu_{ij}^b \right)^2 + \left(v_{0j}^b v_{ij}^b \right)^2 + \left(\pi_{0j}^b \pi_{ij}^b \right)^2$$

is known as the Cosine pseudo-similarity, and it is a pseudo-similarity. Here, $0 \ll \varepsilon < 1$, for example, $\varepsilon = 0.000001$.

Fourthly, it propounds the classification algorithms AIFPIFSC1 and AIFPIFSC2 and provides the pseudocodes of normalize and intuitionistic normalize functions in Algorithms 3 and 4 to be needed for the proposed algorithms' pseudocodes (Algorithms 5 and 6).

Algorithm 3 Normalize function.

Input: $a_{m \times n}$
Output: $\tilde{a}_{m \times n}$

```

1: function NORMALIZE( $a$ )
2:    $[m, n] \leftarrow \text{size}(a)$ 
3:   if  $\max(a) \neq \min(a)$  then
4:      $\tilde{a} \leftarrow (a - \min(a)) / (\max(a) - \min(a))$ 
5:   else
6:      $\tilde{a} \leftarrow \text{ones}(m, n)$ 
7:   end if
8:   return  $\tilde{a}$ 
9: end function

```

Algorithm 4 Intuitionistic normalize function.

Input: $a_{m \times n}, \lambda$
Output: $\tilde{\tilde{a}}_{m \times n \times 2}$

```

1: function INORMALIZE( $a, \lambda$ )
2:    $[m, n] \leftarrow \text{size}(a)$ 
3:   for  $i = 1$  to  $m$  do
4:     for  $j = 1$  to  $n$  do
5:        $\tilde{a}(i, j, 1) \leftarrow 1 - (1 - a(i, j))^\lambda$ 
6:        $\tilde{a}(i, j, 2) \leftarrow (1 - a(i, j))^{\lambda(\lambda+1)}$ 
7:     end for
8:   end for
9:   return  $\tilde{\tilde{a}}$ 
10: end function

```

In the algorithm AIFPIFSC1, parameters λ_1 and λ_2 are determined based on the dataset's characteristics. Then, ifw is computed by measuring the Pearson correlation coefficient between each feature and the class labels. These weights are utilized to construct two *ifpifs*-matrices: one for training data and the other for testing data. Feature fuzzification of the training data features is obtained using the ifw . For each testing sample, a comparison matrix is constructed by calculating pseudo-similarities between the testing *ifpifs*-matrix and training *ifpifs*-matrix. This comparison matrix serves as the basis for parameter weights' computation. Parameter weights are determined by calculating the standard deviation of each column in the comparison matrix. A comparison *ifpifs*-matrix is then generated by combining these parameter weights with the comparison matrix. Next, we apply the algorithm isMBR01 to this comparison *ifpifs*-matrix to identify the

Algorithm 5 AIFPIFSC1's pseudocode algorithm.

Input: $(D_{train})_{m_1 \times n}$, $C_{m_1 \times 1}$, and $(D_{test})_{m_2 \times n}$

Output: $T'_{m_2 \times 1}$

```

1: function AIFPIFSC1(train, C, test)
2:    $\lambda_1 \leftarrow \text{round}(\sqrt{n})$ 
3:    $\lambda_2 \leftarrow \text{round}(\ln(n) / \ln(\text{length}(\text{unique}(C))))$ 
4:   Compute if  $w_{D_{train}}^{\lambda_1 P}$  using  $D_{train}$ , C and  $\lambda_1$ 
5:   Compute feature fuzzification of  $D_{train}$  and  $D_{test}$ , namely  $\tilde{D}_{train}$  and  $\tilde{D}_{test}$ 
6:   Compute  $\tilde{D}_{train}^{\lambda_1}$  and  $\tilde{D}_{test}^{\lambda_1}$ 
7:   for i from 1 to tm do
8:     Compute test ifpifs-matrix  $[a_{ij}]$  using if  $w_{D_{train}}^{\lambda_1 P}$  and  $\tilde{D}_{k-test}^{\lambda_1}$ 
9:     for j from 1 to em do
10:      Compute train ifpifs-matrix  $[b_{ij}]$  using if  $w_{D_{train}}^{\lambda_1 P}$  and  $\tilde{D}_{k-train}^{\lambda_1}$ 
11:       $cm(j, 1) \leftarrow \tilde{s}_H([a_{ij}], [b_{ij}])$ 
12:       $cm(j, 2) \leftarrow \tilde{s}_{Hs}([a_{ij}], [b_{ij}])$ 
13:       $cm(j, 3) \leftarrow \tilde{s}_{Ch}([a_{ij}], [b_{ij}])$ 
14:       $cm(j, 4) \leftarrow \tilde{s}_E([a_{ij}], [b_{ij}])$ 
15:       $cm(j, 5) \leftarrow \tilde{s}_M^3([a_{ij}], [b_{ij}])$ 
16:       $cm(j, 6) \leftarrow \tilde{s}_J([a_{ij}], [b_{ij}])$ 
17:       $cm(j, 7) \leftarrow \tilde{s}_C([a_{ij}], [b_{ij}])$ 
18:    end for
19:    for j from 1 to size(cm, 2) do
20:       $sd(1, j) \leftarrow \text{std}(cm(:, j))$ 
21:    end for
22:     $wm2(1, :) \leftarrow 1 - \text{NORMALIZE}(sd/4)$ 
23:     $dm \leftarrow [wm2; cm]$ 
24:     $idm \leftarrow \text{INORMALIZE}(dm, \lambda_2)$ 
25:     $[ , , , op] \leftarrow \text{isMBR01}(idm)$ 
26:     $t'_{il} \leftarrow C(op(1), 1)$ 
27:  end for
28: end function

```

optimum training sample. The class label of this optimum training sample is assigned to the corresponding testing sample. These steps are repeated for all testing samples to obtain the predicted class labels.

The similar steps in Algorithm 6 are repeated for the algorithm AIFPIFSC2. In the last part, we apply the algorithm iCCE10 to the aforesaid comparison *ifpifs*-matrix to determine the optimum training sample.

Algorithm 6 AIFPIFSC2' pseudocode.

Input: $(D_{train})_{m_1 \times n}$, $C_{m_1 \times 1}$, and $(D_{test})_{m_2 \times n}$

Output: $T'_{m_2 \times 1}$

```

1: function AIFPIFSC2(train, C, test)
2:    $\lambda_1 \leftarrow \text{round}(\sqrt{n})$ 
3:    $\lambda_2 \leftarrow \text{round}(\ln(n)/\ln(\text{length}(\text{unique}(C))))$ 
4:   Compute if  $w_{D_{train}}^{\lambda_1 P}$  using  $D_{train}$ , C and  $\lambda_1$ 
5:   Compute feature fuzzification of  $D_{train}$  and  $D_{test}$ , namely  $\tilde{D}_{train}$  and  $\tilde{D}_{test}$ 
6:   Compute  $\tilde{D}_{train}^{\lambda_1}$  and  $\tilde{D}_{test}^{\lambda_1}$ 
7:   for i from 1 to tm do
8:     Compute test ifpifs-matrix  $[a_{ij}]$  using if  $w_{D_{train}}^{\lambda_1 P}$  and  $\tilde{D}_{k-test}^{\lambda_1}$ 
9:     for j from 1 to em do
10:      Compute train ifpifs-matrix  $[b_{ij}]$  using if  $w_{D_{train}}^{\lambda_1 P}$  and  $\tilde{D}_{k-train}^{\lambda_1}$ 
11:       $cm(j, 1) \leftarrow \tilde{s}_H([a_{ij}], [b_{ij}])$ 
12:       $cm(j, 2) \leftarrow \tilde{s}_{Hs}([a_{ij}], [b_{ij}])$ 
13:       $cm(j, 3) \leftarrow \tilde{s}_{Ch}([a_{ij}], [b_{ij}])$ 
14:       $cm(j, 4) \leftarrow \tilde{s}_E([a_{ij}], [b_{ij}])$ 
15:       $cm(j, 5) \leftarrow \tilde{s}_M^3([a_{ij}], [b_{ij}])$ 
16:       $cm(j, 6) \leftarrow \tilde{s}_J([a_{ij}], [b_{ij}])$ 
17:       $cm(j, 7) \leftarrow \tilde{s}_C([a_{ij}], [b_{ij}])$ 
18:    end for
19:    for j from 1 to size(cm, 2) do
20:       $sd(1, j) \leftarrow \text{std}(cm(:, j))$ 
20:    end for
21:     $wm2(1, :) \leftarrow 1 - \text{NORMALIZE}(sd/4)$ 
22:     $dm \leftarrow [wm2; cm]$ 
23:     $idm \leftarrow \text{INORMALIZE}(dm, \lambda_2)$ 
24:     $[ , , , op] \leftarrow \text{iCCE10}(idm)$ 
25:     $t'_{i1} \leftarrow C(op(1), 1)$ 
26:  end for
27: end function

```

Finally, an illustrative example is presented below to enhance the clarity of the AIFPIFSC1 algorithm concerning its pseudocode. In this study, we use *k*-fold cross-validation to avoid the possibility of the algorithm's results being overly optimistic in a single run.

Illustrative Example:

A data matrix $D = [d_{ij}]_{10 \times 5}$ from the "Iris" dataset (Fisher, 1936) is provided below for implementing AIFPIFSC1. The matrix contains 10 samples divided into three classes

($l = 3$), with the class labels in the last column. Class one consists of three samples, class two has three samples, and class three includes four. In the first iteration of five-fold cross-validation, $(D_{train})_{8 \times 4}$, $C_{8 \times 1}$, $(D_{test})_{2 \times 4}$, and $T_{2 \times 1}$ are obtained as follows:

$$D = \begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 & 1 \\ 4.9 & 3.0 & 1.4 & 0.2 & 1 \\ 4.7 & 3.2 & 1.3 & 0.2 & 1 \\ 7.0 & 3.2 & 4.7 & 1.4 & 2 \\ 6.4 & 3.2 & 4.5 & 1.5 & 2 \\ 6.9 & 3.1 & 4.9 & 1.5 & 2 \\ 6.3 & 3.3 & 6.0 & 2.5 & 3 \\ 5.8 & 2.7 & 5.1 & 1.9 & 3 \\ 7.1 & 3.0 & 5.9 & 2.1 & 3 \\ 6.3 & 2.9 & 5.6 & 1.8 & 3 \end{bmatrix}$$

$$D_{train} = \begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 7.0 & 3.2 & 4.7 & 1.4 \\ 6.9 & 3.1 & 4.9 & 1.5 \\ 6.3 & 3.3 & 6.0 & 2.5 \\ 5.8 & 2.7 & 5.1 & 1.9 \\ 7.1 & 3.0 & 5.9 & 2.1 \end{bmatrix} \quad C = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 3 \\ 3 \end{bmatrix} \quad D_{test} = \begin{bmatrix} 6.4 & 3.2 & 4.5 & 1.5 \\ 6.3 & 2.9 & 5.6 & 1.8 \end{bmatrix} \quad T = \begin{bmatrix} 2 \\ 3 \end{bmatrix}.$$

D_{train} , C , and D_{test} are inputted to AIFPIFSC1. After the classification task, the ground truth class matrix T is utilized to calculate the Acc, Pre, Rec, Spe, MacF, and MicF rates of AIFPIFSC1.

Secondly, λ_1 and λ_2 are calculated as 2 and 1, respectively.

Thirdly, the *ifw* matrix is calculated via D_{train} and C concerning the Pearson correlation coefficient. The main goal of obtaining the *ifw* matrix is to construct the train and test *ifpifs*-matrices in the following phases.

$$[ifw_{ij}] = \begin{bmatrix} 0.9110 & 0.7026 & 0.9975 & 0.9994 \\ 0.0007 & 0.0263 & 0 & 0 \end{bmatrix}.$$

Thirdly, feature fuzzifications of D_{train} and D_{test} are computed as follows:

$$\tilde{D}_{train} = \begin{bmatrix} 0.1667 & 1 & 0.0213 & 0 \\ 0.0833 & 0.3750 & 0.0213 & 0 \\ 0 & 0.6250 & 0 & 0 \\ 0.9583 & 0.6250 & 0.7234 & 0.5217 \\ 0.9167 & 0.5000 & 0.7660 & 0.5652 \\ 0.6667 & 0.7500 & 1 & 1 \\ 0.4583 & 0 & 0.8085 & 0.7391 \\ 1 & 0.3750 & 0.9787 & 0.8261 \end{bmatrix} \quad \tilde{D}_{test} = \begin{bmatrix} 0.7083 & 0.6250 & 0.6809 & 0.5652 \\ 0.6667 & 0.2500 & 0.9149 & 0.6957 \end{bmatrix}.$$

Fourthly, intuitionistic feature fuzzifications of $\tilde{D}_{train}$ and $\tilde{D}_{test}$ are computed as follows:

$$\tilde{D}_{train} = \begin{bmatrix} 0.1667 & 1.0000 & 0.0213 & 0.0000 \\ 0.6944 & 0.0000 & 0.9579 & 1.0000 \\ 0.0833 & 0.3750 & 0.0213 & 0.0000 \\ 0.8403 & 0.3906 & 0.9579 & 1.0000 \\ 0.0000 & 0.6250 & 0.0000 & 0.0000 \\ 1.0000 & 0.1406 & 1.0000 & 0.0000 \\ 0.9583 & 0.6250 & 0.7234 & 0.5217 \\ 0.0017 & 0.1406 & 0.0765 & 0.2287 \\ 0.9167 & 0.5000 & 0.7660 & 0.5652 \\ 0.0000 & 0.2500 & 0.0548 & 0.1890 \\ 0.6667 & 0.7500 & 1.0000 & 1.0000 \\ 0.1111 & 0.0625 & 0.0000 & 0.0000 \\ 0.4583 & 0.0000 & 0.8085 & 0.7391 \\ 0.2934 & 1.0000 & 0.0367 & 0.0681 \\ 1.0000 & 0.3750 & 0.9787 & 0.8261 \\ 0.8403 & 0.3906 & 0.0005 & 0.0302 \end{bmatrix} \quad \tilde{D}_{test} = \begin{bmatrix} 0.7083 & 0.6250 & 0.6809 & 0.5652 \\ 0.0851 & 0.1406 & 1.0190 & 0.1890 \\ 0.6667 & 0.2500 & 0.9149 & 0.6957 \\ 0.1111 & 0.5625 & 0.0072 & 0.0926 \end{bmatrix}.$$

In the next steps, $\tilde{D}_{train}$ and $\tilde{D}_{test}$ are required to constructing the train and test *ifpifs*-matrices.

For $i = 1$, the following steps are performed:

Fifthly, for all j , test *ifpifs*-matrix $[a_{ij}]$ and j^{th} train *ifpifs*-matrix $[b_{ij}]$ are constructed by utilizing $\tilde{D}_{i-test}$ and $\tilde{D}_{j-train}$. For instance ($j = 1$), for $\tilde{D}_{1-test}$ and $\tilde{D}_{1-train}$, test *ifpifs*-matrix $[a_{ij}]$ and 1-train *ifpifs*-matrix $[b_{ij}]$ are constructed as follows:

$$[a_{ij}] = \begin{bmatrix} 0.9110 & 0.7026 & 0.9975 & 0.9994 \\ 0.0007 & 0.0263 & 0 & 0 \\ 0.7083 & 0.6250 & 0.6809 & 0.5652 \\ 0.0851 & 0.1406 & 0.1019 & 0.1890 \end{bmatrix} \quad [b_{ij}] = \begin{bmatrix} 0.9110 & 0.7026 & 0.9975 & 0.9994 \\ 0.0007 & 0.0263 & 0 & 0 \\ 0.1667 & 1 & 0.0213 & 0 \\ 0.6944 & 0 & 0.9579 & 1 \end{bmatrix}.$$

The first rows of $[a_{ij}]$ and $[b_{ij}]$ are the intuitionistic feature weights obtained in the third step. Second rows of $[a_{ij}]$ and $[b_{ij}]$ are the first row (first sample) of $\tilde{D}_{test}$ and first row (first sample) of $\tilde{D}_{train}$, respectively.

Sixthly, cm_{11} , cm_{12} , cm_{13} , cm_{14} , cm_{15} , cm_{16} , and cm_{17} of *CM* are computed by employing the pseudo-similarities of the aforesaid $[a_{ij}]$ and $[b_{ij}]$ as follows:

$$\begin{aligned} cm_{11} &= \tilde{s}_H([a_{ij}], [b_{ij}]) = 0.7432 \\ cm_{12} &= \tilde{s}_{Hs}([a_{ij}], [b_{ij}]) = 0.6708 \\ cm_{13} &= \tilde{s}_{Ch}([a_{ij}], [b_{ij}]) = 0.3416 \\ cm_{14} &= \tilde{s}_E([a_{ij}], [b_{ij}]) = 0.6345 \\ cm_{15} &= \tilde{s}_M^3([a_{ij}], [b_{ij}]) = 0.9742 \\ cm_{16} &= \tilde{s}_J([a_{ij}], [b_{ij}]) = 0.2826 \\ cm_{17} &= \tilde{s}_C([a_{ij}], [b_{ij}]) = 0.4960. \end{aligned}$$

Then, *CM* is calculated as follows:

$$CM = \begin{bmatrix} 0.7432 & 0.6708 & 0.3416 & 0.6345 & 0.9742 & 0.2826 & 0.4960 \\ 0.7516 & 0.6708 & 0.3416 & 0.6280 & 0.9719 & 0.1420 & 0.5495 \\ 0.7614 & 0.6601 & 0.3203 & 0.6132 & 0.9672 & 0.1413 & 0.3758 \\ 0.9589 & 0.8787 & 0.7575 & 0.9165 & 0.9995 & 0.9652 & 0.9887 \\ 0.9523 & 0.8993 & 0.7987 & 0.9201 & 0.9997 & 0.9678 & 0.9893 \\ 0.8881 & 0.7827 & 0.5653 & 0.8065 & 0.9952 & 0.8619 & 0.9736 \\ 0.8677 & 0.7374 & 0.4747 & 0.8077 & 0.9956 & 0.8068 & 0.8931 \\ 0.8719 & 0.8512 & 0.7024 & 0.8204 & 0.9972 & 0.8768 & 0.9766 \end{bmatrix}.$$

Seventhly, the standard deviation matrix $[sd_{1j}]$ and second weight matrix $[wm_{1j}]$ are computed as follows:

$$[sd_{1j}] = [0.0874 \quad 0.0986 \quad 0.1973 \quad 0.1267 \quad 0.0139 \quad 0.3723 \quad 0.2601]$$

and

$$[wm_{1j}] = [0.7947 \quad 0.7634 \quad 0.4882 \quad 0.6851 \quad 1 \quad 0 \quad 0.3130].$$

Eighthly, the decision matrix DM is obtained by employing CM and $[wm_{1j}]$ as follows:

$$DM = \begin{bmatrix} 0.7947 & 0.7634 & 0.4882 & 0.6851 & 1 & 0 & 0.3130 \\ 0.7432 & 0.6708 & 0.3416 & 0.6345 & 0.9742 & 0.2826 & 0.4960 \\ 0.7516 & 0.6708 & 0.3416 & 0.6280 & 0.9719 & 0.1420 & 0.5495 \\ 0.7614 & 0.6601 & 0.3203 & 0.6132 & 0.9672 & 0.1413 & 0.3758 \\ 0.9589 & 0.8787 & 0.7575 & 0.9165 & 0.9995 & 0.9652 & 0.9887 \\ 0.9523 & 0.8993 & 0.7987 & 0.9201 & 0.9997 & 0.9678 & 0.9893 \\ 0.8881 & 0.7827 & 0.5653 & 0.8065 & 0.9952 & 0.8619 & 0.9736 \\ 0.8677 & 0.7374 & 0.4747 & 0.8077 & 0.9956 & 0.8068 & 0.8931 \end{bmatrix}.$$

Ninthly, the intuitionistic decision matrix $iDM = \text{inormalize}(DM, \lambda_2)$ is computed as follows:

$$iDM = \begin{bmatrix} 0.7947 & 0.7634 & 0.4882 & 0.6851 & 1.0000 & 0.0000 & 0.3130 \\ 0.0421 & 0.0560 & 0.2619 & 0.0992 & 0.0000 & 1.0000 & 0.4719 \\ 0.7432 & 0.6708 & 0.3416 & 0.6345 & 0.9742 & 0.2826 & 0.4960 \\ 0.0659 & 0.1084 & 0.4335 & 0.1336 & 0.0007 & 0.5146 & 0.2541 \\ 0.7516 & 0.6708 & 0.3416 & 0.6280 & 0.9719 & 0.1420 & 0.5495 \\ 0.0617 & 0.1084 & 0.4335 & 0.1384 & 0.0008 & 0.7361 & 0.2029 \\ 0.7614 & 0.6601 & 0.3203 & 0.6132 & 0.9672 & 0.1413 & 0.3758 \\ 0.0569 & 0.1155 & 0.4620 & 0.1496 & 0.0011 & 0.7374 & 0.3896 \\ 0.9589 & 0.8787 & 0.7575 & 0.9165 & 0.9995 & 0.9652 & 0.9887 \\ 0.0017 & 0.0147 & 0.0588 & 0.0070 & 0.0012 & 0.0010 & 0.0001 \\ 0.9523 & 0.8993 & 0.7987 & 0.9201 & 0.9997 & 0.9678 & 0.9893 \\ 0.0125 & 0.0472 & 0.1889 & 0.0374 & 0.0000 & 0.0191 & 0.0007 \\ 0.8881 & 0.7827 & 0.5653 & 0.8065 & 0.9952 & 0.8619 & 0.9736 \\ 0.0163 & 0.0418 & 0.2759 & 0.0430 & 0.0000 & 0.0373 & 0.0114 \\ 0.8677 & 0.7374 & 0.4747 & 0.8077 & 0.9956 & 0.8068 & 0.8931 \\ 0.0164 & 0.0221 & 0.0886 & 0.0323 & 0.0000 & 0.0152 & 0.0005 \end{bmatrix}.$$

Tenthly, the soft decision-making algorithm isMBR01 based on *ifpifs*-matrices is applied to iDM and determines the optimum train sample.

Finally, the label of the optimum train sample is assigned to t'_{11} . Because the optimum train sample is the fifth, its label 2 is assigned to t'_{11} .

If t'_{21} is calculated as t'_{11} is, then the predicted class matrix $T'_{2 \times 1}$ is attained as $T' = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. According to these results, the Acc rate of the proposed algorithm is 100% for the considered example.

SIMULATION AND PERFORMANCE COMPARISON

The current section provides a comprehensive overview of the 15 datasets within the UCI-MLR for classification tasks. Six performance metrics that can be used to compare performances are then presented. Subsequently, a simulation is conducted to illustrate that AIFPIFSC1 and AIFPIFSC2 outperform Fuzzy kNN, FPFS-kNN, FPFS-AC, FPFS-EC,

Table 1 Descriptions of UCI datasets.

No.	Ref.	Name	#Instance	#Attribute	#Class	Balanced/Imbalanced
1	Nakai (1996)	Ecoli	336	7	8	Imbalanced
2	Silva & Maral (2013)	Leaf	340	14	36	Imbalanced
3	Dias, Peres & Bscaro (2009)	Libras	360	90	15	Balanced
4	Higuera, Gardiner & Cios (2015)	Mice	1,077	72	8	Imbalanced
5	Lichtinghagen, Klawonn & Hoffmann (2020)	HCV Data	589	12	5	Imbalanced
6	Barreto & Neto (2005)	Column3C	310	6	3	Imbalanced
7	Quinlan (1986)	NewThyroid	215	5	3	Imbalanced
8	Deterding, Niranjana & Robinson (1988)	Vowel	990	13	11	Imbalanced
9	Fisher (1936)	Iris	150	4	3	Imbalanced
10	Charytanowicz et al. (2010)	Seeds	210	7	3	Balanced
11	Bhatt (2017)	Wireless	2,000	7	4	Balanced
12	Aeberhard & Forina (1992)	Wine	178	13	3	Imbalanced
13	Cardoso (2013)	WholesaleR	440	6	3	Imbalanced
14	JP & Jossinet (1996)	Breast Tissue	106	9	6	Imbalanced
15	Breiman et al. (1984)	Led7Digit	500	7	7	Imbalanced

Note:

represents “the number of”.

PPFS-CMC, IFPIFSC, and IFPIFS-HC in terms of classification performance.

Furthermore, the section conducts statistical analyses on the simulation results using the Friedman test, a non-parametric test, and the Nemenyi test, a *post hoc* test.

UCI datasets and features

This subsection provides the characteristics of the datasets utilized in the simulation, as outlined in [Table 1](#). The datasets in [Table 1](#) can be accessed from the UCI-MLR ([Kelly, Longjohn & Nottingham, 2024](#)).

The main problems related to each dataset in [Table 1](#) are as follows: In “Ecoli” predicting the localization site of Ecoli proteins based on biological attributes. In “Leaf”, different species of plants are classified using leaf shape and texture features. In “Libras”, accelerometer data recognizes hand movements in Libras (a Brazilian sign language). In “Mice”, gene expression data is analyzed to distinguish between mouse strains. In “HCV Data”, detect Hepatitis C Virus (HCV) infection stages using blood test data. In “Column 3C”, diagnosing spinal disorders by analyzing biomechanical features of vertebrae. In “NewThyroid”, thyroid conditions (*e.g.*, normal or hypothyroid) are classified based on medical attributes. In “Vowel”, spoken vowels are classified using speech acoustics data. In “Iris”, identify iris species based on flower petal and sepal dimensions. In “Seeds”, classify types of wheat seeds based on their geometric properties. In “Wireless”, predict indoor locations based on Wi-Fi signal strength features. In “Wine”, distinguish wine varieties based on their chemical compositions. In “Wholesale”, predict customer segments based on annual spending on various product categories. In “Breast Tissue”, types of breast tissue samples (*e.g.*, healthy or tumor) are classified using bioimpedance data. In “Led7Digit”, human errors in recognizing handwritten digits are simulated on a seven-segment display.

Performance metrics

This subsection outlines the mathematical notations for six performance metrics (Erkan, 2021; Fawcett, 2006; Nguyen et al., 2019), namely Acc, Pre, Rec, Spe, MacF, and MicF, used to compare the mentioned classifiers. Let $D_{test} = \{x_1, x_2, \dots, x_n\}$, $T = \{t_1, t_2, \dots, t_n\}$, $T' = \{t'_1, t'_2, \dots, t'_n\}$, and l be n samples for classification, ground truth class sets of the samples, prediction class sets of the samples, and the number of classes for the samples, respectively. Here,

$$\begin{aligned} \text{Acc}(T, T') &:= \frac{1}{l} \sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} & \text{Pre}(T, T') &:= \frac{1}{l} \sum_{i=1}^l \frac{TP_i}{TP_i + FP_i} \\ \text{Rec}(T, T') &:= \frac{1}{l} \sum_{i=1}^l \frac{TP_i}{TP_i + FN_i} & \text{Spe}(T, T') &:= \frac{1}{l} \sum_{i=1}^l \frac{TN_i}{TN_i + FP_i} \\ \text{MacF}(T, T') &:= \frac{1}{l} \sum_{i=1}^l \frac{2TP_i}{2TP_i + FP_i + FN_i} & \text{MicF}(T, T') &:= \frac{2 \sum_{i=1}^l TP_i}{2 \sum_{i=1}^l TP_i + \sum_{i=1}^l FP_i + \sum_{i=1}^l FN_i} \end{aligned}$$

where the numbers for true positive, true negative, false positive, and false negative for the class i are, respectively, TP_i , TN_i , FP_i , and FN_i , respectively. Their mathematical notations are as follows:

$$\begin{aligned} TP_i &:= \left| \{x_k | i \in T_k \wedge i \in T'_k, 1 \leq k \leq l\} \right| & TN_i &:= \left| \{x_t | i \notin T_k \wedge i \notin T'_k, 1 \leq k \leq l\} \right| \\ FP_i &:= \left| \{x_t | i \notin T_k \wedge i \in T'_k, 1 \leq k \leq l\} \right| & FN_i &:= \left| \{x_t | i \in T_k \wedge i \notin T'_k, 1 \leq k \leq l\} \right|. \end{aligned}$$

Here, the notation $|\cdot|$ denotes the cardinality of a set.

Simulation results

This subsection conducts a comparative analysis between AIFPIFSC1 and AIFPIFSC2 and well-established classifiers that rely on fuzzy and soft sets, such as Fuzzy kNN, FPFS-kNN, FPFS-AC, FPFS-EC, FPFS-CMC, IFPIFSC, and IFPIFS-HC. The comparison is performed using MATLAB R2021b (The MathWorks, Natick, NY, USA) on a laptop with an Intel(R) Core(TM) i5-10300H CPU @ 2.50 GHz and 8 GB RAM. The mean performance results of the classifiers are derived from random 10 independent runs based on five-fold cross-validation (Stone, 1974). In each cross-validation iteration, the dataset is randomly divided into five parts, whose four parts are used for training and the remaining part for testing (for further details on k -fold cross-validation, refer to Stone (1974)). Table 2 presents the average Acc, Pre, Rec, Spe, MacF, and MicF results for AIFPIFSC1, AIFPIFSC2, Fuzzy kNN, FPFS-kNN, FPFS-AC, FPFS-EC, FPFS-CMC, IFPIFSC, and IFPIFS-HC across the datasets.

Table 2 demonstrates that AIFPIFSC1 and AIFPIFSC2 precisely categorize the dataset “Mice Protein Expression” in the same manner as FPFS-kNN, FPFS-AC, FPFS-EC, FPFS-CMC, IFPIFSC, and IFPIFS-HC. Furthermore, according to all performance measures, AIFPIFSC1’s results for the datasets “NewThyroid”, “Vowel”, “Iris”, “Seeds”, “Wireless”,

Table 2 Comparative simulation results of the aforesaid classifiers.

Datasets	Classifiers	Acc	Pre	Rec	Spe	MacF	MicF
Ecoli	Fuzzy kNN	92.1824	53.5997	60.2825	95.6846	66.3808	68.7296
	FPFS-kNN	94.4340	72.9682	65.3519	96.4741	74.9499	81.1010
	FPFS-AC	94.2172	72.7755	68.6908	96.3172	75.8222	79.3758
	FPFS-EC	94.2085	70.1592	67.0879	96.3468	74.7286	79.2559
	FPFS-CMC	94.0483	69.1430	66.2550	96.2559	73.5161	78.6602
	IFPIFSC	92.2406	63.4527	60.4435	95.1167	68.6435	72.5285
	IFPIFS-HC	92.6453	64.7535	62.4497	95.4705	69.8072	73.6901
	AIFPIFSC1	94.6825	76.8664	71.1918	96.6138	78.6652	81.3745
	AIFPIFSC2	94.4892	76.1900	70.1926	96.4886	77.8141	80.6892
Leaf	Fuzzy kNN	96.2016	32.5347	32.0444	98.0528	62.0025	32.7941
	FPFS-kNN	97.8353	73.3337	67.4556	98.8794	73.8272	67.5294
	FPFS-AC	97.8569	71.8471	67.6889	98.8910	74.4906	67.8529
	FPFS-EC	97.8039	71.8566	66.9333	98.8633	73.7364	67.0588
	FPFS-CMC	97.7686	71.2570	66.5444	98.8451	73.4777	66.5294
	IFPIFSC	97.7608	71.0219	65.9667	98.8419	72.6249	66.4118
	IFPIFS-HC	97.7176	70.1101	65.4611	98.8193	72.3277	65.7647
	AIFPIFSC1	98.0784	75.5176	70.7778	99.0055	75.9554	71.1765
	AIFPIFSC2	98.0843	75.2774	70.9778	99.0086	76.2688	71.2647
Libras	Fuzzy kNN	95.8963	74.0815	69.2600	97.8017	70.1990	69.2222
	FPFS-kNN	96.7037	79.3345	75.2267	98.2343	75.3681	75.2778
	FPFS-AC	97.2815	82.1165	79.6200	98.5435	79.3057	79.6111
	FPFS-EC	97.0667	80.8036	78.0400	98.4288	78.0580	78.0000
	FPFS-CMC	96.9481	79.8471	77.1400	98.3655	77.2585	77.1111
	IFPIFSC	95.9704	73.5069	69.9000	97.8417	70.0397	69.7778
	IFPIFS-HC	96.1889	74.5579	71.4867	97.9589	71.0616	71.4167
	AIFPIFSC1	97.3444	82.8129	80.1467	98.5777	79.7321	80.0833
	AIFPIFSC2	97.2593	82.6086	79.5267	98.5321	79.3585	79.4444
Mice	Fuzzy kNN	99.8608	99.4865	99.4429	99.9200	99.4470	99.4431
	FPFS-kNN	100	100	100	100	100	100
	FPFS-AC	100	100	100	100	100	100
	FPFS-EC	100	100	100	100	100	100
	FPFS-CMC	100	100	100	100	100	100
	IFPIFSC	100	100	100	100	100	100
	IFPIFS-HC	100	100	100	100	100	100
	AIFPIFSC1	100	100	100	100	100	100
	AIFPIFSC2	100	100	100	100	100	100
HCV Data	Fuzzy kNN	97.0934	55.1835	48.1424	96.0803	64.1767	92.7334
	FPFS-kNN	96.9913	56.7283	38.1858	90.3967	84.7871	92.4782
	FPFS-AC	97.7862	70.9108	53.5449	94.7062	74.2838	94.4654
	FPFS-EC	97.0867	59.1956	46.0910	91.5718	77.9663	92.7168

(Continued)

Table 2 (continued)

Datasets	Classifiers	Acc	Pre	Rec	Spe	MacF	MicF
	FPFS-CMC	97.0798	62.4317	47.7072	91.6225	75.4864	92.6994
	IFPIFSC	97.7930	67.9679	55.7708	95.5895	75.1353	94.4824
	IFPIFS-HC	97.8271	67.3802	54.1163	95.8067	76.3137	94.5677
	AIFPIFSC1	98.1460	77.3406	62.4411	95.6001	77.2945	95.3649
	AIFPIFSC2	98.0916	77.7644	62.1039	95.1745	76.7595	95.2289
Column3c	Fuzzy kNN	80.4086	75.9644	61.3111	82.1301	63.4923	70.6129
	FPFS-kNN	84.0430	73.2282	71.4111	87.7623	71.6751	76.0645
	FPFS-AC	82.1290	69.5714	68.6000	86.5810	68.5903	73.1935
	FPFS-EC	82.1720	69.6658	68.6444	86.6150	68.6118	73.2581
	FPFS-CMC	81.8710	69.1010	68.1556	86.3683	68.1437	72.8065
	IFPIFSC	85.0538	72.2589	71.4000	89.1168	71.3761	77.5806
	IFPIFS-HC	84.3441	71.8038	70.3111	88.4744	70.4134	76.5161
	AIFPIFSC1	85.7204	73.5563	72.8556	89.6705	72.7302	78.5806
	AIFPIFSC2	85.6344	73.7498	72.4556	89.5333	72.3830	78.4516
NewThyroid	Fuzzy kNN	95.0388	90.7974	87.8857	95.0752	87.9835	92.5581
	FPFS-kNN	97.0853	96.7062	91.2889	95.7410	93.1413	95.6279
	FPFS-AC	96.3721	93.8027	90.9937	95.3598	91.9502	94.5581
	FPFS-EC	96.4961	93.7248	91.3365	95.5977	92.0784	94.7442
	FPFS-CMC	96.6202	93.8859	91.6921	95.7696	92.4125	94.9302
	IFPIFSC	97.6124	96.1328	94.4222	97.0730	94.9601	96.4186
	IFPIFS-HC	97.8605	97.0014	94.8095	97.2816	95.5104	96.7907
	AIFPIFSC1	98.0775	97.2697	95.1175	97.4475	95.8640	97.1163
	AIFPIFSC2	97.9845	97.1825	94.8000	97.2937	95.6615	96.9767
Vowel	Fuzzy kNN	99.2140	95.8955	95.6768	99.5677	95.6417	95.6768
	FPFS-kNN	99.3352	96.5957	96.3434	99.6343	96.3159	96.3434
	FPFS-AC	99.7337	98.6159	98.5354	99.8535	98.5312	98.5354
	FPFS-EC	99.6474	98.1818	98.0606	99.8061	98.0533	98.0606
	FPFS-CMC	99.6125	98.0247	97.8687	99.7869	97.8614	97.8687
	IFPIFSC	98.9513	94.5523	94.2323	99.4232	94.2318	94.2323
	IFPIFS-HC	99.2507	96.1076	95.8788	99.5879	95.8730	95.8788
	AIFPIFSC1	99.7906	98.9147	98.8485	99.8848	98.8459	98.8485
	AIFPIFSC2	99.7723	98.8142	98.7475	99.8747	98.7437	98.7475
Iris	Fuzzy kNN	97.3778	96.4257	96.0667	98.0333	96.0430	96.0667
	FPFS-kNN	97.5111	96.5657	96.2667	98.1333	96.2529	96.2667
	FPFS-AC	97.4222	96.4202	96.1333	98.0667	96.1199	96.1333
	FPFS-EC	97.4222	96.4202	96.1333	98.0667	96.1199	96.1333
	FPFS-CMC	97.3778	96.3697	96.0667	98.0333	96.0521	96.0667
	IFPIFSC	91.5111	88.0210	87.2667	93.6333	87.1189	87.2667
	IFPIFS-HC	92.8000	90.1718	89.2000	94.6000	89.0407	89.2000
	AIFPIFSC1	97.5111	96.5434	96.2667	98.1333	96.2543	96.2667
	AIFPIFSC2	97.4222	96.4337	96.1333	98.0667	96.1199	96.1333

Table 2 (continued)

Datasets	Classifiers	Acc	Pre	Rec	Spe	MacF	MicF
Seeds	Fuzzy kNN	90.4444	87.2996	85.6667	92.8333	85.4920	85.6667
	FPFS-kNN	92.8571	89.7479	89.2857	94.6429	89.1821	89.2857
	FPFS-AC	93.7143	91.0210	90.5714	95.2857	90.4056	90.5714
	FPFS-EC	93.1429	90.0824	89.7143	94.8571	89.5837	89.7143
	FPFS-CMC	93.3016	90.3377	89.9524	94.9762	89.8107	89.9524
	IFPIFSC	92.9524	89.7158	89.4286	94.7143	89.2777	89.4286
	IFPIFS-HC	92.7302	89.5061	89.0952	94.5476	88.8379	89.0952
	AIFPIFSC1	94.3175	91.7485	91.4762	95.7381	91.4237	91.4762
	AIFPIFSC2	94.7619	92.3747	92.1429	96.0714	92.1077	92.1429
Wireless	Fuzzy kNN	99.0725	98.1721	98.1450	99.3817	98.1485	98.1450
	FPFS-kNN	95.2525	90.6801	90.5050	96.8350	90.5397	90.5050
	FPFS-AC	95.5500	91.2345	91.1000	97.0333	91.0944	91.1000
	FPFS-EC	94.6725	89.4593	89.3450	96.4483	89.3541	89.3450
	FPFS-CMC	94.3875	88.8904	88.7750	96.2583	88.7865	88.7750
	IFPIFSC	98.2200	96.4930	96.4400	98.8133	96.4425	96.4400
	IFPIFS-HC	98.2675	96.6149	96.5350	98.8450	96.5404	96.5350
	AIFPIFSC1	99.0100	98.0371	98.0200	99.3400	98.0204	98.0200
	AIFPIFSC2	99.0875	98.1902	98.1750	99.3917	98.1746	98.1750
Wine	Fuzzy kNN	82.4910	74.0454	72.4131	86.5646	72.5250	73.7365
	FPFS-kNN	97.0063	95.6477	96.2381	97.8322	95.6542	95.5095
	FPFS-AC	95.4011	93.8981	94.1831	96.6168	93.3926	93.1016
	FPFS-EC	97.3090	96.1400	96.6000	98.0425	96.0995	95.9635
	FPFS-CMC	97.1947	95.9917	96.4635	97.9589	95.9366	95.7921
	IFPIFSC	98.0921	97.5585	97.6127	98.5773	97.3807	97.1381
	IFPIFS-HC	98.3915	97.8120	97.9841	98.8081	97.7592	97.5873
	AIFPIFSC1	98.1979	97.3359	97.7556	98.6979	97.4052	97.2968
	AIFPIFSC2	98.5005	97.7649	98.1333	98.9283	97.8170	97.7508
WholesaleR	Fuzzy kNN	64.5455	32.7706	32.1678	67.0562	36.7254	46.8182
	FPFS-kNN	77.2273	34.0997	32.5751	66.0077	56.2602	65.8409
	FPFS-AC	70.6061	33.4320	33.2949	66.6446	39.5025	55.9091
	FPFS-EC	70.5909	33.1335	33.1258	66.3717	39.7357	55.8864
	FPFS-CMC	70.5000	33.4180	33.2277	66.7968	39.1309	55.7500
	IFPIFSC	70.5455	33.4457	33.4324	65.9316	38.5804	55.8182
	IFPIFS-HC	70.1970	32.6805	32.7455	65.7756	39.0998	55.2955
	AIFPIFSC1	71.0303	35.9047	35.8277	67.0121	38.4634	56.5455
	AIFPIFSC2	71.0303	35.9170	35.9634	67.1598	38.1598	56.5455
Breast Tissue	Fuzzy kNN	84.3146	55.5706	51.1056	90.6527	57.0971	52.9437
	FPFS-kNN	88.8066	67.3580	65.0500	93.3213	69.8296	66.4199
	FPFS-AC	90.1833	70.8637	69.6167	94.1207	72.5957	70.5498
	FPFS-EC	88.6768	67.5787	64.8278	93.2355	71.9591	66.0303
	FPFS-CMC	88.3968	66.5644	63.6722	93.0684	70.6789	65.1905

(Continued)

Table 2 (continued)

Datasets	Classifiers	Acc	Pre	Rec	Spe	MacF	MicF
	IFPIFSC	89.2323	69.9230	67.0278	93.5696	68.1063	67.6970
	IFPIFS-HC	89.9942	71.4022	68.9444	94.0127	71.6861	69.9827
	AIFPIFSC1	89.7792	71.5443	68.5667	93.8846	72.6934	69.3377
	AIFPIFSC2	89.5613	71.2471	67.7611	93.7502	72.2527	68.6840
Led7Digit	Fuzzy kNN	92.8360	65.5106	64.3707	96.0143	63.8773	64.1800
	FPFS-kNN	92.3760	63.1691	62.1362	95.7535	63.2381	61.8800
	FPFS-AC	92.8320	65.4730	64.4296	96.0116	64.0844	64.1600
	FPFS-EC	92.8320	65.4730	64.4296	96.0116	64.0844	64.1600
	FPFS-CMC	92.8280	65.4508	64.4010	96.0094	64.0602	64.1400
	IFPIFSC	92.8320	65.4730	64.4296	96.0116	64.0844	64.1600
	IFPIFS-HC	92.8640	65.5359	64.5760	96.0297	64.1747	64.3200
	AIFPIFSC1	92.8480	65.3969	64.5455	96.0238	63.9295	64.2400
	AIFPIFSC2	92.9000	65.6258	64.8141	96.0524	64.1907	64.5000
Mean	Fuzzy kNN	91.1318	72.4892	70.2654	92.9899	74.6155	75.9551
	FPFS-kNN	93.8310	79.0775	75.8214	93.9765	82.0681	83.3420
	FPFS-AC	93.4057	80.1321	77.8002	94.2688	80.6779	83.2745
	FPFS-EC	93.2752	78.7916	76.6913	94.0175	80.6779	82.6885
	FPFS-CMC	93.1957	78.7142	76.5281	94.0077	80.1741	82.4181
	IFPIFSC	93.2512	78.6349	76.5182	94.2836	79.2002	81.9587
	IFPIFS-HC	93.4052	79.0292	76.9062	94.4012	79.8964	82.4427
	AIFPIFSC1	94.3023	82.5859	80.2558	95.0420	82.4851	85.0485
	AIFPIFSC2	94.3053	82.6093	80.1285	95.0217	82.3874	84.9823

Note:

The best performance results are shown in bold.

and “Wine” exceed the performance rates of 95%, 98%, 96%, 91%, 98%, and 97%, respectively. AIFPIFSC2’s results for the datasets “NewThyroid”, “Vowel”, “Iris”, “Seeds”, “Wireless”, and “Wine” exceed the performance rates of 94%, 98%, 96%, 92%, 98%, and 98%. Moreover, AIFPIFSC1 achieves the highest scores across all performance indicators in “Ecoli”, “Libras”, “HCV Data” (excluding Pre value), “Column3C” (excluding Pre value), “NewThyroid”, “Vowel”, and “Iris” (excluding Pre value). AIFPIFSC2 achieves the highest scores across all performance metrics in “Leaf” (excluding Pre value), “Seeds”, “Wireless”, “Wine” (excluding Pre value), “WholesaleR” (excluding Acc, MacF, and MicF value), and “Led7Digit”. As a result, the average performance outcomes presented in Table 2 suggest that AIFPIFSC1 and AIFPIFSC2 are more effective classifiers than other classifiers for the datasets under consideration.

Statistical analysis

This subsection conducts the Friedman test ([Friedman, 1940](#)), a non-parametric method, along with the Nemenyi test ([Nemenyi, 1963](#)), a *post-hoc* test, following the approach suggested by [Demsar \(2006\)](#). The process aims to evaluate all performance outcomes in

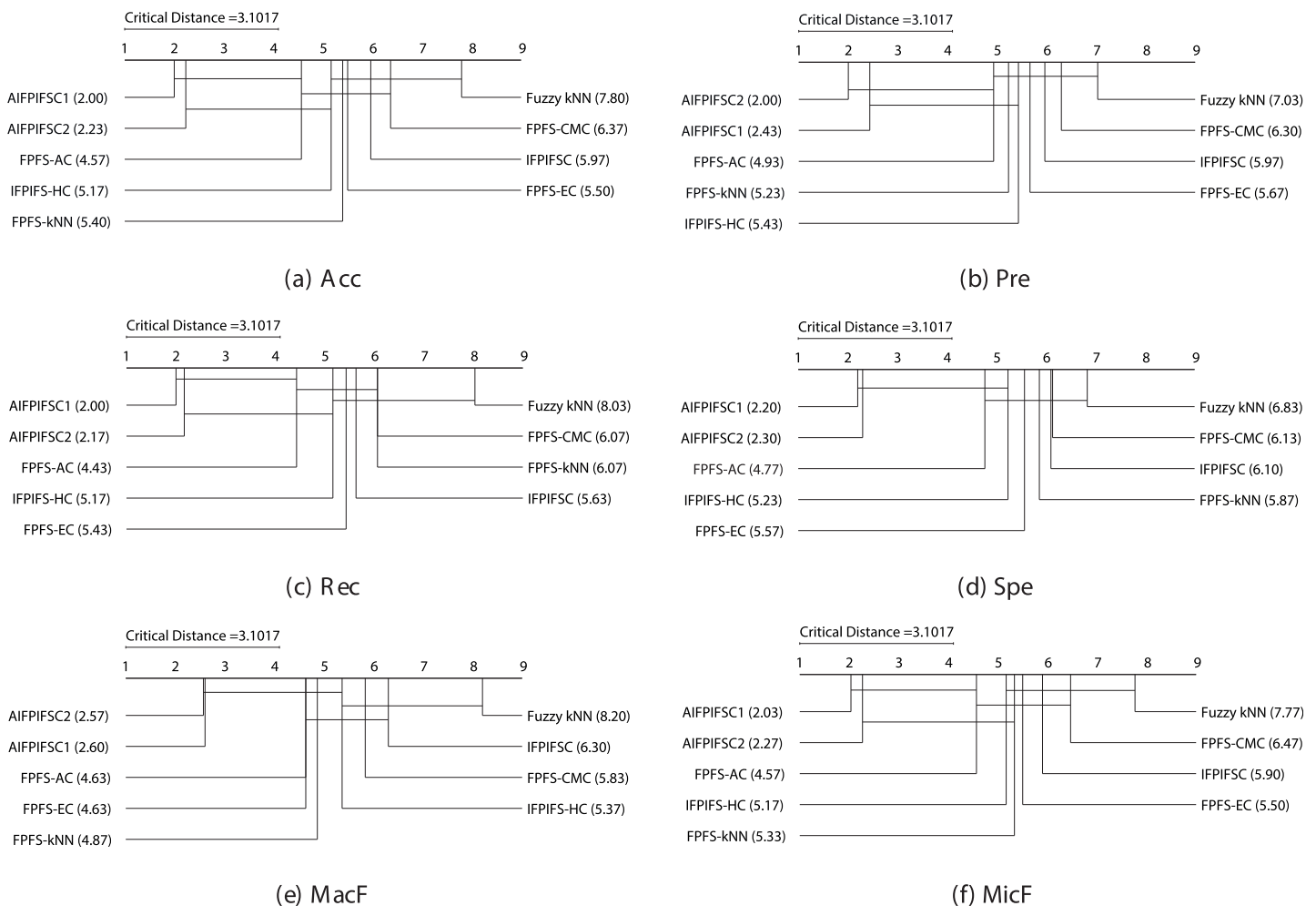


Figure 1 The essential diagrams resulting from the Friedman and Nemenyi tests at a significance level of 0.05 for the six performance criteria in the context of the classifiers mentioned above. Full-size [DOI: 10.7717/peerj-cs.2703/fig-1](https://doi.org/10.7717/peerj-cs.2703/fig-1)

terms of Acc, Pre, Rec, Spe, MacF, and MicF. The Friedman test creates a ranking based on performance for the classifiers across each dataset. Therefore, a rank of 1 indicates the top-performing classifier, followed by a rank of 2 for the next best, and so on. If classifiers exhibit identical performance, the test assigns them the mean of their potential ranks. Subsequently, it assesses the average ranks and computes χ_F^2 , which follows a distribution with $k - 1$ degrees of freedom, where k represents the total number of classifiers. A *post hoc* analysis, such as the Nemenyi test, is applied to identify significant differences among the classifiers. Differences between any pair of classifiers that exceed the critical distance are regarded as statistically significant.

In the statistical analysis, the average ranking for each classifier is computed using the Friedman test. Nine classifiers, which means $k = 9$, including AIFPIFSC1 and AIFPIFSC2, are compared concerning 15 datasets (denoted as $N = 15$) for each of the six performance criteria. The Friedman test yields the following statistics for Acc, Pre, Rec, Spe, MacF, and

MicF: $\chi_F^2 = 59.27$, $\chi_F^2 = 48.50$, $\chi_F^2 = 62.13$, $\chi_F^2 = 46.78$, $\chi_F^2 = 52.15$, and $\chi_F^2 = 57.99$, respectively.

With $k = 9$ and $N = 15$, the Friedman test indicates a critical value of 15.51 at a significance level of $\alpha = 0.05$ (for additional details, refer to [Zar \(2010\)](#)). As the Friedman test statistics for Acc (59.27), Pre (48.50), Rec (62.13), Spe (46.78), MacF (52.15), and MicF (57.99) surpass the critical value of 15.51, it implies a notable distinction in the performances of the classifiers. Consequently, the null hypothesis stating “There are no performance differences between the classifiers” is dismissed, allowing the application of the Nemenyi test. For $k = 9$, $N = 15$, and $\alpha = 0.05$, the critical distance is computed as 3.1017 following the Nemenyi test. [Figure 1](#) illustrates the crucial diagrams generated by the Nemenyi test for each of the six performance metrics.

[Figure 1](#) manifests that the performance differences between the average rankings of AIFPIFSC1 and those of IFPIFS-HC, FPFS-kNN, Fuzzy kNN, FPFS-CMC, IFPIFSC, and FPFS-EC, are more significant than the critical distance (3.1017). Besides, the performance differences between the average rankings of AIFPIFSC2 and those of FPFS-kNN, Fuzzy kNN, FPFS-CMC, IFPIFSC, and FPFS-EC are more significant than the critical distance (3.1017). [Figure 1](#) shows that although the difference between the mean rankings of AIFPIFSC1 and FPFS-AC, as well as AIFPIFSC2 and FPFS-AC and IFPIFS-HC is less than the critical distance (4.0798), AIFPIFSC1 and AIFPIFSC2 outperforms them in terms of all performance measures. Therefore, AIFPIFSC1 and AIFPIFSC2 outperform them in all performance metrics.

CONCLUSION AND FUTURE STUDIES

This study introduced two adaptive machine learning algorithms that concurrently utilize seven pseudo-similarities over *ifpifs*-matrices, applying them to a data classification problem. Besides, it employed two soft decision-making methods, constructed by *ifpifs*-matrices to improve the aforesaid machine learning approaches. In this study, the input values λ_1 and λ_2 , previously provided externally in the algorithm IFPIFSC, are determined adaptively based on the number of classes and instances. Implementing the *ifpifs*-matrices in the proposed algorithms resulted in a noticeable enhancement in classification Acc. The adaptive λ values played a crucial role in fine-tuning the classification process, relying on the specificities of each dataset. The adaptive nature of λ values in AIFPIFSC1 and AIFPIFSC2 proved pivotal in managing the intricacies and variabilities inherent in real-world data. This adaptability ensured the suggested algorithms remained robust and effective across diverse datasets and scenarios. The proposed algorithms are compared with well-known and state-of-the-art fuzzy/soft-set-based classifiers such as Fuzzy kNN, FPFS-kNN, FPFS-AC, FPFS-EC, IFPIFSC, and IFPIFS-HC. The performance results were utilized for a fair comparison and were statistically analyzed. Therefore, the current investigation demonstrates that the suggested approaches yield superior performance outcomes, making it a highly convenient method in supervised learning.

By dynamically adjusting lambda values and integrating *ifpifs*-matrices into two classification algorithms, this study contributes to the theoretical aspects of soft decision-making and machine learning. It provides a practical framework that can be employed in

Table 3 Comparison of the modeling ability and utilized concepts of the considered methods.

Classifier	Ref.	Crisp value	Fuzzy value	Int fuzzy value	Cla metric or sim	<i>fpfs</i> -p metric or sim	<i>ifpifs</i> -p metric or sim	<i>fpfs</i> DM	<i>ifpifs</i> DM	ADA
Fuzzy-kNN	Keller, Gray & Givens (1985)	✓	✗	✗	✓	✗	✗	✗	✗	✗
FPFS-kNN	Memiş, Enginoğlu & Erkan (2022b)	✓	✓	✗	✓	✓	✗	✗	✗	✗
FPFS-AC	Memiş, Enginoğlu & Erkan (2022c)	✓	✓	✗	✓	✓	✗	✓	✗	✗
FPFS-EC	Memiş, Enginoğlu & Erkan (2021)	✓	✓	✗	✓	✓	✗	✗	✗	✗
FPFS-CMC	Memiş, Enginoğlu & Erkan (2022a)	✓	✓	✗	✓	✓	✗	✓	✗	✗
IFPIFSC	Memiş et al. (2023)	✓	✓	✓	✓	✓	✓	✓	✗	✗
IFPIFS-HC	Memiş et al. (2021)	✓	✓	✓	✓	✓	✓	✓	✗	✗
AIFPIFSC1	Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓
AIFPIFSC2	Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note:

Int, Cla, Sim, *fpfs*-p, *ifpifs*-p, DM, and ADA represent intuitionistic, classical, similarity, *fpfs*-pseudo, *ifpifs*-pseudo, decision making, and adaptive, respectively.

various real-world applications. The adaptability and efficiency of the proposed methods make it a valuable addition to machine learning, especially in scenarios where data uncertainty and dynamism are predominant.

Although our proposed algorithms can classify problems with intuitionistic fuzzy values, no fuzzy or intuitionistic fuzzy uncertain data exists in standard machine learning databases. Therefore, to show that the mentioned algorithms can successfully work on data with intuitionistic fuzzy uncertainties, we subject the data in common databases to fuzzification and intuitionistic fuzzification processes, and then apply the proposed approaches. One of the most well-known examples of intuitionistic fuzzy uncertainties can be expressed to explain the usefulness of the methods: If a detector x emits ten signals per second and produces six positive and four negative signals, this situation can be represented by the fuzzy value $\mu(x) = 0.6$. Since intuitionistic fuzzy sets extend fuzzy sets, they can also model this scenario using the intuitionistic fuzzy values $\mu(x) = 0.6$ and $\nu(x) = 0.4$. However, suppose the detector records six positive, three negative, and one corrupted signal. In that case, this cannot be described using fuzzy values alone but can be represented using intuitionistic fuzzy values as $\mu(x) = 0.6$ and $\nu(x) = 0.3$. These examples illustrate the superiority of intuitionistic fuzzy sets over traditional fuzzy sets. Furthermore, soft sets are required to address the challenge of determining the optimal location for constructing a wind turbine when processing data from detectors at various locations. As seen in this example, problems with intuitionistic fuzzy uncertainties are among the types of uncertainties we can encounter daily. Classical machine learning methods cannot work on data with such uncertainty, but the proposed machine learning approaches can efficiently work on issues such as those presented in the study. Moreover, there is no data with such uncertainty in known databases. Therefore, to compare the classical methods with our proposed approaches, we can obtain the performance results of classical machine

learning methods and our proposed methods by converting these data into fuzzy and intuitionistic fuzzy values using the same classical data. The abilities and advantages of the proposed approach compared to the others can be summarized in Table 3.

While the proposed algorithms demonstrate significant improvements in classification Acc by effectively incorporating *ifpifs*-matrices and adaptive λ values, an aspect warranting further attention is their time complexity. Comparative analysis reveals that our robust and adaptable algorithms perform slower than some benchmarked classifiers. This observation underscores the need for optimization in computational efficiency. Future research could focus on refining the algorithmic structure to enhance processing speed without compromising the Acc gains observed. Potential avenues might include integrating more efficient data structures, algorithmic simplification, or parallel processing techniques. Addressing this time complexity issue is crucial for practical applications, especially in real-time data processing scenarios. Further exploration of the scalability of the proposed methods for handling larger datasets could also be a valuable direction for subsequent studies. Moreover, future works can extend this research by integrating other forms of fuzzy and soft sets, *e.g.*, bipolar soft sets (Mahmood, 2020) and picture fuzzy parameterized picture fuzzy soft sets (Memiş, 2023a) into machine learning (Li, 2024), further enhancing the capabilities and applications of adaptive machine learning methods based on soft decision-making.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Samet Memiş conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Ferhan Şola Erduran analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Hivda Aydoğan conceived and designed the experiments, performed the experiments, performed the computation work, prepared figures and/or tables, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The Ecoli dataset is available at UCI Machine Learning Repository: Nakai, K. (1996). Ecoli [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5388M>.

The Leaf dataset is available at UCI Machine Learning Repository: Silva, P. & Maral, A. (2013). Leaf [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C53C78>.

The Libras is available at UCI Machine Learning Repository: Dias, D., Peres, S., & Bscaro, H. (2009). Libras Movement [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5GC82>.

The Mice Protein Expression dataset is available at UCI Machine Learning Repository: Higuera, C., Gardiner, K., & Cios, K. (2015). Mice Protein Expression [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C50S3Z>.

The HCV Data is available at UCI Machine Learning Repository: Lichtinghagen, R., Klawonn, F., & Hoffmann, G. (2020). HCV data [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5D612>.

The Vertebral Column dataset is available at UCI Machine Learning Repository: Barreto, G. & Neto, A. (2005). Vertebral Column [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5K89B>.

The NewThyroid dataset is available at UCI Machine Learning Repository: Quinlan, R. (1986). Thyroid Disease [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5D010>.

The Connectionist Bench dataset is available at UCI Machine Learning Repository: Deterding, D., Niranjan, M., & Robinson, T. (1988). Connectionist Bench (Vowel Recognition-Deterding Data) [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C58P4S>.

The Iris dataset is available at UCI Machine Learning Repository: Fisher, R. (1936). Iris [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C56C76>.

The Seeds dataset is available at UCI Machine Learning Repository: <https://doi.org/10.24432/C5H30K>.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P., & Lukasik, S. (2010). Seeds [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5H30K>.

The Wireless dataset is available at UCI Machine Learning Repository: Bhatt, R. (2017). Wireless Indoor Localization [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C51880>.

The Wine dataset is available at UCI Machine Learning Repository: Aeberhard, S. & Forina, M. (1992). Wine [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5PC7J>.

The Wholesale customers dataset is available at UCI Machine Learning Repository: Cardoso, M. (2013). Wholesale customers [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5030X>.

The Breast Tissue dataset is available at UCI Machine Learning Repository: S, J. & Jossinet, J. (1996). Breast Tissue [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5P31H>.

The LED Display Domain dataset is available at UCI Machine Learning Repository: Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). LED Display Domain [Dataset]. UCI Machine Learning Repository. <https://doi.org/10.24432/C5FG61>.

REFERENCES

- Aeberhard S, Forina M. 1992. *Wine*. UCI Machine Learning Repository. DOI 10.24432/C5PC7J.
- Arslan B, Aydın T, Memiş S, Enginoğlu S. 2021. Generalisations of SDM methods in *fpfs*-matrices space to render them operable in *ifpifs*-matrices space and their application to performance ranking of the noise-removal filters. *Journal of New Theory* (36):88–116 DOI 10.53570/jnt.989335.
- Atanassov KT. 1986. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 20(1):87–96 DOI 10.1016/S0165-0114(86)80034-3.
- Barreto G, Neto A. 2005. *Vertebral column*. UCI Machine Learning Repository. DOI 10.24432/C5K89B.
- Bhatt R. 2017. *Wireless indoor localization*. UCI Machine Learning Repository. DOI 10.24432/C51880.
- Breiman L, Friedman JH, Olshen RA, Stone C. 1984. *LED display domain*. UCI Machine Learning Repository. DOI 10.24432/C5FG61.
- Bustince Sola H, Barrenechea Tartas E, Pagola Barrio M, Fernández Fernández FJ, Xu Z, Bedregal B, Montero J, Hagraş H, Herrera F, Baets Bd. 2016. A historical account of types of fuzzy sets and their relationships. *IEEE Transactions on Fuzzy Systems* 24(1):179–194 DOI 10.1109/TFUZZ.2015.2451692.
- Çağman N, Çıtak F, Enginoğlu S. 2010. Fuzzy parameterized fuzzy soft set theory and its applications. *Turkish Journal of Fuzzy Systems* 1(1):21–35.
- Çağman N, Çıtak F, Enginoğlu S. 2011. FP-soft set theory and its applications. *Annals of Fuzzy Mathematics and Informatics* 2(2):219–226.
- Cardoso M. 2013. *Wholesale customers*. UCI Machine Learning Repository. DOI 10.24432/C5030X.
- Charytanowicz M, Niewczas J, Kulczycki P, Kowalski P, Lukasik S. 2010. *Seeds*. UCI Machine Learning Repository. DOI 10.24432/C5H30K.
- Deli I, Çağman N. 2015. Intuitionistic fuzzy parameterized soft set theory and its decision making. *Applied Soft Computing* 28(3):109–113 DOI 10.1016/j.asoc.2014.11.053.
- Demsar J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7:1–30.
- Deterding D, Niranjana M, Robinson T. 1988. *Connectionist bench (Vowel recognition-deterding data)*. UCI Machine Learning Repository. DOI 10.24432/C58P4S.
- Dias D, Peres S, Bscaro H. 2009. *Libras movement*. UCI Machine Learning Repository. DOI 10.24432/C5GC82.
- El-Yagubi E, Salleh AR. 2013. Intuitionistic fuzzy parameterised fuzzy soft set. *Journal of Quality Measurement and Analysis* 9(2):73–81 DOI 10.3233/JIFS-169805.
- Enginoğlu S, Arslan B. 2020. Intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices and their application in decision-making. *Computational and Applied Mathematics* 39(4):325 DOI 10.1007/s40314-020-01325-1.
- Enginoğlu S, Çağman N. 2020. Fuzzy parameterized fuzzy soft matrices and their application in decision-making. *TWMS Journal of Applied and Engineering Mathematics* 10(4):1105–1115.
- Enginoğlu S, Öngel T. 2020. Configurations of several soft decision-making methods to operate in fuzzy parameterized fuzzy soft matrices space. *Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering* 21(1):58–71 DOI 10.18038/estubtda.562578.

- Enginoğlu S, Aydın T, Memiş S, Arslan B. 2021. Operability-oriented configurations of the soft decision-making methods proposed between 2013 and 2016 and their comparisons. *Journal of New Theory* (34):82–114.
- Enginoğlu S, Memiş S, Çağman N. 2019. A generalisation of fuzzy soft max-min decision-making method and its application to a performance-based value assignment in image denoising. *El-Cezeri Journal of Science and Engineering* 6(3):466–481 DOI 10.31202/ecjse.551487.
- Erkan U. 2021. A precise and stable machine learning algorithm: eigenvalue classification (EigenClass). *Neural Computing and Applications* 33(10):5381–5392 DOI 10.1007/s00521-020-05343-2.
- Fawcett T. 2006. An introduction to roc analysis. *Pattern Recognition Letters* 27(8):861–874 DOI 10.1016/j.patrec.2005.10.010.
- Fisher RA. 1936. *Iris*. UCI Machine Learning Repository. DOI 10.24432/C56C76.
- Fix E, Hodges JL. 1951. *Discriminatory analysis, nonparametric discrimination: consistency properties*. Texas, Randolph Field: USAF School of Aviation Medicine.
- Friedman M. 1940. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics* 11(1):86–92 DOI 10.1214/aoms/1177731944.
- Higuera C, Gardiner K, Cios K. 2015. Mice protein expression. UCI Machine Learning Repository. DOI 10.24432/C50S3Z.
- Hussain A, Ullah K, Pamucar D, Haleemzai I, Tatić D. 2023. Assessment of solar panel using multiattribute decision-making approach based on intuitionistic fuzzy aczel alsina heronian mean operator. *International Journal of Intelligent Systems* 2023(1):6268613 DOI 10.1155/2023/6268613.
- Karaaslan F. 2016. Intuitionistic fuzzy parameterized intuitionistic fuzzy soft sets with applications in decision making. *Annals of Fuzzy Mathematics and Informatics* 11(4):607–619 DOI 10.3233/IFS-2012-0601.
- Karakoç Ö, Memiş S, Sennaroglu B. 2024. A review of sustainable supplier selection with decision-making methods from 2018 to 2022. *Sustainability* 16(1):125 DOI 10.3390/su16010125.
- Keller JM, Gray MR, Givens JA. 1985. A fuzzy k-nearest neighbor algorithm. *IEEE Transactions on Systems, Man, and Cybernetics* 15(4):580–585 DOI 10.1109/TSMC.1985.6313426.
- Kelly M, Longjohn R, Nottingham K. 2024. The UCI Machine Learning Repository. Available at <https://archive.ics.uci.edu>.
- Lichtinghagen R, Klawonn F, Hoffmann G. 2020. *HCV data*. UCI Machine Learning Repository. DOI 10.24432/C5D612.
- Li D. 2024. A pre-averaged pseudo nearest neighbor classifier. *PeerJ Computer Science* 10(1):e2247 DOI 10.7717/peerj-cs.2247.
- Mahmood T. 2020. A novel approach towards bipolar soft sets and their applications. *Journal of Mathematics* 2020(1):4690808 DOI 10.1155/2020/4690808.
- Maji PK, Biswas R, Roy AR. 2001a. Fuzzy soft sets. *The Journal of Fuzzy Mathematics* 9(3):589–602.
- Maji PK, Biswas R, Roy AR. 2001b. Intuitionistic fuzzy soft sets. *The Journal of Fuzzy Mathematics* 9(3):677–692.
- Memiş S. 2023a. Picture fuzzy parameterized picture fuzzy soft sets and their application in a performance-based value assignment problem to salt-and-pepper noise removal filters. *International Journal of Fuzzy Systems* 25(7):2860–2875 DOI 10.1007/s40815-023-01547-5.

- Memiş S. 2023b.** Picture fuzzy soft matrices and application of their distance measures to supervised learning: picture fuzzy soft k-nearest neighbor (PFS-kNN). *Electronics* **12**(19):4129 DOI [10.3390/electronics12194129](https://doi.org/10.3390/electronics12194129).
- Memiş S, Enginoğlu S. 2019.** An application of fuzzy parameterized fuzzy soft matrices in data classification. In: Kılıç M, Özkan K, Karaboyacı M, Taşdelen K, Kandemir H, Beram A, eds. *International Conferences on Science and Technology*. Prizren, Kosovo: Natural Science and Technology ICONST-NST 2019, 68–77.
- Memiş S, Arslan B, Aydın T, Enginoğlu S, Camcı Ç. 2021.** A classification method based on hamming pseudo-similarity of intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices. *Journal of New Results in Science* **10**(2):59–76.
- Memiş S, Arslan B, Aydın T, Enginoğlu S, Camcı Ç. 2023.** Distance and similarity measures of intuitionistic fuzzy parameterized intuitionistic fuzzy soft matrices and their applications to data classification in supervised learning. *Axioms* **12**(5):463 DOI [10.3390/axioms12050463](https://doi.org/10.3390/axioms12050463).
- Memiş S, Enginoğlu S, Erkan U. 2021.** Numerical data classification via distance-based similarity measures of fuzzy parameterized fuzzy soft matrices. *IEEE Access* **9**:88583–88601 DOI [10.1109/ACCESS.2021.3089849](https://doi.org/10.1109/ACCESS.2021.3089849).
- Memiş S, Enginoğlu S, Erkan U. 2022a.** A classification method in machine learning based on soft decision-making via fuzzy parameterized fuzzy soft matrices. *Soft Computing* **26**(3):1165–1180 DOI [10.1007/s00500-021-06553-z](https://doi.org/10.1007/s00500-021-06553-z).
- Memiş S, Enginoğlu S, Erkan U. 2022b.** Fuzzy parameterized fuzzy soft k-nearest neighbor classifier. *Neurocomputing* **500**:351–378 DOI [10.1016/j.neucom.2022.05.041](https://doi.org/10.1016/j.neucom.2022.05.041).
- Memiş S, Enginoğlu S, Erkan U. 2022c.** A new classification method using soft decision-making based on an aggregation operator of fuzzy parameterized fuzzy soft matrices. *Turkish Journal of Electrical Engineering & Computer Sciences* **30**(3):871–890 DOI [10.55730/1300-0632.3816](https://doi.org/10.55730/1300-0632.3816).
- Molodtsov D. 1999.** Soft set theory-first results. *Computers and Mathematics with Applications* **37**(4–5):19–31 DOI [10.1016/S0898-1221\(99\)00056-5](https://doi.org/10.1016/S0898-1221(99)00056-5).
- Nakai K. 1996.** *Ecoli*. UCI Machine Learning Repository. DOI [10.24432/C5388M](https://doi.org/10.24432/C5388M).
- Nemenyi PB. 1963.** *Distribution-free multiple comparisons*. New Jersey: Princeton University.
- Nguyen T, Dang M, Luong A, Liew A, Liang T, McCall J. 2019.** Multi-label classification via incremental clustering on an evolving data stream. *Pattern Recognition* **95**(8):96–113 DOI [10.1016/j.patcog.2019.06.001](https://doi.org/10.1016/j.patcog.2019.06.001).
- Quinlan R. 1986.** *Thyroid disease*. UCI Machine Learning Repository. DOI [10.24432/C5D010](https://doi.org/10.24432/C5D010).
- Silva P, Maral A. 2013.** *Leaf*. UCI Machine Learning Repository. DOI [10.24432/C53C78](https://doi.org/10.24432/C53C78).
- JP S, Jossinet J. 1996.** *Breast Tissue*. UCI Machine Learning Repository. DOI [10.24432/C5P31H](https://doi.org/10.24432/C5P31H).
- Stone M. 1974.** Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)* **36**(2):111–147 DOI [10.1111/j.2517-6161.1974.tb00994.x](https://doi.org/10.1111/j.2517-6161.1974.tb00994.x).
- Zadeh LA. 1965.** Fuzzy sets. *Information and Control* **8**(3):338–353 DOI [10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
- Zar JH. 2010.** *Biostatistical analysis*. Fifth Edition. Upper Saddle River: Prentice Hall, 672.