Neurocomputing 606 (2024) 128368

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Time series features and fuzzy memberships combination for time series classification

Francisco J. Baldán [a,b,*], Luis Martínez [b]

[a] Department of Computer Science and Programming Languages, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Málaga, 29071 Málaga, Spain

[b] Department of Computer Science, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Jaén, 23071 Jaén, Spain

ARTICLE INFO

Communicated by M. Sato-Ilic

Keywords:
Time series
Fuzzy model
Interpretability
Feature-based
Classification

ABSTRACT

Time series classification is an increasingly attractive field with the appearance of new problems in an expanding digitalized world. Most of the proposals in the state-of-the-art have focused just on improving the results' performance, leaving interpretability on a secondary level. The available interpretable proposals do not provide competitive results, which is an issue to be addressed. This paper introduces a new fuzzy feature-based time series classification method, which joins the ability of time series features to capture essential information about the time series with Fuzzy logic. This proposal allows the fuzzy-based approach to incorporate global information about the behavior of time series in the membership calculation with the aim of improving the performance and interpretability of the results by using an interpretable classifier. The proposed method has been evaluated over the 112 state-of-the-art time series classification datasets from the UCR repository, and the results obtained show a better performance. Furthermore, the combination of time series features and fuzzy memberships has also increased the interpretability of final models.

1. Introduction

Nowadays, the world is highly interconnected. Cheaper integrated circuits and sensors have driven the IoT (Internet of Things) to unseen levels [1–3]. Data streams are continuously recorded over time for a wide range of applications: arrhythmia classification [4], predicting residential energy consumption [5], or anomalies detection in industry [6], among others. Data from multiple sources, increasingly complex problems, and new critical applications that require robust and explainable results are some of the challenges of the actual time series classification field. In order to address these challenges, new proposals focused on data processing over time are emerging every day [7–9].

Any magnitude recorded over time produces a specific data type known as a time series. Time series have special characteristics that must be considered during their processing [10,11]. The temporal relationship between all its points prevents typical tasks in other data types, such as reordering or filtering. The application of these techniques without considering the temporal component would destroy the contained time series information.

Traditionally, the time series analysis field consists of two main approaches [12,13]. (1) The traditional method, based on the processing of the original data considering the temporal relationships present within it [14]. (2) The second approach focuses on transforming the original time series into feature vectors representative of the time series itself, removing the time component of the problem [15,16]. The latter approach has proposals specifically focused on improving interpretability, but several proposals can also be found that provide good results for classification [7,17,18]. Currently, the best state-of-the-art time series classification algorithms focus on performance results, ignoring their interpretability. A clear example is the HIVE-COTE meta-ensemble [9], composed of classifiers from different fields, which obtains the best state-of-the-art results, but its interpretability is practically zero.

For some time now, work has been conducted on incorporating fuzzy logic into time series analysis. The fuzzy logic [19–21] allows for handling imprecision and uncertainty in data [22–25], which is helpful when the boundaries between different time series data categories are not clearly defined. Additionally, it enables the incorporation of the linguistic terms [26,27] and fuzzy rules [28,29] into the modeling process that supports and improves the interpretability of the final results. While there are currently proposals grounded in robust theoretical frameworks utilizing fuzzy logic that enable enhancements in outcomes achieved within particular time series-based problems [30,31], these have not been extrapolated to broader scenarios, as general time series

* Corresponding author at: Department of Computer Science and Programming Languages, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Málaga, 29071 Málaga, Spain.

E-mail addresses: fjbaldan@uma.es (F.J. Baldán), martin@ujaen.es (L. Martínez).

https://doi.org/10.1016/j.neucom.2024.128368
Received 13 October 2023; Received in revised form 2 June 2024; Accepted 10 August 2024
Available online 13 August 2024

classification problems because of the complexity involved. One of the main limitations when combining fuzzy logic and time series is the complexity of correctly representing and processing the temporal component in fuzzy models [32]. This limitation leads to uncompetitive results.

The combination of fuzzy tools with a feature-based approach would improve this area. Therefore, we propose PLAYTIME, an approach that combines fuzzy logic with a feature-based approach for time series classification problems to improve performance results and increase their interpretability. The main contributions of this paper, in comparison to previous proposals and existing similar approaches, are:

- PLAYTIME allows the incorporation of temporal information into fuzzy environments, where it is not usually correctly processed [32], through the inclusion of a well-known time series feature-based approach [7,8,16].
- The calculation of the fuzzy membership value considering the time component significantly improves the original performance results.
- The fuzzy- and time-features combination proposed in PLAYTIME allows for the inclusion of tree-based classifiers, which significantly improves classification performance compared with previous majority vote models [32].
- Improvement of interpretability of obtained results through multiple dimensions: it allows the inclusion of a highly interpretable classification model such as tree-based models, the nodes of the decision trees are composed of interpretable features related to the time series, and a significant reduction in the number of nodes of the final decision trees is obtained, simplifying the final model.

The remainder of this paper is organized as follows: Section 2 depicts the fuzzy- and feature-based time series classification state-of-the-art. In Section 3, the proposed method is explained. Section 4 contains the experimentation performed including the interpretability study. Finally, Section 5 concludes the paper.

2. Related works

In this section, we briefly analyze the state-of-the-art of time series classification with fuzzy- (Section 2.1) and feature-based (Section 2.2) approaches.

2.1. Fuzzy time series classification

Time series classification in fuzzy environments is a relatively unstudied field. Traditionally, fuzzy approaches to time series have focused on time series forecasting tasks [33]. Research on time series classification using fuzzy approaches tends to focus on specific applications in which extra interpretability is needed [34]. It is virtually impossible to find proposals [35,36] that test their performance on almost all the datasets included in the reference time series classification repository, the UCR respository [37], which contains 112 widely used datasets.

One of the pioneering methods in this field was proposed in [32], which focuses on obtaining basic statistics of the input time series at each time instant (maximum, minimum, and mean) and creating a membership profile based on these measures for each class of the problem in the training set. Next, the similarity score to each membership profile is calculated for each new input time series. Finally, the time series is classified as belonging to the class with the highest similarity. The main limitations of this approach are as follows:

- First, each point in the time series is evaluated independently of the rest, limiting the analysis of the temporal relationship between different points that compose the same time series. This approach has the same problem as any algorithm for calculating distances with static measurements: any desynchronization in the input time series makes it impossible to detect identical time series belonging to the same class, even if the desynchronization is minimal [14].
- Second, applying a final classification with a simple majority vote significantly limits the options for possible classification results.
- Third, the interpretability of the final results is limited.
- Finally, the results might not be considered robust because only six datasets were used in the final comparison.

In [34], a similar approach to the previous one is proposed for alcoholics and nonalcoholics detection tasks, although eight different statistics are used. They also introduced different classifiers in the final step with the aim of not relying on majority voting and focusing on obtaining the best possible classification results. In the same line, we can find a prototype-based supervised classification algorithm [38] that is able to adapt its architecture to improve the performance of the network. This proposal focuses on optimizing the fuzzy objective function to minimize the output error of the network.

If we look for proposals with a broader experimental section, we can find those that focus on fuzzy cognitive maps. The basic idea behind this type of proposal is that the classifier is able to differentiate the maps constructed from time series belonging to different classes. A first work [35] demonstrated the correct performance of this idea with experimentation carried out on 25 datasets belonging to the UCR repository. Subsequent work [36] included a spatiotemporal feature extraction block that significantly improved the results obtained using 26 UCR datasets. In both proposals, the interpretability of the results is a secondary aspect because no specific studies on interpretability are included.

2.2. Feature-based time series classification

Traditional time series classification tasks require specific models that consider the time component. The feature-based approach arises in order to facilitate the processing of this type of problem by incorporating all vector-based classification algorithms into the time series field. The main advantage of this approach is the possibility of including time information in feature vectors that can be processed by models that do not consider time information in a natural way.

In the feature-based approach, we find two different sub-approaches: the automatic feature extraction process and the features proposed with a strong theoretical basis:

- The automatic feature extraction proposals are usually oriented to obtain the best performance results possible and reduce the computational complexity of traditional time series classification algorithms [15]. We can find proposals like catch22 [39] that start from thousands of different non-specific time series features and optimize the number of final features used, seeking to maintain the best accuracy possible over a specific set of datasets. The Shapelet Transform (ST) [40] is another type of feature-based proposal in which the extracted features are subsequences of the original time series. These subsequences are discriminative enough to be able to differentiate between time series from different classes. In this case, the minimum distance from the subsequence to the time series is used as the input feature to the classifier. The proposals based on Shapelets improve the interpretability of the decision process by relating specific time series patterns to their classification. Another recent proposal that has shown high-performance results is ROCKET [8], which is based on the use of random convolutional kernels. The features extracted, the feature map, for ROCKET are composed of two different subfeatures: the maximum value (equivalent to global max pooling) and the proportion of positive values. Due to the high performance of random convolutional kernels for time series classification tasks, proposals based on shapelets have recently emerged, including fundamental parts of the convolutional kernel approaches, like the notion of dilation. An example of this type of proposal is the Random Dilated Shapelet Transform (RDST) [7], which provides three different features from each extracted shapelet: the minimum distance between the shapelet and the time series, the minimum distance point/location, and the number of shapelet occurrences in the time series. Including additional features that add information to the main extracted feature is an approach that provides significant performance improvement in the final classification.

- The proposals with a fixed number of time series features chosen for their strong theoretical basis are focused on increasing the interpretability of final results [16]. The interpretable concepts extracted through the proposed features allow the users to correlate the final classification performed with the observed specific time series behaviors. The use of an interpretable classifier is necessary. These types of proposals are able to provide competitive results concerning the state-of-the-art time series classification algorithms if more complex and less interpretable classifiers are used [17,18]. The representativeness of the characteristics used in these approaches is so high that they can even be used to generate synthetic time series with realistic behaviors [41].

In summary, time series classification fuzzy approach proposals consider the time component in a limited way. These proposals are quite close to the traditional vector-based classifiers. As our objectives are to increase the interpretability and performance of the final results, the feature-based approach has proven to be the best way to achieve this goal.

3. PLAYTIME. An interpretable Fuzzy Feature-based time series classification approach

This work presents a combination of the feature-based approach with a fuzzy time series classification method to increase the competitivity and interpretability of the fuzzy approach in time series classification problems. In this section, three different mechanisms are presented. Each of them is capable of improving the final results on its own, but the combination of all the mechanisms provides the best possible results and better interpretability:

1. Transform the original input time series (TS) into feature vectors based on the feature-based approach. This approach collects the time information of interest in independent features and removes the existing temporal relationship between the different input points of each time series. This is necessary because traditional vector-based methods, such as the fuzzy time series classification method, cannot correctly process this temporal relationship.
2. Replace the classification mechanism that only considers the highest class membership value as the final classification class for a decision tree. This type of classifier enables us to make more complex decisions and obtain more deeply interpretable results, especially if we combine membership with other types of features.
3. Combine the class fuzzy membership values of each time series with their extracted features to form the input dataset of the proposed classification tree. In this way, our proposal provides an enhanced and more interpretable classification.

An in-depth explanation of the PLAYTIME proposal is provided below. Fig. 1 shows its workflow and Algorithm 1 shows the pseudocode of PLAYTIME, realized with Python-oriented nomenclature.

The **first step** (Fig. 1.1) is focused on transforming the original time series dataset into a traditional vector-based dataset composed of discriminative time series features:

- The proposed method starts from the original time series (Fig. 1.A).

Algorithm 1 PLAYTIME procedure

```
Input:
    inputData: input pandas DataFrame with (tsID, classID,
    tsData) structure
Output:
    outputData: output pandas DataFrame that contains the
    time series features calculated and class membership
    coefficients for each input time series with (tsID, class,
    mem_time_features) structure
1: tsFeatures ← tsFeaturesExtraction(inputData)
2: profileData ← []
3: for each feature in tsFeatures do
4:     varData ← tsFeatures.filter(featureID==feature)
5:     for each class in tsFeatures do
6:         featClassData ← varData.filter(classID==class)
7:         profileData[feature, class] ← featClassData.obtain([max, min,
   mean])
8:     end for
9: end for
10: memCoef ← []
11: for each tsID in tsFeatures do
12:     memCoef[tsID] ← tsFeatures[tsID].memCal(profileData[featureID,
    classID])
13: end for
14: outputData ← tsFeatures.join(memCoef, tsID)
15: return (outputData)
```

- The selected feature extraction mechanism is applied to these data (Fig. 1.1).
- The feature-based representation of the input time series, a traditional vector-based dataset, is obtained (Fig. 1.B) (line 1).

The **second step** (Fig. 1.2) obtains a class profile. In this way, a profile of the behavior of each feature for each class is obtained following the process indicated in the training phase of [32]. The profiles obtained allow us to link the behaviors of each feature to a specific class of the problem (Fig. 1.C):

- The list containing the profile information of each characteristic for each class is initialized (line 2).
- The profile information is calculated (lines 3–9) by analyzing each feature independently (line 4).
- Additionally, the profile information of each feature is associated with each class (lines 5–8).
- Then, the maximum, minimum, and mean values are calculated for each feature and class combination. The data belonging to the analyzed class are selected (line 6), and their profile information is calculated (line 7): maximum, minimum, and average values of a feature in instances of the same class.

The **third step** (Fig. 1.3) processes the class profile information with a simple fuzzy approach to obtaining the class membership coefficient of each input time series (Fig. 1.D) following the process indicated in the classification phase of the [32] but without performing the final classification:

- Once the profile information has been calculated, the membership list of each instance to each class is initialized (line 10) and processed (lines 11–13).
- For each instance, the fuzzy membership of each feature is calculated for each class using the value of the feature and their respective profile information obtained for each class [32].

Finally, in the **last step** (Fig. 1.4), the extracted time series features and the computed class membership coefficients are combined in their respective instances to produce the final dataset (Fig. 1.E):

- The extracted features and the memberships are combined using the original *tsID*, generating the final output dataset (line 14).

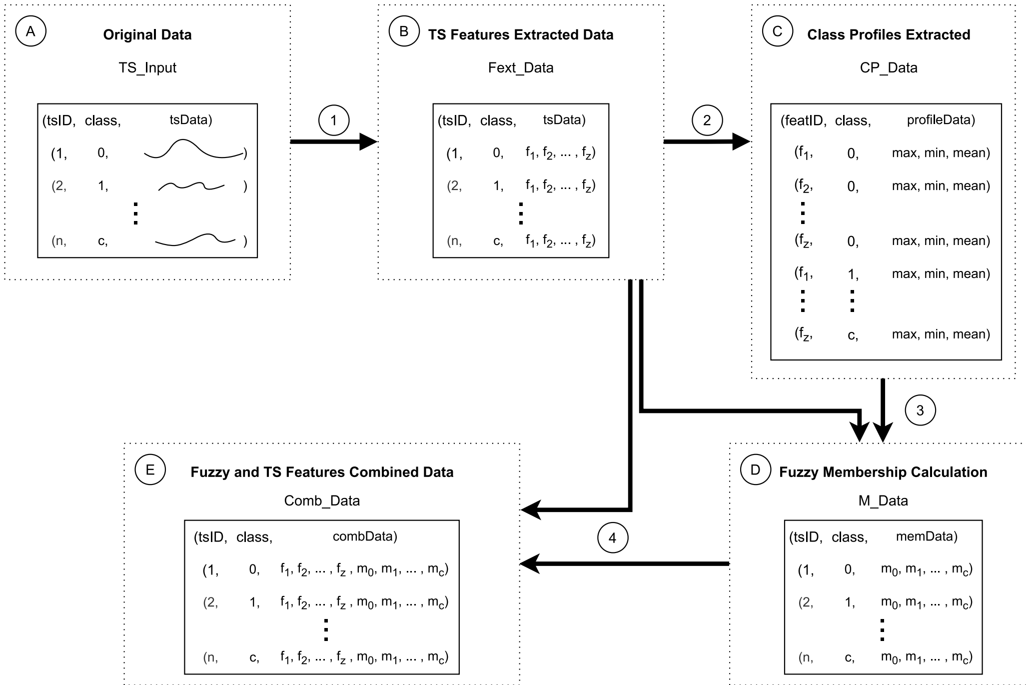

Fig. 1. PLAYTIME schema. The multiple steps of the proposal are differentiated numerically. The data structures are differentiated by letters.

- The selected machine learning model is applied to this dataset. The process commented on until now is the training process. The test set processing is identical but uses the class profiles extracted in the training phase (Fig. 1.C). Therefore, the steps in lines 3–9 would be ignored since the *profileData* generated on the training set is used.

Once the transformed training and test data have been obtained, any traditional vector-based classification algorithm can be applied to address the problem.

4. Experimentation, evaluation and analysis

Since the time series classification state-of-the-art is currently dominated by competitive proposals in terms of performance but leaves the interpretability of the results in second place, the need for models that focus on both objectives is increasing. The aim of this section is to demonstrate that the proposed approach improves the performance and interpretability of the results obtained by the traditional fuzzy vector-based approach.

In this section, the performance and interpretability of our proposal are evaluated. Section 4.1 contains the experimentation setup used, with in-depth selection criterion explanations. In Section 4.2, the results obtained are analyzed.

The source code of our proposal and experimentation has been developed in Python 3.10.10 and is available in the online repository[1]. Additionally, R 4.2.2 has been used for the analysis of the results.

4.1. Experimentation setup

In this section, we motivate the decisions realized in the experimentation carried out. Section 4.1.1 includes the models and parameters selected. In Section 4.1.2, the selection of the dataset performed is explained. Section 4.1.3 includes the performance metrics used.

4.1.1. Models

The main models of the state-of-the-art time series classification of the featured-based approach have been selected. On the one hand, we have chosen models with a feature set defined as CMFTS [16]. On the other hand, we have chosen models that implement an automatic feature generation ROCKET [8] and RDST [7]. The publicly available code[1] has been used to calculate the features in all three cases. Because the number of final features extracted by ROCKET and RDST is a model parameter, we arbitrarily chose to extract 60 features for each time series. This number is slightly higher than the 55 offered by CMFTS. This selection aims to avoid benefiting the proposal with a fixed number of features. Each feature extracted by ROCKET has two components, generating an output dataset of 120 features. RDST generates three components per feature, generating an output dataset of 180 features. Due to the final difference in dimensions of the training and test datasets, in addition to the experimentation proposed up to now, the importance of the number of features in the final results has been analyzed for RDST, which is the best state-of-the-art algorithm among the selected ones [7].

In addition, it has been necessary to fully implement the membership calculation following the indications given by the authors because the code is not available.

4.1.2. Datasets

In order to obtain robust results, the main repository of state-of-the-art datasets for time series classification, the UCR repository [37], has been selected. This repository contains 112 datasets widely used in the univariate time series classification field. The training and test sets are specified in the original repository and are mainly used for the final comparisons.

4.1.3. Metrics and evaluations criteria

Due to the large number of datasets used and their different characteristics, it is necessary to use multiple mechanisms to perform the final comparison. The accuracy measure has been selected as the basic comparison measure because of its predominance in experimental studies in state-of-the-art proposals [9,37]. It is a measure of direct comparison normally accompanied by other types of measures, also included in this

[1] Fuzzy Feature-based Time Series Classification repository. https://github.com/fjbaldan/PLAYTIME.

Table 1
Performance results using majority voting. The best cases by columns are highlighted in bold.

Input data (num. features)	Avg. accuracy	Avg. rank	Win/Loss ratio
CMFTS (55)	59.64	5.085	14/98
fuzzyTsValues (ts lengths)	57.324	5.263	19/93
RDST (180)	**68.198**	**2.839**	**50/62**
RDST (6)	48.286	7.643	2/110
RDST (12)	55.767	6.339	5/107
RDST (18)	59.253	5.192	3/109
RDST (24)	61.152	4.549	1/111
RDST (30)	62.389	4.107	8/104
ROCKET (120)	60.823	3.982	25/87

Table 2
Performance results using a decision-tree classifier. The best cases by columns are highlighted in bold.

Input data (num. features)	Avg. accuracy	Avg. rank	Win/Loss ratio
CMFTS (55)	60.087	4.924	20/92
fuzzyTsValues (ts lengths)	61.758	4.237	31/81
RDST (180)	**65.372**	**3.295**	32/80
RDST (6)	49.554	7.42	6/106
RDST (12)	54.507	6.674	6/106
RDST (18)	57.826	5.513	6/106
RDST (24)	59.564	5.085	8/104
RDST (30)	61.173	4.366	16/96
ROCKET (120)	65.271	3.487	**41/71**

work, which allow for a more in-depth analysis. Accuracy expresses the fraction of correct classifications performed over the total number of classifications.

$$accuracy = \frac{number\ of\ correct\ classifications}{number\ of\ total\ classifications}$$

Considering the large variety in the processed datasets, the average rank measure and the win/loss ratio measures have been included, adding relative performance comparison measures. In addition, the Critical Difference diagram (CD) [42] has been selected to perform a comparison from a statistical point of view. The CD diagram allows easy graphical comparison between different models for a large number of datasets. In this diagram, the models are ranked according to their average positions, which are displayed along the top line. The *CD* included in the Critical Difference diagrams, shown at the top of the graph, is a statistical threshold that determines when the difference in performance between two algorithms is statistically significant. If the difference in performance between two different algorithms is greater than the *CD*, it is considered that there is a statistically significant difference between them. In contrast, the models linked by a bold line are considered to have no statistically significant differences in their results at a specified confidence level α. For statistical comparisons, a confidence level of 95% has been chosen, setting the α parameter to 0.05. The average rank and CD were obtained from the R *scmamp* package.

4.2. Results

This section presents and assesses the performance (Section 4.2.1) and interpretability (Section 4.2.2) of our proposed method. It is important to point out that CMFTS, RDST, and ROCKET are different PLAYTIME applications with different time series feature extraction processes. For this reason, in the rest of the work, CMFTS, RDST, and ROCKET are the results associated with our PLAYTIME proposal. The case fuzzyTSValues represents the original proposal [32] to be improved and with which we have compared ours.

4.2.1. Performance results

First, the performance of the proposed feature-based approach is tested. For this purpose, the results obtained from the fuzzy approach processing the original time series and different time series feature-based approaches using majority voting are compared in Table 1.

The base approach (fuzzyTsValues), which uses the original time series values, is outperformed by all feature-based approaches proposed with a minimum of features. It is important to note that feature-based approaches usually require a minimum number of features to obtain competitive results, especially in cases of automatic extraction. RDST, with 180 features, obtains the best results over the 112 datasets processed, maintaining its position as the best transformation in the state-of-the-art time series classification. We can appreciate that the multiple reduced versions of RDST are able to obtain the best results in some cases. Moreover, the versions with 24 features or more are able to provide the best results in accuracy and rank after RDST (180). ROCKET, with 120 features, provides slightly better results than CMFTS, which has only 55 features and is an interpretability-oriented approach. These results are analyzed in-depth from a statistical point of view in Fig. 2.

As can be seen in Fig. 2, RDST (180) stands out from the rest of the models. ROCKET (120), RDST (30), RDST (24), and CMFTS (55) are in the second performance bracket. RDST(12), RDST (6), and fuzzyTsValues compose the last group. These results show that the feature-based approach with sufficient features significantly improves the base idea.

It is important to note that the number of final features of each proposal has a significant impact on the final results. Each RDST and ROCKET feature is composed of three and two different components, respectively. Therefore, the final classification has a larger number of inputs. RDST has robustly demonstrated its competitiveness; however, we would expect a significant reduction in the performance of ROCKET's results.

Second, to assess the final classifier change in the fuzzy approach, the performance of all the methods in Table 1 was tested by changing the majority vote to a tree-based classifier with its default configuration. The classification tree is run on the membership values obtained for each input feature vector or time series. The results are shown in Table 2.

Analyzing Table 2 and comparing it to Table 1, we can see that the inclusion of the decision tree as a classifier provides significant performance improvements to ROCKET, fuzzyTsValues, and CMFTS approaches. For CMFTS, this behavior can be clearly seen in its Win/Loss ratio, while ROCKET and fuzzyTsValues show improvements in all metrics. On the other hand, almost all cases of RDST suffer performance losses. This behavior indicates that RDST proposals would benefit from the base case of applying majority voting. Additionally, RDST (180) has suffered significant losses in the Win/Loss ratio compared with the previous case, although it remains the best proposal in terms of accuracy and rank. These performance declines are attributed to the multiple versions of RDST included in this experimentation, which have reduced performances in accuracy and rank but have shown significant improvements in their Win/Loss ratios. These results are further analyzed from a statistical point of view in Fig. 3".

In this case, RDST (180) does not stand out from the rest of the models and shares a performance bracket with ROCKET (120), fuzzyTsValues, and RDST(30). These three models have significantly improved their ranking. CMFTS, although improving its ranking, would not enter the first performance bracket.

Finally, the incorporation of the membership information into the feature datasets, or time series values in its case, is checked. For this purpose, the performance obtained by a decision tree-based classifier is compared for each model, where the input data consists of time series features or values, membership values, or a combination of both. Due to the high number of possible combinations and in order to facilitate the visualization of the results, this study has focused on the four main models evaluated in this work: RDST (180), ROCKET

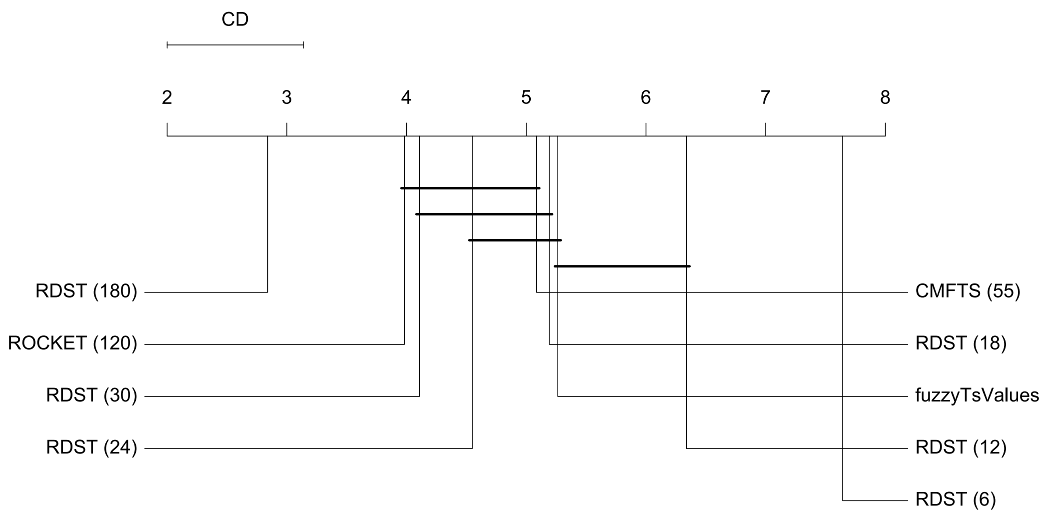

Fig. 2. CD for results shown in Table 1. Confidence level of 95%.

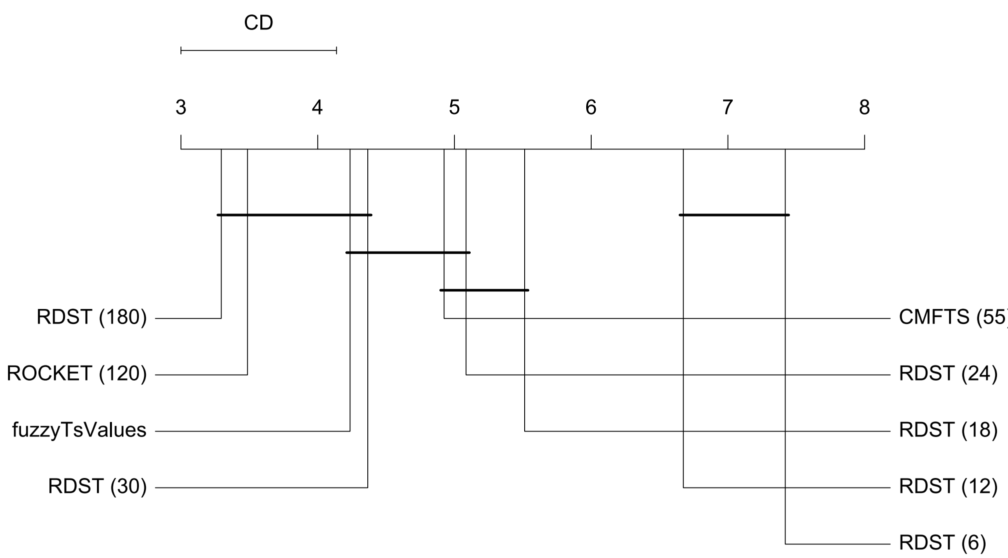

Fig. 3. CD for results shown in Table 2. Confidence level of 95%.

(120), CMFTS (55), and fuzzyTsValues. The performance results are shown in Table 3. It is important to consider that in this case, the CMFTS_comb, RDST_comb, and ROCKET_comb cases are performed with the full PLAYTIME method since they combine the extracted time series features with the membership values of the fuzzy approach.

Table 3 shows that the models using the combination of features with membership provide, in almost all cases, the best results or a very small difference with respect to the best case. ROCKET and the proposed combination obtain the best results, followed very closely by RDST. The use of membership alone provides the worst accuracy and rank results in the four proposed models. This behavior is logical since there would only be as many features as classes in the problem. In this way, in a significant part of the datasets, the input data to the decision tree would not be sufficiently diverse to provide competitive results. The cost associated with the inclusion of membership is low, especially if the number of features included in the final dataset is evaluated. Fig. 4 includes the CD of the Table 3 results.

Fig. 4 shows a clear difference between the fuzzyTsValues approach and the proposed feature-based approach. Furthermore, it is confirmed that the use of membership alone is not able to provide competitive results. The combination of features with membership values provided performance improvement in all cases, except for CMFTS. In this case, the exclusive use of time series features obtains better average performance than the combination of these with the membership values,

Table 3
Performance results using decision-tree classifier with different input data: time series features or values (feat), membership values (mem), or the combination of both (comb). The best cases by columns are highlighted in bold. The best result for each method is shaded in blue.

Input data (num. features)	Avg. accuracy	Avg. rank	Win/Loss ratio
CMFTS_feat (55)	70.375	5.353	15/97
CMFTS_mem (55)	60.087	9.406	4/108
CMFTS_comb (55)	70.509	5.384	15/97
fuzzyTsValues_feat	63.9	7.808	10/102
fuzzyTsValues_mem	61.758	8.281	10/102
fuzzyTsValues_comb	65.504	7.196	9/103
RDST_feat (180)	70.686	5.344	18/94
RDST_mem (180)	65.372	7.54	11/101
RDST_comb (180)	71.613	5.031	19/93
ROCKET_feat (120)	71.387	4.759	16/96
ROCKET_mem (120)	65.271	7.205	11/101
ROCKET_comb (120)	**71.712**	**4.692**	**22/90**

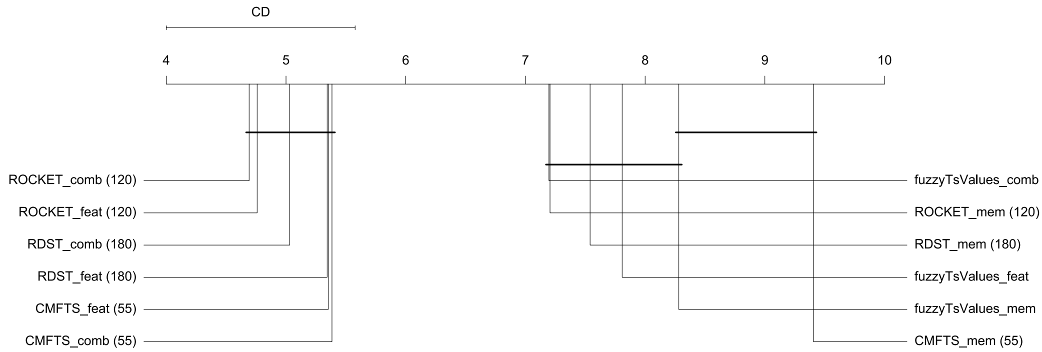

Fig. 4. CD for results shown in Table 3. Confidence level of 95%.

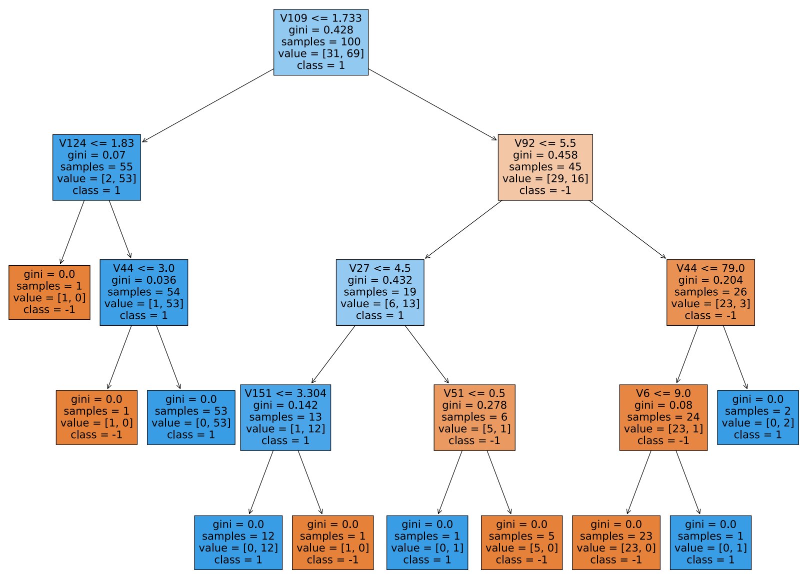

Fig. 5. Decission-tree obtained for ECG200 dataset processing only the RDST extracted features.

although the difference is minimal. The results presented in this section demonstrate the effectiveness of the proposed feature-based approach.

4.2.2. Interpretability

The interpretability of the results of a problem is closely related to the input data and the applied model. In the original case of fuzzyTsValues, majority voting limits the information obtainability of the problem. In the proposal realized, the combination of a decision tree with interpretable features of the time series offers an enormous number of options. Shapelets are one of the main approaches for improving the interpretability of the results. Therefore, this section compares the performance and classification tree obtained using the original proposal and the proposed combination.

Fig. 5 shows the decision tree obtained using only the features extracted by RDST. The tree is composed of nine nodes that have features related to the extracted Shapelets and are able to offer an accuracy of 74%.

Based on the proposed approach, Fig. 6 shows the decision tree obtained from the RDST features and membership values combination. This last tree contains seven nodes, is composed of time series features and membership values, and offers 87% accuracy. If we compare both trees, we can appreciate that the tree of the proposed combination has reduced dimensions. Additionally, its first node includes a membership feature obtaining a simpler decision tree with higher performance. A class membership value of −1 less than 0.5 allows us to correctly separate 61 of 69 instances of class 1 from the rest. This behavior shows the correctness of using time series features instead of time series values, which are subject to multiple problems in the membership calculation.

The results obtained for ECG200 are a clear example of the importance of including membership values in the classification process. These values play an important role in the classification performed, simplifying and improving the interpretability of the decision tree. Moreover, they are able to increase the performance of the obtained model, even if it is simpler.

Based on the analyzed example, we can see:

- First, the original proposal [32] performs the final classification by selecting the class with the highest cumulative membership score. In a three-class problem, the final classifier and the user will only see three membership scores for each instance to be classified, and the one with the highest score will be chosen as the final class. In that case, it is impossible to analyze the contribution of each point in the time series to the final decision since you only see one final cumulative value per class.
- Second, PLAYTIME allows the use of any classifier on the final set of features and memberships obtained, being able to use classifiers as interpretable as a simple classification tree, which shows the importance of each feature or membership score in the final classification obtained. This is not available in the original proposal. In addition, each node can be related to a specific feature extracted from the time series, allowing us to associate behaviors (CMFTS), specific patterns (RDST or ROCKET), or values (fuzzy membership) to a specific class.

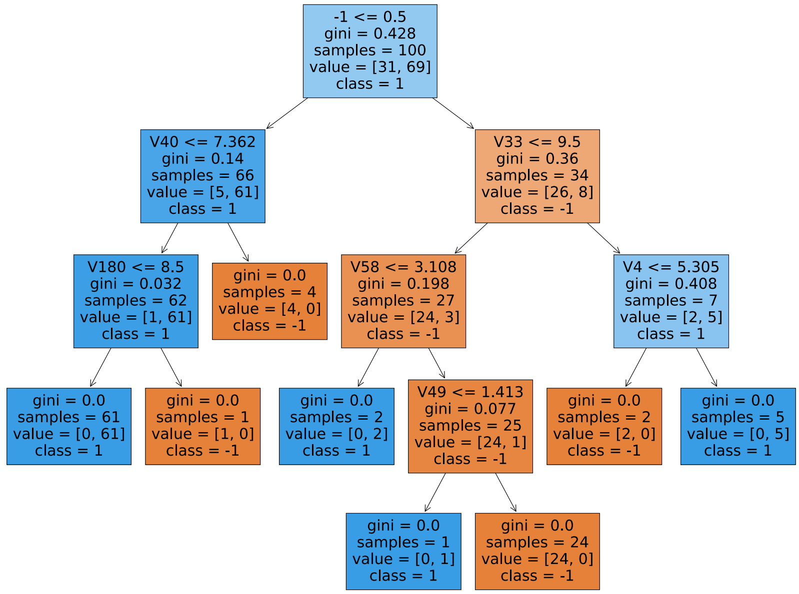

Fig. 6. Decission-tree obtained for ECG200 dataset processing the combination of the RDST extracted features and membership values.

- Finally, it has been shown how the use of membership values can simplify the final decision trees, improving their interpretability, while the original method only provides a single numerical value associated with a class. In addition, the original proposal obtains these values without considering the time component of the original time series.

5. Conclusions

In this work, we have presented a new Fuzzy Feature-based time series classification method. This proposal combines the fuzzy membership extraction approach with the global time series information collected by the feature-based approach. Moreover, it improves the competitiveness and interpretability of the results by combining both types of features and including an interpretable tree-based classifier.

In order to obtain robust results, the proposed method has been tested on the 112 reference datasets of the UCR repository, the state-of-the-art dataset repository for time series classification. The proposal code has been published to facilitate the reproducibility of the results.

Our proposal has shown a significant improvement in the performance of the results obtained by the traditional fuzzy approach. The feature-based approach has been proven to improve the results of the traditional fuzzy approach. The classification tree inclusion provides better performance results than majority voting methods whenever a minimum number of features are involved. The combination of time series features with fuzzy membership features has provided the best results. In addition, the interpretability of the models obtained has been improved.

The observed limitations of the proposed model are related to its interpretability, which is strongly associated with the interpretability of the final classification algorithm and the extracted features. The proposed method is limited to univariate time series. To extend it to multivariate time series problems, the inclusion of multivariate fuzzy membership functions or features is necessary.

The results obtained allow open new lines of research for the fuzzy approach for time series classification as new types of membership calculations adapted to the feature-based approach, the extraction of fuzzy rules from well-defined problems to improve the interpretability of results, or fuzzy proposals to work on multivariate time series, among others.

CRediT authorship contribution statement

Francisco J. Baldán: Conceptualization, Data curation, Methodology, Software, Validation, Writing – original draft. **Luis Martínez:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data are public. The code has been shared via a link to GitHub in the paper.

Acknowledgments

F.J. Baldán was supported by grant FJC2021-047112-I funded by MICIU/AEI/10.13039/501100011033 and by European Union NextGenerationEU/PRTR. Funding for open access charge: Universidad de Málaga / CBUA.

References

[1] H. Ahmadi, G. Arji, L. Shahmoradi, R. Safdari, M. Nilashi, M. Alizadeh, The application of internet of things in healthcare: a systematic literature review and classification, Univ. Access Inf. Soc. 18 (2019) 837–869.
[2] F. Chen, P. Deng, J. Wan, D. Zhang, A.V. Vasilakos, X. Rong, Data mining for the internet of things: literature review and challenges, Int. J. Distrib. Sens. Netw. 11 (8) (2015) 431047.
[3] Y. He, J. Guo, X. Zheng, From surveillance to digital twin: Challenges and recent advances of signal processing for industrial internet of things, IEEE Signal Process. Mag. 35 (5) (2018) 120–129.
[4] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, C. Wu, Arrhythmia classification of LSTM autoencoder based on time series anomaly detection, Biomed. Signal Process. Control 71 (2022) 103228.
[5] T.-Y. Kim, S.-B. Cho, Predicting residential energy consumption using CNN-LSTM neural networks, Energy 182 (2019) 72–81.
[6] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M.S. Hossain, Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach, IEEE Internet Things J. 8 (8) (2020) 6348–6358.

[7] A. Guillaume, C. Vrain, W. Elloumi, Random dilated shapelet transform: A new approach for time series shapelets, in: Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part I, Springer, 2022, pp. 653–664.
[8] A. Dempster, F. Petitjean, G.I. Webb, ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels, Data Min. Knowl. Discov. 34 (5) (2020) 1454–1495.
[9] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, A. Bagnall, HIVE-COTE 2.0: a new meta ensemble for time series classification, Mach. Learn. 110 (11–12) (2021) 3211–3243.
[10] R. Prado, M.A. Ferreira, M. West, Time Series: Modeling, Computation, and Inference, CRC Press, 2021.
[11] M.B. Shrestha, G.R. Bhatta, Selecting appropriate methodological framework for time series data analysis, J. Finance Data Sci. 4 (2) (2018) 71–89.
[12] Z. Xing, J. Pei, E. Keogh, A brief survey on sequence classification, ACM SIGKDD Explor. Newslett. 12 (1) (2010) 40–48.
[13] A. Abanda, U. Mori, J.A. Lozano, A review on distance based time series classification, Data Min. Knowl. Discov. 33 (2) (2019) 378–412.
[14] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. 31 (2017) 606–660.
[15] B.D. Fulcher, N.S. Jones, Highly comparative feature-based time-series classification, IEEE Trans. Knowl. Data Eng. 26 (12) (2014) 3026–3037.
[16] F.J. Baldán, J.M. Benítez, Complexity measures and features for times series classification, Expert Syst. Appl. 213 (2023) 119227.
[17] F.J. Baldán, J.M. Benítez, Multivariate times series classification through an interpretable representation, Inform. Sci. 569 (2021) 596–614.
[18] F.J. Baldán, D. Peralta, Y. Saeys, J.M. Benítez, SCMFTS: scalable and distributed complexity measures and features for univariate and multivariate time series in big data environments, Int. J. Comput. Intell. Syst. 14 (2021) 1–15.
[19] L. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.
[20] E. Trillas, L. Eciolaza, Fuzzy logic, Springer International Publishing, 2015, DOI 10 978–3.
[21] C.W. De Silva, Intelligent Control: Fuzzy Logic Applications, CRC Press, 2018.
[22] J. Wang, H. Li, H. Lu, Application of a novel early warning system based on fuzzy time series in urban air quality forecasting in China, Appl. Soft Comput. 71 (2018) 783–799.
[23] N.G. Dincer, Ö. Akkuş, A new fuzzy time series model based on robust clustering for forecasting of air pollution, Ecol. Inform. 43 (2018) 157–164.
[24] A. Amirkhani, M. Shirzadeh, T. Kumbasar, B. Mashadi, A framework for designing cognitive trajectory controllers using genetically evolved interval type-2 fuzzy cognitive maps, Int. J. Intell. Syst. 37 (1) (2022) 305–335.
[25] A. Amirkhani, M. Shirzadeh, M. Molaie, An indirect type-2 fuzzy neural network optimized by the grasshopper algorithm for vehicle ABS controller, IEEE Access 10 (2022) 58736–58751.
[26] V. Novák, Linguistic characterization of time series, Fuzzy Sets and Systems 285 (2016) 52–72.
[27] R. Efendi, Z. Ismail, M.M. Deris, A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand, Appl. Soft Comput. 28 (2015) 422–430.
[28] P.C. de Lima Silva, H.J. Sadaei, R. Ballini, F.G. Guimarães, Probabilistic forecasting with fuzzy time series, IEEE Trans. Fuzzy Syst. 28 (8) (2019) 1771–1784.
[29] M. Han, K. Zhong, T. Qiu, B. Han, Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview, IEEE Trans. Cybern. 49 (7) (2018) 2720–2731.
[30] A. Amirkhani, M. Shirzadeh, M.H. Shojaeefard, A. Abraham, Controlling wheeled mobile robot considering the effects of uncertainty with neuro-fuzzy cognitive map, ISA Trans. 100 (2020) 454–468.
[31] M. Shirzadeh, A. Amirkhani, N. Tork, H. Taghavifar, Trajectory tracking of a quadrotor using a robust adaptive type-2 fuzzy neural controller optimized by cuckoo algorithm, ISA Trans. 114 (2021) 171–190.
[32] P. Ravikumar, V.S. Devi, Fuzzy classification of time series data, in: 2013 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2013, pp. 1–6.
[33] P. Singh, A brief review of modeling approaches based on fuzzy time series, Int. J. Mach. Learn. Cybern. 8 (2017) 397–420.
[34] V.G. Kaburlasos, E. Vrochidou, F. Panagiotopoulos, C. Aitsidis, A. Jaki, Time series classification in cyber-physical system applications by intervals' numbers techniques, in: 2019 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2019, pp. 1–6.
[35] W. Homenda, A. Jastrzebska, Time-series classification using fuzzy cognitive maps, IEEE Trans. Fuzzy Syst. 28 (7) (2019) 1383–1394.
[36] K. Wu, K. Yuan, Y. Teng, J. Liu, L. Jiao, Broad fuzzy cognitive map systems for time series classification, Appl. Soft Comput. 128 (2022) 109458.
[37] H.A. Dau, A. Bagnall, K. Kamgar, C.-C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, E. Keogh, The UCR time series archive, IEEE/CAA J. Autom. Sin. 6 (6) (2019) 1293–1305.
[38] R.F. Albuquerque, P.D. de Oliveira, A.P.d.S. Braga, Adaptive fuzzy learning vector quantization (AFLVQ) for time series classification, in: Fuzzy Information Processing: 37th Conference of the North American Fuzzy Information Processing Society, NAFIPS 2018, Fortaleza, Brazil, July 4-6, 2018, Proceedings 37, Springer, 2018, pp. 385–397.
[39] C.H. Lubba, S.S. Sethi, P. Knaute, S.R. Schultz, B.D. Fulcher, N.S. Jones, catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis, Data Min. Knowl. Discov. 33 (6) (2019) 1821–1852.
[40] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Min. Knowl. Discov. 28 (2014) 851–881.
[41] Y. Kang, R.J. Hyndman, F. Li, et al., Efficient Generation of Time Series with Diverse and Controllable Characteristics, Technical Report, Monash University, Department of Econometrics and Business Statistics, 2018.
[42] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Francisco Javier Baldán Lozano is Assistant Professor at the University of Málaga, Málaga, Spain. He is a member of the SINBAD2 research group and the Andalusian Interuniversity Insttute in Data Science and Computatonal Intelligence (DaSCI). He graduated in Telecommunicaton Engineering in 2014 and earned a master's degree in Data Science and Computer Engineering. He completed his Ph.D. in 2021 with honors. He has published seven influental journal artcles, partcipated in eight research projects, and conducted a research stay at the University of Ghent. His research focuses on tme series analysis, big data, and decision-making systems.

Luis Martínez is Full Professor at the University of Jaén, Jaén, Spain. He has been main researcher in 19 R&D projects, also has published more than 290 JCR papers and more than 300 contributions in Inter/national Conferences related to his areas. His current research interests include intelligent decision support, fuzzy logic, computing with words. He was a recipient of the IEEE Transactions on fuzzy systems Outstanding Paper Award 2008 and 2012. He is Co-Editor-in-Chief of the International Journal of Computational Intelligence Systems. He is IFSA Fellow 2021. Eventually, he has been appointed as Highly Cited Researcher 2017–2023 in Computer sciences.