Self-Organising Fuzzy Belief Inference System for Classification

Gu, Xiaowei; Angelov, Plamen; Shen, Qiang

Published in:
IEEE Transactions on Fuzzy Systems

DOI:
10.1109/TFUZZ.2022.3179148

Publication date:
2022

Citation for published version (APA):
Gu, X., Angelov, P., & Shen, Q. (2022). Self-Organising Fuzzy Belief Inference System for Classification. *IEEE Transactions on Fuzzy Systems*. https://doi.org/10.1109/TFUZZ.2022.3179148

Self-Organizing Fuzzy Belief Inference System for Classification

Xiaowei Gu, Plamen P Angelov, *Fellow, IEEE* and Qiang Shen

***Abstract*—Evolving fuzzy systems (EFSs) are widely known as a powerful tool for streaming data prediction. In this paper, a novel zero-order EFS with a unique belief structure is proposed for data stream classification. Thanks to this new belief structure, the proposed model can handle the inter-class overlaps in a natural way and better capture the underlying multi-model structure of data streams in the form of prototypes. Utilizing data-driven soft thresholds, the proposed model self-organizes a set of prototype-based IF-THEN fuzzy belief rules from data streams for classification, and its learning outcomes are practically meaningful. With no requirement of prior knowledge in the problem domain, the proposed model is capable of self-determining the appropriate level of granularity for rule base construction, while enabling users to specify their preferences on the degree of fineness of its knowledge base. Numerical examples demonstrate the superior performance of the proposed model on a wide range of stationary and nonstationary classification benchmark problems.**

***Index Terms*—belief structure, classification, data streams, evolving fuzzy system, fuzzy belief rule.**

I. Introduction

FUZZY rule-based (FRB) systems are widely used for modelling real-world problems with uncertainties in the form of IF-THEN fuzzy production rules that are easy to interpret by human [1]. Till now, many algorithms have been proposed in the literature to construct interpretable FRB classifiers and regressors from static data [2]–[4], and there are some recent studies exploring the possibility of improving the interpretability of FRB systems further via the use of extremely simple fuzzy rules [5].

Evolving fuzzy systems (EFSs) are a special class of FRB systems that are designed to self-update the system structure and meta-parameters from streaming data "on the fly" [6]. Since the underlying concept was firstly conceived two decades ago [7], [8], EFSs have received great attention and have been implemented for various real-world applications concerning data streams.

EFSs are based on representing domain knowledge in IF-THEN fuzzy production rules. Key features of EFSs that set them apart from mainstream classification approaches (e.g., neural networks, NNs [10] , support vector machines, SVMs [11], random forests, RFs [12]) include 1) the dynamically evolving structure capable of capturing concept drifts and shifts in data streams [13]; 2) the ability of transforming the learned knowledge from data in the highly intuitive, human-interpretable form of IF-THEN rules. To date, a variety of EFSs with different structure evolving and parameter learning schemes have been proposed for tackling real-world classification and regression problems, where the model interpretability plays a key role [9].

Evolving Takagi-Sugeno (eTS) [6] and dynamic evolving neuro-fuzzy system (DENFIS) [8] are the two earliest and most representative ones in the literature. Both EFSs use recursive clustering algorithms to learn fuzzy rules from data, supported by recursive least square algorithms for updating consequent parameters. eTS and DENFIS as well as their variants have been used in various real-world applications for dynamical system modelling [14]. Sequential adaptive fuzzy inference system (SAFIS) [14] is introduced on the basis of functional equivalence between a radial basis function NN and a FRB system, and the extended Kalman filter algorithm is used by SAFIS to self-update its rule parameters. Parsimonious network based on fuzzy inference system (PANFIS) is proposed in [25], building hyper-ellipsoid clusters in arbitrary positions in the feature space from data and forms the premise parts of fuzzy rules based on these clusters. An extended recursive least square algorithm is also introduced to PANFIS for consequent parameter updating. To enhance the stability of rule base evolution and parameters updating, a correntropy-based evolving fuzzy neural system (CEFNS) is proposed in [17], where correntroy is employed as the main criterion due to its strong non-Gaussian noise rejection ability. CEFNS is further extended in [18] to maximum recursive CEFNS (MRCEFNS) by utilizing the recursive maximum correntropy technique for updating its consequent parameters. A novel and symmetrical approach with self-learning/adaptive thresholds (EFS-SLAT) is introduced in [19] to free EFSs from the requirement of predefined external controlled parameters, utilizing online training errors to help the model select and adjust threshold values automatically. More recently, a statistically evolving fuzzy inference system (SEFIS) is presented in [20] as an further extension of MRCEFNS to tackle the non-Gaussian noises, by employing an adaptive maximum correntropy extend Kalman filter for parameter updating. To ease the effect of "the curse of dimensionality" on high-dimensional problems, a novel AnYa type EFS combining very sparse random projection (VSRP-AnYa-EFS) is proposed in [21]. By using random sparse Bernoulli (RSB) matrix for feature reduction, VSRP-AnYa-EFS is able to learn a more

X. Gu is with the School of Computing, University of Kent, Canterbury, CT2 7NZ, UK. email: X.Gu@kent.ac.uk.

P. Angelov is with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK. email: p.angelov@lancaster.ac.uk.

Q. Shen is with the Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK. email: qqs@aber.ac.uk.

Corresponding author: Xiaowei Gu

Manuscript received XXXX XX, 2022; revised XXXX XX, 2022.

compact, simpler fuzzy rule base from data streams through a more computationally efficient learning process. An online bagging-based ensemble scheme of EFS, named OB-EFS is introduced in [22] to enhance the robustness of individual EFSs on streaming data prediction. OB-EFS uses a probabilistic online sampling strategy to distribute newly observed data samples to ensemble members. This ensemble scheme will autonomously prune these stale ensemble members that produce higher prediction errors, and add new members if unfamiliar data patterns are observed. Other EFSs include, but are not limited to, evolving fuzzy-rule based classifiers (eClass 0 and eClass1) [23], flexible fuzzy inference system (FLEXFIS) [24], generic evolving neuro-fuzzy inference system (GENEFIS) [25], evolving fuzzy model (eFuMo) [26], self-organizing fuzzy inference systems (SOFIS) [27], self-evolving fuzzy system (SEFS) [28], and spatio-temporal fuzzy inference system (SPATFIS) [29]. The interested reader is referred to the recently published literature reviews [9], [30] on the latest development of EFSs and their applications to real-world problems.

Belief rule-based (BRB) systems extend the traditional FRB systems with belief structures to cover credibility uncertainties [31], [32]. The evidential reasoning (ER) algorithm [33] based on the Dempster-Shafer theory of evidence [34] is typically employed by BRB models as the inference engine to perform decision-making. Similar to FRB, BRB systems take the form of belief rules. Belief rules are based on traditional IF-THEN fuzzy rules but with belief degrees introduced to the consequent parts [35], and they provide an informative scheme of formulating expert experience, uncertain knowledge and hybrid information [36], [37]. BRB systems are capable to capture nonlinear causal relationships between premise attributes and consequents. They can integrate various types of uncertain information [38] to produce unified conclusions [39]. Hence, BRB systems have demonstrated superior performance in modelling complex problems [35].

Typically, the structure and parameters of BRB systems are defined by experts based on their experience and domain knowledge. Hence, further optimization is often needed because the initial systems may be inaccurate due to various different factors, e.g., insufficient historical data and imprecise prior knowledge [39], [40]. Most of the existing BRB optimization approaches can be divided into the following three categories [40]: *i)* parameter learning [41], to optimize the system parameters such as membership values, rule weights and belief degrees; *ii)* structure learning [42], to downsize the BRB systems by selecting the most representative rules; and *iii)* parameter and structure joint optimization [43], to iteratively and alternatively optimize the system parameters and structure. However, conventional approaches of constructing BRB systems have two critical drawbacks. Firstly, the scale of BRB systems has to be small because a system with a huge number of rules and parameters is practically infeasible to be constructed and very difficult to be optimized [44]. Secondly, BRB systems are less capable of handling new data samples with unfamiliar patterns. If such data is presented, classical BRB systems require full reconstruction because the optimization process has to be performed offline with static data. The two issues remaining unsolved largely limit the applicability of BRB systems to large-scale, high-dimensional problems concerning data streams.

In this paper, a novel zero-order EFS with a belief structure named Self-Organizing Fuzzy Belief Inference System (SOFBIS) is proposed using SOFIS+ as its implementation foundation. SOFIS+ [45] (built on SOFIS [27]) is a recently introduced zero-order EFS for data stream classification, which learns a set of prototypes from streaming data chunk-by-chunk, exploiting their underlying ensemble properties and mutual distances in a human-interpretable manner. These prototypes are highly representative generalized samples preserving the local structure of the original data [45]. In contrast with SOFIS+ (and SOFIS), SOFBIS has stronger capability of handling uncertainties thanks to the belief degrees introduced to the consequent parts of the learned IF-THEN rules, resulting in self-evolving models from data streams with greater prediction accuracy. Supported with the belief structure, SOFBIS takes data inter-class overlaps into consideration and thus, can better understand the multi-model distributions of the data. It also overcomes the two critical drawbacks suffered by conventional BRB systems as the system structure and meta-parameters are learned from data directly with no requirement of human intervention, being able to self-evolve and self-update with new data samples and continuously expand the induced knowledge base for more accurate classification. To summarize, key features of the proposed SOFBIS are as follows:

1) it extends the conventional zero-order EFS with a belief structure that is continuously self-evolving from data streams;
2) it treats the inter-class overlaps during prototype identification in a natural way, thereby better preserving the underlying structure of data;
3) it utilizes soft thresholds directly calculated from data only to guarantee the objectiveness and practical meaning of the learning outcomes;
4) it allows users to determine the fineness of the learned knowledge base without requiring prior knowledge of the problem domain.

The remainder of this paper is organized as follows. Technical details of the proposed SOFBIS are presented in Section II. A variant of SOFBIS, named SOFBIS+ is further introduced in Section III. Numerical examples are provided in Section IV. This paper is concluded by Section V.

II. Proposed SOFBIS

In this section, technical details of the proposed SOFBIS are presented in detail.

A. General Architecture and Decision-Making Policy

SOFBIS is composed of N fuzzy belief IF-THEN rules in the following form ($k = 1, 2, \ldots, N$; N is the total number of rules):

$$\mathcal{R}_k: \textit{ IF } (\boldsymbol{x} \sim \boldsymbol{a}_{k,1}) \textit{ AND } (\boldsymbol{x} \sim \boldsymbol{a}_{k,2}) \textit{ AND ... AND } (\boldsymbol{x} \sim \boldsymbol{a}_{k,P_k}) \textit{ THEN } \{\boldsymbol{D}, \boldsymbol{\beta}_k\} \tag{1}$$

where " " denotes similarity; $\boldsymbol{x}$ is the M dimensional input sample, $\boldsymbol{x} = [x_1, x_2, \ldots, x_M]^T$ to $\boldsymbol{\mathcal{R}}_k$ in the real space, $\mathfrak{R}^M$; $a_{k,i} = [a_{k,i,1}, a_{k,i,2}, \ldots, a_{k,i,M}]^T$ is the i^{th} antecedent part (prototype) of $\boldsymbol{\mathcal{R}}_k$; P_k is the total number of prototypes associated with $\boldsymbol{\mathcal{R}}_k$; $\boldsymbol{D} = [D_1, D_2, \ldots, D_C]^T$ represents the consequent part; D_k stands for the conclusion "$\boldsymbol{x}$ belongs to class k", namely, the predicted label of $\boldsymbol{x}$, denoted as $\hat{y}$, is $\hat{y} = k$; $\boldsymbol{\beta}_k = [\beta_{k,1}, \beta_{k,2}, \ldots, \beta_{k,C}]^T$ is the vector of belief degrees; $\beta_{k,i}$ is the corresponding belief degree of the i^{th} conclusion D_i, showing the possibility.

Each fuzzy belief rule, $\boldsymbol{\mathcal{R}}_k$ consists of a number of prototypes learned from data with the same belief degrees, and these prototypes associated with the same rule are connected by logic "AND" connectives. Hence, $\boldsymbol{\mathcal{R}}_k$ can be also viewed as an ensemble of multiple parallel simpler fuzzy belief rules sharing the same consequent part as follows ($i = 1, 2, \ldots, P_k$):

$$\boldsymbol{\mathcal{R}}_{k,i}: \ IF \ (\boldsymbol{x} \sim \boldsymbol{a}_{k,i}) \ THEN \ \{\boldsymbol{D}, \boldsymbol{\beta}_k\} \tag{2}$$

The main reason for employing the logic conjunction-based "AND" connectives in the fuzzy belief IF-THEN rules of SOFBIS is due to its unique decision-making scheme that all identified prototypes contribute to the consequent (or decision-making). For conventional EFSs [23], [27], [45], by using the disjunction "OR" connectives, only the nearest prototype of each class is used for computing the rule activation level.

Given a particular data sample $\boldsymbol{x}$, the activation of $\boldsymbol{\mathcal{R}}_k$ is produced based on the distances between all the associated prototypes and $\boldsymbol{x}$:

$$\omega_k = \frac{\theta_k \lambda_k}{\sum_{l=1}^{N} \theta_l \lambda_l} \tag{3}$$

Here θ_k denotes the rule weight of $\boldsymbol{\mathcal{R}}_k$, namely, the relative importance to the overall system output; λ_k is the confidence score produced by $\boldsymbol{\mathcal{R}}_k$ calculated based on the L_1 distances between $\boldsymbol{x}$ and all the prototypes associated as follows:

$$\lambda_k = \sum_{i=1}^{P_k} \lambda_{k,i} \tag{4}$$

where $\lambda_{k,i} = e^{-\frac{||\boldsymbol{x} - \boldsymbol{a}_{k,i}||_1^2}{\delta^2}}$; $||\boldsymbol{x} - \boldsymbol{y}||_1 = \sum_{j=1}^{M} |x_j - y_j|$ denotes the L_1 distance between $\boldsymbol{x}$ and $\boldsymbol{y}$; δ is derived from data directly during the learning process, which will be detailed in Section III.B.

Then, the outputs of individual fuzzy belief rules, $\boldsymbol{\mathcal{R}}_k$ ($k = 1, 2, \ldots, N$) are integrated using Eq. (5), and the predicted label, $\hat{y}$ of $\boldsymbol{x}$ is determined using Eq. (6).

$$\hat{\boldsymbol{\beta}} = \sum_{k=1}^{N} \omega_k \boldsymbol{\beta}_k \tag{5}$$

$$\hat{y} = k^*; \quad k^* = \underset{k=1,2,\ldots,C}{\arg\max} (\hat{\beta}_k) \tag{6}$$

where $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_C]^T$.

Remark 1: Unlike the analytical evidential reasoning approach employed by conventional BRB classification models [33], [41] for aggregating belief rule outputs and making inference, the activations of the fuzzy belief rules used by SOFBIS are calculated utilizing Gaussian type membership functions due to the simplicity, same as [31]. The class labels of data samples are determined based on the fuzzily weighted sum of belief degrees of individual fuzzy belief rules straightforwardly. Such simplifications effectively improve the computational efficiency of SOFBIS, especially when the data dimensionality is high.

B. Learning SOFBIS

Similar to SOFIS+ [45], SOFBIS learns a set of prototype-based fuzzy belief IF-THEN rules from data streams on a chunk-by-chunk basis and can continuously self-improve its knowledge base with new data. There are two unique features that differ SOFBIS from SOFIS+, which include: 1) potential class overlaps are considered during prototype identification naturally via data partitioning; 2) each identified prototype can belong to multiple classes with different belief degrees. These two features enables SOFBIS to better capture the underlying patterns of data and achieve greater prediction accuracy.

First of all, let $\boldsymbol{X}$ be a particular data stream in the M dimensional real space, $\mathfrak{R}^M$. Data samples of $\boldsymbol{X}$ continuously arrive in the form of chunks, denoted as $\boldsymbol{X}_t = \{\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}, \ldots, \boldsymbol{x}_{t,K_t}\}$, where the subscript t denotes the time instance at which $\boldsymbol{X}_t$ is observed, $t = 1, 2, \ldots, T$; T denotes the total number of data chunks available; $\boldsymbol{x}_{t,k}$ represents the k^{th} data sample of $\boldsymbol{X}_t$; K_t is the cardinality of $\boldsymbol{X}_t$, and; there is $\boldsymbol{X}_1 \cup \boldsymbol{X}_2 \cup \ldots \cup \boldsymbol{X}_T = \boldsymbol{X}$. $Y_t = \{y_{t,1}, y_{t,2}, \ldots, y_{t,K_t}\}$ is the corresponding class labels of $\boldsymbol{X}_t$ and there is $y_{t,k} \in \{1, 2, \ldots, C\}, \forall t, k$. Note that the cardinalities of different data chunks are not necessarily to be the same. By default, the level of granularity externally controlled by users is set as G (a non-negative integer).

Each learning cycle of SOFBIS consists of two stages as follows.

Stage 1. Prototype identification: Given the t^{th} data chunk and the corresponding class labels, $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$, SOFBIS firstly calculates the squared L_1 distances between any two data samples within $\boldsymbol{X}_t$ and obtain the following $K_t \times K_t$ dimensional distance matrix, $\mathbf{d}_t$:

$$\mathbf{d}_t = [||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||_1^2]_{1]k=1:K_t}^{2]i=1:K_t} \tag{7}$$

Then, the data-driven threshold, $\sigma_{t,G}^2$ is extracted from $\mathbf{d}_t$ at the predetermined level of granularity, G using Eq. (8) [27], [45]:

$$\sigma_{t,g}^2 = \frac{\sum_{i=1}^{K_t-1} \sum_{k=i+1}^{K_t} w_{g,i,k} ||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||_1^2}{\sum_{i=1}^{K_t-1} \sum_{k=i+1}^{K_t} w_{g,i,k}} \tag{8}$$

where $g = 1, 2, \ldots, G$, and; $w_{g,i,k}$ and $\delta_{t,0}^2$ are defined as follows.

$$w_{g,i,k} = \begin{cases} 1 & if \ ||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||_1^2 \leq \sigma_{t,g-1} \\ 0 & else \end{cases} \tag{9}$$

$$\sigma_{t,0}^2 = \frac{2 \sum_{i=1}^{K_t-1} \sum_{k=i+1}^{K_t} ||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||_1^2}{K_t(K_t - 1)} \tag{10}$$

Note that $\delta_{t,G}^2$ is a soft threshold serving as an estimation of the maximum distance between any two neighbouring data

samples at the G^{th} level of granularity. Since $\delta^2_{t,G}$ is derived directly from the mutual distances between data samples, it is guaranteed to be valid and meaningful.

The distance matrix $\mathbf{d}_t$ is further converted to a $K_t \times K_t$ dimensional sparse adjacency matrix, $\mathbf{A}_t$ using Eq. (11):

$$\mathbf{A}_t = [A_{t,i,k}]^{i=1:K_t}_{k=1:K_t} \tag{11}$$

where $A_{t,i,k} = \begin{cases} 1 & if\ ||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||^2_1 < \sigma^2_{t,G} \\ 0 & else \end{cases}, \forall i,k.$ According to Eq. (11), if the squared L_1 distance between $\boldsymbol{x}_{t,i}$ and $\boldsymbol{x}_{t,k}$ is less than $\sigma^2_{t,G}$, $\boldsymbol{x}_{t,i}$ and $\boldsymbol{x}_{t,k}$ can be seen as a pair of neighbours in the data space $\Re^M$ and $A_{t,i,k}$ is set to be 1. Otherwise, $\boldsymbol{x}_{t,i}$ and $\boldsymbol{x}_{t,k}$ are considered to be distant from each other, and $A_{t,i,k}$ is set to be 0. Hence, one can see that the greater G is, the sparser $\mathbf{A}_t$ will be.

Next, every individual data sample $\boldsymbol{x}_{t,i} (i = 1, 2, \ldots, K_t)$ is treated as a micro-cluster and is assigning membership degrees to nearby data samples including itself using Eq. (12):

$$\mu_{t,i,k} = \begin{cases} e^{-\frac{||\boldsymbol{x}_{t,i} - \boldsymbol{x}_{t,k}||^2_1}{\sigma^2_{t,G}}} & if\ A_{t,i,k} = 1 \\ 0 & else \end{cases}, \ \forall i,k \tag{12}$$

Note that there is $\mu_{t,i,k} = \mu_{t,k,i}$.

Membership degrees assigned to each individual data sample $\boldsymbol{x}_{t,i}$ $(i = 1, 2, \ldots, K_t)$ by its neighbours are aggregated together by Eq. (13):

$$\hat{\mu}_{t,i} = \sum_{k=1}^{K_t} \mu_{t,i,k} \tag{13}$$

These data samples with the highest aggregated membership degrees among their neighbours are identified by Condition 1 as the raw prototypes, denoted as $\mathbf{P}_t$:

$$\begin{aligned} Cond.\ 1:\ & if\ (\hat{\mu}_{t,i} = \underset{\forall k, A_{t,i,k}=1}{\arg\max}\ (\hat{\mu}_{t,k})) \\ & then\ (\mathbf{P}_t \leftarrow \mathbf{P}_t \cup \{\boldsymbol{x}_{t,i}\}) \end{aligned} \tag{14}$$

Condition 1 identifies these data samples that are connected to the most nearby data samples as raw prototypes, representing the local peaks of the multimodal distribution of data. These raw prototypes can be viewed as the hubs of the topology formed by observed data samples, and they have the great potential to well preserve the underlying multi-model structure of data. In general, given a greater G, SOFBIS focuses more on the local patterns of data, and will identify more raw prototypes.

Once all the raw prototypes are identified by Condition 1, they are further refined by Eq. (15) as the centres of connected data samples such that they can better represent the local data patterns:

$$\boldsymbol{a}_{t,i} = \frac{\sum_{j=1}^{K_t} A_{t,i,j} \boldsymbol{x}_{t,j}}{\sum_{j=1}^{K_t} A_{t,i,j}},\ \forall \boldsymbol{x}_{t,i} \in \boldsymbol{P}_t \tag{15}$$

The belief degrees, $\boldsymbol{\beta}_{t,i}$ associated with each individual prototype, $\boldsymbol{a}_{t,i}$ are defined as ($i = 1, 2, \ldots, N_t$; N_t is the cardinality of $\boldsymbol{P}_t$):

$$\boldsymbol{\beta}_{t,i} = [\beta_{t,i,1}, \beta_{t,i,2}, ..., \beta_{t,i,C}]^T \tag{16}$$

where $\beta_{t,i,c} = \frac{\sum_{j=1}^{K_t} A_{t,i,j} \mathbb{I}(y_{t,j}=c)}{\sum_{j=1}^{K_t} A_{t,i,j}}$; $c = 1, 2, ..., C$. The support of $\boldsymbol{a}_{t,i}$, namely, the number of data samples associated with $\boldsymbol{a}_{t,i}$ is obtained as:

$$s_{t,i} = \sum_{j=1}^{K_t} A_{t,i,j} \tag{17}$$

The set of prototypes, the corresponding belief degrees and supports are denoted as $\boldsymbol{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$, respectively. Once $\boldsymbol{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$ have been extracted from $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$, the second stage of the current learning cycle begins.

Remark 2: The level of granularity, G is an externally controlled parameter for SOFBIS to self-learn the soft threshold, $\delta^2_{t,G}$ from the current data chunk, $\boldsymbol{X}_t$, which defines the maximum distance between any two neighbouring data samples. G controls the sparseness of the adjacency matrix, $\mathbf{A}_t$, and has a direct impact on the number of prototypes identified from data. In general, given a greater G, SOFBIS tends to focus more on the local patterns of data due to a sparser $\mathbf{A}_t$ learned from $\boldsymbol{X}_t$ and will identify more prototypes, disclosing finer details of the underlying data structures. This helps SOFBIS to achieve higher prediction accuracy, but may also decrease its computational efficiency as the knowledge base becomes larger. However, it needs to be stressed that G carries clear meaning and is not a user- or problem- specific parameter [27], [45]. The recommended value of G is 6 for small-scale problems, and 9 for large-scale problems.

Stage 2. Knowledge base updating: Given the learned prototypes and the corresponding belief degrees, if the current data chunk $\boldsymbol{X}_t$ (and $\boldsymbol{Y}_t$) is the very first one (namely, $t = 1$), the knowledge base of SOFBIS is initialized as:

$$\tilde{\boldsymbol{\Lambda}} \leftarrow \boldsymbol{\Lambda}_t;\quad \tilde{\boldsymbol{B}} \leftarrow \boldsymbol{B}_t;\quad \tilde{\boldsymbol{S}} \leftarrow \boldsymbol{S}_t \tag{18}$$

The kernel width, δ for calculating the confidence score (Eq. (4)) is initialized as: $\delta \leftarrow \sigma_{t,G}$.

If $\boldsymbol{X}_t$ is not the first data chunk presented to SOFBIS, namely, $t > 1$, its knowledge base will be updated by incorporating the latest knowledge learned from the current data chunk. Firstly, the kernel width, δ is updated using Eq. (19):

$$\delta \leftarrow \sqrt{\frac{\sum_{j=1}^{t-1} K_j \delta^2 + K_t \sigma^2_{t,G}}{\sum_{j=1}^{t} K_j}} \tag{19}$$

Then, Condition 2 is used to select out these prototypes from $\boldsymbol{\Lambda}_t$ that represent new local patterns distinctive from the existing ones to expand the knowledge base ($i = 1, 2, \ldots, N_t$):

$$\begin{aligned} Cond.\ 2:\ & if\ (\min_{\boldsymbol{a} \in \tilde{\boldsymbol{\Lambda}}} (||\boldsymbol{a}_{t,i} - \boldsymbol{a}||^2_1) > \lambda) \\ & then\ \begin{pmatrix} \tilde{\boldsymbol{\Lambda}} \leftarrow \tilde{\boldsymbol{\Lambda}} \cup \{\boldsymbol{a}_{t,i}\};\ \boldsymbol{\Lambda}_t \leftarrow \boldsymbol{\Lambda}_t / \{\boldsymbol{a}_{t,i}\} \\ \tilde{\boldsymbol{B}} \leftarrow \tilde{\boldsymbol{B}} \cup \{\boldsymbol{\beta}_{t,i}\};\ \boldsymbol{B}_t \leftarrow \boldsymbol{B}_t / \{\boldsymbol{\beta}_{t,i}\} \\ \tilde{\boldsymbol{S}} \leftarrow \tilde{\boldsymbol{S}} \cup \{\boldsymbol{s}_{t,i}\};\ \boldsymbol{S}_t \leftarrow \boldsymbol{S}_t / \{s_{t,i}\} \end{pmatrix} \end{aligned} \tag{20}$$

where $\boldsymbol{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$ are the sets of prototypes, belief degrees and supports learned from $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$; $\tilde{\boldsymbol{\Lambda}}$ is the set of prototypes SOFBIS maintains in its knowledge base; $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ are the corresponding belief degrees and supports of $\tilde{\boldsymbol{\Lambda}}$.

For all remaining prototypes in $\mathbf{\Lambda}_t$ that fail to satisfy Condition 2, they represent similar local patterns that SOFBIS have been observed from historical chunks before. These prototypes will not bring fundamentally new knowledge to SOFBIS and, hence, they are subsequently exploited for updating existing prototypes within the knowledge base ($\boldsymbol{a}_{t,i} \in \mathbf{\Lambda}_t$) by Eq. (21). In so doing, SOFBIS is capable of self-adapting to the gradual evolutions of local patterns of data, whilst maintaining a more compact knowledge base.

$$\boldsymbol{a}_{i^*} \leftarrow \frac{s_{i^*}\boldsymbol{a}_{i^*} + s_{t,i}\boldsymbol{a}_{t,i}}{s_{i^*} + s_{t,i}}; \quad \boldsymbol{\beta}_{i^*} \leftarrow \frac{s_{i^*}\boldsymbol{\beta}_{i^*} + s_{t,i}\boldsymbol{\beta}_{t,i}}{s_{i^*} + s_{t,i}}; \quad s_{i^*} \leftarrow s_{i^*} + s_{t,i} \tag{21}$$

where $\boldsymbol{a}_{i^*}$ is the nearest prototype to $\boldsymbol{a}_{t,i}$ within $\tilde{\mathbf{\Lambda}}$ identified by Eq. (22); s_{i^*} and $s_{t,i}$ are the respective supports of $\boldsymbol{a}_{i^*}$ and $\boldsymbol{a}_{t,i}$.

$$\boldsymbol{a}_{i^*} = \arg\min_{\boldsymbol{a} \in \tilde{\mathbf{\Lambda}}} (||\boldsymbol{a}_{t,i} - \boldsymbol{a}||_1^2) \tag{22}$$

Once the knowledge base has been initialized/updated, a set of simpler fuzzy belief rules with singleton premise and consequent parts are built in the form of Eq. (2) based on $\tilde{\mathbf{\Lambda}}$ and $\tilde{\boldsymbol{B}}$. These simpler rules that share the same consequences are combined together using the logical "AND" connectives in the form of Eq. (1). After the fuzzy belief rule base has been constructed, the current learning cycle ends. Then, SOFBIS continues to process the next data chunk by starting a new learning cycle or terminates its learning process is if no new training data is presented.

Remark 3: Zero-order EFSs typically identify prototypes from data samples of different classes separately and ignore potential class overlaps or interactions. Thanks to its unique belief structure, SOFBIS considers data samples of all classes together during prototype identification to gain a better understanding about the structure of data, and quantifies class overlaps via belief degrees. However, different from conventional BRB models that require human expertise to be involved, the belief degrees of SOFBIS are automatically generated by computing the proportions of data samples belonging to different classes within the clusters that are each formed around an identified prototype using Eq. (16). In so doing, SOFBIS handles the class overlap problems in a natural, data-driven way. In addition, since all the prototypes are involved in decision-making rather than the nearest ones only, SOFBIS can better handle the outliers and is able to construct smoother and more precise classification boundaries from data (also see the example given by Fig. 1).

Remark 4: In this study, to facilitate online learning, all the fuzzy belief rules within the rule base of SOFBIS contribute equally to the final decisions. In other words, the rule weights are assumed to be the same ($\theta_1 = \theta_2 = \cdots = \theta_N$). In addition, different attributes also weight equally. In general, one may consider to utilize evolutionary computation techniques (e.g., particle swarm optimization [39], differential evolution [44]) to find the nearly optimal rule weights and attribute weights for classification, but this is beyond the scope of this study.

Remark 5: The chunk-by-chunk learning scheme enables SOFBIS to have a better understanding about the data patterns while aligns closely to the idea of online learning, providing the ability to handle new data patterns efficiently and effectively. SOFBIS can learn from data chunks of various sizes as long as each data chunk contains no less than two data samples, which is essential to derive the soft threshold (Eq. (8)). Hence, the chunk size is not a user- or problem- specific parameter and can be determined without prior knowledge of the problems. Generally, a larger chunk size allows SOFBIS to understand the local data patterns better and produce predictions with greater accuracy.

The learning procedure of SOFBIS is summarized by Algorithm 1 in the form of pseudo-code. Computational complexity analysis of SOFBIS is presented in Supplementary Section A.

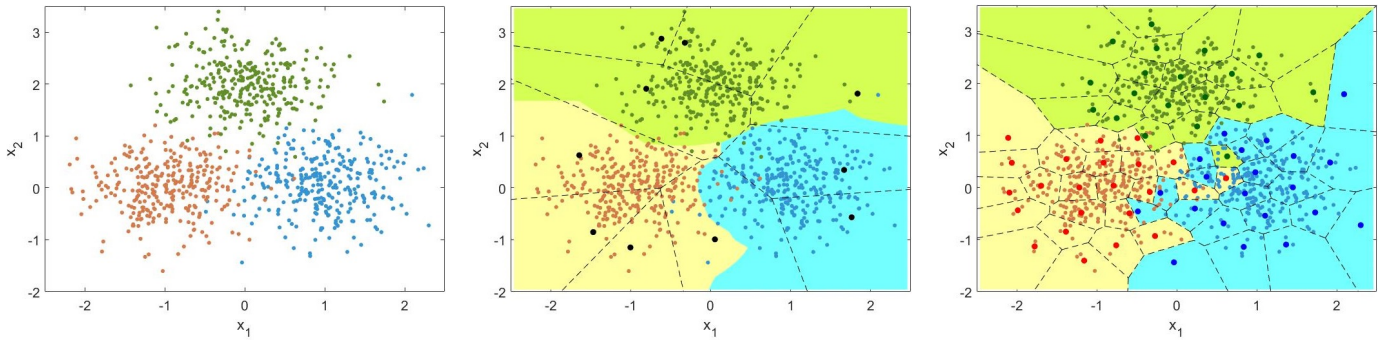
Algorithm 1 Learning of SOFBIS.

while ($\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are available) **do**
  obtain $\mathbf{d}_t$ from $\boldsymbol{X}_t$ using (7);
  estimate $\sigma_{t,G}^2$ using (8);
  convert $\mathbf{d}_t$ to $\mathbf{A}_t$ using (9);
  calculate $\hat{\mu}_{t,i}$ for each $\boldsymbol{x}_{t,i} \in \boldsymbol{X}_t$ using (12) and (13);
  identify $\boldsymbol{P}_t$ from $\boldsymbol{X}_t$ using Condition 1;
  obtain $\mathbf{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$ using (15) and (16)
  if ($t = 1$) **then**
    initialize $\tilde{\mathbf{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using (18);
    $\delta \leftarrow \sigma_{t,G}$;
  else
    update δ using (19);
    expand $\tilde{\mathbf{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using Condition 2;
    update $\tilde{\mathbf{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using (21) and (22);
  end if
  construct $\boldsymbol{\mathcal{R}}_1$, $\boldsymbol{\mathcal{R}}_2$,...,$\boldsymbol{\mathcal{R}}_N$ from $\tilde{\mathbf{\Lambda}}$ and $\tilde{\boldsymbol{B}}$;
  $t \leftarrow t + 1$;
end while

An illustrative example based on synthetic data is given by Fig. 1 to demonstrate the difference between the proposed SOFBIS and SOFIS+ in terms of the learning outcomes. As depicted in Fig. 1a, there are a total of 900 data samples of three different classes in the data space (300 samples per class), which are represented by dots of three different colours (class 1- blue; class 2-orange; class 3-green). These data samples are randomly generated from three different Gaussian distributions, namely, $N([1.0000, 0.0000]^T, \mathbf{\Sigma})$, $N([-1.0000, 0.0000]^T, \mathbf{\Sigma})$ and $N([0.0000, 2.0000]^T, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \begin{bmatrix} 0.2500, & 0.0000 \\ 0.0000, & 0.2500 \end{bmatrix}$.

For visual clarity, $G = 1$ is set for this example and all samples are used as a single chunk. SOFBIS identifies 10 prototypes from data and builds five fuzzy belief IF-THEN rules given by Supplementary Table S1. The prototypes identified by SOFBIS are presented in Fig. 1b (black large dots). These prototypes], as shown in Fig. 1b, partition the data space into 10 shape-free clusters with the boundaries represented by dash lines. The classification boundaries formed by these prototypes are represented by the borders of these shaded regions in different colours.

Using the same experimental setting, SOFIS+ learns 63 prototypes from data samples of three classes and builds



(a) Data samples (b) Classification boundaries by SOFBIS (c) Classification boundaries by SOFIS+

Fig. 1: Illustrative example of classification boundaries constructed by SOFBIS and SOFIS+.

three fuzzy rules given by Supplementary Table S2 (one rule per class). The prototypes identified by SOFIS+ and the classification boundaries derived from these prototypes are depicted in Fig. 1c, where prototypes of different classes are represented by large dots of three different colours.

One can observe by comparing between Figs. 1b and 1c that SOFBIS produces smoother and more precise classification boundaries with much less prototypes than SOFIS+, showing its stronger capability in handling class overlaps. In addition, the classification boundaries constructed by SOFBIS are also less sensitive to outliers.

III. Proposed SOFBIS+

As mentioned earlier, the level of granularity, G is not a user- or problem-specific parameter, and its value can be determined based on users' preferences. However, SOFBIS may not be able to maximize its prediction performance if G is not properly set up. In some application scenarios, it may also be difficult for users to pick an appropriate value of G for the problems satisfying the needs most. To address this issue, in this section, modifications are introduced to SOFBIS, enabling it to automatically self-determine the most suitable level of granularity based on the mutual distances between data. To differentiate from the original algorithm, SOFBIS with this new feature is named as SOFBIS+.

SOFBIS+ has exactly the same architecture and decision-making policy as SOFBIS. The key difference between SOFBIS+ and SOFBIS lies in the way that the data-driven threshold, $\sigma^2_{t,G}$ is learned from data. As described in Section II.B, SOFBIS estimates $\sigma^2_{t,G}$ from $\mathbf{d}_t$ using Eq. (8) directly based on the value of G given by users. In contrast, SOFBIS+ employs the well-known elbow method [47] to help determine the best level of granularity, same as [48].

The learning of SOFBIS+ is detailed as follows.

A. Learning SOFBIS+

For each data chunk, $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$, SOFBIS+ utilizes the following objective function to assess whether a particular level of granularity (assuming the g^{th} level) is sufficient or not to separate well the data such that samples with similar characteristics are grouped together [48].

$$J_1(g) = J_0(g) + \rho\frac{N_{t,g}}{K_t} \tag{23}$$

where $J_0(g) = \frac{\sum_{j=1}^{K_t} \min_{\boldsymbol{p}\in\boldsymbol{P}_{t,g}}(||\boldsymbol{x}_{t,j}-\boldsymbol{p}||_1^2)}{K_t\sigma^2_{t,0}}$, which measures the intra-cluster variance between the clusters formed around the raw prototypes by attracting nearby data samples resembling Voronoi tessellation [49]; $\boldsymbol{P}_{t,g}$ denotes the set of raw prototypes identified by Condition 1 at the g^{th} level of granularity; $N_{t,g}$ is the cardinality of $\boldsymbol{P}_{t,g}$; ρ is the regularization parameter externally controlled by users ($\rho \geq 0$); $\frac{N_{t,g}}{K_t}$ is the penalty term calculated based on the number of prototypes identified from $\boldsymbol{X}_t$ at the g^{th} level of granularity. Hence, ρ controls the trade-off between the intra-cluster variance and the size of the knowledge base in terms of prototypes. In general, a smaller ρ enables SOFBIS+ to self-construct a larger knowledge base from data, and vice versa. The recommended value of ρ is 10^{-1} for small-scale problems, and 10^{-3} for large-scale problems.

The most suitable level of granularity, denoted by g_t^* for $\boldsymbol{X}_t$ is determined by Condition 3 as the local minimum of $J_1(g)$ [48].

$$\begin{aligned} Cond.\ 3:\ &if\ (J_1(g_t^*) < J_1(g_t^*-1)\ and\ (J_1(g_t^*) < J_1(g_t^*+1)) \\ &then\ (g_t^*\ is\ sufficent\ for\ \boldsymbol{X}_t) \end{aligned} \tag{24}$$

The main purpose of Condition 3 is to help SOFBIS+ find out the particular level of granularity at which the intra-cluster variance between the clusters formed around the identified prototypes has been sufficiently small.

If g_t^* satisfies Condition 3, it suggests that prototypes identified at $(g_t^*)^{th}$ level of granularity can well represent the local patterns of the current data chunk $\boldsymbol{X}_t$, and increasing the level of granularity further will not reduce the intra-cluster variance significantly but only increase the model complexity.

Once g_t^* is determined, SOFBIS+ continues to extract $\boldsymbol{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$ from $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$, and initializes/updates its knowledge base (namely, $\tilde{\boldsymbol{\Lambda}}_t$, $\tilde{\boldsymbol{B}}_t$ and $\tilde{\boldsymbol{S}}_t$), following the same algorithmic procedure as SOFBIS. The kernel width, δ will be updated using Eq. (19) by setting $G \leftarrow g_t^*$.

Remark 6: Without the need of human intervention to specify the level of granularity, SOFBIS+ self-determines the most appropriate level of granularity, g_t^* for prototype identification by minimizing the objective function $J_1(g)$ based on the well-known elbow method [47]. Meanwhile, users can still be involved in this process by adjusting the regularization parameter ρ to control the trade-off between the intra-

cluster variance and the number of prototypes, which indirectly influences the degree of fineness of the learned knowledge base. On the other hand, it is worth noting that, as SOFBIS+ estimates the value of g_t^* in an exploratory manner based on the mutual distances between data samples of each individual chunk, drastic changes of underlying data patterns can have a great impact on the correctness of the estimation. Hence, the performance of SOFBIS+ may be limited in nonstationary environments. The recommended value of ρ is 10^{-1} for small-scale problems, and 10^{-3} for problems of larger sizes.

The learning procedure of SOFBIS+ is summarized by Algorithm 2 as follows. Computational complexity analysis of SOFBIS+ is presented in Supplementary Section C.

Algorithm 2 Learning of SOFBIS+.

while ($\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$ are available) **do**
  obtain $\mathbf{d}_t$ from $\boldsymbol{X}_t$ using (7);
  $g \leftarrow 0$;
  while Condition 3 is not met **do**
    $g \leftarrow g + 1$;
    estimate $\sigma_{t,g}^2$ using (8);
    convert $\mathbf{d}_t$ to $\mathbf{A}_{t,g}$ using (9);
    calculate $\hat{\mu}_{t,i}$ for each $\boldsymbol{x}_{t,i} \in \boldsymbol{X}_t$ using (12) and (13);
    identify $\boldsymbol{P}_{t,g}$ from $\boldsymbol{X}_t$ using Condition 1;
  end while
  $G \leftarrow g_t^*$;
  obtain $\boldsymbol{\Lambda}_t$, $\boldsymbol{B}_t$ and $\boldsymbol{S}_t$ using (15) and (16)
  if ($t = 1$) **then**
    initialize $\tilde{\boldsymbol{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using (18);
    $\delta \leftarrow \sigma_{t,G}$;
  else
    update δ using (19);
    expand $\tilde{\boldsymbol{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using Condition 2;
    update $\tilde{\boldsymbol{\Lambda}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{S}}$ using (21) and (22);
  end if
  construct $\boldsymbol{\mathcal{R}}_1$, $\boldsymbol{\mathcal{R}}_2$,...,$\boldsymbol{\mathcal{R}}_N$ from $\tilde{\boldsymbol{\Lambda}}$ and $\tilde{\boldsymbol{B}}$;
  $t \leftarrow t + 1$;
end while

IV. Experimental Investigation

A. Configuration

In this section, numerical examples based on a variety of benchmark datasets are presented as the proof of concept. A total of 24 public datasets from UCI Machine Learning Repository[1], Keel Dataset Repository[2], Scikit-Multiflow[3] and Meta-Experience Replay[4] are used for performance demonstration. Key information of the 24 benchmark datasets are summarized by Supplementary Table S3.

The proposed algorithms are developed on MATLAB2020b platform, and performance evaluation is conducted on a laptop with dual core i7 processer 2.60GHz×2 and 16.0GB RAM. To allow a certain degree of randomness, the reported results are obtained after 10 Monte Carlo experiments. Source codes of the proposed SOFBIS and SOFBIS+ algorithms can be accessed from: https://github.com/Gu-X/Self-Organizing-Fuzzy-Belief-Inference-System-for-Classification.

The proposed SOFBIS takes the level of granularity, G as the externally controlled parameter and SOFBIS+ takes the regularization parameter, ρ as user input. As aforementioned, the recommended value for G is 6 if the scale of the problem is small (less than 1000 samples, i.e., CR, GL, IR, PI, WI) and 9 if the problem scale is larger. The recommended value of ρ is 10^{-1} for small-scale problems, and 10^{-3} for problems of larger sizes. To better understand the influence of G and ρ upon the performances of SOFBIS and SOFBIS+, sensitivity analysis is conducted using GP, IS, OR and PR datasets, and is reported in Supplementary Section E. Since both SOFBIS and SOFBIS+ are capable of online learning on a chunk-by-chunk basis, the influence of chunk size, K_t upon the performances of the two approaches is also investigated. Note that the four datasets involved in sensitivity analysis are not used for performance demonstration.

Supplementary Table S4 that a greater value of G increases the prediction accuracy of SOFBIS because more prototypes are identified from data. However, this will also increase the computational complexity due to the larger knowledge base. One may also notice that the performance improvement is capped once the value of G is sufficiently large. In general, the recommended value range of G is $[6, 11]$. Similarly, Supplementary Table S5 shows that a smaller value of ρ enables SOFBIS+ to obtain finer partitions of data and mine more knowledge about the local patterns in the form of prototypes. This helps SOFBIS+ to achieve greater classification accuracy. Nevertheless, the learning process of SOFBIS+ becomes more computationally expensive as it takes more iterations before the algorithm identifies the local minimum of the objective function, $J_1(g)$ (Eq. (13)). Hence, the recommended value range of ρ is $[10^{-1}, 10^{-4}]$. It can be seen from Table S6 that the chunk size has little impact on classification accuracy of the proposed approaches unless it is too small, but can largely influence their computational efficiency, especially for SOFBIS+. In general, using a smaller chunk size can speed up the learning processes of SOFBIS and SOFBIS+. However, to achieve good performance, a data chunk needs to contain no less than 40 samples, as suggested by Supplementary Table S6. Hence, it can be concluded that the observations from the sensitivity analysis results coincide with Remarks 2, 5 and 6.

B. Performance Demonstration

In this subsection, numerical examples are conducted based on the aforementioned benchmark datasets for performance investigation.

Firstly, SOFBIS and SOFBIS+ are tested on five small-scale datasets, which include CR, GL, IR, PI and WI. These five datasets are commonly used by mainstream BRB models [40], [44] for performance evaluation. Following the commonly used experimental protocol [40], [44], the prediction accuracy rates (Acc) of SOFBIS and SOFBIS+ are obtained after five-fold cross validation and reported in Table I. Two state-of-the-art BRB models, 1) belief rule based expert system

[1]Available at: https://archive.ics.uci.edu/ml/index.php
[2]Available at: https://sci2s.ugr.es/keel/index.php
[3]Available at: https://scikit-multiflow.github.io/
[4]Available at: https://nlp.stanford.edu/projects/mer/#download

TABLE I
PERFORMANCE COMPARISON ON FIVE SMALL-SCALE DATASETS

Algorithm	IR	WI	GL	CR	PI
SOFBIS	0.9667	0.9662	0.6506	0.9613	0.7461
SOFBIS+	0.9600	0.9662	0.6599	0.9713	0.7356
BRBES	1.0000	0.9944	0.7009	0.9842	0.7910
PMP-BRBS	0.9920	0.9730	0.6720	0.9796	0.7841
SOFIS	0.9600	0.9327	0.5909	0.9541	0.6601
SOFIS+	0.9667	0.9493	0.6257	0.9426	0.6653
SAFIS	0.7533	0.7616	0.2171	0.9699	0.6524
ESAFIS	0.9600	0.9265	0.5697	0.9599	0.7695
ALMMo1	0.8333	0.9660	0.5423	0.9542	0.7669
ALMMo0	0.9733	0.9323	0.5656	0.8694	0.5898
PALM	0.8200	0.9660	0.6263	0.9585	0.7773
eClass0	0.9267	0.9541	0.4189	0.9685	0.5665
eClass1	0.9733	0.9490	0.6234	0.9586	0.7630
SEFIS	0.7467	0.7641	0.2126	0.9570	0.6693

(BRBES) [44] and 2) parallel multi-population belief rule-based system (PMP-BRBS) [40] are used for benchmark comparison, and the prediction results by the two models are directly obtained from [40] and presented in Table I for comparison. In addition, the following 10 EFSs are involved in benchmark comparison under the same experimental protocol, and their performances in terms of Acc are given in Table I as well, which include 1) SOFIS [27]; 2) SOFIS+ [45]; 3) SAFIS [15]; 4) extended sequential adaptive fuzzy inference system (ESAFIS) [50]; 5) autonomous learning multi-model system of first-order (ALMMo1) [51]; 6) autonomous learning multi-model system of zero-order (ALMMo0) [52]; 7) parsimonious learning machine (PALM) [53]; 8) eClass0 [23]; 9) eClass1 [23], and; 10) SEFIS [20].

In this numerical example, the level of granularity, G is set to be 12 and 9 for SOFIS and SOFIS+, respectively, as suggested by [27], [45]. Due to the smaller problem size, SOFBIS, SOFBIS+ and SOFIS+ takes the entire training set as a data chunk, namely, $T = 1$. for SAFIS, its externally controlled parameters are set as follows based on the recommendation of [15]: $\gamma = 0.999$; $\varepsilon_{min} = 0.1$; $\varepsilon_{max} = 1.2$; $K = 1.5$; $e_g = 0.01$, and; $e_p = 0.001$. The parameter setting of ESAFIS is as follows: $\gamma = 0.999$; $\varepsilon_{min} = 0.1$; $\varepsilon_{max} = 1.2$; $K = 1.5$; $e_g = 0.01$, and; $e_p = 0.01$, and; $M = 20$. ALMMo1, ALMMo0, eClass0 and eClass1 follow the exact same settings as [23], [51], [52]. The first-order PALM with local updating strategy is used for experimental comparison and its parameter setting is as follows: $a = 0.1$; $b_1 = 0.002$; $b_2 = 0.01$; $c_1 = 0.01$, and; $c_2 = 0.01$. For SEFIS, its externally controlled parameters are determined as: $K = 0.5$; $\delta_1 = 0.5$; $\delta_2 = 0.5$, and $p_0 = 2$. Among the 10 EFSs used for experimental comparison, SOFIS, SOFIS+, ALMMo0 and eClass0 are zero-order EFSs designed for classification tasks. For the remaining six first-order models, only eClass1 is designed for classification, the other five are multi-input single-output models designed for regression purposes. Hence, in this study, SAFIS, ESAFIS, ALMMo1, PALM and SEFIS employ the "one-versus-all" strategy for classification.

For visual clarity, average classification accuracy rates on the five problems by the 14 rule-based models are depicted in

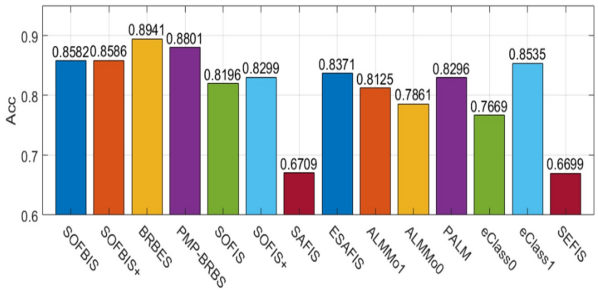


Fig. 2: Average classification accuracy (Acc) rates of different rule-based models on five small-scale datasets.

Fig. 2. One can see from Fig. 2 that the proposed SOFBIS and SOFBIS+ outperform the 10 alternative EFSs on the five small-scale datasets in terms of average classification accuracy. It can also be seen from Fig. 2 that BRBES and PMP-BRBS are able to provide greatest classification accuracy thanks to the iterative parameter optimization processes utilizing evolutionary algorithms. In contrast, EFSs (including the proposed SOFBIS and SOFBIS+) are implemented for data streams. Typically, a EFS learns from data streams in a "single pass", sample-by-sample or chunk-by-chunk manner without revisiting processed data to maintain its computational efficiency at a high level. As a result, the system structure and learned parameters of EFSs are often not optimal. The key difference in the learning schemes leads to the performance gap between EFSs and BRB models.

Next, the performances of SOFBIS and SOFBIS+ are evaluated on the following 10 widely used benchmark classification problems: CA, GC, LR, MA, MF, MG, PB, PW, SH, and WF. In this example, for each dataset, 50% of data samples are randomly selected out to construct the training set, and the remaining samples are used for testing. The 10 EFSs used in the previous numerical example are also involved for benchmark comparison using the same parameters settings as before. During the experiments, each training set is divided into three chunks evenly for SOFBIS, SOFBIS+ and SOFIS+, namely, $T = 3$. Prediction accuracy rates of SOFBIS, SOFBIS+ and 10 EFS competitors on the 10 datasets are reported in Table II. The average Acc rates and training time costs (t_{tra} in seconds) of 12 approaches are given in Figs. 3 and 4, respectively.

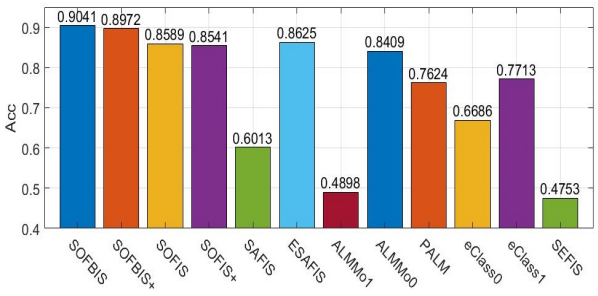


Fig. 3: Average classification accuracy (Acc) comparison on 10 benchmark datasets

It can be observed from Table II and Figs. 3-4 that SOFBIS and SOFBIS+ are able to achieve the top performance

TABLE II
PERFORMANCE COMPARISON ON 10 BENCHMARK DATASETS

Algorithm	Dataset									
	CA	GC	LR	MA	MF	MG	PB	PW	SH	WF
SOFBIS	0.8967	0.7118	0.9462	0.9843	0.9633	0.8159	0.9589	0.9550	0.9060	0.9028
SOFBIS+	0.8916	0.6986	0.9441	0.9825	0.9632	0.7827	0.9491	0.9546	0.8951	0.9106
SOFIS	0.8792	0.6488	0.9289	0.8330	0.9200	0.7695	0.9414	0.9361	0.8918	0.8406
SOFIS+	0.8838	0.6530	0.9412	0.7180	0.9393	0.7697	0.9494	0.9423	0.8854	0.8591
SAFIS	0.7833	0.6958	0.0900	0.9772	0.1000	0.7159	0.9067	0.8918	0.2971	0.5556
ESAFIS	0.8988	0.6574	0.9266	0.9823	0.9722	0.8537	0.9496	0.9418	0.5957	0.8466
ALMMo1	0.6021	0.5316	0.1568	0.5334	0.4216	0.5324	0.8983	0.5452	0.3259	0.3503
ALMMo0	0.8432	0.6662	0.9190	0.7040	0.9329	0.7248	0.9492	0.9426	0.8918	0.8352
PALM	0.8611	0.7608	0.1729	0.9576	0.9741	0.7833	0.7670	0.9199	0.7892	0.6384
eClass0	0.6747	0.5684	0.4905	0.9044	0.7861	0.6313	0.6899	0.7758	0.6946	0.4707
eClass1	0.8792	0.7298	0.7052	0.2156	0.9515	0.8302	0.8842	0.9286	0.7829	0.8056
SEFIS	0.7024	0.5546	0.0980	0.9283	0.3786	0.6502	0.0216	0.6592	0.3761	0.3842

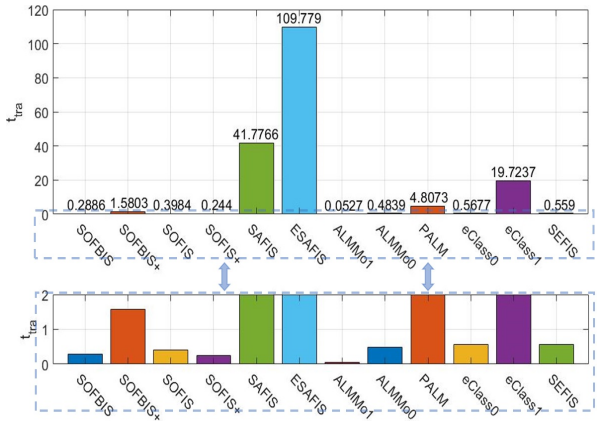


Fig. 4: Average training time cost (t_{tra}) comparison on 10 benchmark datasets

across 10 different benchmark problems for classification, outperforming 10 rule-based competitors. In addition, the computational efficiency of SOFBIS is also among the highest. To examine the statistical significance of the performance improvements achieved by SOFBIS and SOFBIS+, over the 10 competitors on the 10 benchmark problems, cross-group Kruskal-Wallis H-tests [54] are firstly conducted, where the cascaded classification results by each approach across the 10 experiments are used. The outcomes of the statistical tests in terms of p-value are presented in Supplementary Table S7. It can be seen from the table that all the p-values returned from the cross-group tests are equal to 0, whereas a p-value smaller than the level of significance ($\alpha = 0.05$) indicates that the null hypothesis is rejected, suggesting that there is no significant difference between populations. The results of the cross-group tests reveal that the predictions made by SOFBIS, SOFBIS+ and others are significantly different. To further examine the statistical significance, pairwise Kruskal-Wallis H-tests [54] are then conducted, and the p-values adopted from the tests are reported in Supplementary Tables S8 and S9. It can be observed that 75.00% of the p-values returned by the pairwise tests are below 0.05 for both SOFBIS and SOFBIS+. The results further confirm that the performances of SOFBIS and SOFBIS+ are significantly better than the other 10 rule-based models. This numerical example as well as the previous one (shown in Table I and Fig. 2) demonstrates the superiority of the proposed approaches over alternative EFSs by offering the greatest classification accuracy. This justifies the effectiveness and validity of the proposed concept and general principles. It is also interesting to notice that performances of zero-order EFSs are generally stronger than first-order ones , e.g., SAFIS, ALMMo1, PALM, SEFIS, on classification problems.

The classification performances of SOFBIS and SOFBIS+ are further compared with 12 alternative mainstream single-model and ensemble classifiers, including SVM, RF and XGBoost [55], etc., on the 10 benchmark problems and full details of the performance comparison are reported in Supplementary Section G. Supplementary Table S9 and Fig. S1 show that SOFBIS achieves greater performance than all single-model classifiers as well as the majority of ensemble models, and is only outperformed by XGBoost.

Finally, the performance of SOFBIS is tested on five large-scale nonstationary classification problems, including HP, PMN, RMN, SE and SU. Experiments are performed under the "prequential test-then-train" environment [56], and the chunk size for SOFBIS is set as $K_t = 1000$. Performance of SOFBIS in terms of accumulated one-chunk ahead prediction accuracy, Acc and t_{tra} (in seconds) is reported in Table III. Note that SOFBIS+ is not suitable for handling drastic changes in data patterns and it is not involved in the experiments.

The performance of SOFBIS is further compared with the following state-of-the-art classifiers for data stream classification: 1) multilayer self-evolving recurrent neural network (MUSE-RNN) [56]; 2) neural network with dynamically evolved capacity (NADINE) [57]; 3) autonomous deep learning (ADL) [58]; 4) parsimonious ensemble+ (pENsemble+) [59], and; 5) deep evolving denoising autoencoder (DEVDAN) [60]. The results of these comparative approaches are obtained directly from [56], [58], [60] and reported in Table III as well. One can see from Table III that SOFBIS outperforms the five classification approaches on four out of five cases, showing its strong capability to handle large-scale data stream problems in nonstationary environments.

In short, all the numerical experiments carried out so far (Tables I-III, Supplementary Table S9 and Fig. S1) collectively demonstrate the superior performances of SOFBIS and SOF-

TABLE III
TEST-THEN-TRAIN PERFORMANCE COMPARISON ON NONSTATIONARY PROBLEMS

Algorithm	Meas.	Dataset				
		HP	PMN	RMN	SE	SU
SOFBIS	Acc	0.9846±0.0011	0.9558±0.0000	0.9527±0.0005	0.9924±0.0000	0.7783±0.0002
	t_{tra}	7	290	250	5	5773
MUSE-RNN	Acc	0.9264±0.0215	0.8387±0.1342	0.7627±0.0490	0.9237±0.0611	0.7814±0.0160
	t_{tra}	250	416	190	116	21,000
NADINE	Acc	-	0.7764±0.1509	0.7451±0.0750	0.9224±0.0640	0.7803±0.0300
	t_{tra}	-	202	192	15	1455
ADL	Acc	0.9233±0.0263	0.6840±0.2417	0.7290±0.0935	0.9282±0.0579	0.7826±0.0280
	t_{tra}	22	212	199	18	2500
pEnsemble+	Acc	0.8760±0.0620	-	-	0.9200±0.0600	0.7699±0.0460
	t_{tra}	120	-	-	230	35,000
DEVDAN	Acc	0.9119±0.0328	0.7667±0.1400	0.7648±0.0097	0.9112±0.0711	-
	t_{tra}	-	-	-	-	-

BIS+ over their competitors, showing their significant potential as powerful tools for data stream classification.

V. CONCLUSION

This paper has presented a novel EFS with a belief structure named SOFBIS for data stream classification. SOFBIS learns from data streams on a chunk-by-chunk basis and self-organizes the resulting knowledge base in the form of prototypes that reflect the underlying ensemble properties while considering the inter-class overlaps. SOFBIS enables users to determine the level of granularity of the learning outcomes with guaranteed objectiveness and meaningfulness. In addition, SOFBIS can self-determine the most suitable level of granularity for classification based on the mutual distances of data. Numerical examples have been provided, demonstrating the superior classification performance of SOFBIS and its variant over alternative EFSs and other mainstream classifiers. Particularly, the proposed approaches have been shown to be of great applicability for practically challenging classification problems that involve data streams of a nonstationary nature.

There are several considerations for future work. First, the proposed SOFBIS and SOFBIS+ assume that each rule and each input attribute are of an equal weight respectively, only learning the premise and consequent parts and the associated belief degrees of fuzzy rules. By incorporating a certain evolving scheme to automatically update the rule and attribute weights, one can expect a significant improvement in terms of their classification accuracy. Second, the proposed SOFBIS+ can self-determine the most suitable level of granularity for classification from data, but drastic changes of underlying data patterns can severely damage the performance of the learned model because the soft thresholds derived from different data chunks may differ significantly. Hence, it would be very helpful to design an alternative scheme to allow SOFBIS+ to estimate the soft thresholds in a more robust way. Last but not the least, one can see from numerical examples that BRB models achieve more accurate classification than EFSs thanks to their iterative optimisation process. Yet, as with other EFSs, both SOFBIS and SOFBIS+ generalize data in a "single pass" manner, and their structure and parameters are not necessarily optimized. Therefore, it would be interesting to investigate how SOFBIS and SOFBIS+ may perform if their parameters are further optimized by an evolutionary algorithm. Designing a computationally efficient approach for near real-time optimisation would make such further research practically more useful.

REFERENCES

[1] D. Leite, I. Skrjanc, and F. Gomide, "An overview on evolving systems and learning from stream data," *Evol. Syst.*, vol. 11, no. 2, pp. 181–198, 2020.

[2] Z. Chi, H. Yan, and T. Pham, *Fuzzy algorithms*: *with applications to image processing and pattern recognition*. World Scientific., 1996.

[3] A. Fernandez, F. Herrera, O. Cordon, M. Jose Del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: why, when, what for, and where to?," *IEEE Comput. Intell. Mag.*, vol. 14, no. 1, pp. 69–81, 2019.

[4] F. Aghaeipoor, M. M. Javidi, and A. Fernandez, "IFC-BD: an interpretable fuzzy classifier for boosting explainable artificial intelligence in big data," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 3, pp. 830–840, 2022.

[5] J. Kluska and M. Madera, "Extremely simple classifier based on fuzzy logic and gene expression programming," *Inf. Sci. (Ny).*, vol. 571, no. 27, pp. 560–579, 2021.

[6] P. Angelov and D. Filev, "An approach to online identification of Takagi-Sugeno fuzzy models," *IEEE Trans. Syst. Man, Cybern. - Part B Cybern.*, vol. 34, no. 1, pp. 484–498, 2004.

[7] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 667–677, 2002.

[8] N. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, 2002.

[9] I. Skrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey," *Inf. Sci. (Ny).*, vol. 490, pp. 344–368, 2019.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Crambridge, MA: MIT Press, 2016.

[11] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.

[12] L. Breiman, "Random forests," *Mach. Learn. Proc.*, vol. 45, no. 1, pp. 5–32, 2001.

[13] E. Lughofer and P. Angelov, "Handling drifts and shifts in on-line data streams with evolving fuzzy systems," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2057–2068, 2011.

[14] E. Lughofer, Evolving fuzzy systems-methodologies, advanced concepts and applications. Berlin: Springer, 2011.

[15] H. Rong, N. Sundararajan, G. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction," *Fuzzy Sets Syst.*, vol. 157, no. 9, pp. 1260–1275, 2006.

[16] M. Pratama, S. Anavatti, P. Angelov, and E. Lughofer, "PANFIS: a novel incremental learning machine," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 1, pp. 55–68, 2014.

[17] R. Bao, H. Rong, P.Angelov, B. Chen, and P. Wong, "Correntropy-based evolving fuzzy neural system," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1324–1338, 2018.

[18] H. Rong, Z. Yang, and P. K. Wong, "Robust and noise-insensitive recursive maximum correntropy-based evolving fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2277–2284, 2019.

[19] D. Ge and X. Zeng, "Learning data streams online - an evolving fuzzy system approach with self-learning/adaptive thresholds," *Inf. Sci. (Ny).*, vol. 507, pp. 172–184, 2020.

[20] Z. Yang, H. Rong, P. Angelov, and Z. Yang, "Statistically evolving fuzzy inference system for non-Gaussian noises," *IEEE Trans. Fuzzy Syst.*, DOI: 10.1109/TFUZZ.2021.3090898, 2021.

[21] H. Huang, H. Rong, Z. Yang, and C. Vong, "Jointly evolving and compressing fuzzy system for feature reduction and classification," *Inf. Sci. (Ny).*, vol. 579, pp. 218–230, 2021.

[22] E. Lughofer, M. Pratama, and I. Skrjanc, "Online bagging of evolving fuzzy systems," *Inf. Sci. (Ny).*, vol. 570, pp. 16–33, 2021.

[23] P. Angelov and X. Zhou, "Evolving fuzzy-rule based classifiers from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1462–1474, 2008.

[24] E. Lughofer, "FLEXFIS: a robust incremental learning approach for evolving Takagi-Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1393–1410, 2008.

[25] M. Pratama, S. Anavatti, and E. Lughofer, "Genefis: toward an effective localist network," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 3, pp. 547–562, 2014.

[26] D. Dovzan, V. Logar, and I. Skrjanc, "Implementation of an evolving fuzzy model (eFuMo) in a monitoring system for a waste-water treatment process," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1761–1776, 2015.

[27] X. Gu and P.Angelov, "Self-organising fuzzy logic classifier," *Inf. Sci. (Ny).*, vol. 447, pp. 36–51, 2018.

[28] D. Ge and X. Zeng, "A self-evolving fuzzy system which learns dynamic threshold parameter by itself," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 8, pp. 1625–1637, 2018.

[29] S. Samanta, M. Pratama, and S. Sundaram, "A novel spatio-temporal fuzzy inference system (SPATFIS) and its stability analysis," *Inf. Sci. (Ny).*, vol. 505, pp. 84–99, 2019.

[30] P. de Campos Souza, "Fuzzy neural networks and neuro-fuzzy networks: a review the main techniques and applications used in the literature," *Appl. Soft Comput.*, vol. 92, p. 106275, 2020.

[31] J. Liu, J.Yang, D. Ruan, L. Martinez, and J. Wang, "Self-tuning of fuzzy belief rule bases for engineering system safety analysis," *Ann. Oper. Res.*, vol. 163, no. 1, pp. 143–168, 2008.

[32] Q. Shen and R. Zhao, "A credibilistic approach to assumption-based truth maintenance," *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 41, no. 1, pp. 85–96, 2011.

[33] J. Yang, J. Liu, J. Wang, H. Sii, and H. Wang, "Belief rule-base inference methodology using the evidential reasoning approach - RIMER," *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 36, no. 2, pp. 266–285, 2006.

[34] G. Shafer, *A mathematical theory of evidence*. Princeton University Press, 1976.

[35] Z. Zhou, G. Hu, C. Hu, C. Wen, and L. Chang, "A survey of belief rule-base expert system," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 8, pp. 4944–4958, 2021.

[36] Y. You, J. Sun, Y. wang Chen, C. Niu, and J. Jiang, "Ensemble Belief Rule-Based Model for complex system classification and prediction," *Expert Syst. Appl.*, vol. 164, p. 113952, 2021.

[37] J. Liu, L. Martinez, A. Calzada, and H. Wang, "A novel belief rule base representation, generation and its inference methodology," *Knowledge-Based Syst.*, vol. 53, pp. 129–141, 2013.

[38] X. Fu and Q. Shen, "Fuzzy compositional modeling," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 823–840, 2010.

[39] B. Qian, Q. Wang, R. Hu, Z. Zhou, C. Yu, and Z. Zhou, "An effective soft computing technology based on belief-rule-base and particle swarm optimization for tipping paper permeability measurement," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 3, pp. 841–850, 2019.

[40] W. Zhu, L. Chang, J. Sun, G. Wu, X. Xu, and X. Xu, "Parallel multipopulation optimization for belief rule base learning," *Inf. Sci. (Ny).*, vol. 556, pp. 436–458, 2021.

[41] J. Yang, J. Liu, D.Xu, J. Wang, and H. Wang, "Optimization models for training belief-rule-based systems," *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 37, no. 4, pp. 569–585, 2007.

[42] L. Chang, Y. Zhou, J. Jiang, M. Li, and X. Zhang, "Knowledge-based systems structure learning for belief rule base expert system: a comparative study," *Knowledge-Based Syst.*, vol. 39, pp. 159–172, 2013.

[43] L. Chang, Z. Zhou, Y. Chen, T. Liao, Y. Hu, and L. Yang, "Belief rule base structure and parameter joint optimization under disjunctive assumption for nonlinear complex system modeling," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1542–1554, 2018.

[44] L. Chang, Z. Zhou, Y. You, L. Yang, and Z. Zhou, "Belief rule based expert system for classification problems with new rule activation and weight calculation procedures," *Inf. Sci. (Ny).*, vol. 336, pp. 75–91, 2016.

[45] X. Gu, P. Angelov, and Z. Zhao, "Self-organizing fuzzy inference ensemble system for big streaming data classification," *Knowledge-Based Syst.*, vol. 218, p. 106870, 2021.

[46] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *Int. J. Gen. Syst.*, vol. 41, no. 2, pp. 163–185, 2012.

[47] T. Kodinariya and P. Makwana, "Review on determining number of cluster in K-means clustering," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 1, no. 6, pp. 2321–7782, 2013.

[48] X. Gu and M. Li, "A multi-granularity locally optimal prototype-based approach for classification," *Inf. Sci. (Ny).*, vol. 569, pp. 157–183, 2021.

[49] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*, 2nd ed. Chichester, England: John Wiley & Sons., 1999.

[50] H. Rong, N. Sundararajan, G. Huang, and G.Zhao, "Extended sequential adaptive fuzzy inference system for classification problems," *Evol. Syst.*, vol. 2, no. 2, pp. 71–82, 2011.

[51] P. Angelov, X. Gu, and J. Principe, "Autonomous learning multimodel systems from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 2213-2224, 2018.

[52] P. Angelov and X. Gu, "Autonomous learning multi-model classifier of 0-order (ALMMo-0)," in *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 2017, pp. 1-7.

[53] M. Ferdaus, M. Pratama, S. Anavatti, and M. Garratt, "PALM: an incremental construction of hyperplanes for data stream regression," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2115–2129, 2019.

[54] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Am. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952.

[55] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[56] M. Das, M. Pratama, S. Savitri, and J. Zhang, "MUSE-RNN: a multilayer self-evolving recurrent neural network for data stream classification," in *IEEE International Conference on Data Mining*, 2019, pp. 110–119.

[57] M. Pratama, C. Zaiin, A. Ashfahani, Y. S. Ong, and W. Ding, "Automatic construction of multi-layer perceptron network from streaming examples," in *International Conference on Information and Knowledge Management*, 2019, pp. 1171–1180.

[58] A. Ashfahani and M. Pratama, "Autonomous deep learning: continual learning approach for dynamic environments," in *SIAM International Conference on Data Mining*, 2019, pp. 666–674.

[59] M. Pratama, E. Dimla, T. Tjahjowidodo, W. Pedrycz, and E. Lughofer, "Online tool condition monitoring based on parsimonious ensemble+," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 664–677, 2020.

[60] A. Ashfahani, M. Pratama, E. Lughofer, and Y. Ong, "DEVDAN: deep evolving denoising autoencoder," *Neurocomputing*, vol. 390, pp. 297–314, 2020.