Autonomous Learning Multi-Model Classifier of 0-Order (ALMMo-0)

Plamen Angelov, *Fellow, IEEE* and Xiaowei Gu, *Student Member, IEEE*
Data Science Group,
School of Computing and Communications,
Lancaster University, Lancaster, UK
E-mail: {p.angelov, x.gu3}@lancaster.ac.uk

***Abstract*—In this paper, a new type of 0-order multi-model classifier, called Autonomous Learning Multiple-Model (ALMMo-0), is proposed. The proposed classifier is non-iterative, feedforward and entirely data-driven. It automatically extracts the data clouds from the data per class and forms 0-order AnYa type fuzzy rule-based (FRB) sub-classifier for each class. The classification of new data is done using the "winner takes all" strategy according to the scores of confidence generated objectively based on the mutual distribution and ensemble properties of the data by the sub-classifiers. Numerical examples based on benchmark datasets demonstrate the high performance and computation-efficiency of the proposed classifier.**

Keywords—multi-model classifier; autonomous; data-driven; AnYa fuzzy rule-based (FRB) system

I. INTRODUCTION

As a general problem of pattern recognition, classification is considered to be a supervised machine learning technique of building a mathematical function to determine whether a data sample is a part of a set (or probably several sets) or not. Based on a training set of observed data whose category memberships are known, the system can learn to identify the category to which new observations belong. Nowadays, classification techniques have been widely used in different fields like natural language processing [1], [2], image processing [3], [4], etc. Fuzzy logic is now often used in the classification tasks [5], [6]. There are many methods for automatically generating and learning fuzzy IF-THEN rules from data for pattern recognition problems [7]–[11].

Autonomous Learning Multiple-Model (ALMMo) system [12] was recently introduced in the form of an AnYa FRB system [10], [11] within the Empirical Data Analytics (EDA) [13]–[15] framework to extract shape-free data clouds [10], [11] and autonomously form simple linguistic rules from the empirically observed data. The ALMMo system forms its multi-model structure in an entirely data-driven way without making *prior* assumptions. All the meta-parameters within ALMMo system are obtained directly from the data and can be updated recursively, which ensures its memory- and computational efficiency.

In this paper, we introduce the 0-order Autonomous Learning Multiple-Model (ALMMo-0) classifier on the basis of the 0-order AnYa type fuzzy rule-based (FRB) systems [10], [11] in a multiple-model architecture [9]. This classifier is non-parametric, non-iterative and fully autonomous. There is no need to train any parameters due to its feedforward structure. The proposed classifier automatically identifies the focal points from the empirically observed data and forms data clouds resembling Voronoi tessellation [16] per class. Then, sub-classifiers corresponding to different classes are built in a form of a set of AnYa type of fuzzy rules formed from the non-parametric data clouds. For any new data sample, each AnYa FRB sub-classifier generates a score of confidence objectively and the label is assigned to the new data sample based on the "winner takes all" rule. As the proposed ALMMo classifier learns from the data and conducts classification based on very fundamental principles, a variety of modification and extension can further be done, i.e. using the fuzzy rules with 1st order consequent part. Numerical examples in this paper demonstrate the excellent performance of the proposed 0-order ALMMo classifier and show the proposed classifier to be a strong alternative to the well-known classical classifiers [17]–[20].

The remainder of this paper is organized as follows. Section II briefly recalls the concepts of the 0-order AnYa FRB system and the nonparametric EDA estimator. The details of the proposed 0-order ALMMo classifier are described in section III. Numerical examples based on benchmark problems are presented in section IV followed by a discussion in section V. This paper is concluded by section VI.

II. BASIC CONCEPTS

The ALMMo system [12] was recently introduced within the EDA framework [13]–[15]. In this section, the concepts of the 0-order AnYa FRB system and the EDA estimator will be briefly recalled.

A. 0-Order AnYa Fuzzy Rule-Based Systems

The structure of an ALMMo system is composed of a set of AnYa fuzzy rules. AnYa type fuzzy system was introduced by Angelov and Yager [10], [11]. Compared with the two widely used FRB systems, namely, the Mamdani type [21], [22] and the Takagi-Sugeno type [23], [24], the antecedent part of the AnYa fuzzy rule is revised and simplified into a vector, which includes the focal points of the data clouds on

This work was supported by The Royal Society (Grant number IE141329/2014)

which the rules are built upon. The concept of a data cloud was also introduced in [10], [11]. Data clouds are sets of data samples with common properties grouped around the focal points resembling Voronoi tessellation [16].

In AnYa, the data clouds and the respective focal points are used as the basis of the antecedent (IF part) of the fuzzy rule. A 0-order AnYa fuzzy rule is expressed as follows [8], [10], [11].

$$\text{Rule}^{\,i}\text{:}\ \ IF\ \left(\boldsymbol{x} \sim \boldsymbol{x}^{*i}\right)\quad THEN\ \left(Label^{i}\right) \tag{1}$$

where $\boldsymbol{x}^{*i}$ is the focal point of the i^{th} data cloud; $Label^{i}$ is the corresponding label. The inference in the 0-order AnYa fuzzy rule can be done following the well-known "winner takes all" principle when classification is considered.

B. EDA Estimator

In this paper, we will employ the *unimodal density* [12] from the EDA framework as the main estimator for disclosing the ensemble properties from the observed data in a fully autonomous way.

Firstly, let us consider the data set/stream in the Euclidean data space $\mathbf{R}^{d}$ as $\{\boldsymbol{x}\} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k\}$ and the subscripts indicate the time instances at which the data samples were observed. In this paper, the Euclidean distance is used for mathematical derivation for simplicity, but other types of distances can be used as well.

The *unimodal density* of the i^{th} data sample at the k^{th} time instance is calculated as [12]–[15]:

$$D_k\left(\boldsymbol{x}_i\right) = \frac{1}{1 + \frac{\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\right\|^2}{\sigma_k^2}} = \frac{1}{1 + \frac{\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\right\|^2}{X_k - \left\|\boldsymbol{\mu}_k\right\|^2}} \tag{2}$$

where $\boldsymbol{\mu}_k$ is the global mean of all the data samples at the k^{th} time instance and X_k is the average scalar product; $\sigma_k^2 = X_k - \left\|\boldsymbol{\mu}_k\right\|^2$. It is worth to be noticed that, using Euclidean distance, the *unimodal density* is in the form of a Cauchy function in its nature, but this was not a *prior* assumption of a Cauchy distribution.

For streaming data processing, recursive calculation is very important for improving the memory- and computation-efficiency. $\boldsymbol{\mu}_k$ and X_k can be updated recursively as equations (3) and (4), which allows the *unimodal density* to be calculated recursively without any loop and iterations:

$$\boldsymbol{\mu}_k = \frac{k-1}{k}\boldsymbol{\mu}_{k-1} + \frac{1}{k}\boldsymbol{x}_k;\quad \boldsymbol{\mu}_1 = \boldsymbol{x}_1 \tag{3}$$

$$X_k = \frac{k-1}{k}X_{k-1} + \frac{1}{k}\left\|\boldsymbol{x}_k\right\|^2;\quad X_1 = \left\|\boldsymbol{x}_1\right\|^2 \tag{4}$$

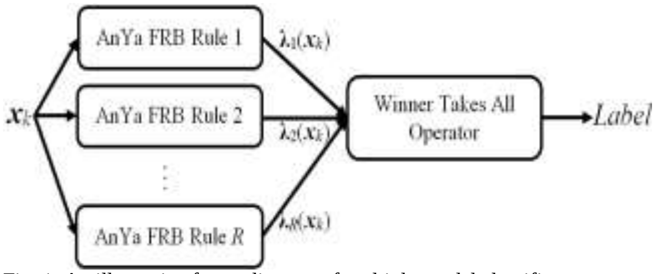

Fig. 1. An illustrative frame diagram of multiple-model classifier

III. THE PROPOSED 0-ORDER ALMMO CLASSIFIER

A. Multiple-Model Architecture

The multiple-model architecture is based on R FRB rules ($R \geq C$, where C represents the different classes in the data set/stream). An illustrative diagram of the classifier with multiple- model architecture is depicted in Fig. 1. As it is demonstrated in Fig. 1, each new coming data sample, denoted as $\boldsymbol{x}_k$, is sent to all the existing sub-classifiers and each sub-classifier generates a score of confidence, denoted as λ_i ($i = 1, 2, ..., R$) indicating the degree of confidence by the FRB rule in the claim that the new data sample belongs to a specific class. Then, the "winner takes all" rule is used for assigning $\boldsymbol{x}_k$ to the class it is most likely to be associated with:

$$Label = \underset{i=1,2,...,R}{\arg\max}\left(\lambda_i\right) \tag{5}$$

In the proposed classifier, we use the multiple-model structure to enhance the ability of the proposed classifier to handle complex problems.

B. Learning stage

In this subsection, the learning stage of the proposed ALMMo classifier will be described in detail.

Due to the multiple-model architecture of the proposed classifier, only the AnYa FRB rules corresponding to the new data sample's class will be updated. For each newly arrived data sample, it will be normalized by its norm, namely:

$$\boldsymbol{x} \leftarrow \frac{\boldsymbol{x}}{\left\|\boldsymbol{x}\right\|} \tag{6}$$

This type of normalization enhances the classifier's ability for high-dimensional data processing [25].

Let us assume the new data sample is the k^{th} data sample of the i^{th} class, thus, the normalized data sample is denoted as $\boldsymbol{x}_k^i$.

Firstly, the global mean $\boldsymbol{\mu}_{k-1}^{i}$ of the i^{th} class is updated to $\boldsymbol{\mu}_k^i$ using equation (3). There is no need to update the average scalar product anymore because $X_k^i = \left\|\boldsymbol{x}_k^i\right\|^2 = 1$ since the data is being normalized. The *unimodal densities* of the data sample $\boldsymbol{x}_k^i$ and all the identified focal points of the i^{th} class, denoted as $\boldsymbol{x}_j^{*i}$ ($j = 1, 2, ..., F^i$) are calculated using equation (2), where F^i is the number of focal points.

Then, the following principle (Condition 1) is checked to see whether $\boldsymbol{x}_k^i$ will generate a new rule/data cloud [11]:

$$\begin{aligned}&IF\ \left(D_k\left(\boldsymbol{x}_k^i\right) > \max_{j=1,2,\dots,F_i}\left(D_k\left(\boldsymbol{x}_j^{*i}\right)\right)\right)\\ &OR\ \left(D_k\left(\boldsymbol{x}_k^i\right) < \min_{j=1,2,\dots,F_i}\left(D_k\left(\boldsymbol{x}_j^{*i}\right)\right)\right)\\ &THEN\ \left(\boldsymbol{x}_k^i\ is\ a\ new\ focal\ point\right)\end{aligned} \tag{7}$$

If Condition 1 is triggered, a new fuzzy rule/data cloud is being formed around $\boldsymbol{x}_k^i$ and its parameters are being updated as follows:

$$\begin{cases} F^i \leftarrow F^i + 1 \\ \boldsymbol{x}_{F^i}^{*i} \leftarrow \boldsymbol{x}_k^i \\ M_{F^i}^{*i} \leftarrow 1 \\ r_{F^i}^{*i} \leftarrow r_o \end{cases} \tag{8}$$

where $M_{F^i}^{*i}$ is the number of members of the data cloud; $r_{F^i}^{*i}$ is the radius of the influence area; r_o is a small value to stabilize the initial status of the newborn data clouds. In this paper, we use $r_o = \sqrt{2\left(1-\cos(15^o)\right)}$ [25]. We need to stress that, r_o is not a problem-specific parameter and requires no *prior* knowledge to decide. It is for preventing the newborn data clouds from attracting data samples that are not close enough. It defines a degree of closeness that is interesting and distinguishable.

If Condition 1 (equation (7)) is not satisfied, then the algorithm continues by finding the nearest data cloud to $\boldsymbol{x}_k^i$:

$$\boldsymbol{x}_N^{*i} = \arg\min_{j=1,2,\dots,F^i}\left(\left\|\boldsymbol{x}_k^i - \boldsymbol{x}_j^{*i}\right\|\right) \tag{9}$$

where $\boldsymbol{x}_N^{*i}$ denotes the focal point of the nearest data cloud.

Before $\boldsymbol{x}_k^i$ is assigned to the nearest data cloud, Condition 2 is being checked to see whether $\boldsymbol{x}_k^i$ is close to the data cloud or not:

$$\begin{aligned}&IF\ \left(\left\|\boldsymbol{x}_k^i - \boldsymbol{x}_N^{*i}\right\| \le r_N^{*i}\right)\\ &THEN\ \left(\boldsymbol{x}_k^i\ is\ assigned\ to\ the\ nearest\ data\ cloud\right)\end{aligned} \tag{10}$$

If Condition 2 is satisfied, the meta-parameters of the nearest fuzzy rule/data cloud are updated as follows:

$$\begin{cases} \boldsymbol{x}_N^{*i} \leftarrow \dfrac{M_N^{*i}}{M_N^{*i}+1}\boldsymbol{x}_N^{*i} + \dfrac{1}{M_N^{*i}+1}\boldsymbol{x}_k^i \\ M_N^{*i} \leftarrow M_N^{*i} + 1 \\ r_N^{*i} \leftarrow \sqrt{0.5\left(\left(r_N^{*i}\right)^2 + \left(1-\left\|\boldsymbol{x}_N^{*i}\right\|^2\right)\right)} \end{cases} \tag{11}$$

On the contrary, if Condition 2 is not met, a new fuzzy rule/data cloud is formed around $\boldsymbol{x}_k^i$ using equation (8).

For the data clouds that do not receive new members, the parameters of the other fuzzy rules/data clouds stay the same for the next processing cycle.

The main procedure of the learning stage of the proposed classifier is summarized as follows.

The learning stage of the ***0-order ALMMo classifier***

While the new data sample of the i^{th} class $\boldsymbol{x}_k^i$ is available

i. $\boldsymbol{x}_k^i \leftarrow \dfrac{\boldsymbol{x}_k^i}{\left\|\boldsymbol{x}_k^i\right\|}$

ii. **If** ($k=1$) **Then**

1. $\boldsymbol{\mu}_1^i \leftarrow \boldsymbol{x}_1^i$
2. $F^i \leftarrow 1$
3. $\boldsymbol{x}_1^{*i} \leftarrow \boldsymbol{x}_1^i$
4. $M_1^{*i} \leftarrow 1$
5. $r_1^{*i} \leftarrow r_o$

iii. **Else**

1. Update $\boldsymbol{\mu}_{k-1}^i$ to $\boldsymbol{\mu}_k^i$ using eq. (3);
2. Calculate $D_k\left(\boldsymbol{x}_k^i\right)$ using eq. (2);
3. Update $D_k\left(\boldsymbol{x}_j^{*i}\right)$ ($j=1,2,\dots,F^i$) using eq. (2);
4. **If** (Condition 1 (eq. (7)) is met) **Then**
   - Add a new data cloud using eq. (8);
5. **Else**
   - Find the nearest data cloud using eq. (9);
   - **If** (Condition 2 (eq. (7)) is met) **Then**
     * Update the meta-parameters of the nearest data cloud using eq. (11);
   - **Else**
     * Add a new data cloud using eq. (8);
   - **End If**
6. **End If**

iv. **End If**

End While

C. *Validation Stage*

In this subsection, we will describe the procedure of the proposed ALMMo classifier to generate labels for the validation data samples.

Each validation data sample is being sent to all the AnYa FRB sub-classifiers corresponding to the C classes of the dataset. As each class may have several AnYa type of fuzzy rules (R may be larger than C), the output, namely, the score of confidence of each AnYa FRB rule is given in the following way ($j=1,2,\dots,R$):

$$\text{Rule}^{\,j}: \begin{aligned}&IF\ \left(\boldsymbol{x}_k \sim \boldsymbol{x}_j^*\right)\\ &THEN\ \left(\lambda_j = \exp\left(-\frac{1}{2}\left\|\boldsymbol{x}_k - \boldsymbol{x}_j^*\right\|^2\right)\right)\end{aligned} \tag{12}$$

After all the R AnYa FRB rules generate their scores of confidence, the “winner takes all” operator (equation (5)) will be used to select out the most confident rule and assign the validation data sample the corresponding label.

IV. Numerical Examples

In this section, a number of benchmark problems will be considered as numerical examples for evaluating the performance of the proposed 0-order ALMMo classifier.

A. *MONK’s Problem Dataset* [26]

The first numerical example is based on the well-known MONK’s 2nd problem dataset [26]. It contains 432 data samples with 6 attributes (a1 to a6) and 1 label. There are 169 data samples in the training set and 432 samples in the testing set.

The performance of the proposed classifier is further compared with the following well-known widely-used algorithms:

i) SVM classifier with Gaussian kernel [17];

ii) Naïve Bayes classifier [18];

iii) KNN classifier [19];

iv) Decision tree classifier [20];

and the performance comparison between the five classifiers is based on the following three criteria:

i) Confusion matrix of the classification result;

ii) Overall accuracy;

iii) Training time (in seconds).

The confusion matrices of classification results are tabulated in Table I. The overall accuracies and the time consumptions for training of the five classifiers are depicted in Fig. 2 and Fig. 3, respectively.

B. *Banknote Authentication Dataset* [27]

This dataset was extracted from images that were taken from genuine and forged banknote-like specimens. Wavelet Transform tool were used to extract features from images [27]. This dataset contains 1372 samples and each sample has 4 attributes:

i) variance of wavelet transformed image;

ii) skewness of wavelet transformed image;

iii) curtosis of wavelet transformed image;

iv) entropy of image

and 1 label: class (0 and 1). 762 data samples are in class 0 and 610 samples are in class 1. Since the structure of this dataset is relatively simple, we use the first 20% of the data samples of each class (152 samples in class 0 and 122 samples in class 1) as the training set and use the rest of the dataset as the validation set. Similar as the first numerical example, our approach will be compared with the four approaches [17]–[20] used in section IV.A. The confusion matrices of the results

TABLE I. Confusion Matrices

Methods	Actual	Classification	
		0	***1***
ALMMo[a]	***0***	82.07% 238 samples	17.93% 52 samples
	1	15.49% 22 samples	84.51% 120 samples
SVM	***0***	85.17% 247 samples	14.83% 43 samples
	1	47.89% 68 samples	52.11% 74 samples
Naïve Bayes	***0***	90.34% 262 samples	9.66% 28 samples
	1	88.73% 126 samples	11.27% 16 samples
KNN	***0***	82.07% 238 samples	17.93% 52 samples
	1	26.06% 37 samples	73.94% 105 samples
Decision Tree	***0***	71.03% 206 samples	28.97% 84 samples
	1	35.21% 50 samples	64.79% 92 samples

[a] The proposed 0-order ALMMo classifier

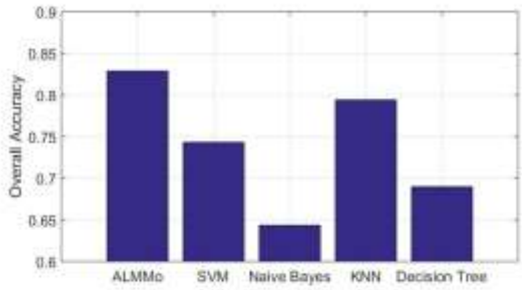

Fig.2. Overall accuracy

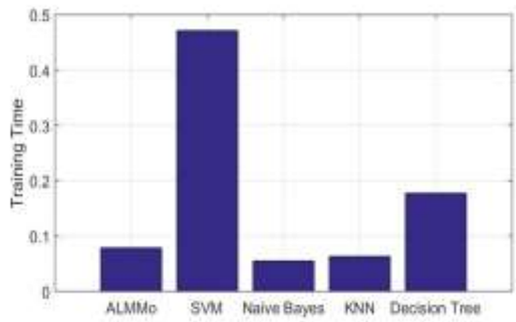

Fig.3. Time consumption

obtained by the five classifiers are tabulated in Table II. The overall accuracies of the five classifiers are:

i) ALMMo-0 Classifier: 0.9918;

ii) SVM classifier: 0.9672;

iii) Naïve Bayes classifier: 0.8370;

iv) KNN classifier: 0.9909;

*v)*Decision tree classifier: 0.9508.

For a deeper comparison, we used the 10-fold cross-validation method by training the classifiers with the randomly selected 20% of the data samples of each class and using the rest for validation 10 times. The average overall accuracies of the classification results obtained by the five classifiers are depicted in Fig. 4 and the corresponding average amount of time consumptions for training are presented in Fig. 5.

C. *Tic-Tac-Toe Endgame Dataset* [28]

The Tic-Tac-Toe Endgame dataset encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "winning for x" (i.e., true when "x" has 1 of 8 possible ways to create a "three-in-a-row"). This dataset contains 958 data samples with 9 attributes and 1 class label [28]:

i) top-left-square: {x, o, b}

ii) top-middle-square: {x, o, b}

iii) top-right-square: {x, o, b}

iv) middle-left-square: {x, o, b}

v) middle-middle-square: {x, o, b}

vi) middle-right-square: {x, o, b}

vii) bottom-left-square: {x, o, b}

viii) bottom-middle-square: {x, o, b}

ix) bottom-right-square: {x, o, b}

x) Class: {positive, negative}

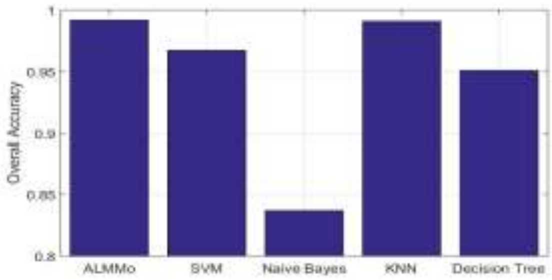

Fig. 4. Overall accuracy

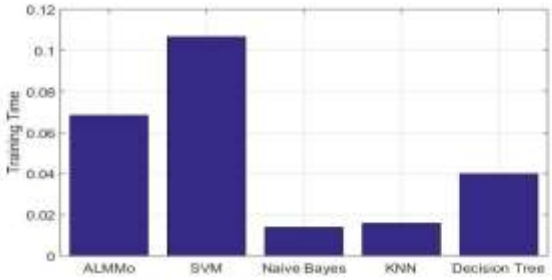

Fig. 5. Time consumption

TABLE II. CONFUSION MATRICES

Methods	**Actual**	**Classification**	
		0	***1***
ALMMo	***0***	100.00% 610 samples	0.00% 0 sample
	1	15.49% 9 samples	84.51% 479 samples
SVM	***0***	99.18% 605 samples	0.82% 5 samples
	1	6.35% 31 samples	93.65% 457 samples
Naïve Bayes	***0***	87.21% 532 samples	12.79% 78 samples
	1	20.70% 101 samples	79.30% 387 samples
KNN	***0***	98.36% 600 samples	1.64% 10 samples
	1	0.00% 0 samples	100.00% 488 samples
Decision Tree	***0***	96.89% 591 samples	3.11% 19 samples
	1	7.17% 35 samples	92.83% 453 samples

TABLE III. CONFUSION MATRICES

Methods	**Actual**	**Classification**	
		Positive	***Negative***
ALMMo	***Positive***	100.00% 125 samples	0.00% 0 sample
	Negative	0.00% 0 sample	100.00% 66 samples
SVM	***Positive***	100.00% 125 samples	0.00% 0 sample
	Negative	100.00% 66 samples	0.00% 0 sample
Naïve Bayes	***Positive***	88.80% 111 samples	11.20% 14 samples
	Negative	65.15% 43 samples	34.85% 23 samples
KNN	***Positive***	100.00% 125 samples	0.00% 0 sample
	Negative	0.00% 0 sample	100.00% 66 samples
Decision Tree	***Positive***	88.00% 110 samples	12.00% 15 samples
	Negative	27.27% 18 samples	72.73% 48 samples

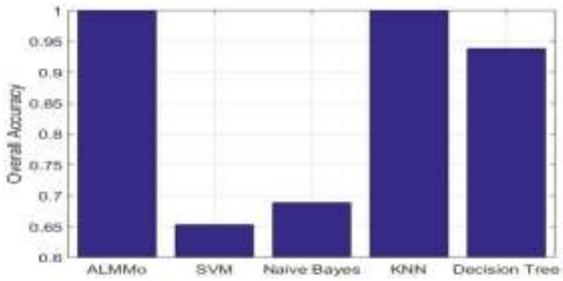

Fig. 6. Overall accuracy

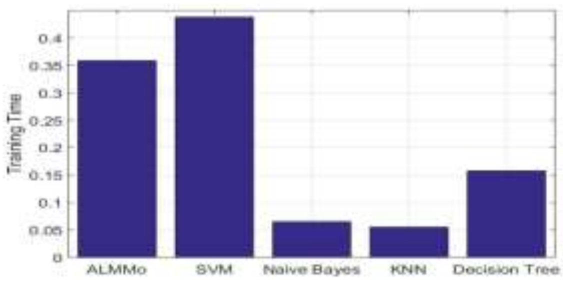

Fig. 7. Time consumption

In this experiment, we further encode “x” as “1”, “o” as “5” and “b” as “3”. This dataset is divided into two parts. The first 80% samples of each class (767 samples in total) are used for training, and the rest of them are used for validation. The confusion matrices of the results obtained by the proposed classifier and the four comparative classifiers are tabulated in Table III.

We also used the 10-fold cross-validation by randomly selecting 80% of the data samples of each class for training the classifiers and using the rest for validating the classifiers. The average overall accuracy and the time consumptions of the training process of the five classifiers are depicted in Fig. 6 and Fig. 7, respectively.

D. *CNAE-9 Dataset* [29]

CNAE-9 dataset is a well-known benchmark NLP dataset. It contains 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of 9 categories catalogued in a table called National Classification of Economic Activities (Classificação Nacional de Atividades Econômicas- CNAE) [29]. Each data sample in this dataset has 856 attributes with word frequency and 1 class label.

In this experiment, the first 80% (865 samples) of the dataset are used for training and the 20% are used as the validation set. The confusion matrix of the classification result using the proposed classifier is visualized in Fig. 8.

The 10-fold cross-validation is conducted by randomly selecting 80% of the data samples of each class for training the classifiers and using the rest for validating the classifiers.

Because of the very high dimensionality, the naïve Bayes classifier [18] failed to generate any result, we compared the proposed multiple-class feedforward ALMMo classifier with the other three approaches [17], [19], [20]. The average classification accuracies and the time consumptions of the training process of the 4 classifiers are presented in Figs. 9 and 10, respectively.

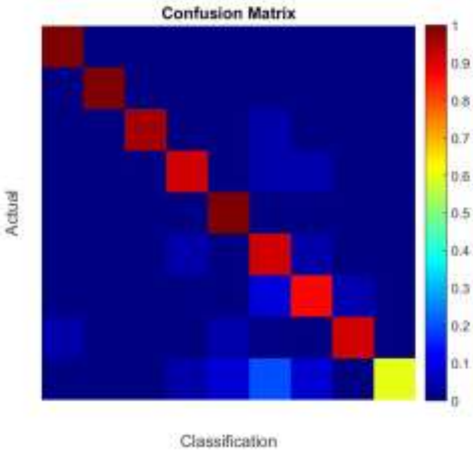

Fig. 8. Confusion matrix

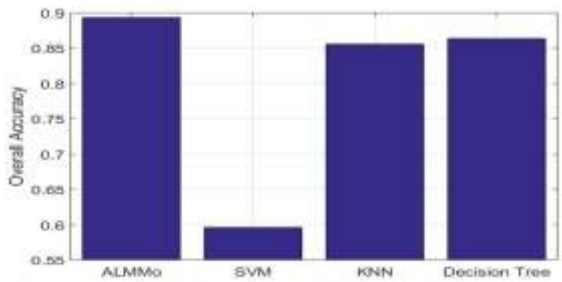

Fig. 9. Overall accuracy

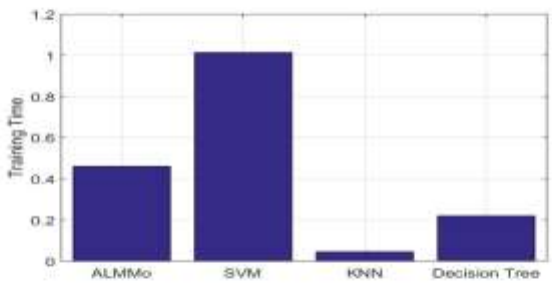

Fig. 10. Time consumption

E. *Discussion*

From the four numerical examples in section IV.A, B, C and D we can see that the SVM classifier with Gaussian kernel [17] requires more time for training and it is less effective in handling high dimensional problems, see Figs. 6 and 9. The naïve Bayes classifier [18] is the fastest one due to its simplicity and its performance is quite stable, though not high. The KNN classifier [19] is also very efficient and its classification accuracies in some problems are comparable to the proposed 0-order ALMMo classifier, but it is not effective in handling high-dimensional datasets with complex structure, see Fig. 9. In addition, its interpretability is not high because it does not reveal an internal structure. The classification accuracy of decision tree classifier [20] is relatively low and it is less efficient in handling lower dimensional problems.

In contrast, the proposed 0-order ALMMo classifier can exhibit excellent performance in all the four real benchmark problems and, at the same time, still keeps its high computational efficiency. It is fully autonomous and offers good interpretability. Moreover, it is evolving in nature.

Therefore, one can conclude that, the proposed classifier is a strong alternative to the existing well-known classifiers.

V. Conclusion

In this paper, we introduced a new type of 0-order FRB classifier called ALMMo-0. This new classifier has the multiple-model architecture and is very efficient in handling complex problems. With the feedforward learning technique based on the EDA estimator, the classifier can identify its structure non-iteratively based on the ensemble properties and mutual distribution of the data without making any *prior* assumptions or training any parameters. Numerical examples based on benchmark datasets show an excellent performance of the proposed classifier and prove it to be a strong alternative to the existing well-known approaches.

As future work, we will focus on modifying the consequent part of the proposed classifier to improve its performance and apply the classifier to more complex problems, i.e. handwriting classification.

REFERENCES

[1] R. Collobert, J. Weston, and L. Bottou, "Natural language processing (almost) from scratch," *J. Mach. …*, vol. 12, pp. 2493–2537, 2011.

[2] L. M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, 2002.

[3] S. B. Park, J. W. Lee, and S. K. Kim, "Content-based image classification using a neural network," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 287–300, 2004.

[4] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and SVM training," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1689–1696.

[5] H. Mohamadi, J. Habibi, M. S. Abadeh, and H. Saadi, "Data mining with a simulated annealing based fuzzy classification system," *Pattern Recognit.*, vol. 41, no. 5, pp. 1841–1850, 2008.

[6] T. Nakashima, G. Schaefer, Y. Yokota, and H. Ishibuchi, "A weighted fuzzy classifier and its application to image processing tasks," *Fuzzy Sets Syst.*, vol. 158, no. 3, pp. 284–294, 2007.

[7] H. Ishibuchi, T. Murata, and M. Gen, "Performance evaluation of fuzzy rule-based classification systems obtained by multi-objective genetic algorithms," *Comput. Ind. Eng.*, vol. 35, no. 3–4, pp. 575–578, 1998.

[8] 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.

[9] P. Angelov, E. Lughofer, and X. Zhou, "Evolving fuzzy classifiers using different model architectures," *Fuzzy Sets Syst.*, vol. 159, no. 23, pp. 3160–3182, 2008.

[10] 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, 2011.

[11] P. Angelov, *Autonomous Learning Systems: From Data Streams to Knowledge in Real Time*. John Willey, 2012.

[12] P. P. Angelov, X. Gu, and J. C. Principe, "Autonomous Learning Multi-model Systems from Data Streams," *IEEE Trans. Fuzzy Syst.*

[13] P. Angelov, "Outside the box: an alternative data analytics framework," *J. Autom. Mob. Robot. Intell. Syst.*, vol. 8, no. 2, pp. 53–59, 2014.

[14] P. P. Angelov, X. Gu, J. Principe, and D. Kangin, "Empirical data analysis - a new tool for data analytics," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 53–59.

[15] P. Angelov, X. Gu, and D. Kangin, "Empirical data analytics," *Int. J. Intell. Syst.*, 2016, to appear.

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

[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines : and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press, 2000.

[18] C. M. Bishop, *Pattern Recognition*. New York: Springer, 2006.

[19] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, pp. 427–438, 2000.

[20] S. R. Safavian and D. Landgrebe, "A survey of decsion tree clasifier methodology," *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1990.

[21] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst. Man Cybern.*, no. 1, pp. 28–44, 1973.

[22] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man. Mach. Stud.*, vol. 7, no. 1, pp. 1–13, 1975.

[23] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. Syst. Man. Cybern.*, vol. 15, no. 1, pp. 116–132, 1985.

[24] W. H. Ho and J. H. Chou, "Design of optimal controllers for Takagi-Sugeno fuzzy-model-based systems," *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans*, vol. 37, no. 3, pp. 329–339, 2007.

[25] X. Gu, P. P. Angelov, D. Kangin, and J. C. Principe, "A new type of distance metric and its application for NLP problems," *submitted to Pattern Recognit.*.

[26] "MONK's Problems Dataset," *https://archive.ics.uci.edu/ml/datasets/MONK's+Problems*.

[27] "Banknote Authentication Dataset," *https://archive.ics.uci.edu/ml/datasets/banknote+authentication*.

[28] "Tic-Tac-Toe Endgame Dataset," *https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame*.

[29] "CNAE-9 Dataset," *http://archive.ics.uci.edu/ml/datasets/CNAE-9*.