Information Fusion 91 (2023) 412–423

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

A distributed evolutionary fuzzy system-based method for the fusion of descriptive emerging patterns in data streams

Á.M. García-Vico [a,*], C.J. Carmona [b,c], P. González [b], M.J. del Jesus [b]

[a] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, 18071, Granada, Spain
[b] Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Jaén, 23071, Jaén, Spain
[c] Leicester School of Pharmacy, DeMontfort University, Leicester, United Kingdom

ARTICLE INFO

Keywords:
Evolutionary fuzzy systems
Distributed computing
Data stream mining
Fuzzy rule-based systems

ABSTRACT

Nowadays the amount of networks of devices and sensors, such as smart homes or smart cities, is rapidly increasing. Each of these devices generates massive amounts of data on a continuous basis where an interpretable description of its state is interesting for the experts. This knowledge can be extracted by means of emerging pattern mining techniques. In fact, it can be extracted locally on each device and joined together afterwards in order to obtain a global vision of the system without transferring any data. However, the traditional massive data processing frameworks are focused on the extraction of this global model, which produces huge amounts of data transfers throughout the network.

This paper proposes a distributed method based on evolutionary fuzzy systems for both the extraction and subsequent fusion of descriptive emerging patterns in data streams from different sources of the same kind. First, an evolutionary algorithm following an informed approach for efficient data processing is presented for the extraction of emerging patterns on the data stream generated by each device, in order to obtain a local model for each stream. Then, several fusion methods are proposed for the aggregation of these patterns in order to extract the global model. A wide experimental study has been carried out to analyse the suitability of the evolutionary algorithm for the extraction of high-quality emerging patterns and its capacity to deal with concept drift. Finally, the quality of the proposed fusion methods is also analysed.

1. Introduction

The demand for data science-related processes is currently growing exponentially. This is mainly produced by the huge increase in data availability due to, among other reasons, the advent of Internet of Things (IoT) devices [1,2]. These data, which continuously arrive in the system over time at an undetermined speed, are known in the literature as data streams [3]. Furthermore, these data often have a very short lifespan, requiring an immediate response. In this scenario, it is interesting that learning models are updated and adapted as the data arrive in the system to avoid learning from obsolete information.

In recent years there has been an important development of static, massive data analytics [4–6] and data stream analysis techniques [7–10]. This is evidenced, for example, in the amount and popularity of the massive data processing frameworks that have been developed so far such as Apache Spark, Apache Flink or Apache Storm [11–13]. This software, which represents the current dominant paradigm for massive data processing, is mainly based on a centralised model in large data centres. This centralised paradigm has a fundamental problem in areas such as IoT: the latency produced by this data delivery prevents an acceptable response time [14]. This problem has increased the demand for Artificial Intelligence (AI) services at the edge of the network, which aims to reduce this latency, and other related problems, by computing AI models on the IoT device itself [15]. Nevertheless, these AI models can only learn using the local data generated in that device, providing a local vision of the system. Therefore, it is interesting to fusion several local models at different levels of granularity to obtain a model that contains a global vision of the system. In this way, the knowledge in this global model can also be leveraged by the local models by giving them useful information about the environment. This stratified extraction of knowledge in several levels is especially useful when applied to organisations, as they also present a hierarchy. For example, in healthcare environments [16–19], or in smart industry [20,21]. The knowledge can be extracted from the lowest level of the system and then fused to provide higher-level knowledge to different organisational areas. For

* Corresponding author.
E-mail addresses: agvico@decsai.ugr.es (Á.M. García-Vico), ccarmona@ujaen.es (C.J. Carmona), pglez@ujaen.es (P. González), mjjesus@ujaen.es (M.J. del Jesus).

https://doi.org/10.1016/j.inffus.2022.10.028
Received 25 July 2022; Received in revised form 7 October 2022; Accepted 27 October 2022
Available online 3 November 2022
1566-2535/

example, from data extracted from the sensors of each patient in a hospital, it is possible to extract knowledge about the behaviour of each patient, whereas the fusion of these patients allows us to extract knowledge about the state of each room, each floor or the whole hospital. Also, geographical applications such as smart homes [22], smart cities [23], smart agriculture [24], among others can benefit from this approach as knowledge can be extracted from each local area, such as a home, and the fused to extract the general behaviour of a wider area, such as a street, neighbourhood or a city.

An interesting kind of knowledge can be extracted through Emerging Pattern Mining (EPM) [25,26] in this kind of problem. The objective of EPM is the description of discriminative features in data with respect to a variable of interest, or the description of emergent behaviours in data. EPM has been successfully applied in different fields such as disease management [27,28], renewable energies [29] and social networks [30], among others. For data stream mining, several algorithms where presented for EPM following a window-based strategy were proposed [31–36]. However, they are computationally expensive, especially when the number of features increases. Metaheuristic algorithms for EPM such as Evolutionary Fuzzy Systems (EFSs) were proposed to improve the efficiency on large datasets and to improve the descriptive capacities with respect to traditional models [37]. According to these results, big data [38,39] and data stream mining EFS-based approaches were developed with interesting results [40,41]. However, all these models are not able to extract knowledge for each data stream individually and they are based on a centralised paradigm in which knowledge retrieval implies unnecessary information transfer. Therefore, the work presented in this paper is, to our knowledge, the first EFS-based system that allows the extraction and fusion of different local pattern models minimising the information transfer for the EPM task.

The main contributions of this work are:

1. An EFS-based learning method for the extraction of emerging patterns models in the data stream generated by each device that constitutes the system. These are the local models that compose the system. For improving the performance in the extraction of knowledge, an informed approach is employed. This only executes the evolutionary process when the quality of the current model decays significantly. Therefore, the extracted patterns are updated and adapted only when necessary. In this way, there is a good balance between resource consumption and the quality of the extracted solutions.
2. A restart system in the evolutionary process inspired by boosting methods to maximise the description of data stream instances.
3. Several fusion methods of the local models previously extracted in order to obtaining a global, high-level knowledge.
4. A scalable and robust approach thanks to an asynchronous and bidirectional learning approach, where global knowledge can be further used by the local models and vice versa to improve the quality of the extracted knowledge.
5. A comprehensive experimental study of the proposed method.

This paper is organised as follows: firstly, the main concepts presented in this work are shown in Section 2. Next, the main components of the proposed system and its working scheme are shown in Section 4. After that, the experimental study, the results extracted and its discussion are depicted in Section 5. Finally, the conclusions of this work are presented in Section 6.

2. Preliminaries

The proposed system is composed of two elements: an EFS which extracts a pattern model from the raw data streams generated on each device and several fusion methods. Hence, several computational methodologies are employed in this work. In this section, the main concepts related to this proposal are presented below: data stream mining and big data (Section 2.1) and EPM (Section 2.2).

2.1. Data stream mining and big data analysis

A data stream $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots\}$ is defined as an ordered and potentially infinite sequence of instances $\mathbf{x}_t$ arriving at the system over time at a variable rate [3]. Due to its nature, data stream mining does not fit well with classical machine learning problems and algorithms for the following reasons:

1. All the instances in the stream cannot be stored in memory since $|S| = \infty$. Therefore, only a subset $B \subset S$ can be stored for a short period of time t_{lim} [42].
2. The learning model must be periodically adjusted as the instances arrive into the system.
3. The response time of the algorithm should always be less than t_{lim}.
4. Learning models have to deal with concept drift due to its continuous and possibly non-stationary nature. Concept drift is defined as an alteration of the underlying distribution of the data over time [9]. Therefore, it causes a significant failure of the trained models over time if it is not correctly handled [42–44].

Considering these features, the development of data stream mining models must determine the following elements:

1. Memory. It is in charge of collecting the data coming from the stream and storing them in the subset B to be later processed and discarded [45].
2. Learning process. It is in charge of extracting knowledge from the data stream, so it must be adapted to be updated incrementally while taking into consideration the effects of concept drift [7,10,46].
3. Concept drift detector. This is an optional element that supports the learning algorithm on the detection of concept drifts [47–49]. It aims to perform the replacement, updating or re-training of the learning algorithm, or any of its components, only when necessary.
4. Evaluation strategy. In data stream mining, classical evaluation strategies cannot be applied as they require the entire dataset. One of the most widely used evaluation strategies in this field is the test-then-train method [50]. This approach is based on evaluating the current model with incoming data. Afterwards, this data is used to update the model if necessary.

Data stream mining can be considered a subset within the concept of big data analysis defined through the so-called 3-V's model: a large volume of data, coming from a variety of sources at a high velocity [51]. In particular, the processing of huge volumes of data requires the employment of specialised paradigms. One of the most popular massive data processing frameworks is MapReduce [52] due to its ease of use, fault tolerance, automation and transparency in the deployment of computer clusters [4,52,53]. Currently, there are several frameworks widely employed for processing big data streams such as Apache Spark Streaming [11], Apache Flink [12], or Apache Storm [13], among others. Despite its merits, one of the main drawbacks of the MapReduce paradigm is that it follows a data-centric approach. In this approach, data travels from the place where it was generated, for example, a wearable device, to a central server. This transfer of information between machines produces a delay between data generation and the possible intelligent response that may be unfeasible in some scenarios. Recently, the concept of edge computing has emerged, whose aim is to compute the AI model and its response on the device itself to minimise these data transfers [15]. However, these AI models are trained only using their local data. Thus, in this paper, we present a system that extracts knowledge in the form of local emerging pattern models from each device. These are then further refined using the application of different fusion methods to obtain a global vision of the system.

2.2. Emerging pattern mining

EPM [25,26] is a task that belongs to the Supervised Descriptive Rule Discovery (SDRD) framework [54] whose objective is the extraction of patterns that possess a significant change of support from one dataset $\mathcal{D}_1$ to another $\mathcal{D}_2$. In particular, the set of emerging patterns is defined as $EP = \{P|GR(P) > \rho\}$, where P is a pattern, GR is the growth rate metric, described in Eq. (1) and $\rho > 1$ is a threshold value.

Through the specialised literature [26,54,55], a wide set of quality measures have been defined to determine the quality of the patterns extracted. All these metrics can be defined as a function $g : \mathbb{N}^4 \rightarrow \mathbb{R}$, which takes as input the values of a two-class contingency table where the number of instances correctly/incorrectly covered/not covered by a pattern is represented as true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). A function $h : \mathcal{P} \times \mathcal{D} \rightarrow \mathbb{N}^4$ is also defined to calculate this contingency table, which takes as input a pattern $P \in \mathcal{P}$ and a dataset $D \in \mathcal{D}$.

In general, for all tasks within SDRD, the descriptive capacity of the extracted patterns is key to obtain a model that is easy to understand, as it facilitates the analysis by the expert. This descriptive capacity is characterised by the search for a balance between three fundamental aspects:

1. The reliability of the patterns, i.e., how accurate are the descriptions made. It can be determined by the following quality measures:
   - Growth Rate (GR) [25]. It determines the support change from one class (or dataset) to another.
   $$GR(P) = \frac{tp(fp+tn)}{fp(tp+fn)} \quad (1)$$
   - Confidence (Conf) [56]. It computes the accuracy of the pattern with respect to its covered instances.
   $$Conf(P) = \frac{tp}{tp+fp} \quad (2)$$
   - False Positive Rate (FPR) [57]. It calculates the percentage of incorrectly covered instances by the pattern with respect to the total number of negative instances. This measure must be minimised.
   $$FPR(P) = \frac{fp}{fp+tn} \quad (3)$$
2. Generality, in the sense of the number of instances affected by the descriptions made by the set of patterns. It is mainly determined by the True Positive Rate (TPR) measure [58] as it computes the percentage of correctly covered instances by a pattern with respect to the total number of positive instances.
$$TPR(P) = \frac{tp}{tp+fn} \quad (4)$$
3. The novelty or interest of the extracted knowledge to the expert. It is measured by the Unusualness, or Weighted Relative Accuracy (WRAcc) [59], which calculates the accuracy of the pattern with respect to the percentage of positive instances.

$$WRAcc(P) = \frac{tp+fp}{tp+fp+tn+fn}\left(\frac{tp}{tp+fp} - \frac{tp+fn}{tp+fp+tn+fn}\right) \quad (5)$$

3. Related works

EPM is closely related to frequent pattern mining and the respective algorithms share many similarities. One of the key differences is that the search space in EPM is not convex [60]. This means that super-patterns of a given pattern can present a higher GR than itself. Therefore, the successful association rule mining algorithms based on the downward closure principle [61] cannot be applied. Dong and Li [25] introduced the first algorithm for EPM where a border-based representation is employed to extract all the EPs with a GR higher than a given threshold using a compact representation. However, this approach is computationally expensive, especially when the number of attributes increases. After that, the approaches presented by the authors are based on extracting subsets of EPs with interesting properties. In [62] an incremental border-based method is proposed for the extraction of jumping EPs (jEPs), i.e., those whose GR = ∞. Despite its incremental learning approach, which is interesting from the data stream mining perspective, it is still computationally expensive as it extracts all the jEPs.

Later on, inspired by FP-Growth [63], several tree-based approaches were proposed for EPM [64–68]. Broadly speaking, these methods employ a compact representation of the dataset in memory using a tree structure, usually, a CP-Tree [66]. Then, they mine EPs of interest by applying several pruning strategies. In general, these approaches are faster than their predecessors but they require recursive reinsertion of the nodes in the tree (this process is called merge) to correctly extract the EPs when mining. This is a large bottleneck in processing time and memory, which is a great limitation when processing large datasets. Other methods leverage current decision-tree methods such as C4.5, Random Forest, among others [69–73] to extract EPs from them. This approach avoids the previous discretisation phase and extracts EPs with more descriptive capacity than previous approaches through additional comparison operators in the attribute–value pair.

On the other hand, bio-inspired metaheuristics are another approach for solving these kinds of problems. These approaches can be relatively more efficient than previous approaches on massive datasets. This paradigm has been widely applied in tasks similar to EPM such as association rule mining [74–76], subgroup discovery [77–79] and in EPM as well [37]. In fact, in [26] the authors demonstrated that the descriptive ability of the EPs extracted by evolutionary algorithms is significantly better than that of traditional methods. These algorithms do not guarantee the extraction of the optimal set of EPs that satisfy the imposed constraints. They also present scalability issues due to the computation of the objective function.

In an attempt to solve these scalability issues several distributed computing approaches, mainly developed under the MapReduce paradigm, have been developed for EPM and similar tasks. For example, several implementations of the Apriori method [80–82], as well as for FP-Growth [83,84] have been proposed for association rule mining. For EPM, in [85] a MapReduce version of the CP-Tree is introduced where local CP-Trees are computed on each mapper and the reduce phase merges them accordingly. Although this proposal allows the application of this tree structure to big data, it still requires the computationally expensive merge procedure which limits its application. Metaheuristic algorithms were also applied in these tasks for big data. In [86] an evolutionary algorithm following a local approach is proposed for subgroup discovery where an NSGA-II algorithm [87] runs in parallel and the extracted rules on each mapper are combined on the reduce phase employing a token competition procedure [88]. In [89] a two-steps genetic programming-based method is presented for associative classification rules. The first step extracts the rules and the next step generates the classifier. Here only the evaluation procedure of each rule is parallelised using MapReduce. Following a similar approach, in [90] an evolutionary fuzzy system based on NSGA-II is shown where only the evaluation process is parallelised. In [39] another evolutionary fuzzy system is proposed following an adaptive NSGA-II approach and an improved evaluation procedure based on bitwise computations that significantly improves the computation time as well as the scalability with respect to the number of attributes.

All the approaches previously presented were mainly designed for batch processing and they struggle when they are applied within data stream environments. Many of the methods focused on data stream mining on related tasks such as association rule mining have been oriented towards the extraction of frequent items, without the subsequent extraction of patterns [91–93]. Several approaches for EPM have been presented throughout the literature for handling this kind of data. In [34,94] incremental border-based algorithms were proposed for EPM but they are very inefficient. In [31] a block-based, border-based algorithm is shown where items are sorted by the strength measure and EPs with the highest average strength are extracted. Tree-based algorithms were also applied for EPM in data streams. For example, in [32] a version of the CP-Tree for data stream mining is presented where a new merge procedure is applied to improve efficiency but it is not efficient enough for high-speed data streams. In [33] a modified version of the P-Tree called IEP-TFP is shown for the extraction of EPs on ECG data. This approach maintains two trees and the EP extraction is made by merging both trees and pruning accordingly. In [95] a method is proposed for streaming features where an online feature selection algorithm keeps the best feature while the EP extraction process is carried out in an offline fashion. Also, several tree-based algorithms following a window-based approach were presented to tackle this problem recently [35,36,96]. Finally, metaheuristic algorithms were also applied in different data stream mining tasks. In [97] a streaming clustering algorithm is proposed where the FEAC evolutionary algorithm [98] extracts new clusters only when a concept drift alarm signal is triggered. In [99] a steady-state evolutionary algorithm is shown for online learning of association rules. For EPM, in [40] an evolutionary process is presented where the algorithm runs in an offline phase for all arriving batches of data. In this method, previous knowledge is employed for both guiding and updating the model towards good results on the current data. In [41] an evolutionary algorithm is proposed for the processing of massive data streams arriving from different sources employing Apache Kafka for data ingestion and Apache Spark for distributed processing of data.

The majority of works presented in this section are focused on extracting knowledge from batch datasets, massive or not, or in single data streams with several techniques and adaptations to handle this kind of data. In data stream scenarios, the efficiency of these methods is limited especially when the number of features increases, so distributed approaches are mandatory when the amount of arriving instances exceeds the computing capabilities of the system. Only the approach presented in [41] shows an attempt to handle at the same time massive data streams from multiple sources. This approach presents several disadvantages: 1) it does not extract local knowledge for each data stream source individually, it only extracts the global one, and 2) it is constantly transferring data from the source to the processing data centre. Therefore, the model proposed in this paper is, to the best of our knowledge, the first EPM-specific work to extract both local and global information from multiple data streams in a distributed environment.

4. Description of the proposed system

The algorithm proposed in this work is a distributed EFS-based method for the fusion of descriptive emerging patterns models in data streams. This method is capable of explaining or monitoring the behaviour of dynamical systems at different levels of granularity. This is possible because initially, the proposed EFS-based method extracts local models of emerging patterns in each device. These provide a local view of each element of the system. Notice that this computation is performed on the device itself, so there is no massive transfer of data. Finally, these local models are fused through different methods proposed in this work to obtain a global view of the system. It is also important to highlight that these global insights can be subsequently exploited by the different devices to guide them towards a higher quality local model.

The system is described in the following subsections. The proposed architecture and its main components are described in Section 4.1. The main features of the proposed EFS are presented in Section 4.2. The proposed fusion methods are defined in Section 4.3. Finally, Section 4.4 shows the operational schema where all the components presented above are unified.

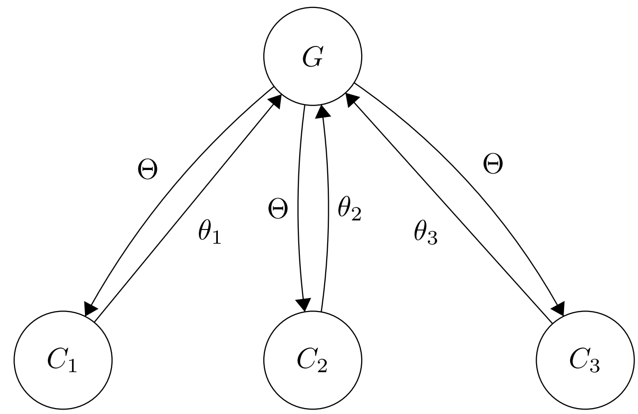

Fig. 1. General scheme of the proposed system. Each C_i represents a device and G represents the fusion node. Each arrow represents an information interchange.

4.1. System architecture and components

The proposed system is made up of a number of components which are briefly defined below and which will be detailed in the following subsections:

- A set of C_i devices that acquire data through a data stream for each one, called S_j. It is important to highlight that in this work any kind of device that can collect data, contains some computing capability, and has some network connectivity can be suitable for use. This ranges from high-performance computers to embedded IoT devices. For communication efficiency, this data will be kept available only in the device itself. Therefore data is not sent over the network. This is key in IoT applications as the connectivity of these devices generally relies on connections with limited bandwidth such as 2G or 3G. These devices extract a local model $\theta_i = \{p_1, p_2, \dots, p_n\}$ which contain a set of emerging patterns $p_i \in \mathcal{P}$ by means of the proposed EFS. The fuzzy patterns extracted describe the local behaviour of the data stream.
- The highest node of the hierarchy, called fusion node G, will store a model Θ with more refined knowledge obtained from the fusion of the local models. Specifically, it will be in charge of receiving a series of local models θ_i and join them through a fusion method Δ, so that $\Theta = \Delta(\theta_1, \dots, \theta_n)$.

A schematic of the proposed architecture is presented in Fig. 1. In this work, a hierarchical client–server architecture is proposed where the elements communicate asynchronously with each other to avoid deadlocks. Due to the changing nature of each S_j, it is of interest that the system can take advantage at the lower levels of the information obtained in the upper layers. In this way, each C_i should acquire knowledge from the surrounding context to improve its learning. To achieve this, the proposed system performs these steps:

1. C_i, following a test-then-train scheme, evaluates its model with respect to the block of data it receives from its data stream S_j. This is done by computing $Q = g\left(h\left(\theta_i, B\right)\right)$, where Q is a value that determines the quality of the local model θ_i with respect to the data block B.
2. If Q is below a threshold Q_{thr}, then the local model θ_i is updated. This updating process is carried out by a single-objective evolutionary algorithm in order to find patterns that provide a better description of the data than the current ones.
3. After the execution of the proposed evolutionary algorithm, the new model is sent asynchronously to G in order to be fused with the global model. C_i continues to process the incoming data blocks from point 1.

4. Once G receives the local model from C_i it merges it with the existing global model Θ stored at G. Namely, G performs $\Theta = \Delta\left(\Theta, \theta_i\right)$, where Δ is a fusion method. Finally, G sends Θ to C_i for further use. Specifically, C_i will keep those patterns of the global model that can be useful for it, e.g., those that provide a good description of its data stream. In this work, the patterns with a confidence value (Eq. (2)) higher than a given threshold are kept.

Finally, it is important to remark that a new device can join the system at any time during the previous steps. In this case, the fusion node G just sends Θ to it in order to provide context, so it does not operate blindly.

4.2. EFS for the extraction of patterns in data streams

The core of this proposal is a learning algorithm for the extraction of descriptive patterns in data streams. Specifically, it is an EFS based on a single-objective genetic algorithm with an archive that will store the best solutions found so far together with a restart process to avoid the stagnation that promotes fast convergence while exploring new areas of the search space. It employs a "chromosome = pattern" approach where an individual in the population represents a potential pattern to be extracted [100]. These patterns are encoded following a disjunctive normal form that groups items belonging to the same variable through disjunctions, while the different variables are connected through conjunctions [101]. In addition, numerical variables are represented by means of triangular fuzzy linguistic labels (LLs) [102]. The genetic operators of this method are the classical binary tournament selection [103], the two-point crossover [104], and an SDRD-specific biased mutation operator [37]. After this process, the offspring population replace the current population following an elitist scheme of size 2, where the two best parents are kept, while the two worst offspring are discarded. Using this process, high selective pressure is performed, so the method will converge faster. Due to the fast convergence, a restart operator is necessary. This is in charge of filling the archive of patterns and then moving to other areas of the search space that contains examples that have not been correctly described by the patterns currently found in the archive. To achieve this, the best individual in the population is firstly included in the archive. After that, the instances correctly covered by this individual are removed. Finally, a new random population is created. The evolutionary cycle will run until a maximum number of iterations are reached or all the instances of the block B are correctly covered by at least one pattern of the archive. Finally, it is important to remark that the initial population includes the current pattern model θ_i so that it contains a memory of the state of the stream [105].

In SDRD it is required to optimise different objectives. In this method, a single-objective scheme is proposed in order to speed-up the evolutionary process. One of the most common approaches to include different objectives is by means of a weighted sum of them as presented in Eq. (6).

$$Obj\left(\mathcal{P}, B\right) = \sum_{i=1}^{m} w_i \cdot g_i\left(h\left(\mathcal{P}, B\right)\right) \tag{6}$$

where m is the number of objectives considered and w_i is the weight given to objective i, subject to $\sum_{i=1}^{m} w_i = 1$. As this evaluation is intensive, the Bit-LUT [39] technique is employed in order to speed up this process as much as possible.

4.3. Fusion methods

The second fundamental component of the proposed architecture is the fusion node, G, which is responsible for filtering and merging the local models received from the different devices of the network to obtain global insights about the state of the system. For this purpose, a fusion method Δ is used so that the model obtained is $\Theta = \Delta(\theta_1, \dots, \theta_n)$.

In many cases, a fusion method requires the use of additional information about the different local models to be merged correctly. For example, it is required to determine some quality measures to perform the fusion. However, the computation of these measures on each device could bias the results. To avoid this bias, the use of a reduced set of validation data D_v associated with G is proposed. This set can be built, for example, by means of sampling a subset of data from the different devices. Then the local models θ_i can be previously evaluated with this validation set D_v, so the fusion method can discard unwanted knowledge. The advantage of using this validation set is that it can be easily created by the experts according to their needs. Thus, the Θ model can be redefined as presented in Eq. (7), in which the Δ operator is conditioned on the validation data D_v.

$$\Theta = \Delta(\theta_1, \dots, \theta_n; D_v) \tag{7}$$

This paper proposes several fusion methods that can be used by experts according to their needs. Specifically, the proposed methods are:

1. Pareto front extraction. Given a set of quality measures $\{g_1, g_2, \dots, g_m\}$ of interest to the expert, calculated on the validation data D_v, this fusion method consists of extracting the patterns that are non-dominated. That is those patterns where at least one of the target metrics is better than the rest of the patterns in the set.

$$\Theta = \{p_i \in \bigcup_{i=0}^{n} \theta_i | \nexists\ p_j \in \bigcup_{i=0}^{n} \theta_i \ : \ g_k(p_j) > g_k(p_i) \forall k \in \{1, \dots, m\}\} \tag{8}$$

where n is the number of local models received and m is the set of objectives set by the expert. Using this procedure, the expert can obtain a set of patterns that describe the dataset based on the given objectives. To perform this method efficiently, the fast sorting method of the NSGA-II [87] algorithm has been used. This fusion model is interesting as it can extract a set of patterns that cover the whole spectrum of the objective space. This can be useful, for example, when the current global model Θ becomes very specialised on one objective and it gets stagnated, In this way, the diversity within the objectives' search space is improved.

2. Confidence filter. This consists of keeping only those patterns whose value in the confidence measure (Eq. (2)), calculated on the validation data, exceeds an established threshold value α. In this way, the expert extracts only those patterns that are of quality with respect to the validation data. This filter is defined as presented in Eq. (9).

$$\Theta = \{p \in \bigcup_{i=0}^{n} \theta_i | Conf(p, D_v) > \alpha\} \tag{9}$$

This fusion model is interesting when it is required a global model which precisely describes the validation data. These patterns can also be useful to local models θ_i as this knowledge can be employed to better explore an area of the search space. Nevertheless, it is highlighted that possibly the number of patterns extracted can be high, which decreases its interpretability.

3. Coverage Ratio. This process consists of the elimination of redundant patterns with respect to the validation dataset. To do so, the algorithm performs two main operations: searching for highly overlapping patterns and eliminating those ones that are redundant. To determine the highly redundant patterns, the set O is calculated as presented in Eq. (10) [106].

$$O = \{(p_i, p_j) \in \bigcup_{i=0}^{n} \theta_i | \frac{TPs(p_i) \cap TPs(p_j)}{TPs(p_i) \cup TPs(p_j)} > \alpha\} \tag{10}$$

where $\alpha \in [0,1]$ is the minimum overlap percentage and $TPs(p)$ is the set of instances correctly covered by pattern p. Then, for each pair of patterns of O, the odds ratio of each pattern is computed. This is calculated as presented in Eq. (11):

$$OddsRatio(p) = \left[\frac{tp \cdot tn}{fp \cdot fp} exp(-\omega), \frac{tp \cdot tn}{fp \cdot fp} exp(\omega) \right] \tag{11}$$

where $\omega = Z_{\alpha/2}\sqrt{\frac{1}{tp} + \frac{1}{fp} + \frac{1}{tn} + \frac{1}{fn}}$ y $Z_{\alpha/2} = 1.96$ for a 95% confidence interval. If the intervals p_i and p_j overlap, then the pattern containing the largest number of variables is eliminated. Otherwise, the one with the lowest maximum OddsRatio value is eliminated. This fusion model allows the extraction of a reduced set of patterns which is spread throughout the whole validation data in order to cover all possible instances. It is interesting for reducing the redundancy and complexity of the model.

4. Token Competition [107]. This process yields a set of patterns whose goal is to describe as many instances as possible with as few patterns as possible. For this purpose, a niche strategy is employed in which the best-adapted patterns obtain large parts of the instance space that cannot be invaded by other less strong patterns. For this, we first define a set $T = \{t_1, t_2, \dots, t_{|D_v|}\}$ where each $t_i \in \{0,1\}$ represents whether instance i of the validation set has been captured. An order in the patterns is also defined from a quality measure Q. Initially, token competition sorts the patterns by the Q metric from best to worst. After this, the token capture process begins, where each pattern, in order, will capture those instances that have not been previously captured. This is determined according to Eq. (12). Next, T is updated according to Eq. (13).

$$c(p) = (T \oplus TPs(p)) \wedge \neg T \tag{12}$$

$$T \leftarrow T \vee TPs(p) \tag{13}$$

where $\oplus$ and $\neg$ are the binary XOR operations and the complement, respectively. The final result contains only those patterns whose $c(p)$ value is greater than zero.

$$\Theta = \{p_i \in \bigcup_{i=0}^{n} \theta_i | c(p_i) > 0\} \tag{14}$$

Similar to the previous approach, this method is interesting for obtaining a reduced and dispersed set of patterns across the space of validation instances. However, the reduction of redundant patterns is guided by those patterns that are best adapted with respect to the objectives. This produces a reduced set of patterns oriented towards the interests of the expert, which can be useful in many scenarios. However, the selection of the quality measure Q significantly influences the result, and its election is not straightforward.

5. Composition of Token Competition and Confidence. This method aims to obtain a model that contains a set of high precision patterns that maximises the coverage of the instance space with a minimum quantity of patterns. For this purpose it is firstly applied the confidence filter (Eq. (9)), after that the token competition is applied (Eqs. (12)–(14)). This process has been widely applied in several related works with good results [37–39].

$$\Theta = TC\left(Conf\left(\bigcup_{i=0}^{n} \theta_i\right)\right) \tag{15}$$

where $TC()$ is the token competition process and $Conf()$ is the confidence filter. The combination of both techniques is interesting when it is required a trade-off between precise description, coverage of validation instances, and complexity of the model extracted.

4.4. Operational scheme

The pseudocode of the proposed system is presented in Algorithm 1 and Algorithm 2 for the description of the devices C_i and the fusion node G, respectively, in which all the elements described above are unified.

Algorithm 1 Operational scheme of the device C_i

Input:
B: A batch of data collected from the stream.
θ_i: Current local model.
Θ: Global model asynchronously received from G. It is empty if its not received.
Q_{thr}: Minimum value for the chosen quality measure to trigger the learning algorithm.
Output:
θ_i: An updated model for the current batch.

1: **if** Θ is not $\emptyset$ **then**
2: $\Theta \leftarrow$ filter(Θ; B) {filter by confidence}
3: $\theta_i \leftarrow \theta_i \cup \Theta$
4: **end if**
5: **for all** $p \in \theta_i$ **do**
6: $Q \leftarrow$ Compute quality measure of θ_i with respect to B
7: **end for**
8: **if** $Q < Q_{thr}$ **then**
9: $\theta_i \leftarrow$ EFS(θ_i) {Launch evolutionary learning process presented at Section 4.2}
10: Send θ_i to the corresponding fusion node G asynchronously.
11: **end if**
12: **return** θ_i

Algorithm 2 Operational scheme of the aggregator G

Input:
θ_i: A local model asynchronously sent from device C_i.
Θ: Current global model in G.
D_v: Validation data in G, if required.
$\Delta, \ldots$: The aggregation function and additional parameters required to run that function.
Output:
Θ: An updated set of patterns.

1: $\Theta \leftarrow \Theta \cup \theta_i$
2: $\Theta \leftarrow \Delta(\Theta, D_v)$ {Section 4.3, one of the proposed fusion methods}
3: Send Θ to device C_i asynchronously.
4: **return** Θ

Regarding the device C_i presented in Algorithm 1, firstly it is checked if the global model Θ has been received from the fusion node. If so, such a model must be included in θ_i. To do so, those patterns in Θ whose confidence with respect to block B is less than a threshold value are removed (line 2), and then the result is added directly to θ_i (line 3). In this way, the C_i device will only use those patterns that are of interest for its local data. After this, the quality measure Q responsible for triggering the learning algorithm is computed (lines 5–7). If the value of Q is below the given threshold value Q_{thr}, then the EFS learning algorithm is executed, which will return a new θ_i model. This new model has immediately been sent to the fusion node asynchronously (lines 8–11).

On the other hand, the operation of the fusion node G is presented in Algorithm 2. Once it has received a local model θ_i (sent in line 10 of Algorithm 1), the whole model is included in Θ (line 1). Subsequently, the chosen fusion method Δ is applied with respect to the validation data (line 2). Finally, the result of this operation is sent back to the device C_i asynchronously.

5. Experimental study

In this section, a comprehensive experimental study is carried out to determine the quality of the proposed algorithm. The experimental framework is shown in Section 5.1. Next, in Section 5.2 the quality of the global knowledge extracted by the fusion node is analysed.

A similar analysis is carried out in Section 5.3 with respect to the local models obtained for each device. In this case, both the capacity to adapt the current global model to the context of the device and its adaptation to concept drift are analysed. In addition, the execution time of the proposed method is determined for each device.

Table 1
Datasets employed in the experimental study.

Dataset	#Instances	#Attributes.	#Classes
Aggrawal	1000 000	9	2
Hyperplane	1000 000	10	2
Mixed	1000 000	4	2
RandomRBF	1000 000	10	2
RandomTree	1000 000	10	2
SEA	1000 000	3	2

Table 2
Parameters of the algorithm in the experimental study.

Parameter	Value
Number of LLs	3
Population size	50
Crossover probability	0.7
Mutation probability	0.05
Max. evaluations	5000
Confidence threshold	0.6
Support threshold	0.1
Objectives	WRAcc, Support Diff and Confidence
Weight for each objective (w_i)	$\frac{1}{3}$

5.1. Experimental Framework

In this experimental study, a setup composed of four devices C_1, C_2, C_3 and C_4 and a fusion node G is proposed. Each device will generate data from a data stream. In all the cases analysed, each C_i will process one million instances, divided into 200 blocks of 5000 instances for each one. In this study, both non-concept drift and concept drift environments are analysed to determine the behaviour of the proposed algorithm in both circumstances. Therefore, it is mandatory to have full control over when, where and the kind of concept drift that is produced to determine the adaptation capabilities of the proposed algorithm. In this study, artificial data streams will be employed as they allow us to have full control over these characteristics, so a better analysis can be carried out. These data have been generated through the MOA software [108] which provides us with a set of well-established artificial data streams widely employed throughout the literature.

Specifically, concept drift will occur, if required, in each device randomly and independently of each other between block 10 and block 50 of the current concept in order to be as realistic as possible. For example, device C_1 will produce an abrupt concept drift at blocks 20, 50, 97, 120 and 165 whereas device C_2 will generate a drift at blocks 44, 83, 130 and 162, etc. Finally, it is important to remark that each C_i employs the same data generator, but uses a different configuration on each one. In this way, data present different distributions on each device. Using this approach an environment where several devices of the same kind, located at different places, can be simulated. For example, the same model of smartband is worn by different people, while performing different activities at different times.

Regarding the fusion node, it will contain a validation dataset D_v of 500000 instances. These instances were sampled using the same MOA generator employed on the devices, but also using different configurations with respect to them in order to generate different data. Following the previous example, in this case, the validation data can be seen as a sample of data taken from other persons wearing the same band. In this study, this environment is simulated on a single computer, where each component is individually executed on an execution thread. All the experiments were carried out on a laptop with an Intel i7-7700HQ at 3.8 GHz and 64GBs of RAM on Linux.

The characteristics of each dataset used are shown in Table 1. The configuration of the parameters used for the experimental study is shown in Table 2.

The source code of the proposed algorithm, together with additional information regarding the reproducibility of this study is available under the AGPLv3 license in our GitHub repository.[1]

5.2. Analysis of the results of the global model

This section presents the results obtained by the global model extracted in the fusion node G. This study aims to determine the average quality of the global model obtained after applying the different fusion methods proposed in Section 4.3 with respect to the validation data D_v. These fusion methods will be compared against the union of all the received local models, i.e., $\Theta = \bigcup_{i=1}^{n} \theta_i$ as it is the most simple fusion method. This method is the baseline for this analysis.

The average results extracted are shown in Table 3. An analysis of the different key aspects of EPM can be extracted from this table and is shown below.

- Complexity of the model. Determined by the number of patterns and the average number of variables they have. In general, the complexity of the extracted pattern model is significantly reduced when applying an operator with respect to the baseline method. In particular, non-dominance filtering offers the greatest reduction in the number of patterns, possibly due to the mono-objective approach employed in the devices, which extracts a set of patterns that are highly dominated by each other, making the Pareto front contain very few patterns. On the other hand, confidence filtering is the one that obtains the lowest average number of variables, at the cost of a significant increase in the number of rules. In any case, there is no clear conclusion on this point, so these results should be considered together with the analysis of the rest of the objectives.
- Interest. Determined by the WRAcc value. In this case, the confidence filter, the token competition operator and the composition of both allow increasing the interest in the patterns obtained with respect to the baseline method. Therefore, these fusion methods allow retaining only those patterns of interest. In particular, the composition of the confidence filter and token competition shows a very relevant trade-off value, as the complexity of the model is significantly reduced with respect to the separate methods while maintaining the value of WRAcc. This synergy occurs because, firstly, confidence filtering retains those patterns that are highly reliable, while token competition allows maximising the coverage of instances while minimising the number of patterns.
- Generality. Determined by TPR. At this point, confidence filtering, token competition and the combination of both still improve on the baseline method. However, in this case, the confidence filter wins. This may be because this filter maintains any pattern with high confidence even if it is less interesting, i.e., it has a confidence value close to the threshold with a higher TPR. However, it is observed that the combination of operators maintains a very positive balance by maintaining a high TPR while significantly reducing the number of patterns, indicating that the application of confidence filtering alone extracts patterns that are highly redundant with each other.
- Reliability. Determined by the CONF, GR and FPR values. This last measure must be minimised. In this case, the combination of operators obtains the best results in FPR and GR, while the confidence filter obviously obtains the best results in CONF. These results follow a similar conclusion to the previous points, where the combination of operators produces a synergy necessary to obtain a reduced, reliable and interesting set of patterns for the expert.

[1] https://github.com/agvico/2022_InfFus

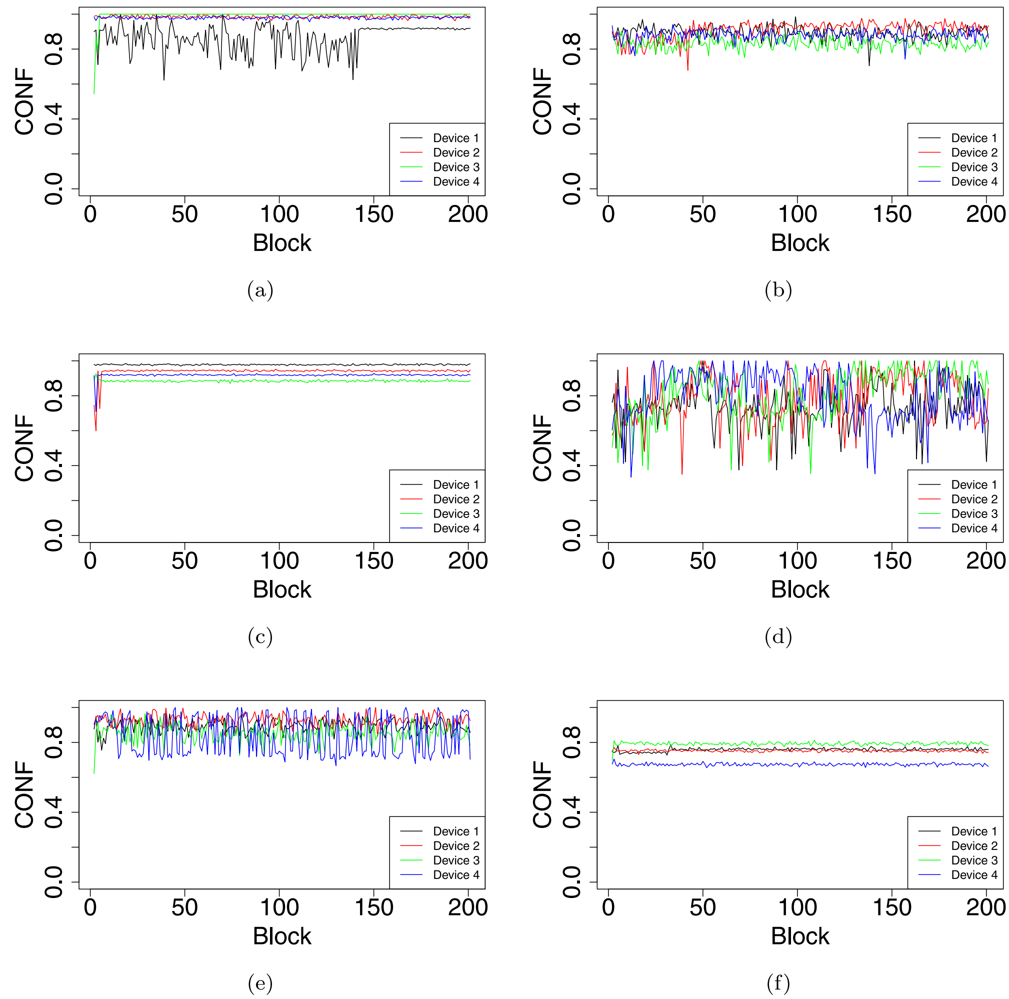

Fig. 2. Average confidence of the local models θ_i for each analysed dataset: (a) Agrawal, (b) Hyperplane, (c) Mixed, (d) RandomRBF, (e) RandomTree, (f) SEA. It is highlighted that these data do not present concept drift.

Table 3
Average results of the different fusion methods analysed.

Aggregation	# Patt.	# Vars.	WRAcc	CONF	GR	TPR	FPR
Union (baseline)	104	5.026	0.515	0.639	0.718	0.109	0.142
Coverage Ratio	62	4.879	0.501	0.614	0.683	0.096	0.187
Confidence	76	**4.703**	**0.565**	**0.866**	0.978	**0.176**	0.046
Pareto Front	**3**	5.400	0.507	0.580	0.668	0.100	0.193
TC	39	4.844	0.528	0.667	0.800	0.142	0.087
TC + Confidence	30	4.804	0.562	0.827	**0.984**	0.168	**0.045**

In EPM it is necessary to find a good balance between the above objectives. Based on the average results obtained, in general, it can be concluded that the operator composed of the confidence filter and token competition obtains the best model with a good balance between quality and interpretability. This is due to the fact that, despite not obtaining the best results in most metrics, it presents quality levels very close to these with significantly fewer patterns. Therefore, experts may be able to obtain relevant and easily analysable global knowledge about the underlying distribution of the different devices that make up the system.

5.3. Analysis of the performance of the local models

This section presents the results obtained by the local models extracted on each device $\theta_1, \ldots, \theta_4$. The aim of this section is twofold. Firstly, it is analysed whether the global model Θ can be useful for improving the adaptation of each local model θ_i to its local data. Finally, the capacity to extract a new local model θ_i when a concept drift is produced by the proposed EFS learning method is analysed. The concept drift can be defined as a change in the posterior probability of the target class [9], which in EPM is determined by the confidence of the pattern. It is important to note that this section only shows the results obtained by the token competition and confidence filter aggregation operator, as it is the one that presents the best balance with respect to the results obtained in the previous analysis.

In Fig. 2, for each analysed dataset, the performance of each local model θ_i is presented as its average confidence for each data block B without applying any concept drift. Overall, it can be observed that the local models have excellent average confidence, approximately around a value of 0.8. This indicates a good adaptation of the global model to each particular device, together with high accuracy in the description provided by the obtained patterns. It is highlighted that in Fig. 2(d) and Fig. 2(e) there is a high variation in quality over time. Nevertheless, it can be observed that, in general, the accuracy in terms of confidence

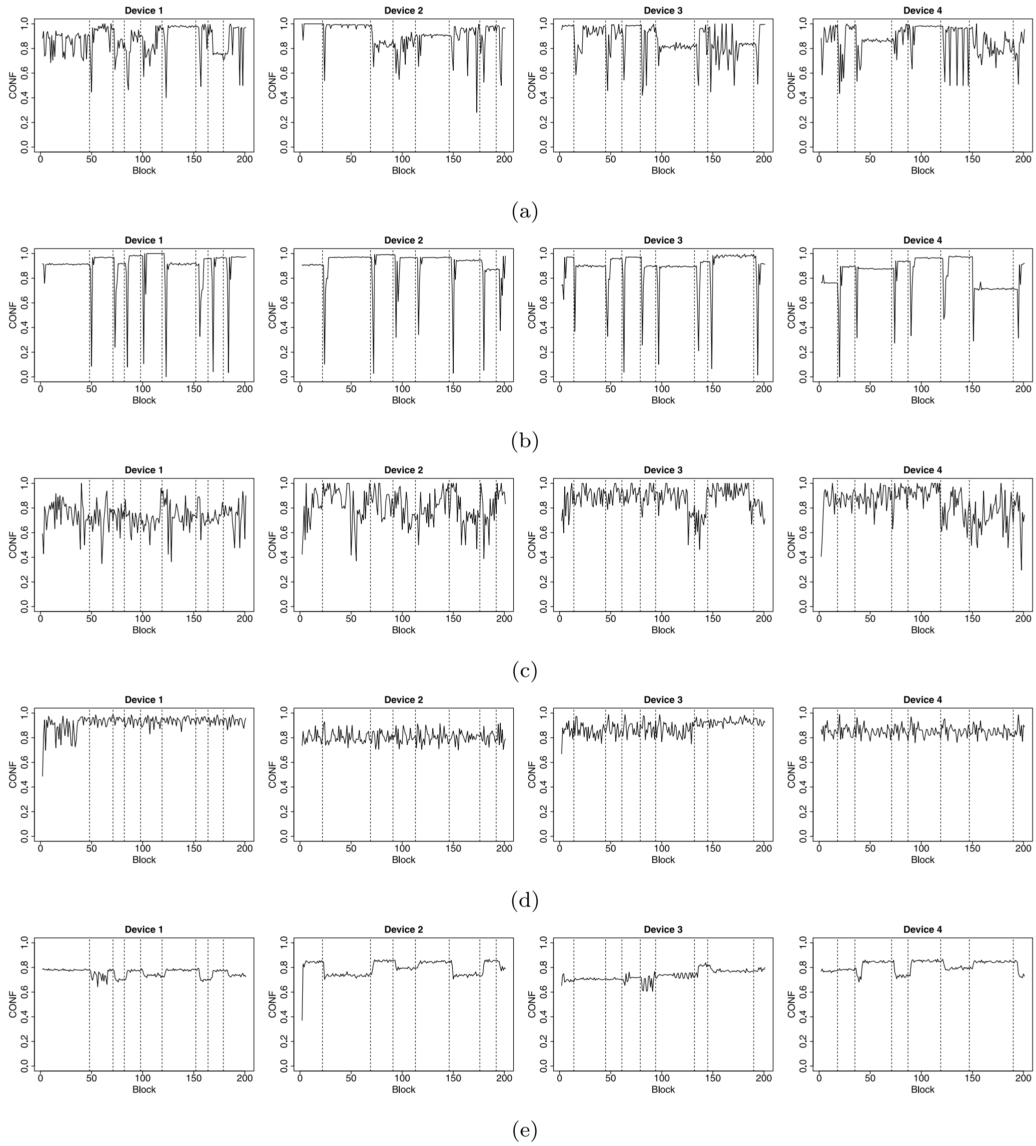

Fig. 3. Average confidence of the local models extracted on each device for each of the analysed datasets: (a) Agrawal, (b) Mixed, (c) RandomRBF, (d) RandomTree, (e) SEA. It is important to remark that these data present concept drift, which is marked as a dashed vertical line on each plot.

is good in both cases. In fact, the usefulness of the global model Θ received by the fusion node is more relevant, as the information coming from other devices contains useful information to improve the local model fitting. Furthermore, it can be observed in Fig. 2(d), that when the quality drops below the minimum confidence threshold, there is an immediate improvement due to the launching of the proposed EFS algorithm to extract new knowledge.

The performance of the algorithm with respect to data sets with concept drift is presented in Fig. 3. This is measured as the average confidence of each local model on each data block. It is important to note that each concept drift is marked with a dashed vertical line on each graph. It can be seen that, after a concept drift, the average confidence of the local model suddenly decays. This is normal behaviour after a concept drift, given its definition. However, the quality of the local model significantly improves on the following data blocks. This is due to the concept drift monitoring system reacting correctly by running the EFS learning algorithm, which extracts a new model adapted to the new circumstances. While keeping the same concept, a very similar behaviour to the one presented in Fig. 2 can be observed.

Finally, Fig. 4 presents the processing time of each data block. In these plots, the horizontal dashed lined marks an execution time deadline of one second which should not be constantly surpassed. It can be observed that the informed strategy proposed in this work improves the average execution time as the processing of data is very fast when no concept drift is detected. The peaks observed are due to the execution of the evolutionary process. However, in the majority of cases, the execution time is far below the given threshold of 1 s of execution time. This means that the proposed evolutionary process can extract good results in a reasonable time thanks to its fast convergence. Therefore, the proposed algorithm is a good alternative for the extraction of robust and quality patterns with a great balance between its descriptive capacity and its ability to adapt to changes.

6. Conclusions

In this paper, a distributed EFS-based method for the extraction and fusion of emerging patterns in data streams coming from different generating sources is presented. On the one hand, an EFS which is in charge of extracting emerging patterns describing the local behaviour of each particular stream is presented. On the other hand, several fusion methods of these local emerging patterns are presented to obtain global knowledge. This way, the local models can obtain a local vision of the system without the transmission of huge volumes of data, whereas the

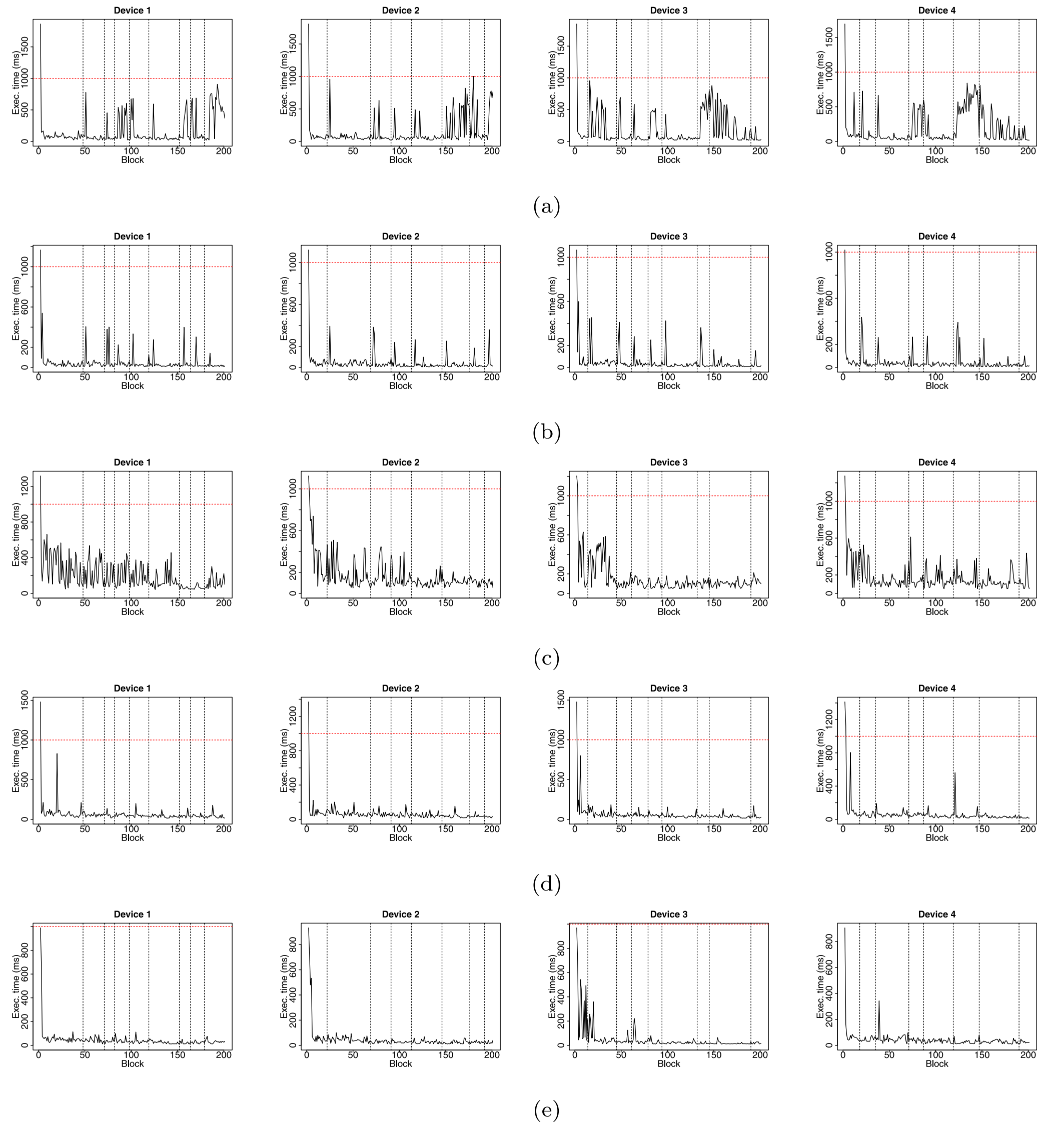

Fig. 4. Average execution time of each data block, on each device, for each dataset analysed: (a) Agrawal, (b) Mixed, (c) RandomRBF, (d) RandomTree, (e) SEA. It is important to remark that these data present concept drift, which is marked as a dashed vertical line on each plot. Also, the dashed, horizontal red line shows the execution time deadline.

global model gives us a whole vision of the system. To the best of our knowledge, the method proposed in this work is the first EPM method that uses this working scheme.

The EFS learning method follows an informed strategy based on a change monitoring system to find a good balance between the quality of the extracted knowledge and the computational efficiency. It is based on a single-objective evolutionary algorithm with a restart system that allows redirecting the search for new patterns to areas of the space not covered by the best patterns extracted so far.

The proposed system has been analysed in an experimental study. Firstly, the quality of the global model obtained by the application of the different fusion methods proposed is analysed. The conclusion is that the fusion method based on the combination of confidence filter and token competition obtains the best balance between global model complexity, interest, reliability and generality.

Finally, the usefulness of the knowledge within the global model to be employed by the local models has been analysed. Also, the capacity of the learning method to extract an updated local model when a concept drift occurred is analysed. It has been demonstrated that the proposed method can effectively adapt to abrupt concept drifts and correctly extract useful information from the global model to improve the description of the data of each local model.

Overall, the proposal is a promising approach for the extraction of emerging patterns in this type of environment. Therefore, this work opens an interesting research line regarding the development of new methodologies for extracting emerging patterns in these local environments, fusion methods and the combination of these techniques to improve the extraction of higher-level knowledge.

CRediT authorship contribution statement

Á.M. García-Vico: Writing – original draft, Conceptualization, Methodology, Software, Investigation. **C.J. Carmona:** Writing – review & editing, Conceptualization, Methodology. **P. González:** Writing – review & editing, Supervision. **M.J. del Jesus:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the Regional Government of Andalusia, program "Personal Investigador Doctor", reference DOC_00235 and by the Spanish Ministry of Economy and Competitiveness under the project PID2019-107793GB-I00.

References

[1] O.B. Sezer, E. Dogdu, A.M. Ozbayoglu, Context-aware computing, learning, and big data in internet of things: A survey, IEEE Internet Things J. 5 (1) (2017) 1–27.
[2] J.H. Nord, A. Koohang, J. Paliszkiewicz, The internet of things: Review and theoretical framework, Expert Syst. Appl. 133 (2019) 97–108.
[3] J. Gama, Knowledge Discovery from Data Streams, CRC Press, 2010.
[4] A. Fernández, S. Río, V. López, A. Bawakid, M. del Jesus, J. Benítez, F. Herrera, Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks, WIREs Data Min. Knowl. Discov. 5 (4) (2014) 380–409.
[5] A. Fernández, C.J. Carmona, M.J. del Jesus, F. Herrera, A View on Fuzzy Systems for Big Data: Progress and Opportunities, Int. J. Comput. Intell. Syst. 9 (1) (2016) 69–80.
[6] A. L'heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data: Challenges and approaches, IEEE Access 5 (5) (2017) 777–797.
[7] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, Inf. Fusion 37 (2017) 132–156.
[8] E. Lughofer, On-line active learning: A new paradigm to improve practical useability of data stream modeling methods, Inform. Sci. (2017) 356–376.
[9] I. Khamassi, M. Sayed Mouchaweh, M. Hammami, K. Ghédira, Discussion and review on evolving data streams and concept drift adapting, Evol. Syst. 9 (1) (2018) 1–23.
[10] I. Škrjanc, J.A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, Inf. Sci. 490 (2019) 344–368.
[11] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, in: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, 2010, p. 10.
[12] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas, Apache flink: Stream and batch processing in a single engine, Bull. IEEE Comput. Soc. Techn. Committee Data Eng. 36 (4) (2015).
[13] A.S. Foundation, Apache storm, 2021, https://storm.apache.org/. (Accessed 15 April 2021).
[14] I. Sittón-Candanedo, R.S. Alonso, J.M. Corchado, S. Rodríguez-González, R. Casado-Vara, A review of edge computing reference architectures and a new global edge proposal, Future Gener. Comput. Syst. 99 (2019) (2019) 278–294.
[15] F. Samie, L. Bauer, J. Henkel, From cloud down to things: An overview of machine learning in internet of things, IEEE Internet Things J. 6 (3) (2019) 4921–4934.
[16] R. Dautov, S. Distefano, R. Buyya, Hierarchical data fusion for smart healthcare, J. Big Data 6 (1) (2019).
[17] M. Hartmann, U.S. Hashmi, A. Imran, Edge computing in smart health care systems: Review, challenges, and research directions, Trans. Emerg. Telecommun. Technol. (July) (2019) 1–25.
[18] F. Ali, S. El-Sappagh, S.R. Islam, D. Kwak, A. Ali, M. Imran, K.-S. Kwak, A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion, Inf. Fusion 63 (2020) 208–222.
[19] M.A. Rahman, M.S. Hossain, An internet-of-medical-things-enabled edge computing framework for tackling covid-19, IEEE Internet Things J. 8 (21) (2021) 15847–15854.
[20] A. Diez-Olivan, J. Del Ser, D. Galar, B. Sierra, Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0, Inf. Fusion 50 (2019) 92–111.
[21] O.C. Abikoye, A.O. Bajeh, J.B. Awotunde, A.O. Ameen, H.A. Mojeed, M. Abdulraheem, I.D. Oladipo, S.A. Salihu, Application of internet of thing and cyber physical system in industry 4.0 smart manufacturing, in: Emergence of Cyber Physical System and IoT in Smart Automation and Robotics, Springer, 2021, pp. 203–217.
[22] B.L. Risteska Stojkoska, K.V. Trivodaliev, A review of internet of things for smart home: Challenges and solutions, J. Cleaner Prod. 140 (2017) 1454–1464.
[23] P. Chamoso, A. González Briones, S. Rodríguez, J. Corchado Rodríguez, Tendencies of technologies and platforms in smart cities: A state-of-the-art review, Wireless Commun. Mob. Comput. 2018 (2018) 1–17.
[24] A. Srivastava, D.K. Das, A comprehensive review on the application of internet of thing (iot) in smart agriculture, Wirel. Pers. Commun. (2021) 1–31.
[25] G. Dong, J. Li, Efficient mining of emerging patterns: Discovering trends and differences, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 1999, pp. 43–52.
[26] A.M. García-Vico, C.J. Carmona, D. Martín, M. García-Borroto, M.J. del Jesus, An overview of emerging pattern mining in supervised descriptive rule discovery: Taxonomy, empirical study, trends and prospects, WIREs: Data Min. Knowl. Discov. 8 (1) (2018).
[27] M. Piao, H.G. Lee, G.Y. Sohn, G. Pok, K.H. Ryu, Emerging patterns based methodology for prediction of patients with myocardial ischemia, in: Proc. of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, 2009, pp. 174–178.
[28] G. Tzanis, I. Kavakiotis, I.P. Vlahavas, Polya-iep: A data mining method for the effective prediction of polyadenylation sites, Expert Syst. Appl. 38 (10) (2011) 12398–12408.
[29] A.M. García-Vico, J. Montes, J. Aguilera, C.J. Carmona, M.J. del Jesus, Analysing Concentrating Photovoltaics Technology through the use of Emerging Pattern Mining, in: Proc. of the 11th International Conference on Soft Computing Models in Industrial and Environmental Applications, Springer, San Sebastián, Spain, 2016, pp. 1–8.
[30] M. Peng, S. Ouyang, J. Zhu, J. Huang, H. Wang, J. Yong, Emerging topic detection from microblog streams based on emerging pattern mining, in: Proc. of the IEEE 22nd International Conference on Computer Supported Cooperative Work in Design, CSCWD, Nanjing, China, 2018, pp. 259–264.
[31] H. Alhammady, Mining streaming emerging patterns from streaming data, in: 2007 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2007, 2007, pp. 432–436.
[32] S.K. Tanbeer, C.F. Ahmed, B.S. Jeong, Y.K. Lee, CP-tree: A tree structure for single-pass frequent pattern mining, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5012 LNAI, Springer Berlin Heidelberg, 2008, pp. 1022–1027.
[33] J.H. Park, H.G. Lee, J.H. Park, Real-time diagnosis system using incremental emerging pattern mining, in: 2010 Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications, CUTE 2010, 2010, pp. 1–5.
[34] J. Bailey, E. Loekito, Efficient incremental mining of contrast patterns in changing data, Inform. Process. Lett. 110 (3) (2010) 88–92.
[35] T.M. Akhriza, Y. Ma, J. Li, A novel Fibonacci windows model for finding emerging patterns over online data stream, in: Proc. of the 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications, 2015, pp. 1–8.
[36] Mining discriminative itemsets in data streams using the tilted-time window model, Knowl. Inf. Syst. 63 (5) (2021) 1241–1270.
[37] A.M. García-Vico, C.J. Carmona, P. González, M.J. del Jesus, MOEA-EFEP: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2861–2872.
[38] A.M. García-Vico, C.J. Carmona, P. González, M.J. d. J. H. Seker, Fepds: A proposal for the extraction of fuzzy emerging patterns in data streams, IEEE Trans. Fuzzy Syst. 28 (12) (2020) 3193–3203.
[39] A. García-Vico, F. Charte, P. González, D. Elizondo, C.J. Carmona, E2pamea: A fast evolutionary algorithm for extracting fuzzy emerging patterns in big data environments, Neurocomputing 415 (2020) 60–73.
[40] Á.M. García-Vico, C.J. Carmona, P. González, H. Seker, M.J. d. Jesus, Fepds: A proposal for the extraction of fuzzy emerging patterns in data streams, IEEE Trans. Fuzzy Syst. 28 (12) (2020) 3193–3203.
[41] Ángel M. García-Vico, C. Carmona, P. González, M.J. del Jesus, A cellular-based evolutionary approach for the extraction of emerging patterns in massive data streams, Expert Syst. Appl. 183 (2021) 115419.
[42] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Wozniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57.
[43] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 44:1–44:37.
[44] A. Sayuri-Iwashita, J. Papa, An overview on concept drift learning, IEEE Access 7 (2019) 1532–1547.
[45] W. Ng, M. Dash, Discovery of frequent patterns in transactional data streams, in: Transactions on Large-Scale Data-and Knowledge-Centered Systems II, Springer, 2010, pp. 1–30.
[46] E. Lughofer, Evolving fuzzy systems fundamentals, reliability, interpretability, useability, applications, in: Handbook on Computational Intelligence: Volume 1: Fuzzy Logic, Systems, Artificial Neural Networks, and Learning Systems, World Scientific, 2016, pp. 67–135.
[47] I. Khamassi, M. Sayed Mouchaweh, Drift detection and monitoring in non-stationary environments, in: 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2014, Linz, Austria, June 2-4 2014, 2014, pp. 1–6.
[48] A. Shaker, E. Lughofer, Self-adaptive and local strategies for a smooth treatment of drifts in data streams, Evol. Syst. 5 (4) (2014) 239–257.
[49] H. Toubakh, M. Sayed-Mouchaweh, Hybrid dynamic data-driven approach for drift-like fault detection in wind turbines, Evol. Syst. 6 (2) (2015) 115–129.
[50] A. Wald, Sequential Analysis, Courier Corporation, 1973.
[51] V. Mayer-Schonberger, K. Cukier, Big Data: The Essential Guide to Work, Life and Learning in the Age of Insight, Hachette UK, 2013.

[52] J. Dean, S. Ghemawat, Mapreduce: Simplified data processing on large clusters, in: Operating Systems Design and Implementation, OSDI, 2004, pp. 137–150.
[53] S. Ramírez-Gallego, A. Fernández, S. García, M. Chen, F. Herrera, Big data: Tutorial and guidelines on information and process fusion for analytics algorithms with mapreduce, Inf. Fusion 42 (2018) 51–61.
[54] P. Kralj-Novak, N. Lavrac, G.I. Webb, Supervised Descriptive Rule Discovery: A Unifying Survey of Constrast Set, Emerging Pattern and Subgroup Mining, J. Mach. Learn. Res. 10 (2009) 377–403.
[55] F. Herrera, C.J. Carmona, P. González, M.J. del Jesus, An overview on Subgroup Discovery: Foundations and Applications, Knowl. Inf. Syst. 29 (3) (2011) 495–525.
[56] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery: An overview, in: Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Menlo Park, CA, USA, 1996, pp. 1–34.
[57] D. Gamberger, N. Lavrac, Expert-guided subgroup discovery: Methodology and application, J. Artif. Intell. Res. 17 (2002) 501–527.
[58] W. Kloesgen, Explora: A multipattern and multistrategy discovery assistant, in: Advances in Knowledge Discovery and Data Mining, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996, pp. 249–271.
[59] C.J. Carmona, M.J. del Jesus, F. Herrera, A Unifying Analysis for the Supervised Descriptive Rule Discovery via the Weighted Relative Accuracy, Knowl.-Based Syst. 139 (2018) 89–100.
[60] L. Wang, H. Zhao, G. Dong, J. Li, On the complexity of finding emerging patterns, in: Proc. of the 28th Annual International Computer Software and Applications Conference, Vol. 2, 2004, pp. 126–129.
[61] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. Very Large Data Bases, in: VLDB, vol. 1215, 1994, pp. 487–499.
[62] J.Y. Li, K. Ramamohanarao, G.Z. Dong, The space of jumping emerging patterns and its incremental maintenance algorithms, in: Proc. of the 17th International Conference on Machine Learning, 2000, pp. 551–558.
[63] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, ACM Sigmod Record 29 (2) (2000) 1–12.
[64] J. Bailey, T. Manoukian, K. Ramamohanarao, Fast Algorithms for Mining Emerging Patterns, in: Principles of Data Mining and Knowledge Discovery, vol. 2431, Springer, 2002, pp. 187–208.
[65] H. Fan, K. Ramamohanarao, Efficiently mining interesting emerging patterns, in: Proc. of the 4th International Conference on Web-Age Information Management, 2003, pp. 189–201.
[66] H. Fan, K. Ramamohanarao, Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers, IEEE Trans. Knowl. Data Eng. 18 (6) (2006) 721–737.
[67] P. Terlecki, K. Walczak, Efficient Discovery of Top-K Minimal Jumping Emerging Patterns, in: Proc. of the 6th International Conference Rough Sets and Current Trends in Computing, Springer, 2008, pp. 438–447.
[68] Q. Liu, P. Shi, Z. Hu, Y. Zhang, A novel approach of mining strong jumping emerging patterns based on BSC-tree, Internat. J. Systems Sci. 45 (3) (2014) 598–615.
[69] M. García-Borroto, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, M.A. Medina-Pérez, J. Ruiz-Shulcloper, LCMine: An efficient algorithm for mining discriminative regularities and its application in supervised classification, Pattern Recognit. 43 (9) (2010) 3025–3034.
[70] L. Wang, Y. Wang, D. Zhao, Building emerging pattern (ep) random forest for recognition, in: Proc. of the 17th IEEE International Conference on Image Processing, 2010, pp. 1457–1460.
[71] M. García-Borroto, J. Martínez-Trinidad, J. Carrasco-Ochoa, Fuzzy emerging patterns for classifying hard domains., Knowl. Inf. Syst. 28 (2) (2011) 473–489.
[72] O. Loyola-González, M.A. Medina-Pérez, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, R. Monroy, M. García-Borroto, PBC4cip: A new contrast pattern-based classifier for class imbalance problems, Knowl.-Based Syst. 115 (2017) 100–109.
[73] D.L. Aguilar, O. Loyola-González, M.A. Medina-Pérez, L. Cañete-Sifuentes, K.-K.R. Choo, PBC4occ: A novel contrast pattern-based classifier for one-class classification, Future Gener. Comput. Syst. 125 (2021) 71–90.
[74] M. Nandhini, M. Rajalakshmi, S. Sivanandam, Experimental and statistical analysis on the performance of firefly based predictive association rule classifier for health care data diagnosis, J. Control Eng. Appl. Inform. 19 (2) (2017) 101–110.
[75] R.A. Mohammed, M.G. Duaimi, Association rules mining using cuckoo search algorithm, Int. J. Data Min., Model. Manag. 10 (1) (2018) 73–88.
[76] Y. Djenouri, D. Djenouri, A. Belhadi, P. Fournier-Viger, J.C.-W. Lin, A. Bendjoudi, Exploiting gpu parallelism in improving bees swarm optimization for mining big transactional databases, Inform. Sci. 496 (2019) 326–342.
[77] C.J. Carmona, P. González, M.J. del Jesus, F. Herrera, NMEEF-SD: Non-dominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery, IEEE Trans. Fuzzy Syst. 18 (5) (2010) 958–970.
[78] J.M. Luna, J.R. Romero, C. Romero, S. Ventura, On the Use of Genetic Programming for Mining Comprehensible Rules in Subgroup Discovery, IEEE Trans. Cybern. 44 (12) (2014) 2329–2341.
[79] V. Pachón, J. Mata, J. Domínguez, Searching for the most significant rules: An evolutionary approach for subgroup discovery, Soft Comput. 21 (10) (2017) 2609–2618.
[80] J.-D. Cryans, S. Ratté, R. Champagne, Adaptation of apriori to mapreduce to build a warehouse of relations between named entities across the web, in: 2010 Second International Conference on Advances in Databases, Knowledge, and Data Applications, 2010, pp. 185–189.
[81] W. Jiang, V.T. Ravi, G. Agrawal, A map-reduce system with an alternate api for multi-core environments, in: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE, 2010, pp. 84–93.
[82] X. Zhou, Y. Huang, An improved parallel association rules algorithm based on mapreduce framework for big data, in: 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD, IEEE, 2014, pp. 284–288.
[83] Q. Ding, Q. Ding, W. Perrizo, Parm—an efficient algorithm to mine association rules from spatial data, IEEE Trans. Syst. Man Cybern. B 38 (6) (2008) 1513–1524.
[84] H. Li, Y. Wang, D. Zhang, M. Zhang, E.Y. Chang, Fp: Parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 107–114.
[85] D. Savage, X. Zhang, P. Chou, X. Yu, Q. Wang, Distributed mining of contrast patterns, IEEE Trans. Parallel Distrib. Syst. 28 (7) (2016) 1881–1890.
[86] F. Pulgar-Rubio, A.J. Rivera-Rivas, M.D. Pérez-Godoy, P. González, C.J. Carmona, M.J. del Jesus, MEFASD-BD: Multi-Objective Evolutionary Fuzzy Algorithm for Subgroup Discovery in Big Data Environments - A MapReduce Solution, Knowl.-Based Syst. 117 (2017) 70–78.
[87] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[88] T.H. Li, C. Lucasius, G. Kateman, Optimization of calibration data with the dynamic genetic algorithm, Anal. Chim. Acta 2768 (1992) 123–134.
[89] F. Padillo, J.M. Luna, S. Ventura, A grammar-guided genetic programing algorithm for associative classification in big data, Cogn. Comput. 11 (3) (2019) 331–346.
[90] A.M. García-Vico, C.J. Carmona, P. González, M.J. del Jesus, A big data approach for extracting fuzzy emerging patterns, Cogn. Comput. 11 (3) (2019) 400–417.
[91] H. Li, N. Zhang, J. Zhu, Y. Wang, H. Cao, Probabilistic frequent itemset mining over uncertain data streams, Expert Syst. Appl. 112 (2018) 274–287.
[92] S. Saleti, S. R.B.V, A mapreduce solution for incremental mining of sequential patterns from big data, Expert Syst. Appl. 133 (2019) 109–125.
[93] V.Q.P. Huynh, J. Küng, Fpo tree and dp3 algorithm for distributed parallel frequent itemsets mining, Expert Syst. Appl. 140 (2020) 112874.
[94] H. Alhammady, K. Ramamohanarao, Mining emerging patterns and classification in data streams, in: Proceedings - 2005 IEEE/WIC/ACM InternationalConference on Web Intelligence, WI 2005 2005, 2005, pp. 272–275.
[95] K. Yu, W. Ding, D. Simovici, H. Wang, J. Pei, X. Wu, Classification with streaming features: An emerging-pattern mining approach, ACM Trans. Knowl. Discov. Data 9 (4) (2015) 30:1–30:31.
[96] M. Seyfi, Mining Discriminative Itemsets in Data Streams using Different Window Models (Ph.D. thesis), Queensland University of Technology, 2018.
[97] J. de Andrade Silva, E.R. Hruschka, J. Gama, An evolutionary algorithm for clustering data streams with a variable number of clusters, Expert Syst. Appl. 67 (2017) 228–238.
[98] V. Alves, R. Campello, E. Hruschka, Towards a fast evolutionary algorithm for clustering, in: 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 1776–1783.
[99] E. Ruiz, J. Casillas, Adaptive fuzzy partitions for evolving association rules in big data stream, Internat. J. Approx. Reason. 93 (2018) 463–486.
[100] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, 2001.
[101] E. Loekito, J. Bailey, Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams, in: Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 307–316.
[102] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Parts I, II, III, Inform. Sci. 8–9 (1975) 199–249, 301–357, 43–80.
[103] B.L. Miller, D.E. Goldberg, Genetic Algorithms, Tournament Selection, and the Effects of Noise, Complex Syst. 9 (1995) 193–212.
[104] J.H. Holland, Adaptation in Natural and Artificial Systems, second ed., University of Michigan Press, 1975.
[105] A. García-Vico, C.J. Carmona, P. González, M.J. del Jesus, Fepds: A proposal for the extraction of fuzzy emerging patterns in data streams, IEEE Trans. Fuzzy Syst. 28 (12) (2020) 3193–3203.
[106] J. Li, J. Liu, H. Toivonene, K. Satou, Y. Sun, B. Sun, Discovering statistically non-redundant subgroups, Knowl.-Based Syst. 67 (2014) 315–327.
[107] K.S. Leung, Y. Leung, L. So, K.F. Yam, Rule Learning in Expert Systems Using Genetic Algorithm: 1, Concepts, in: K. Jizuka (Ed.), Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks, 1992, pp. 201–204.
[108] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, MOA: Massive online analysis, J. Mach. Learn. Res. 11 (2010) 1601–1604, URL https://moa.cms.waikato.ac.nz/.