International Journal of Approximate Reasoning 139 (2021) 69–87

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Incremental fuzzy rough sets based feature subset selection using fuzzy min-max neural network preprocessing ☆,☆☆

Anil Kumar, P.S.V.S. Sai Prasad *

School of CIS, University of Hyderabad, Hyderabad, India

ARTICLE INFO

Article history:
Received 6 May 2021
Received in revised form 25 July 2021
Accepted 16 September 2021
Available online 22 September 2021

Keywords:
Fuzzy rough set
Fuzzy discernibility matrix
Incremental learning
Fuzzy min-max neural network
Feature subset selection

ABSTRACT

Fuzzy rough sets (FRS) provides effective ways for selecting the compact/relevant feature subset for hybrid decision systems. However, the underlying complexity of the existing FRS methods through batch processing is often costly or intractable on large data and also suffer from continuous model adaptation on dynamic data. This paper proposes a FRS based incremental feature subset selection (IvFMFRS) framework using fuzzy min-max neural network (FMNN) as a preprocessor step in aiding to deal with data dynamically without sacrificing classification performance. FMNN is a single epoch learning algorithm employed to construct fuzzy hyperboxes (information granules) of pattern spaces very fast. Fuzzy hyperboxes facilitate the formation of interval-valued decision system (IDS) from the numerical decision system of much smaller size. In IvFMFRS, on each sample subset arrival, an incremental mechanism for updating fuzzy discernibility matrix (FDM) based on constructed IDS is first formulated and then update feature subset by adding and deleting features based on updated FDM. A comparative analysis has been conducted comprehensively to assess the performance of the proposed algorithm with the existing FRS methods on numerical datasets. And, the results show that the IvFMFRS obtained the relevant feature subsets with similar classification accuracy with significantly less computational time than existing FRS methods.

1. Introduction

Data mining is an essential process to infer the underlying structural pattern and knowledge from given data. The majority of data mining applications are restricted to classical batch setting; i.e., the entire data are provided prior to training for learning. Sometimes batch procedures can not work for large data that easily exceed the memory limit. Moreover, they may also lack in model adaptability according to constantly arriving new information/data thus resulting in the reconstruction of new models from scratch, which is repeatedly a time-consuming task. Incremental learning, in contrast, is a solution for dynamic data with properties of gradual model adaptation on sequentially arriving data without sacrificing model accu-

☆ The work is supported by DST, Government of India under ICPS project [Grant Number: File No. DST/ICPS/CPS-Individual/2018/579] and UoH-IoE by MHRD, Government of India [Grant Number: F11/9/2019-U3(A)].

☆☆ Authors would like to acknowledge Zhang et al. [1], Yang et al. [2] and Zhang et al. [3] for providing the source code for PARA, IV-FS-FRS and AIFWAR.

* Corresponding author.
E-mail addresses: anilhcu@uohyd.ac.in (A. Kumar), saics@uohyd.ac.in (P.S.V.S. Sai Prasad).
URL: http://scis.uohyd.ac.in/~saics/ (P.S.V.S. Sai Prasad).

https://doi.org/10.1016/j.ijar.2021.09.006
0888-613X/

racy and works with limited memory resources. The important challenge of incremental learning is to retain the previously acquired knowledge while getting new information [4].

In recent times, there has been ubiquitous growth of data in every field. And, it remains a challenge for feature selection techniques to cope with higher complexity in data. Feature selection is a core concept in data mining that hugely impacts the performance of the learning model. It is necessary to preprocess the data to exclude irrelevant features before applying them in the learning model. In the 1980s, Z. Pawlak [5] has proposed a theoretical framework called rough set theory (RST) for feature subset selection (called reduct) and knowledge discovery in data. However, RST is usually applied to categorical decision systems. RST can work on numeric decision systems but result in finer granularity in the feature subset. So, the induced rules from the feature subset suffer poor generalizability to test data.

Later, Dubois et al. [6] have presented the theory of fuzzy rough sets (FRS) as a further extension of RST to work on numeric or hybrid decision systems. FRS has proven its remarkable role in feature subset selection even though data is imprecise and uncertain [7,8]. Several aspects of improvement and extensions [9–21] for FRS reduct computation have been developed. However, despite being these modifications and extensions, the implementation of FRS approaches is restricted majorly to batch processing, and there is a lack of dealing with dynamic data for reduct computation. Even for processing large datasets, these existing FRS methods are intractable due to exceeding the memory limitation.

In the last decades, researchers have been exploring to process dynamic data or large volume data through incremental learning methodologies to avoid unnecessary recomputations. Incremental learning, meanwhile, minimizes the complexities of time and space regarding processing and storage. This idea has prompted several researchers who investigated the incremental perspective of reduct computation in the framework of RST for categorical decision systems to deal with a dynamic dataset. These algorithms have been basically investigated in various scenarios such as variation of feature set (adding and deleting features) and the sample set (adding and deleting objects), respectively. For incrementally adding and deleting features, some incremental reduct computation algorithms are introduced based on information entropy [22], discernibility matrix [23], knowledge granularity [24,25] and positive domain [26]. For incrementally adding and deleting objects, there are some incremental algorithms based on information entropy [27–29], discernibility matrix [30,31], knowledge granularity [25,32], positive domain [33], bijective soft sets [34] and representative instances [35].

There have been a few studies based on FRS based incremental feature selection algorithm [2,36,37,3,38] so far. In 2015, Zeng et al. [37] have presented an incremental feature selection based on the gaussian kernelized FRS in hybrid decision systems with updating features sequentially. In 2017, Yang et al. [2] have proposed an approach based on incremental feature selection for FRS on arriving subset. The approach is to update the relative discernibility relation with an update in selected subsets on each arriving subset. Still, this approach suffers in terms of memory space on the construction of relative discernibility relation on large data. To overcome the space complexity, in 2020, Zhang et al. [3] have proposed an incremental feature selection algorithm (AIFWAR) employing FRS based information entropy with the incoming subsets. In [3], the author aims first to generate representative instances from the arriving subset and then compute an incremental mechanism to calculate reduct by using FRS based information entropy. After that, a wrapper procedure is applied to the resultant reduct to select the best feature subset, achieving maximum accuracy by employing a classification model. In 2020, Peng et al. [38] have introduced an incremental feature selection algorithm based on FRS (PIAR) using representative instances. On each incoming sample, the key instance set (representative instances) is selected and then update the positive region and dependency function to compute the feature subset using the key instance set [38].

In [16,35,2,3,38] the authors have calculated an approximate reduct. Slezak [39] first introduces the idea of the approximate reduct having the potential to achieve near to exact reduct capability.

The existing RST or FRS methods require an object-based computation that impacts an increase in space and computation overhead. Nevertheless, they do not concern processing through information granules (refer as Granular Computing) aspect in FRS methods to perform feature subset selection which establishes a significant reduction in space and computational cost in large data.

In 2020, Anil et al. [40] have introduced a novel scalable FRS feature subset selection method (FMNN-FRS) utilizing a fuzzy min-max neural network as a preprocessor. Simpson [41] has proposed a powerful single pass dynamic neural network technique, known as “fuzzy min-max neural network” (FMNN), employed to create information granules representatives (called fuzzy hyperboxes) to understand the pattern space better. In FMNN-FRS, the given numeric decision system is first transformed into an interval-valued decision system (IDS) using FMNN preprocessor. And, then the fuzzy discernibility matrix (FDM) is constructed based on IDS to compute an approximate reduct. Here, the author’s idea is to build FDM based on hyperbox space instead of object space. Consequently, this approach significantly reduces the time and space complexity regarding the processing and storage of large data. Even in [40], authors assess the capability of constructing the model with those datasets where the existing FRS models are intractable to compute in a given system. Motivated from the above observations [40] using FMNN and to deal with dynamic datasets, our main contribution is as follows:

1. We propose an FRS-based incremental feature subset selection using FMNN as a preprocessor to deal with dynamic datasets named IvFMFRS.
2. FDM is incrementally constructed upon the sample arrival using updated fuzzy hyperboxes, which are viewed as a preprocessing step in IvFMFRS.

3. Based on updated FDM, a sequential forward selection (SFS) [14] based algorithm is used for updating the current reduct restricted to modified or newly added entries in FDM. Then sequential backward elimination (SBE) [42] strategy is employed to remove redundant features in the current reduct.

The difference between our proposed incremental algorithm IvFMFRS and FMNN-FRS are summarized as follows:

1. Our proposed algorithm is an incremental adaptation of the method in [40] to deal with a dynamic dataset with retaining the comparative classification performance.
2. FMNN-FRS performs in entire data, whereas the proposed algorithm computes with subsequently arriving instances (dynamic data).
3. FMNN-FRS employs SFS strategy for reduct computation, whereas the proposed algorithm applies SFS strategy followed by an SBE strategy to compute reduct on subsequently arriving instances.

Experimental comparison of IvFMFRS has been demonstrated by comparing with existing state-of-art FRS based incremental algorithms and non-incremental algorithms including FMNN-FRS algorithm to validate the comparable classification performance on real-valued datasets.

The rest of this paper is designed as follows. Section 2 briefly provides the basics of fuzzy discernibility matrix based on fuzzy rough sets, fuzzy min-max neural network and overview of the related work. Section 3 consists of the proposed IvFMFRS algorithm. Section 4 shows the comparative experiments and analysis of results with other FRS approaches. Lastly, the paper closes with a conclusion and directions for future work.

2. Preliminaries and related work

In this section, we briefly review the basics of fuzzy discernibility matrix and attribute reduction based on fuzzy rough sets. Also, we overview the concepts of fuzzy min-max neural network. Last, we present a summary of related work that is useful for presenting our work.

2.1. Preliminaries

2.1.1. Decision-relative fuzzy discernibility matrix with fuzzy rough sets

RST can only operate effectively on datasets containing discrete attributes. Dubois et al. [6] introduce the fuzzy rough sets (FRS) as an extension of RST to work on real-valued attributes. RST generalizes to a fuzzy set by allowing that objects can belong to a concept to varying degrees. The notion of FRS appears completely in fuzzy context, having fuzzy sets on a universe U to represent greater indiscernibility, hence, allowing greater flexibility in modeling.

Let $DT = (U, C \cup D)$ with $C \cap D = \emptyset$ be a decision system where U is a non-empty finite set of samples (or, termed as Universe) and C is a non-empty real-valued conditional attributes, and D is a set of symbolic decision attributes and usually contains a single decision variable d.

In FRS, fuzzy similarity relation (R) succeeds on crisp equivalence relation in RST [14]. A fuzzy similarity relation should hold reflexive, symmetry and T-transitivity properties. Many approaches exist in the literature to construct similarity relations. An example of R_a for $(x, x') \in U$ is given as:

$$\mu_{R_a}(x, x') = 1 - \frac{|a(x) - a(x')|}{|a_{\max} - a_{\min}|} \tag{1}$$

where μ_{R_a} is the degree to which objects x and x' are similar for feature 'a'.

Fuzzy discernibility relation is a generalization of crisp discernibility relation in RST [14]. Fuzzy discernibility relation ($DR_a(x, x')$) is the fuzzy negation of fuzzy similarity relation R_a between pair of objects x and x', as given in Equation (2).

$$\mu_{DR_a}(x, x') = Neg(\mu_{R_a}(x, x')) \tag{2}$$

where Neg is a fuzzy negation, and $Neg(\mu_{R_a}(x, x'))$ is the degree of the fuzzy discernibility w.r.t. attribute 'a'. Usually, the fuzzy negation of similarity measure between two objects is taken as standard negation,

$$Neg(\mu_{R_a}(x, x')) = 1 - \mu_{R_a}(x, x')$$

Fuzzy discernibility relation is represented in the form of a Fuzzy discernibility matrix (FDM). Each entry (known as clauses) of FDM is represented as $M(x, x')$. $M(x, x')$ is a set of values of discernibility between x and x' over all conditional attributes. This FDM construction considers only those entries corresponding to a pair of objects belonging to different classes. Each entry in FDM $M(x, x')$, $\forall (x, x') \in U$ is defined as:

$$M(x, x') = \begin{cases} \{a_s |\ a \in C, s = Neg(\mu_{R_a}(x, x'))\}, & \text{if d(x)} \neq \text{d(x')} \\ \emptyset, & \text{otherwise} \end{cases} \tag{3}$$

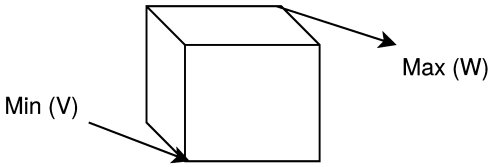

Fig. 1. The min-max hyperbox in 3-D.

The entry in $M(x, x')$ shows that objects x and x' are discerning with a certain degree in range between 0 to 1 w.r.t. all attributes. For example, $M(x, x') = \{a_{0.64}, b_{0.35}, c_{0.87}\}$ indicates x and x' have the degree of fuzzy indiscernibility of 0.64 based on attribute 'a', 0.35 based on attribute 'b' and 0.84 based on attribute 'c'.

The degree of satisfaction for $M(x, x')$ for a given subset of attributes P w.r.t. D is defined as:

$$SAT_{P,D}(M(x, x')) = S_{a \in P}\{Neg(\mu_{R_a}(x, x'))\} \tag{4}$$

where S is a t-conorm. Thus, the total satisfiability of entire clauses for subset P is defined as:

$$SAT(P) = \frac{\sum_{x,x' \in U, x \neq x'} SAT_{P,D}(M(x, x'))}{\sum_{x,x' \in U, x \neq x'} SAT_{C,D}(M(x, x'))} \tag{5}$$

A minimal subset of conditional attributes $P \subseteq C$ is referred to as FRS based reduct if and only if the following conditions hold:

- $SAT(P) = SAT(C) = 1$
- $\forall P' \subset P, SAT(P') < SAT(P)$

2.1.2. Fuzzy min-max neural network

Simpson [41] introduces a powerful supervised learning process called "Fuzzy Min-Max Neural Network" using an aggregation of n-dimensional hyperbox fuzzy sets to represent the multiple classes. Each n-dimensional hyperbox (B_j) bounds a subregion characterized by a minimum point (V) and maximum point (W) with corresponding fuzzy membership function (b_j), as depicted in Fig. 1. The hyperbox is represented as:

$$B_j = \{X_h, V_j, W_j, b(X_h)\} \quad \forall X_h \in I^n \tag{6}$$

where $X_h = (x_{h1}, x_{h2}, ..., x_{hn})$ is input pattern in n-dimensional space and, $V_j = (v_{j1}, v_{j2}, ..., v_{jn})$ and $W_j = (w_{j1}, w_{j2}, ..., w_{jn})$ are the corresponding minimum points and maximum points for hyperbox B_j. I^n is the n-dimensional unit pattern space.

The fuzzy membership function ($b_j(X_h)$) that describes the membership value of the input pattern X_h w.r.t. hyperbox B_j is defined as:

$$b_j(X_h) = \frac{1}{2n}\sum_{i=1}^{n}[max(0, 1 - max(0, \gamma\ min(1, x_{hi} - w_{ji}))) + max(0, 1 - max(0, \gamma\ min(1, v_{ji} - x_{hi})))] \tag{7}$$

where γ is the sensitive parameter that regulates how fast the membership decreases as the distance between X_h and B_j increases and $0 \leq b_j(X_h) \leq 1$.

For each input pattern, the learning in the network is happening through three phases, i.e., expansion, overlap tests, and contraction steps. During the training process, each input pattern explores existing fuzzy hyperbox with the same class that gives full membership value. Otherwise, the network tries to find a corresponding class hyperbox with the highest membership for expansion. For the hyperbox B_j to expand and to include the input pattern X_h, the following criterion must be satisfied:

$$\sum_{i=1}^{n}\left(max(w_{ji}, x_{hi}) - min(v_{ji}, x_{hi})\right) \leq n\theta \tag{8}$$

In Equation (8), *theta* (θ) is a user-defined expansion parameter with a range of ($0 \leq \theta \leq 1$) and controls the maximum size of the hyperbox.

If the expansion criterion is satisfied, then the adjustment of minimum and maximum points of B_j to incorporate input pattern is given in Equation (9) and Equation (10).

$$v_{ji}^{new} = min(v_{ji}^{old}, x_{hi}) \quad \forall i = 1, 2, 3, \ldots, n. \tag{9}$$

$$w_{ji}^{new} = max(w_{ji}^{old}, x_{hi}) \quad \forall i = 1, 2, 3, \ldots, n. \tag{10}$$

If the input pattern on the selected hyperbox does not fulfill the expansion condition, then a new point hyperbox will be generated to contain the input pattern.

After expansion, the expanded hyperbox is further examined for overlap with existing hyperboxes of different classes. This overlap test is performed dimension by dimension, as defined in [41]. If an overlapping region occurs in each dimension during the overlap test, then the smallest overlap dimension is retained for the contraction steps. The selected dimension is adjusted to min-max points through contraction rule to remove ambiguity between hyperboxes, as defined in [41]. FMNN learning process is achieved through the placing and adjustment of hyperboxes by storing their min-max points.

2.2. Related work

2.2.1. FMNN-FRS algorithm

Anil et al. [40] introduce an algorithm FMNN-FRS to scale the FRS reduct computation using FMNN as a preprocessor. The FMNN model creates hyperboxes as learning in the pattern spaces. Authors restrict FMNN learning until the expansion step of hyperboxes only because it tends to lose the information of the shared boundary region between hyperboxes of different classes through contraction steps, as described in Section 2.1.2.

The author's idea is first to construct the interval-valued decision system (IDS) through hyperboxes of numeric decision system. Let IDS = $(HS, C \cup \{d\})$ be a interval-valued decision system, where $HS = \{h_1, h_2, \ldots, h_r\}$ represents the universe of hyperboxes. Let $[V^h, W^h]$ denote the minimum and maximum points of hyperbox h. In IDS, the value of an hyperbox $h \in HS$ over an attribute $c \in C$ is represented by the interval v_c^h to w_c^h ($[v_c^h, w_c^h]$), where, v_c^h is component of minimum point V^h and w_c^h is component of maximum point W^h corresponding to the attribute c. The value of decision attribute d is taken as per the decision class to which h belongs.

Then, instead of object wise creation of FDM, here, the author's purpose is to build FDM based on IDS. Each entry (clause) in FDM, denoted as $M(h_x, h_y)$, corresponds to a pair of hyperboxes belonging to different classes. Each clause $M(h_x, h_y)$ contains the set of fuzzy discernibility measures of all attributes in C between h_x and h_y. In FMNN-FRS, authors used negation of Jaccard similarity measure [43] to compute discernibility between intervals (I_x, I_y). Jaccard similarity J_S is defined as:

$$J_S(I_x, I_y) = \frac{|I_x \cap I_y|}{|I_x \cap I_y| + |I_x \setminus I_y| + |I_y \setminus I_x|} \tag{11}$$

where, $|I_x \cap I_y|$ is the size of intersection between I_x and I_y, $|I_x \setminus I_y|$ is the size of the interval segment of I_x that are not overlapping with I_y and $|I_y \setminus I_x|$ is the size of the interval segment of I_y that is not overlapping with I_x. Hence, $J_S(I_x, I_y) = 0$, when I_x and I_y are not overlapping to each other, means both intervals are completely different from each other and $J_S(I_x, I_y) = 1$, when I_x and I_y are fully overlapping to each other, means both intervals are completely identical.

The FDM entry between h_x and h_y belonging to different classes is defined by using J_S as:

$$M(h_x, h_y) = \begin{cases} \{c_s \mid c \in C, s = Neg(J_S([v_c^{h_x}, w_c^{h_x}], [v_c^{h_y}, w_c^{h_y}]))\}, & \text{if } \mathrm{d}(h_x) \neq \mathrm{d}(h_y) \\ \emptyset, & \text{otherwise} \end{cases} \tag{12}$$

The component corresponding to $c \in C$ is:

$$M^c(h_x, h_y) = Neg(J_S([v_c^{h_x}, w_c^{h_x}], [v_c^{h_y}, w_c^{h_y}])) \tag{13}$$

This FDM construction is an approximation of FDM through object-wise. After construction, authors followed the SFS strategy (SAT measure from [14]) for finding an approximate reduct. In each iteration, SAT is calculated Equation (5) for $(red \cup \{c\})$ $\forall c \in C - red$. The attribute having maximum SAT values is included in reduct red. When $SAT(Red)$ becomes to $SAT(C)$, then the algorithm terminates and returns a final reduct red.

3. Proposed IvFMFRS algorithm

This section describes the incremental IvFMFRS algorithm to compute an approximate FRS reduct utilizing FMNN learning model. The proposed algorithm has extended the FMNN-FRS algorithm from [40] to an incremental perspective for computing a reduct for the real-valued dynamic decision system, where data sample subsets are arriving in the sequence.

3.1. Incremental environment description and notation

This section describes the incremental environment and the used symbols/notations in algorithms, which is our proposed IvFMFRS approach. The description of function and notation inside the Algorithm 1, Algorithm 2 and Algorithm 3 are mentioned in Table 1. Also, we present a flowchart of IvFMFRS algorithm for better understandability and is depicted in Fig. 2.

Here, we assume that the data is presented in sample subsets $(U_1, U_2, U_3, \ldots)$ that arriving sequentially and condition $(U_i \cap U_j = \emptyset \ for \ i \neq j)$ must be satisfied. So, in each iteration, a new sample subset is provided to an algorithm to perform incrementally.

Table 1
Description of Function Name and Notation in Algorithm.

Notation	Meaning
FM	Represents FMNN learning model
$FM.Belongs(x)$	Check absolute membership value of x on any existing hyperboxes of same class.
$Next$	Break the current iteration and continues the next iteration in the loop.
$FM.HMemb(x)$	Find the highest membership value correspond to x with existing hyperbox of same class label.
$FM.Exp(h, x)$	Check expansion of h to include x is possible or not using expansion criterion Equation (8)
$Remove(HS, h)$	Remove h from set HS.
$FM.Expand(h, x)$	Expand the hyperbox h to include x using Equation (9) and Equation (10).
$Insert(HS, h)$	Insert newly h into set of hyperbox HS.
$Update(HS, h)$	Update the expanded hyperbox h in the set HS.
$FM.Create(x)$	Create a new hyperbox to include x.
$A \cup B$	Merge of A and B.

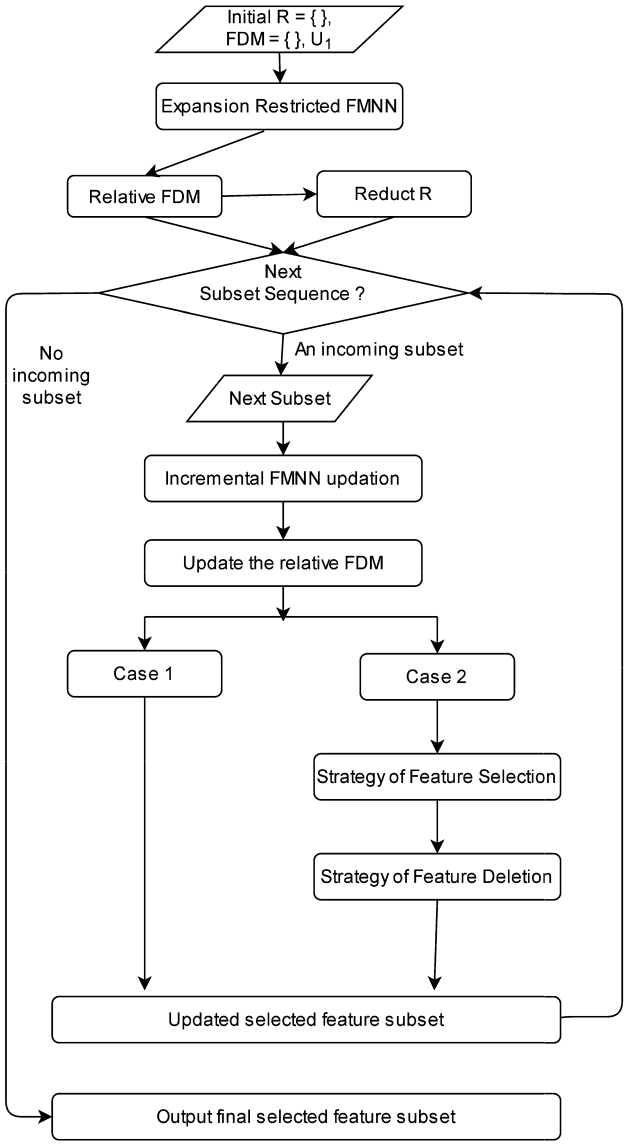

Fig. 2. Flow chart of Incremental IvFMFRS.

Initially, we compute the base reduct R_1 and fuzzy discernibility matrix M_1 through FMNN-FRS algorithm for a sample U_1 to our algorithm. For the next sample subset arrival U_2, we apply our incremental IvFMFRS with given R_1 and M_1 to result next reduct R_2 based on updated fuzzy discernibility matrix M_2. Similarly, the algorithm is repeated for subsequent samples.

Basically, for arriving sample U_{i+1}, Algorithm 1, Algorithm 2 and Algorithm 3 begin by giving as input HS_i, M_i and R_i for further calculation and get output HS_{i+1}, M_{i+1} and R_{i+1} respectively, where R_i is the current reduct, HS_i is the current set of hyperboxes, M_i is the current fuzzy discernibility matrix.

3.2. Updating fuzzy hyperboxes through FMNN learning model

This section shows the updation of existing hyperboxes HS_i by training FMNN with a new batch sample U_{i+1}. Algorithm 1 performs updation of hyperboxes. For incubation of an input pattern x in hyperbox space, if x is giving an absolute membership value with any existing hyperbox of the same class using Equation (7), then no modification on hyperbox takes place. If x is outside of hyperboxes of the same class, then an hyperbox h correspond to the highest membership value is selected to verify whether it can be expanded or not using expansion criterion Equation (8). If yes, then hyperbox h is expanded to accommodate input x by adjusting their min and max points of h using Equation (9) and Equation (10). If not, then a new point hyperbox is created to incorporate input pattern x.

After completion of training with all input patterns in U_{i+1}, hyperboxes are divided into three categories. First, hyperboxes which are existing in HS_i but not yet modified. Second, those hyperboxes which are existing in HS_i but they are modified as part of the expansion process, symbolized as HS^{mod}. These hyperboxes are removed from HS_i and kept in set HS^{mod}. Third, new hyperboxes are created for representing new training patterns and also may be updated as part of the expansion process. These hyperboxes are saved in set HS^{new}. Hence, the final hyperbox space $HS_{i+1} = HS_i \cup HS^{mod} \cup HS^{new}$.

Algorithm 1: Updating Fuzzy Hyperboxes through FMNN.

Input : U_{i+1}, HS_i
Output: HS_{i+1}, HS^{mod}, HS^{new}
Initialize, $HS^{mod} = \emptyset, HS^{new} = \emptyset$;
Let FM represents FMNN model comprises of $HS_i \cup HS^{mod} \cup HS^{new}$;
for *every x in* U_{i+1} **do**
  if *FM.Belongs(x)* $== True$ **then**
    Next;
  end
  h = FM.HMemb(x);
  if *h exist* **then**
    if $FM.Exp(h, x) == True$ **then**
      if $h \in HS_i$ **then**
        Remove(HS_i, h); FM.Expand(h, x);
        Insert(HS^{mod}, h);
      else if $h \in HS^{new}$ **then**
        FM.Expand(h, x); Update(HS^{new}, h);
      else if $h \in HS^{mod}$ **then**
        FM.Expand(h, x); Update(HS^{mod}, h);
      end
    else
      h^n = FM.Create(x); Insert(HS^{new}, h^n);
    end
  else
    h^n = FM.Create(x); Insert(HS^{new}, h^n);
  end
end
$HS_{i+1} = \{HS_i \cup H^{mod} \cup H^{new}\}$;
Return $HS_{i+1}, HS^{mod}, HS^{new}$

3.3. Updating fuzzy discernibility matrix

From the discussion in the above section, three categories of hyperboxes are generated, which is a necessary step in updating current FDM (M_i) incrementally. Algorithm 2 performs an update of FDM. Hyperboxes in the first category HS_i are unmodified; hence they don't contribute to changes into any existing FDM entries. In an ideal scenario, we have the hyperboxes in HS_i that are representative of all new patterns in U_{i+1}. In that case, no modification of hyperboxes structure occurs which mean $HS_{i+1} = HS_i$. Hence, there is no updation of FDM, and reduct remains unchanged. So, R_{i+1} becomes R_i and the algorithm immediately returns reduct. The chance for this ideal scenario increases as more training data arrives.

Whereas in the second category, modified hyperboxes HS^{mod} change the learning model with adjusting of their V (min point) and W (max point) values, so that their respective entries in current FDM will also be modified.

For the third category, the hyperboxes in HS^{new} are the new objects augmented to current IDS. For their FDM entries, corresponding new entries are added into existing FDM. These hyperboxes are compared with different class hyperboxes of HS_{i+1}.

Modified entries of FDM resulting from second and third category hyperboxes are placed in a new collection M^{new} representing either new or updated entries of current FDM. The final FDM is $M_{i+1} = M_i \cup M^{new}$.

Algorithm 2: Updating Fuzzy Discernibility Matrix.

Input : HS_i, HS^{mod}, HS^{new}, M_i
Output: M_{i+1}, M^{new}
1 Initialize, $M^{new} = \emptyset$;
2 **for** *each* $h^m \in HS^{mod}$ **do**
3 **for** *each* $h \in (HS_i \cup HS^{mod})$ **do**
4 **if** $d(h^m) \neq d(h)$ **then**
5 Update $M_i(h^m, h)$ using Equation (12);
6 $M^{new} = M^{new} \cup \{M_i(h^m, h)\}$;
7 **end**
8 **end**
9 **end**
10 **for** *each* $h^n \in HS^{new}$ **do**
11 **for** *each* $h \in \{HS_i \cup HS^{mod} \cup HS^{new}\}$ **do**
12 **if** $d(h^n) \neq d(h)$ **then**
13 Compute $M_i(h^n, h)$ using Equation (12);
14 $M^{new} = M^{new} \cup \{M(h^n, h)\}$;
15 **end**
16 **end**
17 **end**
18 $M_{i+1} = M_i \cup M^{new}$;
19 Return M_{i+1}, M^{new}

3.3.1. Remark

There is an advantage of the granularity concept in our proposed algorithm over any object-based incremental learning approach. In existing object-based incremental approaches [2,37,36], for every new object arrival, new FDM entries must be created which is a time-consuming task. But in the proposed approach, for all the new training input patterns which have obtained absolute membership into any of the existing hyperboxes, no update of FDM is required reducing frequent alterations of FDM. This significantly diminishes the computational times and space requirements of the proposed approach.

3.4. Incremental manner to compute reduct

Algorithm 3 performs an update of current reduct R_i to become R_{i+1}. After updating FDM, as discussed in the above Section, the incremental process for updating current reduct R_i is performed by using two case strategies, as summarized below:

Case 1: $SAT_{M^{new}}(R_i) == SAT_{M^{new}}(C)$;
Case 2: $SAT_{M^{new}}(R_i) \neq SAT_{M^{new}}(C)$;

If the *Case* 1 holds, then the current reduct R_i satisfies the newly added or modified entries in M^{new}. Hence, the updation of current reduct is not required and existing R_{i+1} becomes reduct of M_{i+1}.

If *Case* 2 is satisfied then we initially assign R_i to R_{i+1}. The current reduct R_{i+1} doesn't satisfy some of the newly added entries in M^{new} leading to requirement for updation of R_{i+1} for becoming reduct. This is done in two phases: In the first phase, the additional attributes are added ($\forall c \in C - R_{i+1}$) into R_{i+1} using SFS strategy only on M^{new}. Here in each iteration, SAT measure is computed Equation (5) for ($R_{i+1} \cup \{c\}$) $\forall c \in C - R_{i+1}$. Then, the attribute having a maximum SAT measure is included in R_{i+1}. This strategy for attribute selection is repeated till $SAT_{M^{new}}(R_{i+1}) = SAT_{M^{new}}(C)$. The SFS strategy of reduct updation is only restricted to M^{new} as R_{i+1} already satisfies unmodified entries in M_i.

The modified R_{i+1} is a super reduct for $M_{i+1}(= M_i \cup M^{new})$ and can contain the redundant attributes. Hence, SBE strategy in [42] which is an efficient third order complexity approach, is followed on M_{i+1} to remove redundant attributes in R_{i+1} as given in Steps 11-17. Here for each attribute 'c' in R_{i+1}, it is checked whether omission of the attribute 'c' affects SAT measure. It is verified whether $SAT_{M_{i+1}}(R_{i+1} - \{c\})$ is one or not. If one, then the attribute 'c' is redundant and hence removed from R_{i+1}. Otherwise, the attribute 'c' is indispensable and retained in R_{i+1}.

Finally, the current R_{i+1} is the final reduct for sample U_{i+1}.

Algorithm 3: Incremental Way to Compute Reduct.

Input : R_i, M^{new}, M_{i+1}
Output: R_{i+1}
if $SAT_{M^{new}}(R_i) == SAT_{M^{new}}(C)$ **then**
  $R_{i+1} = R_i$;
else
  $R_{i+1} = R_i$;
  while $SAT_{M^{new}}(R_{i+1}) \neq SAT_{M^{new}}(C)$ **do**
    For each $c \in C - R_{i+1}$, Compute $SAT_{M^{new}}(R_{i+1} \cup \{c\})$;
    Select feature $c_o \in C - R_{i+1}$, satisfying
    $SAT_{M^{new}}(R_{i+1} \cup \{c_o\}) = max_{c \in C - R_{i+1}} SAT_{M^{new}}(R_{i+1} \cup \{c\})$;
    $R_{i+1} = R_{i+1} \cup \{c_o\}$;
  end
  Compute $SAT_{M_{i+1}}(C)$;
  for *each* $c \in C - R_{i+1}$ **do**
    Compute $SAT_{M_{i+1}}(R_{i+1} - \{c\})$;
    if $SAT_{M_{i+1}}(C) == SAT_{M_{i+1}}(R_{i+1} - \{c\})$ **then**
      $R_{i+1} = R_{i+1} - \{c\}$;
    end
  end
end
Return R_{i+1}

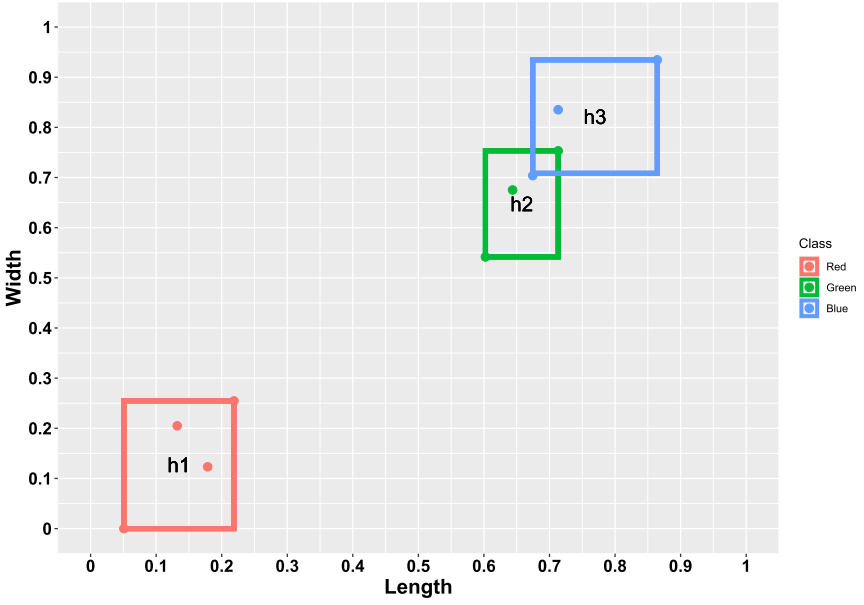

Fig. 3. Hyperboxes created through FMNN learning process for U_1. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.5. Illustration for IvFMFRS algorithm on toy dataset

We are illustrating the IvFMFRS algorithm with a toy dataset (25 objects, 2 conditional attributes (*length* and *width*), 3 decision classes (red, green and blue)). Fig. 3 and Fig. 4 are depicting the 2-D visualization of FMNN hyperboxes between *length* and *width* attributes for each sample arriving. *a* and *b* represent *length* and *width* attributes in FDM.

Here, we partition the given dataset randomly into two sample subsets U_1 (10 objects) and U_2 (15 objects). Initially, we apply the base algorithm FMNN-FRS, describe in Section 2.2.1, to sample U_1 to get base reduct R_1, set of hyperboxes HS_1 and FDM M_1.

In FMNN-FRS, FMNN first transforms data into IDS through hyperboxes for sample U_1, as shown in Table 2 and visual representation in Fig. 3. Hence, HS_1 is equal to $\{h_1, h_2, h_3\}$. Then based on IDS, a FDM M_1 is constructed corresponding to three hyperboxes (h_1, h_2 and h_3) of different classes, as shown in Table 3. After that, a reduct R_1 is computed based on M_1 using SFS strategy (SAT measure). The obtained approximate reduct R_1 for U_1 is {b,a}.

For a sample U_2, we apply IvFMFRS with given HS_1, M_1 and R_1 to result HS_2, R_2 and M_2. During incremental FMNN learning for the sample U_2, a new hyperbox h_4 is created and added in HS^{new} and also two hyperboxes $\{h_2, h_3\}$ belongs to

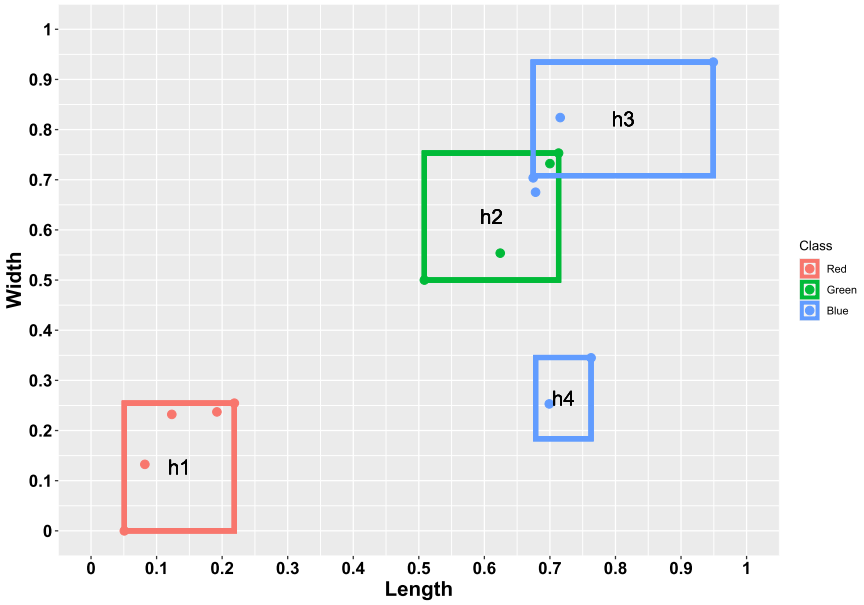

Fig. 4. Hyperboxes through FMNN learning process for U_2.

Table 2
Interval valued decision system based on U_1.

Hyperboxes	length (a)	width (b)	Class
h_1	[0.0508,0.2186]	[0.0000,0.2546]	Red
h_2	[0.6023,0.7132]	[0.5417,0.7532]	Green
h_3	[0.6745,0.8644]	[0.7083,0.9346]	Blue

Table 3
Fuzzy discernibility matrix M_1 based on IDS.

	h_1	h_2	h_3
h_1	$\emptyset$	$\{a_1, b_1\}$	$\{a_1, b_1\}$
h_2	$\{a_1, b_1\}$	$\emptyset$	$\{a_{0.78}, b_{0.93}\}$
h_3	$\{a_1, b_1\}$	$\{a_{0.85}, b_{0.88}\}$	$\emptyset$

Table 4
Updated Interval valued decision system for U_2.

Hyperboxes	length (a)	width (b)	Class
h_1	[0.0508,0.2186]	[0.0000,0.2546]	Red
h_2	[0.5085,0.7132]	[0.5000,0.7532]	Green
h_3	[0.6745,0.9492]	[0.7083,0.9346]	Blue
h_4	[0.6781,0.7628]	[0.1836,0.3451]	Blue

HS_1 are updated due to expansion for accommodating input patterns, describe in Algorithm 1. These modified hyperboxes are kept in HS^{mod} and removed from HS_1. Hence, we have $HS^{new} = \{h_4\}$, $HS^{mod} = \{h_2, h_3\}$, $HS_1 = \{h_1\}$

Now, HS_1 contains only unmodified hyperbox h_1. The resulting visualization of hyperboxes is depicted in Fig. 4 and modified IDS due to hyperboxes is shown in Table 4.

After updating IDS for U_2, the required modification of entries corresponding to HS^{mod} hyperboxes in M_1 are revised and also placed separately in M^{new}. Also, the FDM entries corresponding to newly introduced hyperbox h_4 are placed in M^{new}, describe in Algorithm 2. Furthermore,

$$M_2 = M_1 \cup M^{new}$$

$$HS_2 = HS_1 \cup H^{mod} \cup H^{new}$$

Table 5
Update Fuzzy discernibility matrix M_2 based on updated IDS.

	h_1	h_2	h_3	h_4
h_1	∅	$\{a_1, b_1\}$	$\{a_1, b_1\}$	$\{a_1, b_{0.79}\}$
h_2	$\{a_1, b_1\}$	∅	$\{a_{0.91}, b_{0.89}\}$	$\{a_{0.13}, b_1\}$
h_3	$\{a_1, b_1\}$	$\{a_{0.91}, b_{0.89}\}$	∅	∅
h_4	$\{a_1, b_{0.79}\}$	$\{a_{0.13}, b_1\}$	∅	∅

Using Algorithm 3, the entries in M^{new} are only used for reduct updation using sequential forward selection strategy, and the entries in M_2 are used in SBE strategy for filtering out the redundant attributes. The modified fuzzy discernibility matrix M_2 is shown in Table 5.

Based on the illustration mentioned above, Two cases are summarized for current reduct R_1 based on M^{new}.

Case 1: $SAT_{M^{new}}(R_1) == SAT_{M^{new}}(C)$;
Case 2: $SAT_{M^{new}}(R_1) \neq SAT_{M^{new}}(C)$;

Current reduct R_1 is satisfied the M^{new} (Case 1 satisfied). Hence, the recomputation of reduct is not required and existing R_1 becomes reduct of M_2. Hence, the obtained approximate reduct R_2 for U_2 is {b,a}.

3.6. Analysis of complexity

Algorithm 1, Algorithm 2 and Algorithm 3 compute a reduct starting from given HS_i, M_i and R_i. Algorithm 1 and Algorithm 2 initialize HS^{mod}, HS^{new} and M^{new} as an empty set. Algorithm 1 performs updation of IDS using FMNN, with time complexity of $O(|U_{i+1}||HS_{i+1}||C|)$. Algorithm 2 incrementally computes FDM, with time complexity of $O(|HS^{mod} \cup HS^{new}||HS_{i+1}||C|)$. In Algorithm 3, Steps 4-9 perform SFS computation for adding features into the current reduct, with time complexity of $O(|M^{new}||C - R_i|^2) == O(|HS^{mod} \cup HS^{new}|^2|C - R_i|^2)$. Algorithm 3, Steps 11-17 perform the strategies of deleting redundant features in current reduct, with third order time complexity of $O(|M_{i+1}||R_{i+1}|) == O(|HS_{i+1}|^2|R_{i+1}|)$. Hence, the time complexity is $O(|U_{i+1}||HS_{i+1}||C|) + O(|HS^{mod} \cup HS^{new}||HS_i||C|) + O(|M^{new}||C - R_i|^2) + O(|M_{i+1}||R_{i+1}|)$.

The space complexity of IvFMFRS is $O(|HS_{i+1}|^2|C|)$. For the entire dataset, IvFMFRS and FMNN-FRS algorithms are having the same space complexity.

The following **limitations** are observed in IvFMFRS approach: IvFMFRS algorithm only works on numeric decision systems, not hybrid decision systems. This is owing to the FMNN algorithm being only applicable to numerical attributes.

One important point is that IvFMFRS is an incremental adaptation of FMNN-FRS algorithm. So the output of IvFMFRS incrementally on U_1, U_2,... U_i sample subsequences and output of FMNN-FRS on $U_1 \cup U_2 \cup \cdots \cup U_i$ are not same.

In the next Section 4, the relevance of obtained approximate reduct by IvFMFRS with the aspects of computational time, reduct size and utility of reduct for classification are studied in depth through experimental and comparative study.

4. Experiments

This section evaluates the experimental performance of the proposed incremental algorithm IvFMFRS. The comparative analysis of proposed incremental algorithm is conducted with the recent (published in 2018-20) incremental FRS reduct approaches namely IV-FS-FRS [2], AIFWAR [3] and PIAR [38]. As the proposed IvFMFRS algorithm is an incremental adaptation of batch algorithm FMNN-FRS ([40]), the comparative analysis is also conducted with FMNN-FRS algorithm.

4.1. Environment and objectives of experimentation

Twelve benchmark datasets of different sizes have collected from the UCI machine learning repository [44] for experimental evaluation, as outlined in Table 6. The hardware environment of the system applied for experiments is CPU: Intel(R) i7-8500, Clock Speed: 3.40 GHz × 6, RAM: 32 GB DDR4, OS: Ubuntu 18.04 LTS 64 bit and Software: Matlab R2017a. The proposed algorithm is implemented in the Matlab environment. For IvFMFRS, the Lukasiewicz t-conorm ($S(x, y) = min\{x+y, 1\}$) for Equation (5) and fuzzy standard negation ($Neg(x) = 1 - x$) for Equation (12) are employed.

We have selected the value of the sensitive parameter *gamma* (γ) to 4 as recommended from the original FMNN paper [41]. Also, we have chosen the *theta* (θ) parameter to 0.3 based on experimental results obtained for base algorithm FMNN-FRS [40] for restricting hyperboxes size in FMNN learning model. Moreover, the compared algorithms (IV-FS-FRS, AIFWAR, PIAR and FMNN-FRS) follow their fuzzy model of t-norm, t-conorm and fuzzy similarity relations for computing as given in the respective publications and experiments are conducted in the same environment stated above.

4.2. The relevance of IvFMFRS algorithm in construction of classifiers

This section analyzes the performance of given algorithms through the significance of reduct in inducing a classifier through 10-fold cross-validation (10-FCV). CART, kNN(k =3) and RBF-SVM (SVM using Radial Basis function) classifiers, with

Table 6
Details of Numeric Benchmark Datasets.

Dataset	Attributes	Objects	Classes
Ionosphere	32	351	2
Spectfheart	44	267	2
Vehcile	18	846	4
Ozone Layer	72	1848	2
Segment	16	2310	2
Steel	27	1941	7
Page	10	5472	5
Thyroid	21	7200	3
Texture	41	5500	2
Gamma	10	19020	2
Satimage	36	6435	6
Shuttle	9	57999	7

having default optimal values. Also, we have also induced the classification model construction over unreduced datasets (named as 'Unred' in Tables 7, 8, 9) for verifying the relevance of reducts over 10-FCV. Moreover, a paired t-test with a significance level of 0.05 is conducted to evaluate the statistical significance of IvFMFRS results over other comparative algorithms and Unred.

Each column in Tables reports the results of the corresponding algorithm in the form of mean and standard deviation along with the p-value except IvFMFRS column. The $p - value$ is the significance level between the respective algorithm with the IvFMFRS algorithm. If the $p - value$ is greater than 0.05, there is no statistically significant difference, marked with the symbol 'o'. If the $p - value$ is less than 0.05, and the result obtained by the respective algorithm is less than IvFMFRS, then the particular algorithm is inferior to IvFMFRS and marked as a loss '-'. Otherwise, it is representing a win with '+'. Also, in each Table, the last two rows summarize total average results for given datasets and the count of loss '-', win '+' and no difference 'o' as competing with proposed IvFMFRS. The '*' sign in all tables shows the corresponding algorithm is intractable to a particular dataset to compute the reduct due to insufficient memory. And, '#' sign represents the scenario of non termination of the code even after several hours of computation.

Table 7, Table 8 and Table 9 illustrate the comparative results of 10-FCV for classification accuracies of CART, kNN and RBF-SVM classifiers, respectively. Table 10 and Table 11 present 10-FCV experiment results on computational time (in Seconds) and 10-FCV experiment results on reduct length obtained by algorithms.

For incremental algorithms, we randomly divided the training dataset into ten equal subsets for computing in each fold of 10-FCV. Each subset sequentially updates the incremental learning model for reduct computation. The last subset outcome of an algorithm is the final reduct for each fold. For FMNN-FRS, we provide an entire training dataset for computing in each fold of 10-FCV.

4.2.1. Analysis of results

Table 7 shows the classification results of CART classifier. Based on Table 7, it is clear that IvFMFRS returns significantly similar results to compared algorithms and Unred in most of the datasets. In Texture dataset, IvFMFRS performed statistically inferior to AIFWAR, although the difference in average classification accuracy for both algorithms is insignificant. In thyroid dataset, IvFMFRS performed statistically significant than AIFWAR, PIAR and Unred.

Similar conclusions can be obtained in the kNN and RBF-SVM classifiers from Table 8 and Table 8. In both classifiers, IvFMFRS achieved statistically significant than IV-FS-FRS in Vehicle, Segment and Steel datasets. In gamma dataset, PIAR performed statistically inferior to IvFMFRS in both classifiers. Also, PIAR could not be able to compute reduct in a reasonable amount of time in Satimage dataset, where IvFMFRS algorithm could compute reduct with comparative classification accuracies. Moreover, IvFMFRS showed statistically equivalent to other algorithms and Unred in most of the datasets. AIFWAR performed significantly better in Texture dataset than IvFMFRS, but the difference in mean value is very less.

Eventually, it can be seen that the idea of computing the approximate reduct by IvFMFRS is satisfactory and effective in terms of classification results in given classifiers. As we can see that the total average classification accuracy of the IvFMFRS algorithm for all datasets is quite similar to AIFWAR, PIAR, FMNN-FRS and Unred except IV-FS-FRS. It is further observed that, in Gamma, Texture, Satimage and Shuttle datasets, IV-FS-FRS could not obtain reduct due to memory overflow at given system configuration where IvFMFRS, AIFWAR obtained reduct in reasonable computational time. This is due to the aspect of representative instances in AIFWAR and fuzzy hyperboxes based granularization in IvFMFRS, achieving a significant reduction in space utilization.

In terms of computational times, as shown in Table 10, IvFMFRS incurred significantly less computational time than compared incremental algorithms (IV-FS-FRS, AIFWAR and PIAR) except base algorithm FMNN-FRS for all datasets. Also, the proposed method IvFMFRS obtained the lowest total average computational time (0.66 seconds) on all datasets, whereas as compared incremental algorithms total average computational time range showed from 8 to 252 seconds. Even, the resulting standard deviation of computation time presented very less variation, thus showing that methodology is reliable as

Table 7
Classification accuracies results (%) with CART classifier.

Datasets	IvFMFRS	IV-FS-FRS		AIFWAR		PIAR		FMNN-FRS		Unred	
	Mean ± Std	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val
Ionosphere	89.15 ± 3.27	88.32 ± 7.66	0.73^{o}	89.15 ± 3.76	1.00^{o}	88.30 ± 4.57	0.63^{o}	87.74 ± 4.68	0.44^{o}	87.46 ± 7.01	0.49^{o}
Spectfheart	75.65 ± 13.80	75.04 ± 7.91	0.90^{o}	76.03 ± 9.02	0.94^{o}	71.96 ± 9.41	0.49^{o}	74.27 ± 10.07	0.80^{o}	73.58. ± 4.03	0.65^{o}
Vehicle	69.53 ± 5.03	52.07 ± 8.75	0.00^{-}	71.98 ± 5.23	0.29^{o}	66.92 ± 5.59	0.28^{o}	69.13 ± 4.19	0.84^{o}	70.91 ± 5.63	0.57^{o}
Segment	95.02 ± 2.50	63.93 ± 12.83	0.00^{-}	96.45 ± 1.57	0.14^{o}	95.15 ± 1.55	0.89^{o}	95.49 ± 2.73	0.69^{o}	95.93 ± 1.88	0.36^{o}
Steel	91.08 ± 2.19	90.93 ± 2.02	0.87^{o}	91.54 ± 2.12	0.63^{o}	91.60 ± 3.39	0.68^{o}	92.16 ± 1.15	0.18^{o}	92.06 ± 2.05	0.31^{o}
Page	96.41 ± 0.80	94.95 ± 2.77	0.12^{o}	96.25 ± 0.61	0.62^{o}	96.63 ± 0.72	0.52^{o}	96.52 ± 0.53	0.72^{o}	96.41 ± 0.80	1.00^{o}
Thyroid	98.81 ± 0.77	99.61 ± 0.21	0.01^{+}	97.20 ± 2.08	0.03^{-}	99.52 ± 0.21	0.01^{+}	97.97 ± 0.65	0.01^{-}	99.61 ± 0.21	0.01^{+}
Ozone	95.29 ± 0.86	95.84 ± 1.21	0.25^{o}	95.23 ± 2.03	0.93^{o}	94.54 ± 1.76	0.24^{o}	94.64 ± 1.57	0.26^{o}	94.65 ± 1.50	0.25^{o}
Gamma	82.45 ± 0.57	*		81.97 ± 1.46	0.34^{o}	79.03 ± 1.58	0.00^{-}	82.45 ± 0.57	1.00^{o}	82.45 ± 0.57	1.00^{o}
Texture	91.50 ± 1.23	*		92.92 ± 1.43	0.02^{+}	90.47 ± 1.23	0.07^{o}	90.83 ± 1.28	0.24^{o}	92.40 ± 1.42	0.14^{o}
Satimage	85.12 ± 1.77	*		85.09 ± 1.80	0.97^{o}	#		85.36 ± 1.16	0.72^{o}	85.53 ± 1.76	0.60^{o}
Shuttle	99.97 ± 0.02	*		99.96 ± 0.02	0.25^{o}	99.94 ± 0.03	0.01^{-}	99.97 ± 0.02	1.00^{o}	99.97 ± 0.01	1.00^{o}
Average	89.12	82.58		89.48		88.55		88.87		89.24	
Lose/Win/Tie		2/1/5		1/1/10		2/1/8		1/0/11		0/1/11	

Notes: * represents non-executable due to memory overflow. # represents non-termination of program even after several hours.

Table 8
Classification accuracies results (%) with kNN (k=3) classifier.

Datasets	IvFMFRS	IV-FS-FRS		AIFWAR		PIAR		FMNN-FRS		Unred	
	Mean ± Std	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val
Ionosphere	90.30 ± 4.51	85.76 ± 8.26	0.14^{o}	87.47 ± 2.99	0.11^{o}	86.06 ± 5.00	0.06^{o}	87.48 ± 5.82	0.24^{o}	84.59 ± 6.65	0.04^{-}
Spectfheart	74.96 ± 7.09	73.58 ± 10.84	0.74^{o}	74.96 ± 9.81	1.00^{o}	75.04 ± 11.33	0.98^{o}	73.42 ± 8.39	0.66^{o}	73.20 ± 8.76	0.62^{o}
Vehicle	66.06 ± 8.45	54.09 ± 11.74	0.01^{-}	63.76 ± 7.53	0.52^{o}	62.57 ± 5.61	0.29^{o}	66.78 ± 5.61	0.82^{o}	69.88 ± 7.51	0.29^{o}
Segment	96.10 ± 1.69	62.90 ± 12.31	0.00^{-}	96.58 ± 1.14	0.46^{o}	96.49 ± 1.43	0.58^{o}	95.28 ± 2.21	0.36^{o}	96.88 ± 1.15	0.24^{o}
Steel	93.40 ± 1.61	89.43 ± 2.96	0.00^{-}	91.90 ± 1.77	0.06^{o}	93.25 ± 1.62	0.83^{o}	92.37 ± 1.82	0.19^{o}	92.68 ± 1.64	0.33^{o}
Page	96.14 ± 0.62	94.95 ± 2.59	0.17^{o}	95.88 ± 0.83	0.43^{o}	95.96 ± 1.14	0.66^{o}	96.01 ± 1.17	0.75^{o}	95.94 ± 0.78	0.53^{o}
Thyroid	94.79 ± 1.76	93.98 ± 0.59	0.18^{o}	94.69 ± 1.77	0.90^{o}	94.00 ± 0.70	0.20^{o}	94.70 ± 1.19	0.89^{o}	93.98 ± 1.26	0.25^{o}
Ozone	96.63 ± 1.24	96.11 ± 1.29	0.37^{o}	96.22 ± 1.15	0.45^{o}	96.11 ± 1.48	0.40^{o}	96.10 ± 1.19	0.34^{o}	96.27 ± 1.36	0.54^{o}
Gamma	83.15 ± 0.49	*		83.02 ± 1.57	0.80^{o}	79.28 ± 1.58	0.00^{-}	83.15 ± 0.49	1.00^{o}	83.15 ± 0.49	1.00^{o}
Texture	97.20 ± 1.41	*		98.78 ± 0.56	0.00^{+}	96.27 ± 0.97	0.10^{o}	97.18 ± 1.07	0.97^{o}	98.70 ± 0.30	0.00^{+}
Satimage	89.50 ± 1.70	*		90.55 ± 1.44	0.15^{o}	#		89.77 ± 0.98	0.66^{o}	90.06 ± 1.09	0.39^{o}
Shuttle	99.91 ± 0.02	*		99.91 ± 0.03	1.00^{o}	99.91 ± 0.02	1.00^{o}	99.91 ± 0.03	1.00^{o}	99.91 ± 0.02	1.00^{o}
Average	89.84	81.35		89.47		88.63		89.34		89.60	
Lose/Win/Tie		3/0/5		0/1/11		1/0/10		0/0/12		1/1/10	

Notes: * represents non-executable due to memory overflow. # represents non-termination of program even after several hours.

Table 9
Classification accuracies results (%) with RBF-SVM classifier.

Datasets	IvFMFRS	IV-FS-FRS		AIFWAR		PIAR		FMNN-FRS		Unred	
	Mean ± Std	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val
Ionosphere	91.75 ± 4.07	90.62 ± 4.73	0.57^{o}	90.33 ± 4.99	0.49^{o}	94.30 ± 3.53	0.15^{o}	91.73 ± 4.15	0.99^{o}	95.15 ± 3.30	0.06^{o}
Spectfheart	79.58 ± 10.91	79.18 ± 10.62	0.93^{o}	80.54 ± 10.21	0.81^{o}	79.27 ± 10.01	0.94^{o}	79.57 ± 10.76	0.99^{o}	79.96 ± 11.02	0.93^{o}
Vehicle	73.51 ± 3.47	42.69 ± 6.97	0.00^{-}	65.65 ± 13.24	0.08^{o}	67.84 ± 4.79	0.00^{-}	72.00 ± 5.48	0.47^{o}	75.62 ± 3.67	0.20^{o}
Segment	93.24 ± 1.63	43.33 ± 12.51	0.00^{-}	94.24 ± 1.14	0.12^{o}	93.72 ± 1.17	0.45^{o}	92.16 ± 3.68	0.40^{o}	94.06 ± 1.35	0.23^{o}
Steel	92.42 ± 2.13	91.80 ± 1.97	0.05^{-}	92.27 ± 2.07	0.87^{o}	93.56 ± 1.85	0.21^{o}	92.37 ± 1.82	0.95^{o}	93.56 ± 1.79	0.20^{o}
Page	93.91 ± 1.44	93.11 ± 2.46	0.38^{o}	94.02 ± 1.44	0.86^{o}	94.02 ± 1.45	0.85^{o}	93.87 ± 1.42	0.95^{o}	94.06 ± 1.45	0.81^{o}
Thyroid	93.09 ± 0.77	93.09 ± 0.79	1.00^{o}	92.86 ± 0.79	0.51^{o}	93.09 ± 0.79	1.00^{o}	93.08 ± 0.80	1.00^{o}	93.09 ± 0.79	1.00^{o}
Ozone	96.92 ± 1.66	96.91 ± 1.69	1.00^{o}	96.84 ± 1.64	0.90^{o}	96.76 ± 1.85	0.84^{o}	96.19 ± 1.91	0.38^{o}	96.98 ± 1.82	0.94^{o}
Gamma	85.57 ± 1.11	*		85.33 ± 1.04	0.62^{o}	80.01 ± 1.49	0.00^{-}	85.57 ± 1.11	1.00^{o}	85.57 ± 1.11	1.00^{o}
Texture	96.12 ± 1.84	*		98.50 ± 0.76	0.00^{+}	94.38 ± 1.37	0.02^{-}	95.80 ± 1.63	0.68^{o}	98.94 ± 0.40	0.00^{+}
Satimage	89.09 ± 1.29	*		89.89 ± 1.44	0.20^{o}	#		88.18 ± 1.29	0.13^{o}	90.16 ± 1.04	0.06^{o}
Shuttle	97.67 ± 0.39	*		98.17 ± 1.05	0.17^{o}	97.40 ± 0.12	0.05^{o}	97.68 ± 0.34	0.95^{o}	98.56 ± 0.12	0.00^{+}
Average	90.23	78.84		89.89		89.48		89.84		91.30	
Lose/Win/Tie		3/0/5		0/1/11		3/0/8		0/0/12		0/2/10	

Notes: * represents non-executable due to memory overflow. # represents non-termination of program even after several hours.

Table 10
10-FCV Experiment Results on Computational Time (in Seconds).

Datasets	IvFMFRS	IV-FS-FRS		AIFWAR		PIAR		FMNN-FRS	
	Mean ± Std	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val
Ionosphere	0.17 ± 0.03	0.21 ± 0.01	0.00^{-}	4.82 ± 0.15	0.00^{-}	1.23 ± 0.12	0.00^{-}	0.11 ± 0.01	0.00^{+}
Spectfheart	0.06 ± 0.03	0.26 ± 0.01	0.00^{-}	5.20 ± 0.52	0.00^{-}	0.68 ± 0.07	0.00^{-}	0.08 ± 0.00	0.05^{o}
Vehicle	0.22 ± 0.04	0.59 ± 0.01	0.00^{-}	4.41 ± 0.33	0.00^{-}	2.08 ± 0.22	0.00^{-}	0.10 ± 0.01	0.00^{+}
Segment	0.09 ± 0.00	5.61 ± 0.10	0.00^{-}	3.95 ± 0.30	0.00^{-}	13.97 ± 1.24	0.00^{-}	0.04 ± 0.01	0.00^{+}
Steel	0.76 ± 0.08	9.04 ± 0.79	0.00^{-}	4.79 ± 0.22	0.00^{-}	5.98 ± 0.40	0.00^{-}	0.65 ± 0.03	0.00^{+}
Page	0.07 ± 0.03	21.47 ± 1.05	0.00^{-}	4.01 ± 0.37	0.00^{-}	19.02 ± 1.70	0.00^{-}	0.07 ± 0.01	1.00^{o}
Thyroid	0.11 ± 0.02	62.10 ± 0.16	0.00^{-}	10.28 ± 0.78	0.00^{-}	37.13 ± 1.91	0.00^{-}	0.08 ± 0.01	0.00^{+}
Ozone	1.36 ± 0.09	21.50 ± 1.45	0.00^{-}	8.72 ± 0.79	0.00^{-}	14.70 ± 0.95	0.00^{-}	1.41 ± 0.07	0.18^{o}
Gamma	1.51 ± 0.06	*		10.85 ± 0.17	0.00^{-}	575.11 ± 75.01	0.00^{-}	1.82 ± 0.07	0.00^{-}
Texture	0.32 ± 0.05	*		11.58 ± 0.58	0.00^{-}	67.28 ± 7.18	0.00^{-}	0.11 ± 0.02	0.00^{+}
Satimage	3.07 ± 0.47	*		16.22 ± 0.34	0.00^{-}	#		1.12 ± 0.04	0.00^{+}
Shuttle	0.23 ± 0.03	*		19.66 ± 0.82	0.00^{-}	2032.2 ± 0.09	0.00^{-}	0.44 ± 0.02	0.00^{-}
Average	0.66	15.09		8.70		251.76		0.50	
Lose/Win/Tie		8/0/0		12/0/0		11/0/0		2/7/3	

Notes: * represents non-executable due to memory overflow. # represents non-termination of program even after several hours.

Table 11
10-FCV Experiment Results on Reduct Length.

Datasets	IvFMFRS	IV-FS-FRS		AIFWAR		PIAR		FMNN-FRS	
	Mean ± Std	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val	Mean ± Std	p-Val
Ionosphere	7.50 ± 0.56	9.30 ± 1.33	0.00^{-}	7.30 ± 1.88	0.75^{o}	16.90 ± 0.99	0.00^{-}	7.00 ± 0.47	0.04^{+}
Spectfheart	7.40 ± 0.69	15.90 ± 2.13	0.00^{-}	9.50 ± 4.85	0.19^{o}	19.80 ± 1.39	0.00^{-}	6.00 ± 0.66	0.00^{+}
Vehicle	12.80 ± 0.78	2.70 ± 0.82	0.00^{+}	9.00 ± 4.52	0.00^{+}	8.90 ± 0.73	0.00^{+}	13.06 ± 1.33	0.60^{o}
Segment	8.30 ± 1.63	4.60 ± 4.42	0.00^{+}	9.20 ± 1.22	0.17^{o}	9.30 ± 0.48	0.07^{o}	9.30 ± 1.49	0.16^{o}
Steel	12.70 ± 0.82	2.10 ± 0.47	0.00^{+}	8.00 ± 3.52	0.00^{+}	14.90 ± 0.31	0.00^{-}	12.80 ± 1.13	0.82^{o}
Page	7.80 ± 0.78	6.60 ± 2.91	0.22^{o}	9.60 ± 0.96	0.00^{-}	8.90 ± 0.31	0.00^{-}	7.30 ± 0.82	0.17^{o}
Thyroid	9.50 ± 1.64	20.00 ± 0.00	0.00^{-}	13.40 ± 4.47	0.01^{-}	18 ± 0.00	0.00^{-}	9.10 ± 0.73	0.49^{o}
Ozone	10.30 ± 1.05	22.60 ± 1.95	0.00^{-}	11.50 ± 8.50	0.66^{o}	28.50 ± 0.97	0.00^{-}	8.70 ± 0.67	0.00^{+}
Gamma	10.00 ± 0.00	*		6.60 ± 0.51	0.00^{+}	6.20 ± 0.42	0.00^{+}	10.00 ± 0.00	1.00^{o}
Texture	10.30 ± 1.33	*		17.10 ± 2.53	0.00^{-}	6.90 ± 0.31	0.00^{+}	9.20 ± 1.03	0.05^{+}
Satimage	15.40 ± 1.50	*		29.50 ± 4.71	0.00^{-}	#		15.70 ± 2.21	0.72^{o}
Shuttle	6.50 ± 0.52	*		5.60 ± 1.64	0.11^{o}	6.00 ± 0.00	0.00^{-}	5.90 ± 0.52	0.01^{+}
Average	9.87	10.47		11.35		13.11		9.50	
Lose/Win/Tie		4/3/1		4/3/5		7/3/1		0/5/7	

Notes: * represents non-executable due to memory overflow. # represents non-termination of program even after several hours.

compared to others. FMNN-FRS performed significantly better in computational time than IvFMFRS in most of the datasets. These substantial reductions of computational time of IvFMFRS and FMNN-FRS are due to the dealing with hyperboxes constructed by the FMNN model where $|HS| << |U|$. Thus, the speed-up computation and performance demonstrate the potential of the IvFMFRS algorithm and its suitability for larger datasets.

As comparing our proposed algorithm with FMNN-FRS algorithm in Table 10, IvFMFRS has incurred slightly more computational time than FMNN-FRS in most of the datasets due to incremental adaptation. For each of new subset, IvFMFRS needs to update existing hyperboxes and their corresponding entries in FDM where FMNN-FRS constructs FDM on entire data once. Hence, FMNN-FRS will perform slightly better in term of computational time than IvFMFRS.

From the results on reduct length shown in Table 11, IvFMFRS performed statistically extremely significant, which means computed relevant attributes with smaller reduct size than IV-FS-FRS, AIFWAR and PIAR algorithms in most of the datasets. Even, IvFMFRS obtained a statistically similar reduct as compared to FMNN-FRS. IvFRMS performed statistically inferior in terms of reduct size from IV-FS-FRS in some datasets. But, the quality of reduct from IV-FS-FRS algorithms in term of average classification accuracies are statistically inferior to IvFMFRS.

In summary, the relevance of IvFMFRS is significantly validated as it computes incremental reduct with lesser length and incurring less computational time while preserving similar or better classification accuracies than compared incremental approaches.

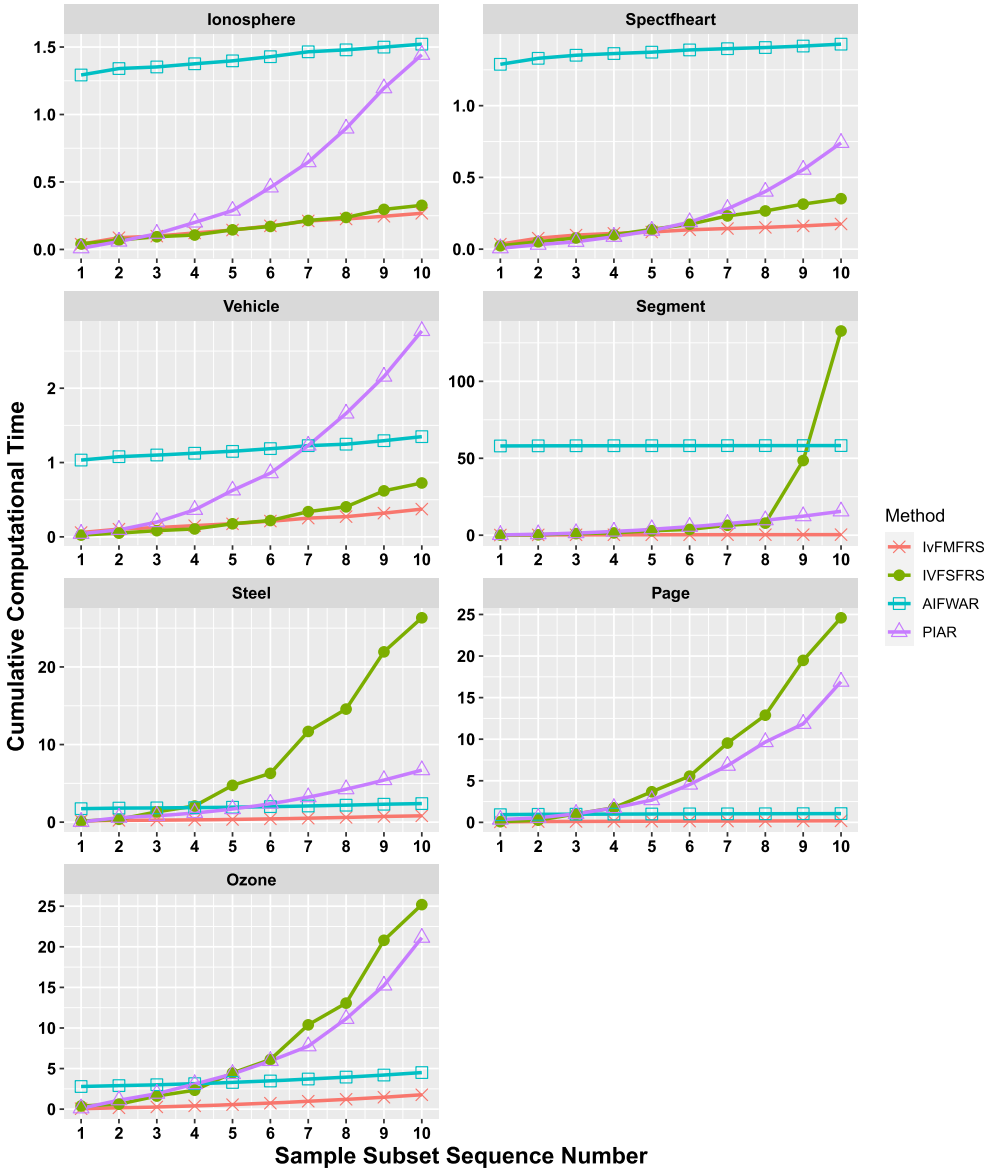

Fig. 5. Cumulative Computational results in Incremental Reduct Computation Stages.

4.3. Comparative analysis of incremental reduct algorithms

This section investigates the comparative analysis of the incremental algorithms (IvFMFRS, IV-FS-FRS, AIFWAR and PIAR) in aspects of reduct length and computational time in the incremental step of reduct computation. We are presenting two figures (Fig. 5 and Fig. 6) depicting the detailed change of the computational time and reduct size of IvFMFRS, IV-FS-FRS, AIFWAR and PIAR with subset continuously entering. Each dataset is randomly partitioned into the ten equal subsets for an experiment. In each iteration, we incrementally update an algorithm with a subset to learn and find the corresponding approximate reduct. Here, we are depicting the cumulative computational time till that iteration and reduct size at that iteration in Fig. 5 and Fig. 6 respectively.

In both figures, the x-axis is the sequence size of the data. And, the y-axis is the computational time (in seconds) in Fig. 5 and reduct length in Fig. 6. The dashed line shows the results of IvFMFRS; the dotted line shows IVFSFRS; the solid line shows AIFWAR in figures. This experiment is conducted on only eight datasets because, in the rest of the datasets, IV-FS-FRS could not compute the reduct on the given system. Both figures illustrate the efficiency of incremental algorithms on arriving subset sequences one by one.

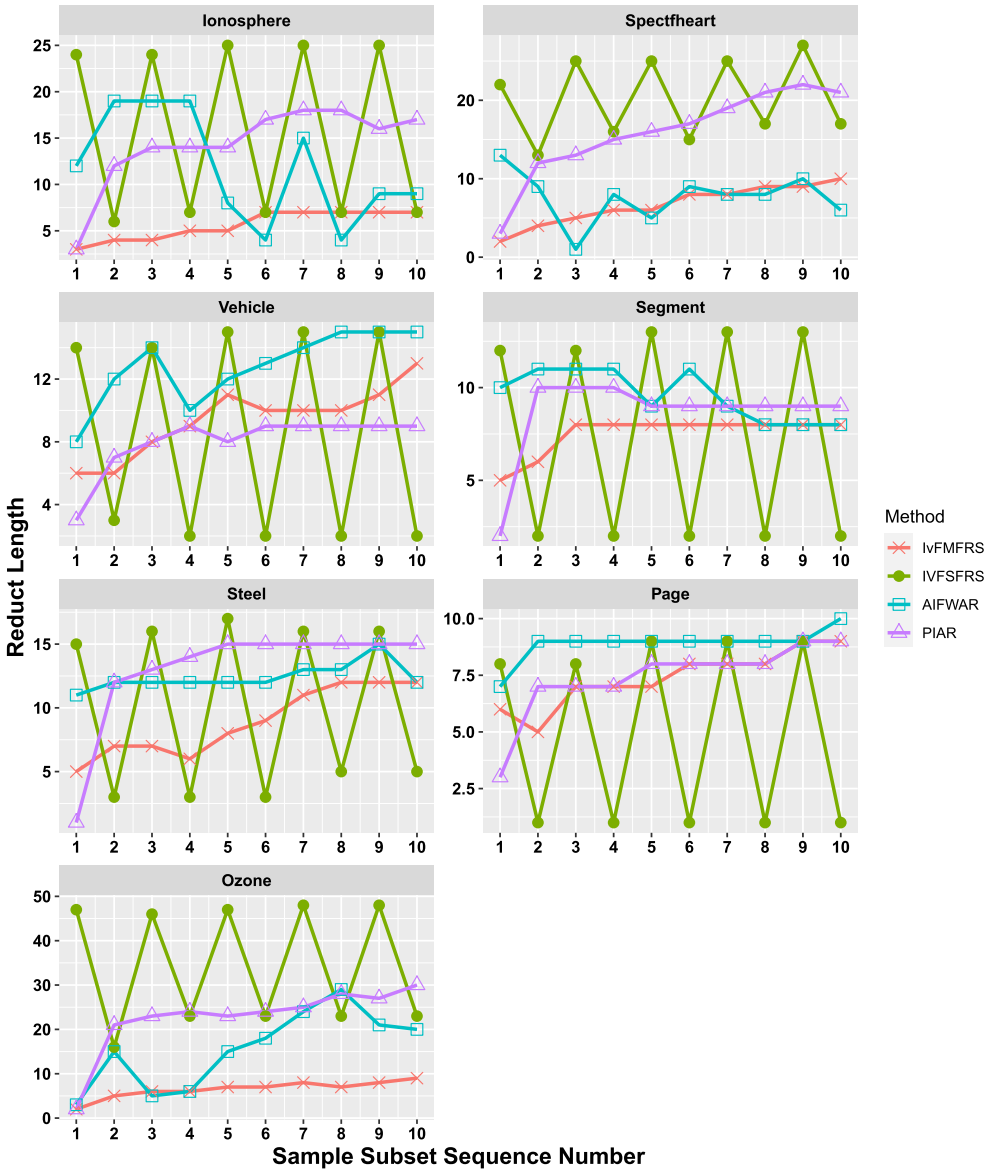

Fig. 6. Reduct length results in Incremental Reduct Computation Stages.

4.3.1. Analysis of results

In Fig. 5, the computational time is starting with base reduct computation from the first base part, and the rest of the timestamps are the time that is incurred for updating the reduct when a next sample subset has arrived. It can be seen from Fig. 5 that in most of the given datasets, the computational time for each subsequent sample as the number of samples increases result in a significant increase in computational time for both IV-FS-FRS and PIAR algorithms. However, in IvFMFRS and AIFWAR, the computational time is showing almost like a flat line for most datasets, indicating a roughly negligible amount of time is incurred when a subsequent sample is added after the base reduct computation on U_1.

In IvFMFRS, the changes that have happened to the FDM and computational effort are actually very much minimal when it comes to our proposed algorithm. The size of FDM signifies the computational time that is involved when a new subset is added. This process further is attributed to the utilizing of FMNN as a preprocessor in FDM construction. FMNN is absorbing many new objects accommodated into the existing hyperboxes result in no changes in FDM and the changes that are happening almost equivalent when the subsequent subset is added. Hence, the computation effort is seeming roughly equal in each step. Usually, changes in FMNN tend to update in FDM construction. So in our case, the FDM size is not significantly growing from one sample to subsequent sample arriving. However, in respective algorithms, FDM sizes are significantly changing, increasing in computation time as the sample subset is growing.

From Fig. 6, it can be seen that both AIFWAR and IV-FS-FRS reduct exhibit a significant fluctuation in reduct size when a new sample is added. However, in IvFMFRS, the change in reduct size is very gradual, and it is going for a smaller reduct length to a little bigger length as sample subset arriving. This gradual increase in reduct size is perhaps due to the following reason. In both IvFMFRS and IV-FS-FRS algorithms, whenever a sample is entering, the SFS algorithm adds new attributes in the existing reduct followed by the SBE algorithm to remove redundant attributes. In the AIFWAR algorithm, the attributes that are included in the existing reduct through the SFS process are followed by the wrapper technique for searching for the best attribute subset in reduct.

So, in the case of SBE inclusion or wrapper technique, removing most of the earlier present reduct attributes exhibits a lot of variance in reduct size, which can be seen in IV-FS-FRS and AIFWAR algorithms in Fig. 6. But in our case, the change is not much significant and not much variation in the reduct size observed. The attributes that are added in our approach are significant even after new attributes are included in the SFS process, which is getting retained in SBE process. As in our approach, attributes are selected based on discernibility over hyperboxes, which represent a set of objects of the decision system, leading to the selection of highly significant attributes. This aspect of selecting significant attributes as part of the SFS process due to FMNN preprocessing is aiding in making very less oscillation in reduct length, as seen in Fig. 6.

5. Future work

Experiments have demonstrated that the proposed IvFMFRS algorithm achieved better scalability than some of the other incremental approaches owing to granularization resulting in significant reduction in space utilization. Still this approach is not scalable to very large scale datasets such as Susy, Hepmass, Higgs and Cifar datasets [45] as memory becomes insufficient even after hyperbox space transformation. This requires Distributed/parallel solutions, which will be investigated in future.

6. Conclusion

The proposed IvFMFRS is an incremental adaptation of FMNN-FRS [40] for incremental reduct computation using Fuzzy Rough sets. The incremental updation of reduct with the onset of new training data involves three phases: updation of hyperboxes through FMNN to include new training patterns, updation of FDM based on updated hyperboxes and update current reduct using SFS strategy followed by SBE strategy. FMNN preprocessing results in relatively fewer changes to the discernibility matrix than object-based, resulting in IvFMFRS being efficient from the aspects of both computational time and space utilization simultaneously. The detailed comparative experimental study is conducted with state of the art incremental FRS approaches and established the relevance of IvFMFRS in obtaining reduct with increased scalability and comparable or improved generalizability of the classifier models induced. It is also observed that the changes to the reduct in incremental learning of IvFMFRM are gradual in nature with better stability. IvFMFRS can scale to much larger datasets than the compared approaches. In future, we will investigate distributed/parallel algorithms for IvFMFRS for achieving scalability to very large scale decision systems.

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.

References

[1] P. Ni, S. Zhao, X. Wang, H. Chen, C. Li, Para: a positive-region based attribute reduction accelerator, Inf. Sci. 503 (2019) 533–550, https://doi.org/10.1016/j.ins.2019.07.038.
[2] Y. Yang, D. Chen, H. Wang, X. Wang, Incremental perspective for feature selection based on fuzzy rough sets, IEEE Trans. Fuzzy Syst. 26 (3) (2018) 1257–1273, https://doi.org/10.1109/TFUZZ.2017.2718492.
[3] X. Zhang, C. Mei, D. Chen, Y. Yang, J. Li, Active incremental feature selection using a fuzzy-rough-set-based information entropy, IEEE Trans. Fuzzy Syst. 28 (5) (2020) 901–915.
[4] A. Gepperth, B. Hammer, Incremental learning algorithms and applications, in: European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2016, pp. 1–12.
[5] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[6] D. Dubois, H. Prade, Putting Rough Sets and Fuzzy Sets Together, vol. 11, Springer, Netherlands, Dordrecht, 1992, pp. 203–232.
[7] A.M. Radzikowska, E.E. Kerre, A comparative study of fuzzy rough sets, Fuzzy Sets Syst. 126 (2) (2002) 137–155, https://doi.org/10.1016/S0165-0114(01)00032-X.
[8] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognit. Lett. 24 (6) (2003) 833–849, https://doi.org/10.1016/S0167-8655(02)00196-4.
[9] R. Jensen, N.M. Parthaláin, Towards scalable fuzzy–rough feature selection, Inf. Sci. 323 (2015) 1–15, https://doi.org/10.1016/j.ins.2015.06.025.
[10] R. Jensen, A. Tuson, Q. Shen, Finding rough and fuzzy-rough set reducts with sat, Inf. Sci. 255 (2014) 100–120, https://doi.org/10.1016/j.ins.2013.07.033.
[11] R. Jensen, Q. Shen, Fuzzy–rough attribute reduction with application to web categorization, Fuzzy Sets Syst. 141 (3) (2004) 469–485.
[12] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowinski (Ed.), Intelligent Decision Support, in: Theory and Decision Library, vol. 11, Springer, 1992, pp. 331–362.
[13] R.B. Bhatt, M. Gopal, On fuzzy-rough sets approach to feature selection, Pattern Recognit. Lett. 26 (7) (2005) 965–975.
[14] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 824–838, https://doi.org/10.1109/TFUZZ.2008.924209.

[15] A. Ohrn, Discernibility and Rough Sets in Medicine: Tools and Applications, Department of Computer and Information Science, 2000, pp. 1–239.
[16] X. Zhang, C. Mei, D. Chen, Y. Yang, A fuzzy rough set-based feature selection method using representative instances, Knowl.-Based Syst. 151 (2018) 216–229.
[17] C. Wang, Y. Huang, M. Shao, Q. Hu, D. Chen, Feature selection based on neighborhood self-information, IEEE Trans. Cybern. 50 (9) (2020) 4031–4042, https://doi.org/10.1109/TCYB.2019.2923430.
[18] P. Maji, P. Garai, Fuzzy–rough simultaneous attribute selection and feature extraction algorithm, IEEE Trans. Cybern. 43 (4) (2013) 1166–1177, https://doi.org/10.1109/TSMCB.2012.2225832.
[19] C. Wang, Y. Huang, W. Ding, Z. Cao, Attribute reduction with fuzzy rough self-information measures, Inf. Sci. 549 (2021) 68–86, https://doi.org/10.1016/j.ins.2020.11.021.
[20] C. Wang, Y. Huang, M. Shao, X. Fan, Fuzzy rough set-based attribute reduction using distance measures, Knowl.-Based Syst. 164 (2019) 205–212, https://doi.org/10.1016/j.knosys.2018.10.038.
[21] C. Wang, Y. Wang, M. Shao, Y. Qian, D. Chen, Fuzzy rough attribute reduction for categorical data, IEEE Trans. Fuzzy Syst. 28 (5) (2020) 818–830, https://doi.org/10.1109/TFUZZ.2019.2949765.
[22] F. Wang, J. Liang, Y. Qian, Attribute reduction: a dimension incremental strategy, Knowl.-Based Syst. 39 (2013) 95–108.
[23] J. Zhang, T. Li, D. Ruan, D. Liu, Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, Int. J. Approx. Reason. 53 (4) (2012) 620–635, https://doi.org/10.1016/j.ijar.2012.01.001.
[24] W. Qian, W. Shu, C. Zhang, Feature selection from the perspective of knowledge granulation in dynamic set-valued information system, J. Inf. Sci. Eng. 32 (2016) 783–798.
[25] Y. Jing, T. Li, J. Huang, Y. Zhang, An incremental attribute reduction approach based on knowledge granularity under the attribute generalization, Int. J. Approx. Reason. 76 (2016) 80–95.
[26] W. Shu, H. Shen, Updating attribute reduction in incomplete decision systems with the variation of attribute set, Int. J. Approx. Reason. 55 (3) (2014) 867–884, https://doi.org/10.1016/j.ijar.2013.09.015.
[27] F. Wang, J. Liang, C. Dang, Attribute reduction for dynamic data sets, Appl. Soft Comput. 13 (1) (2013) 676–689, https://doi.org/10.1016/j.asoc.2012.07.018.
[28] J. Liang, F. Wang, C. Dang, Y. Qian, A group incremental approach to feature selection applying rough set technique, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 294–308, https://doi.org/10.1109/TKDE.2012.146.
[29] W. Shu, W. Qian, Y. Xie, Incremental feature selection for dynamic hybrid data using neighborhood rough set, Knowl.-Based Syst. 194 (2020) 105516, https://doi.org/10.1016/j.knosys.2020.105516.
[30] W. Wei, X. Wu, J. Liang, J. Cui, Y. Sun, Discernibility matrix based incremental attribute reduction for dynamic data, Knowl.-Based Syst. 140 (2018) 142–157, https://doi.org/10.1016/j.knosys.2017.10.033.
[31] Y. Liu, L. Zheng, Y. Xiu, H. Yin, S. Zhao, X. Wang, H. Chen, C. Li, Discernibility matrix based incremental feature selection on fused decision tables, Int. J. Approx. Reason. 118 (2020) 1–26.
[32] Y. Jing, T. Li, C. Luo, S.-J. Horng, G. Wang, Z. Yu, An incremental approach for attribute reduction based on knowledge granularity, Knowl.-Based Syst. 104 (2016) 24–38.
[33] W. Shu, H. Shen, Incremental feature selection based on rough set in dynamic incomplete data, Pattern Recognit. 47 (12) (2014) 3890–3906, https://doi.org/10.1016/j.patcog.2014.06.002.
[34] N. Mac Parthalain, R. Jensen, R. Diao, Fuzzy-rough set bireducts for data reduction, IEEE Trans. Fuzzy Syst. (2019) 1.
[35] Y. Yang, D. Chen, H. Wang, Active sample selection based incremental algorithm for attribute reduction with rough sets, IEEE Trans. Fuzzy Syst. 25 (4) (2017) 825–838, https://doi.org/10.1109/TFUZZ.2016.2581186.
[36] Y. Yang, D. Chen, H. Wang, E.C. Tsang, D. Zhang, Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving, in: Theme: Fuzzy Rough Sets, Fuzzy Sets Syst. 312 (2017) 66–86, https://doi.org/10.1016/j.fss.2016.08.001.
[37] A. Zeng, T. Li, D. Liu, J. Zhang, H. Chen, A fuzzy rough set approach for incremental feature selection on hybrid information systems, in: Special Issue: Uncertainty in Learning from Big Data, Fuzzy Sets Syst. 258 (2015) 39–60, https://doi.org/10.1016/j.fss.2014.08.014.
[38] P. Ni, S. Zhao, X. Wang, H. Chen, C. Li, E.C. Tsang, Incremental feature selection based on fuzzy rough sets, Inf. Sci. 536 (2020) 185–204.
[39] D. Slezak, Approximate reducts in decision tables, in: Proc. Sixth Int'l Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems, 1996, pp. 1159–1164.
[40] A. Kumar, P.S.V.S.S. Prasad, Scalable fuzzy rough set reduct computation using fuzzy min–max neural network preprocessing, IEEE Trans. Fuzzy Syst. 28 (5) (2020) 953–964.
[41] P.K. Simpson, Fuzzy min-max neural networks. I. Classification, IEEE Trans. Neural Netw. 3 (5) (1992) 776–786, https://doi.org/10.1109/72.159066.
[42] S. Ghosh, P.S.V.S.S. Prasad, C.R. Rao, Third order backward elimination approach for fuzzy-rough set based feature selection, in: PReMI, vol. 10597, 2017, pp. 254–262.
[43] P. Jaccard, Nouvelles recherches sur la distribution florale, Bull. Soc. Vaud. Sci. Nat. 44 (1908) 223–270, https://doi.org/10.5169/seals-268384.
[44] D. Dua, C. Graff, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017.
[45] O.U. Lenz, D. Peralta, C. Cornelis, A scalable approach to fuzzy rough nearest neighbour classification with ordered weighted averaging operators, in: T. Mihálydeák, F. Min, G. Wang, M. Banerjee, I. Düntsch, Z. Suraj, D. Ciucci (Eds.), Rough Sets, Springer International Publishing, Cham, 2019, pp. 197–209.