Applied Intelligence (2022) 52:5561–5583
https://doi.org/10.1007/s10489-021-02741-4

# Missing information in imbalanced data stream: fuzzy adaptive imputation approach

Bohnishikha Halder[1] · Md Manjur Ahmed[1] · Toshiyuki Amagasa[2] · Nor Ashidi Mat Isa[3] · Rahat Hossain Faisal[1] · Md. Mostafijur Rahman[4]

Accepted: 4 August 2021 / Published online: 16 August 2021


## Abstract

From a real-world perspective, missing information is an ordinary scenario in data stream. Generally, missing data generate diverse problems in recognizing the pattern of data (i.e., clustering and classification). Particularly, missing data in data stream is a challenging topic. With imbalanced data, the problem of missing data greatly affects pattern recognition. As a solution to all these issues, this study puts forward an adaptive technique with fuzzy-based information decomposition method, which simultaneously solves the problem of incomplete data and overcomes the imbalanced data stream in a dataset. The main purpose of the proposed fuzzy adaptive imputation approach (FAIA) is to represent the effect of missing values in imbalance data stream and handle the missing data problem in imbalance data stream. FAIA is a single pass method. It considers adaptive selection of intervals based on all observed instances by using the interrelationship of attributes to identify correct interval for computing missing instances. Here, the interrelationship of two attributes means one attribute's value of an instance depends on another attribute's value of the same instance. In FAIA, after measuring all interval distances from a certain missing value, the least distance is selected for this missing value. Synthetic data of minority class are generated using the same process of missing value imputation for balancing data that is called oversampling. Instances of the datasets are divided into the chunks in data stream to balance data without any ensemble of previous chunks because missing values may misguide the future chunks. To demonstrate the performance of FAIA, the experiment is divided into three parts: missing data imputation, imbalanced information for offline data for data stream, and imbalanced information with missing value for offline data. Eleven numerical datasets with different dimensions from various repositories are considered for the computing performance of missing data imputation and imbalanced data without data stream. Four different datasets are also used to measure the performance of imbalanced data stream. In maximum measuring cases, the proposed method outperforms.

**Keywords** Data imputation · Missing information · Fuzzy adaptive approach · Pattern recognition · Imbalanced data · Data stream

## 1 Introduction

Missing data essentially refer to the absence of certain information from a dataset. When a dataset has missing data, several problems must be addressed to obtain the desired output. In real-life scenarios, missing information is normal. Missing information may result in outside or inside disturbances of systems and communication structures, such as sensor fault and malicious attack [1]. Medical diagnosis, biological research with DNA microarrays, and industrial fields are accustomed to incomplete data [2]. Imbalanced data represent another common problem in a dataset. When one class has outnumbered instances than another, then data imbalance exists [3]. Data stream pertains to the continuous and sequential reading of data items for pattern recognition [4]. Examples of data stream are wireless sensor network, web click stream, and scientific data.

✉ Md Manjur Ahmed
manjur_39@yahoo.com

✉ Nor Ashidi Mat Isa
ashidi@usm.my

[1] Department of Computer Science and Engineering, University of Barishal, Barishal 8200, Bangladesh

[2] Center for Computational Sciences, University of Tsukuba, Tsukuba, Japan

[3] School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

[4] Department of Software Engineering, Daffodil International University, Dhaka, Bangladesh

In recent years, concept drift with imbalanced data stream is one of the most challenging issues in data mining [5]. Recognizing a pattern is more challenging with missing data.

Data stream is a dynamic process in which data are collected over time, and when these data change over time, concept drift occurs [4]. Data distribution, class prior, class-conditional probability, and posterior probability are four components that cause concept drift [5]. These four types of drifts may be present in data stream simultaneously. To solve this problem, several ensemble methods are considered, and all of them are easy and high performing [6]. Another two problems in a data stream are incomplete data and imbalanced data.

Incomplete data generate a biased effect on the quality of classification in a dataset [7] (i.e., imbalanced data). Addressing this problem is a means to enhance the performance of pattern recognition. Deleting missing data instances (i.e., complete case analysis) or imputing missing data on the basis of observed instances (i.e., recovery of missing values) can overcome the problem [8]. Recovering missing data shows better results than deleting them. The three ways to estimate missing data are interpolation, imputation, and matrix completion [9, 10]. Many machine learning imputation techniques are available, and each provides better results than the other. Among these techniques, k-nearest neighbors (KNN) is an easy and lazy instance-based method [11]. In KNN, k number of closer distance instances are considered to measure missing values [1]. In some cases where rate of missing data is high, measuring missing data is impossible with KNN method. By contrast, some methods weight these distances on the basis of mutual information (MI) or gray relational coefficient (GRA) [1, 12].

Imbalanced data also occur commonly in real world. The high number of instances in one class is called majority class, and the least number of instances in another class is called minority class. In training phase, the overall performance is very poor for a small number of instances in minority class in datasets [3]. To improve the performance, several imbalance methods are available to alleviate the aforementioned problem in training phase. A basic solution s random oversampling (ROS), wherein randomly instances are selected from minority class that may face overfitting problem [13]. Synthetic minority oversampling technique (SMOTE) generates synthetic instances of minority class using "feature space" [5]. Specifically, SMOTE faces borderline issues to create new instances. Therefore, borderline-SMOTE selects instances that are very close to majority class [6].

Many cost-sensitive methods consider cost during learning, but all of them overlook missing information. Fuzzy-based information decomposition (FID) [3] solves both problems simultaneously. FID [3] considers two issues in which data imbalance is reduced by oversampling of minority class and missing data of minority class is imputed to obtain proper outcome. The outcome of this existing method is better than those of other methods. In addition, FID [3] shows the importance of imputing missing values. For estimating the missing value, FID considers all instances in the same attribute as a distinct column vector. However, for most datasets, FID does not consider the interrelationship among attributes of each instance. Such an interrelationship is basically the relationship among attributes for classifying a specific instance. For example, the Iris dataset [14] has three classes of data (setosa, versicolor, and virginica) and four attributes (sepal length, sepal width, petal length, and petal width). For *I. setosa*, the attribute petal width value is between 0.1 cm and 0.6 cm, and the petal length value is between 1 cm and 1.9 cm. For *I. versicolor*, the attribute petal width value is between 1 cm and 1.8 cm, and the petal length value is between 3 cm and 5.1 cm (though most have a petal length ranging between 3 cm and 4.9 cm). Similarly, for *I. virginica*, the petal width value is between 1.8 cm and 2.5 cm (except one), and the petal length value is between 4.8 cm and 6.9 cm (except one). A relationship exists between petal width and length, that is, if the petal width is between 0.1 cm and 0.6 cm and the length is from 3 cm to 5.1 cm, then the flower is *I. setosa*. The problem is that FID [3] only uses single attributes values for imputing missing instances, so the imputed missing data sometimes create confusion. Moreover, at the time of interval selection, the sequential manner may divert to choose correct intervals.

For data stream, the solution of imbalanced data with concept drift is trickier. The two types of solutions are online based and chunk based. Under the online-based solution, the prediction method is updated for each sample. On the contrary, a number of samples are combined in a certain time and based on that, the classification process that takes place is known as a chunk-based approach [5]. Online-based approach is regarded difficult because it processes an instance at a time [15], and it may face false alarms, delays on the detectors, or even missed detections situations [5]. Chunk-based leaning may be inappropriate when memory is limited and high speed is a main requirement [15]. Moreover, both approaches ignore missing values in data stream. Therefore, we propose a chunk-based missing data imputation method for imbalanced data stream. The limited memory problem for chunk-based learning is solved by avoiding previous data chunks details. The proposed fuzzy adaptive imputation approach (FAIA) for missing data uses adaptive interval selection to weight instances for imputing missing data and balancing data. Unlike FID method [3] for missing data imputation, FAIA also imputes missing data for majority class. FAIA considers adaptive selection of intervals based on the interrelationship of attributes to identify the correct interval for weighting the observed instances for imputing missing value. For example, from the same Iris dataset [14], which has a missing instance with four attributes, and one is missing among these four attributes. FAIA computes distances of all attributes of observed instances from the missing instance's

attributes besides using the FID method. The measured distances aid to find the closest observed instances. After measuring all intervals of distances, the least distance is selected for a certain missing value, making the FAIA adaptive. This adaptive selection minimizes the sequential increasing issue for imputing missing value. To solve the imbalance problem, given some missing values in the data stream, the previous chunks histories are not that much important for balancing the recent chunk of data that can misguide the current data. As such, FAIA is a single pass method.

## 2 Related works

Classifying data from incomplete and imbalanced data streams is challenging because missing data must be imputed and imbalanced data with concept drift should be balanced. Researchers already found several ways to attenuate the issues of missing data and nonstationary imbalanced data streams separately. For missing data, the methods are often divided into "complete case analysis" or "recovery of missing values" [3]. In complete case analysis, only observed instances are considered. With an outsized missing rate, crucial information is overlooked which cannot provide a satisfying output. A correct missing data imputation method may be a better decision than case deletion. These imputation methods may be in forms of single imputation, multiple imputations, fractional imputation, or iterative imputation method [1]. Recovering missing values by means or mode of observed values may be a basic technique for single imputation. Hot deck (HD) and cold deck (CD) are almost similar single imputation methods, except HD uses input vector with similar pattern, and CD's source dataset differs from the current dataset. For example, in a survey context, the external source can be a previous realization of the same survey [8, 12]. Consistent with the article [11], single imputation weakens the uncertainty of the imputed data. By contrast, multiple imputation restores the natural validities and solves the uncertainty, but the data need to keep knowledge about missing mechanism. In comparison, iterative imputation methods can use all useful information, including missing information [1].

Statistical imputation methods and soft computing with machine learning technologies are other two representative imputation methods. Expectation maximization (EM), regression imputation (RI), MI-algorithm, and fuzzy rough set are examples of statistical imputation method. EM is an iterative algorithm that can be used in machine learning and data mining [8]. In RI, multiple regression imputation method is used with the parametric and non-parametric methods to impute missing data [1]. For the parametric method, if incomplete data cannot be modeled parametrically, then performance will be very low [1].

KNN [1], decision tree (DT) [1], self-organizing map (SOM) [16], and SVM [17] imputation can be categorized under the machine learning method. SOM is trained without missing data then imputes missing values [16]. DT can be applied to impute numerical and categorical data. KNN measures k nearest neighbors from missing samples using distance matric [1]. Some developed KNN imputation methods, such as sequential (SKNN) [1] and iterative KNN imputation (IKNNI) [18], are introduced to get more valid results for missing data. To improve classification with estimated missing values, [12] considered feature-weighted distance matric using MI. Authors [1] considered feature relevance and MI-weighted GRA matric to retrieve missing values. GRA presents a relationship between a referential observation and compared observation using gray relational coefficient (GRC) and gray relational grade (GRG) [1]. On the contrary, author [11] considered the gray distance to compute the closest neighbors of missing data. All of these methods try to find out more accurate missing data from datasets. Some of them perform better than other methods, but all of them cannot deal with imbalanced data in datasets.

Imbalance problem is another critical issue in the data mining field where data of several class labels are not balanced. To solve this problem, several methods are introduced by researchers. All of these methods can be categorized into three levels, namely, data level, algorithm level, and hybrid approach [9, 19]. Example of data-level solutions are ROS of minority class or randomly under-sampling (RUS) of majority class [13], SMOTE [2], and borderline-SMOTE [20]. Among them, SMOTE and borderline-SMOTE create synthetic samples for balancing the data, for examples. Other methods randomly pick samples from original dataset with replicate manner, which may lead to an overfitting problem [3]. In majority weighted minority oversampling technique (MWMOTE), minority samples are selected in an appropriate manner, and these instances are clustered to derive correct synthetic data [21]. In algorithm-level approach, minority class samples are weighted using learning procedure [6]. Adaboost is the first method under this approach. All of these methods are binary class-based classifier. Although multi-class imbalance data methods are available, very few suitable tools can the classification of multi-class imbalance data easily and properly. Author [22] solved this problem by introducing a multi-imbalance software, which consists of 18 algorithms for multi-class learning. The 18 algorithms are divided into seven categories; using five binarization techniques, this software is implemented for multi-class imbalance data [22]. However, if data are missing in these imbalanced datasets, then the situation becomes complex. With missing values, synthetic samples may not be correct for minor class, and multi-class software may fail to predict appropriate classes; as a result, the predicted output of the method will be inaccurate [3].

Imbalance problem in data streams is more challenging. Several existing ensemble methods can balance the data in each data chunk with different tricks. With Learn++.NSE method, SMOTE [2] is considered for balancing the minority data by creating synthetic data, and it creates individual classifier for each chunk. Using time-decay and their performance on the current chunk, classifiers are weighted in [15]. Furthermore, authors [4] solves the imbalanced data stream problem, and it is known as Learn++.CDS (Learn++ with concept drift and SMOTE). Another method [23] combines SMOTE [2] and Adaboost SVM ensemble integrated with time weighting method for balancing dynamic financial distress prediction model. To balance the current chunk, uncorrelated bagging (UB) [24] ensembles minority class samples from previous chunks, and under samples the majority class samples from current chunk. In the future, for the concept drift data, majority class can be converted into minority class. To solve this issue, SERA [25] uses a selective ensemble method to balance the data distribution. It preserves minority class data of previous chunk to use in latest chunk by selecting the most similar data of current chunk from preserved minority class data. It uses Mahalanobis distance to find similar data from previous chunks. Multiple selectively recursive approach toward imbalanced stream data mining (MuSeRA) [26] is the improved version of SERA [25]. MuSeRA ensembles all hypotheses built over previous chunk to make predictions. However, these methods do not consider complex data distribution. The selection-based resampling ensemble (SRE) [4] method calculates the probability of minority samples of previous chunks on the basis of difficulty factors for current data chunk. Nonetheless, to ensemble previous minority class data, there is an overfitting risk with majority class in the current data chunk. To overcome this, gradual resampling ensemble (GRE) [27] uses DBSCAN cluster to select the minority class samples that do not have any overlapping problem with the current majority class data. These existing methods use fixed chunk to classify the data streams, which becomes a problem when the amount of minority class data is very low or absent in the current data. As a result, adaptive chunk-based dynamic weighted majority (ACDWM) [5] method adaptively chooses the chunk size by statistical hypothesis tests. It does not need to store any previous data for its incremental manner.

All of these methods solve the imbalanced data stream problem by using ensemble method that are regarded as memory-inefficient methods. If there are missing values in an imbalanced data stream, then the previous ensemble data are not that much important because the missing values contain lack of information, and the lacking information at the previous ensemble data can misguide the processing of current data. To solve this issue, imputing missing data and balancing data for the current chunk are the best solutions. Although missing data greatly affect imbalanced data streams, there are not enough methods that consider the effect of both imbalance and missing data. Existing FID [3] imputes missing values with oversampling method by creating synthetic instances of minority class using fuzzy based decomposition [28] system, so it solves the imbalance problem. However, when it measures missing data, it sequentially increases the values using fuzzy-based decomposition [28]. This sequential manner of intervals, which is discussed further in the next section, sometimes predicts wrong values for missing data. As a solution, the proposed FAIA improves the sequential problem by using adaptive selection of intervals with information decomposition for data streams.

## 3 Proposed fuzzy adaptive imputation approach

### 3.1 Motivation

Missing data and imbalanced data are common in field with data stream and without data stream (i.e. offline data), so the best solution should be one that can solve both problems simultaneously and more correctly. Many existing chunk-based methods for imbalanced data stream exist, but they are not considered to address the problem of missing data. Fewer works take into account missing data and imbalanced data concurrently, and these are not available for data stream. These issues motivate the proposal of FAIA, a fuzzy adaptive decomposition approach for imbalanced data stream. FAIA can address the missing data and imbalanced data problems in data stream at the same time. It is even as stable for offline data field as in data stream. This study demonstrates how effectively FAIA can handle the missing values and imbalanced data simultaneously in data stream field.

### 3.2 Proposed methodology

In this section, the proposed online-based FAIA is presented in detail. This method not only can impute missing data presented in dataset but also balance the imbalanced data. Furthermore, it can be used for data stream that does not make use of history. The result section reveals how FAIA effectively works in offline dataset as well. For classifying data, decision tree C4.5 is used in this method because it is more popular for classifying. FAIA is a single pass model wherein previous data are not considered in the current chunk. Given the missing data, the single pass model is more suitable than ensemble model, which considers previous chunks values. In addition, with missing values, the previous chunks cannot provide detailed information for the current chunk. The proposed FAIA is divided into two parts: balancing data and imputing missing data.

### 3.2.1 Balancing data

If the number of instances in one class is more than that in another class, then an imbalance problem arises, which misclassifies instances in dataset and reduces the accuracy of classifier. Many methods are available for balancing a dataset. However, with missing values, balancing data is more challenging. To solve this problem, a new oversampling method is generated.

First, $|P_m|$ and $|N_m|$, the number of majority and minority instances, respectively, are computed from the dataset. Depending on the percentage of oversampling $f$, the amount of synthetic minority instances is then generated.

$$s_m = (|N_m| - |P_m|) \times f \tag{1}$$

Here, $s_m$ is the number of synthetic instances, and these are considered as missing values for the minority class. The missing data are imputed (as described in the following subsection). Fig. 1 shows the proposed method. Figure 1(a) illustrates an arrival data chunk with missing values. The blue circles represent majority class data without missing values, and yellow circles present majority class data with missing values. The black stars are minority class data without missing values, and yellow stars present minority class data with missing data. The number of stars is less than that of circles. To balance the data chunk from data stream, extra yellow stars are added in the chunk as missing values (Fig. 1(b)). These additional yellow stars are basically synthetic data which will balance the chunk of data stream using Eq. (1). After balancing all missing data in the chunk of data stream, these are imputed. The green stars are synthetic and partially imputed data in minority class (Fig. 1(c)). The green circle with yellow are imputed data in majority class (Fig. 1(c)).

### 3.2.2 Imputing missing data

With missing values in a dataset, it is difficult to classify properly because the missing values of a certain instance may represent vital characteristics for categorizing it under the correct class. In addition, if the dataset is imbalanced, then it is more difficult to get correct classes for the data. Undoubtedly, measuring the missing data is crucial. However, computing the exact values of missing data is challenging.

For example, a dataset $M$ has $m$ instances and $n$ attributes with missing values.

The proposed FAIA uses column vector $x$ with $m$ features and $t$ numbers of missing values at vector $x$. Here, column vector $x$ is basically all the instances' values in a certain attribute from $n$. The total missing values are $t = s_m + N$, where $N =$ number of missing values present in main data at vector $x$. Therefore, t number of missing values must be computed. In addition,

$$x = (x_1, \ldots, NaN, \ldots., x_i, \ldots, NaN, \ldots, NaN, \ldots, x_m)^T \tag{2}$$

where $x_i$ is a feature value, $NaN$ represents not a number which is basically considered as missing values, and $T$ is transpose of the vector. $D$ represents the index set of observed features values.

$$D = \{i | x_i \neq NaN, i = 1, 2, \ldots, m\} \tag{3}$$

Moreover, $a$ and $b$ denote the minimum and maximum values of the observed data, respectively.

$$a = min\{x_i | i \in D\}, b = max\{x_i | i \in D\} \tag{4}$$

The mean of each column vector is calculated with respect to all observed value $D$ for entire dataset $M$.

$$M_e = \frac{1}{m} \sum_1^m x_i; x_i \in D \tag{5}$$

$M_e$ value is used in all $NaN$ positions according to the columns. For column vector $x$, interval $I = [a, b]$ and $h$ denote step length determined as $t$. Thus, $h = (b - a)/t$.

Interval $I$ is divided into $t$ parts to determine the weights for recovering missing values.

Therefore, $I = \cup_{s=1}^t I_s$ where

$$I_s = [a + (s-1) \times h, a + s \times h), s = 1, 2, .., t-1 \tag{6}$$

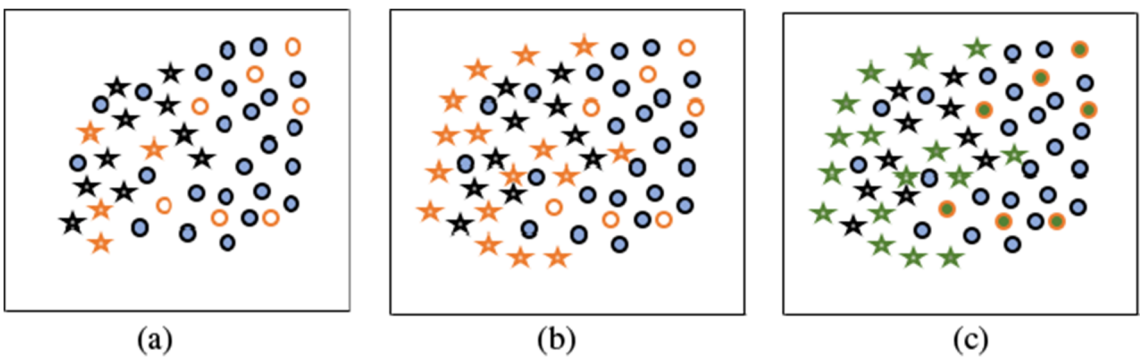



**Fig. 1** Resampling procedure for data stream using the proposed method: (a) imbalanced data with missing values, (b) balanced data with missing values, and (c) imputed missing data

$$I_t = [a + (t-1) \times h, a + t \times h] \tag{7}$$

For contribution weighting, the discrete of universe for $x$, $U = \{u_1, u_2, \ldots, u_t\}$ [28] is considered where

$$u_s = \frac{a + (s-1) \times h + a + s \times h}{2}, s = 1, 2, \ldots, t. \tag{8}$$

To achieve the goal (a new FAIA), the fuzzy of $x$ is represented in the following form:

$\mu : x \times U \rightarrow [0, 1]$ and $(x_i, u_s) \rightarrow \mu(x_i, u_s)$, where $\mu(x_i, u_s)$ denotes membership degree.

Fuzzy adaptive imputation is calculated as follows:

At first, distances between $x_i$ and $U = \{u_1, u_2, \ldots, u_t\}$ are calculated using $\|x_i - u_s\|$. The smallest distances are selected, and this is the first step in making the method adaptive and finding $(x_i, u_s) = e^{-\frac{\|x_i - u_s\|}{h}}$. If the number of smallest distances are more than 1, then $\mu(x_i, u_s)$ values are attributed to all of those $U$ that belongs to the smallest distance. Hence,

$$\mu(x_i, u_s) = \begin{cases} e^{-\frac{\|x_i - u_s\|}{h}}, & if\ min\{|x_i - u_s|\} > 0 \\ 0, & Otherwise \end{cases} \tag{9}$$

The minimum numbers of instances present in the discrete of universe is determined, with $U$ *as* $Min_{intances}$ = *min* {*number of instances* $u_k | k \in U$ *AND* $\mu(x_i, u_k) ! = 0$.}

To make adaptive information decomposition, the distance $d$ of discrete of universe, $U$ from each missing valued instance is measured by using Euclidean distance, as given in Eq. (10). For each column, the entire row of corresponding instance is considered with attribute values that aid to find the total minimum distance from corresponding missing instances to observed instances. The entire row consists of some attributes, and these attribute values represent the instance into a specific class. For a certain class, all instances' attribute values are closer than those of other classes. To compare the attribute relationship, we use Euclidean distance, which is actually measured as the total distance. If instances belong to the same class, then the attribute values are the closest than other classes' instances. As a result, the total distance will be the smallest. This is the reason the equation used involves the interrelationships of attributes.

$$d(x_s, u_s) = \begin{cases} \sqrt{\sum_{j=1}^{n} \left(M(x_s, y_j) - M(x_i, y_j)\right)^2}, & if\ \mu(x_i, u_s) != 0 \\ 0, & otherwise \end{cases} \tag{10}$$

Here, $y_j$ represents $j^{th}$ attribute of the dataset, and $x_s$ is the missing value. $M(x_s, y_j)$ and $M(x_i, y_j)$ are the attribute values of corresponding instances in main dataset $M$. In addition, $u_s \in U$. Here, for each $u_s$ if membership function $\mu(x_i, u_s)$ is not equal to 0 then the Euclidean distance $d(x_s, u_s)$ is calculated.

The computed distances $d(x_s, u_s)$ are sorted in an ascending order for corresponding $u_s$. The average of each $d(x_s, u_s)$ is measured by

$$A(x_s, u_s) = \frac{\sum_{i=1}^{Min_{intances}} d(x_{si}, u_{si})}{Min_{intances}} \tag{11}$$

$$(u_s, p_s) = \min\{A(x_s, u_s)\} \tag{12}$$

Here, $p_s$ is the position of all corresponding $u_s$ values. Thus, the $u_s$ is measured adaptively in the proposed FAIA where $u_s$ of FID is selected sequentially such that the first element of $U$ is considered for the first missing data imputation.

Finally, the information decomposition to retrieve the $s^{th}$ missing value of $x$ is given as follows:

$$\widetilde{m_s} = \frac{\sum_{j \in D} m_{i,\ p_s}}{\sum_{i \in D} \mu\left(x_i, u_{p_s}\right)}, \text{if } \sum_{i \in D} \mu\left(x_i, u_{p_s}\right) > 0 \tag{13}$$

Here, $\bar{x}$ is the mean of all observed feature values, $\sum_{i \in D} \mu(x_i, u_s) = 0$ occurs when no observed feature values contribute to $I_s$, and $m_{i,\ p_s} = \mu\left(x_i, u_{p_s}\right) \times x_{i,}, \in D$.

**Theorem:** Let $I_s = [a_s, b_s]$, where $I_s$ is the $s^{th}$ closest interval. Then, the corresponding observed missing value $\widetilde{m_s} \in (a_s, b_s)$, which means for adaptive fuzzy imputation, it must remain in the range of $a_s$ to $b_s$. This range is the closest interval among all intervals (i.e., $a_s \leq \widetilde{m_s} \leq b_s$).

***Proof:***

As previously mentioned, $u_s$ is the center of $I_s$, thus $\|a_i - u_s\| = \|b_i - u_s\|$ . According to Eq. (9), if $\|x_i - u_s\|$ is not the minimum, then $\mu(x_i, u_s) = 0$. Moreover, if $\|x_i - u_s\|$ is the minimum for more than one interval of $I_s$, then the closest interval must be determined using Eqs. (10) to (12).The missing value is obtained using adaptive fuzzy imputation based on the proposed FAIA as $\widetilde{m_s}$ . Thus,

$$\widetilde{m_s} = \frac{\sum_{j \in D} \mu\left(x_i, u_{p_s}\right) \times x_i}{\sum_{i \in D} \mu\left(x_i, u_{p_s}\right)} \leq \frac{\sum_{j \in D} \mu\left(x_i, u_{p_s}\right) \times b_s}{\sum_{i \in D} \mu\left(x_i, u_{p_s}\right)}$$

Similarly,

$$\widetilde{m_s} = \frac{\sum_{j \in D} \mu\left(x_i, u_{p_s}\right) \times x_i}{\sum_{i \in D} \mu\left(x_i, u_{p_s}\right)} \geq \frac{\sum_{j \in D} \mu\left(x_i, u_{p_s}\right) \times a_s}{\sum_{i \in D} \mu\left(x_i, u_{p_s}\right)}$$

This is how other attributes of a certain instance with missing value help in choosing the nearest interval for imputing missing value. This ends the proof. □

The proposed FAIA is described by Algorithms 1 and 2, which are both used for data stream. However, if offline data is considered, then only Algorithm 2 is taken into account. At first, data stream $S$ is considered in Algorithm 1. For each current data chunk $B_m$ ($B_m \in S$), a certain amount of data is considered as training data ($P_m$ represents positive data and $N_m$ represents negative data). To balance the training data,

how much minority data need to be generated is calculated by using Eq. (1) at step 3 (in Algorithm 1). To impute missing data, Algorithm 2 is called at step 4. In Algorithm 2, for each column of $\boldsymbol{D}$, maximum and minimum values are measured at step 2 for dividing the entire length $h$ into $t$ number of parts to get intervals using step 5 for corresponding feature vector. For each missing value, the suitable interval $I$ is selected, and MF is calculated for imputing missing value. For selecting suitable interval and making the method adaptive, the first step is identifying the proper $U$ and its' MF using Eq. (9) at step 7. The minimum distance from corresponding missing instance is then calculated using Eq. (10). This step basically considers all attributes of instances in dataset $M$, which actually represents the interrelationship among attributes for the instance. The proposed adaptive process is represented in Algorithm 2 from steps 6 to 14. This process is repeated until all missing values are measured. When all missing values are imputed, then Algorithm 2 returns these values to step 5 of Algorithm 1 to predict the result. Finally, FAIA not only imputes missing values but also reduces the imbalance issues in data stream by oversampling the minority class. By contrast, for the dataset without streaming (i.e., offline data), the training data is passed to Algorithm 2 for balancing the data and imputing the missing data.

## 4 Iv. Performance evaluation

In this section, the performance of FAIA is evaluated in detail. The evaluation is divided in three parts. The first part (Section IV: A) represents the capacity to impute missing values with several missing rates in dataset. The performance of FAIA is compared with those of KNN [1] and FID [3]. Although KNN [1] is a basic missing data imputation method, it is the most popular. FID [1] is a recently published method, and its imputation capacity can easily be shown from its algorithm. In this part, 11 datasets are used to compare FAIA with the existing methods. The second part (Section IV: B) measures how much stable the proposed FAIA is for missing value imputation for imbalanced information with respect to data streaming scenario. Two synthetic datasets and two real-world datasets with different rates of missing values (a total of four datasets) are used. The third part (Section IV: C) considers the performance of resampling the minority data imbalance without data streaming (i.e., for offline data).

**A. missing data imputation** This section measures the imputation capacity of missing data of the proposed FAIA.

A.1) **Datasets:** Eleven real-world datasets are used from

**Algorithm 1:** Fuzzy adaptive imputation approach

**Input:** $S$ = data stream, $B_m$ = current data chunk, $f$ = percentage of oversampling
**Output:** predicted labels of testing sets of data in the dataset

**Step 1:** ***for all current*** data chunk $B_m \in S$ ***do***
**Step 2:** split $B_m$ into $P_m$ *and* $N_m$, which are positive and negative data in the current chunk, respectively.
**Step 3:** find number of synthetic data needed to evaluate using $s_m = (|N_m| - |P_m|) \times f$
**Step 4:** assign $s_m$ number of missing values in the training data and call Algorithm 2 to impute missing data
**Step 5:** predict the label for each testing instance using C 4.5 classification algorithm.
**Step 6:** ***end for***

**Algorithm 2:** Missing data imputation

**Input:** a set of positive instances with the missing data $D$
**Output:** a set of positive instances with synthetic positive instances and imputed missing data using *function* $FAIA(D)$

**Step 1:** ***while*** each feature vector $x$ has missing values
**Step 2:** calculate numbers of missing values $t$ in $x$ and find $a$ and $b$ using $Eq^n(4)$
**Step 3:** mean of observed data using $Eq^n(5)$ and put in $NaN$ positions
**Step 4:** step length, $h = (b-a)/t$
**Step 5:** compute intervals using $Eq^n(6)$ and $Eq^n(7)$
**Step 6:** ***for*** each missing number from 1 to $t$
**Step 7:** calculate discrete universe using $Eq^n(8)$ and membership function (MF) using $Eq^n(9)$
**Step 8:** ***for*** each interval (Steps 8, 9, 10, and 12, make the FAIA adaptive to find the exact interval)
**Step 9:** distance of discrete universe using $Eq^n(10)$ through $\boldsymbol{m}$ , this step helps choose proper interval for missing data
**Step 10:** find average of all distances from Step 9 using $Eq^n(11)$, which makes adaptive selection.
**Step 11:** ***end for***
**Step 12:** measure the least distance considering $Eq^n(12)$
**Step 13:** finally recover missing data using $Eq^n(13)$
**Step 14:** ***end for***
**Step 15:** ***end while***

**Table 1** Database information for missing data imputation and offline data with missing values

| Data Number | Data Name | #Size | #Attributes |
|---|---|---|---|
| 1 | KC1 | 2109 | 18 |
| 2 | MC2 | 161 | 39 |
| 3 | PC1 | 1109 | 21 |
| 4 | PC3 | 1563 | 37 |
| 5 | Sonar | 208 | 60 |
| 6 | Vehicle | 846 | 18 |
| 7 | Wine | 168 | 14 |
| 8 | Glass (1) | 214 | 9 |
| 9 | Glass (2) | 214 | 9 |
| 10 | Page Block (1) | 5473 | 10 |
| 11 | Page Block (2) | 5472 | 10 |

several databases. Table 1 represents the details of all 11 datasets. The size of datasets varies from 161 to 5473, and the attribute ranges from 6 to 60. To measure the results of datasets, multiple classes of datasets are considered as binary classes. The smallest class is considered as the minority class, and remaining other classes are presented as majority class in the dataset.

A.2) **Data Missing Mechanism and Missing Rate:** According to [1], the three types of missingness mechanisms are as follows:

- In missing completely at random (MCAR) case, missing information does not rely on either observed or unobserved data.
- In missing at random (MAR) case, missing information can be predicted from other observed data in datasets.
- In not missing at random (NMAR) case, missing information is slightly complicated to measure. Basically, missing data depend on unobserved data in datasets.

To measure performance, we follow the process for creating incomplete datasets that is used in [3, 7]. Datasets with 5%, 10%, 15%, 20%, and 30% missing values are considered to evaluate the performance of all existing methods.

A.3) **Performance Measure for Missing Values:** The performance of missing data imputation methods can be measured in a number of ways. One is the difference between original and measured values called RMSE [1]. The difference is the distance between two values. To compute these distances, Euclidean distance is used. Most of the missing data imputation techniques employ this method. When the RMSE values of a method are smaller than other existing methods then, the former is considered much stable.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(e_i - \breve{e}_i)^2} \qquad (14)$$

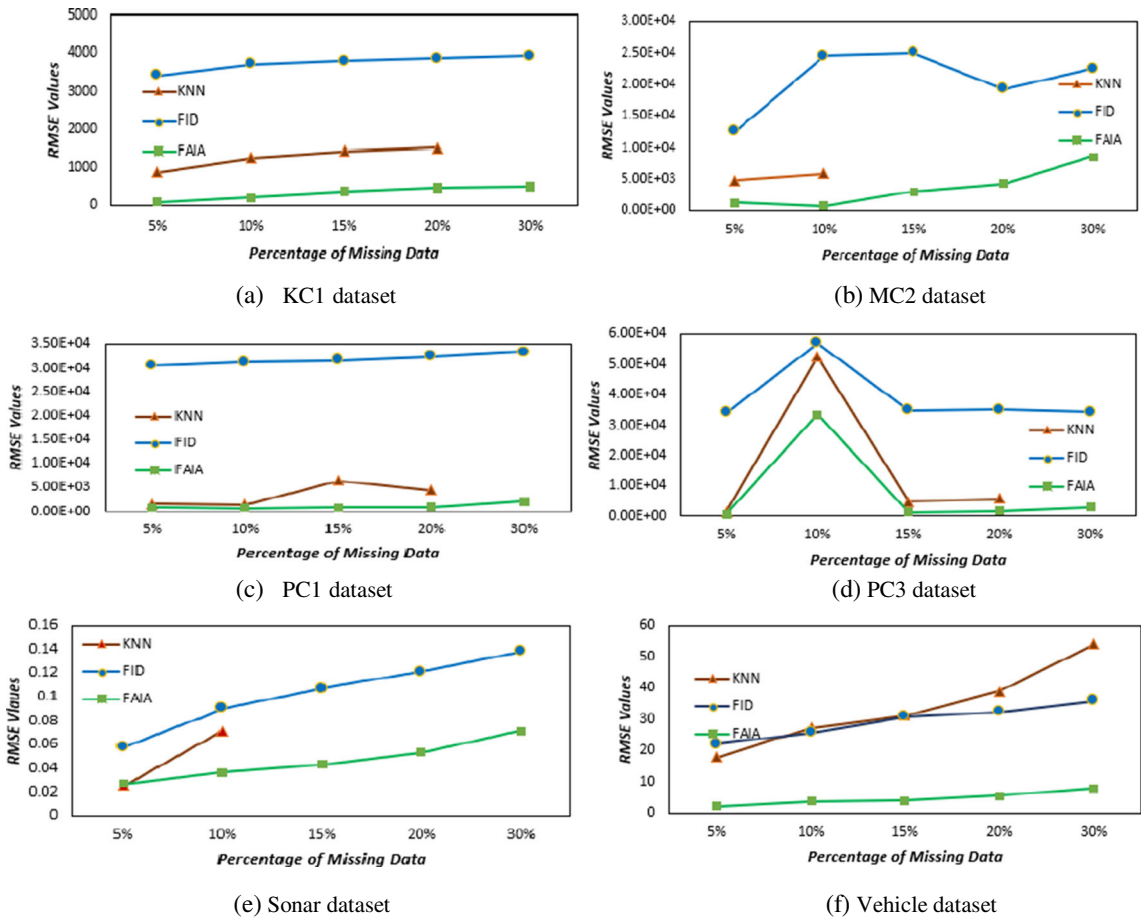
Here, $e_i$ and $e_i$ are the actual and imputed values, respectively. To evaluate the performance of FAIA, its performance is compared with those of KNN [1] and FID [3] method. These two methods are used to compare because KNN [1] is known as a very popular method for missing data imputation, and the proposed FAIA tries to solve the limitation of FID [3]. FAIA's main purpose is to solve the missing values in imbalanced data and to balance the dataset simultaneously, but recent methods that solve these problems at the same time are scant. KNN [1] finds the k number of nearest neighbors from observed instances, and the average result is taken into account as the value of corresponding missing value.

A.4) **Result for Missing Value Imputation:** Fig. 2 shows the RMSE values of all datasets, which are mentioned in Table 1. Figure 2(a) represents the RMSE of KC1 dataset with 5%, 10%, 15%, 20%, and 30% missing values. The outputs of the proposed FAIA are much lower than those of KNN [1] and FID [3]. Using Eqs. (9), (10), (11), and (12) makes the proposed FAIA adaptive by considering the attribute values of corresponding missing instances. These attribute values not mentioned in FID [3] denote an interrelationship. Equation (10) measure distances from a certain missing instance to all observed instances that belong to different MFs. By using Eq. (11), the average values are calculated, and finally, using Eq. (12), the minimum distance and corresponding $u_s$ are selected. Thus, FAIA adaptively selects $u_s$ and based on this value, the missing data are computed for the corresponding instance using Eq. (13). In addition, when missing values are more than 20%, KNN cannot impute the missing values because of the missing data of each row. This problem is created when instances are less but the rate of missing data is high. In Fig. 2(b), the proposed FAIA provides lower RMSE than KNN [1] and FID [3]. Similar to the previous figure, when missing values are more than 10%, then KNN cannot impute missing values because instances are less but the rate of missing data is high. Similar to the previous reason, Figs. 2(c), 2(d), 2(e), 2(f), and 2(g) indicate that FAIA provides better results than KNN and FID. For KNN, when missing values are more than 20%, 20%, 10%, and 20% in PC1, PC3, Sonar, and Wine datasets, respectively, it cannot impute

the missing values for the same reasons. In Figs. 2 (h) and 2(i), FID and the proposed FAIA obtain much closer results, thus showing the significance of the proposed method because FAIA is considered as the algorithm for data stream. Figures 2(j) and 2 (k) show differences from previous results. When the rate of missing data is increasing, FAIA provides almost the same or higher values than KNN but better than FID. The overall observation concludes that the proposed FAIA can reduce the problem of missing data.

**B. data stream with missing values** This section measures the stability of the proposed FAIA for imputing missing values in imbalanced information when considering data stream.

B.1) **Datasets:** The datasets used in this experiment are two synthetic datasets and two real-world datasets. Two synthetic datasets are SEA [29] and Hyperplane [29]. The SEA dataset has three attributes, and among them, third is considered as noise [4], and two are relevant. This dataset has 60,000 instances with two classes. Concept drift is also considered in this dataset. The Hyperplane [29] generator contains an incremental drift. It generates 200,000 instances with 10 attributes and two classes. Weather and Electricity market datasets are the two real-world datasets, and these are used to measure and compare the stability of the existing methods w.r.t the proposed FAIA. Weather dataset consists of weather details of the Offutt Air Force Base in Bellevue, Nebraska in the period of 1949–1999 [29]. This dataset comprises 18,159 instances, eight features, and two classes (1,0). Electricity market dataset contains information of the Australian New South Wales Electricity Market with 45,312 instances, eight features, and two classes (1,0) [29]. Table 2 shows the details about the datasets used for data stream.




(a) KC1 dataset (b) MC2 dataset

(c) PC1 dataset (d) PC3 dataset

(e) Sonar dataset (f) Vehicle dataset

**Fig. 2** RMSE values with different percentages of missing data: (a) KC1 dataset, (b) MC2 dataset, (c) PC1 dataset, (d) PC3 dataset, (e) Sonar dataset, (f) Vehicle dataset, (g) Wine dataset, (h) Glass (1) dataset, (i) Glass (2) dataset, (j) Page Block (1) dataset, and (k) Page Block (2) dataset

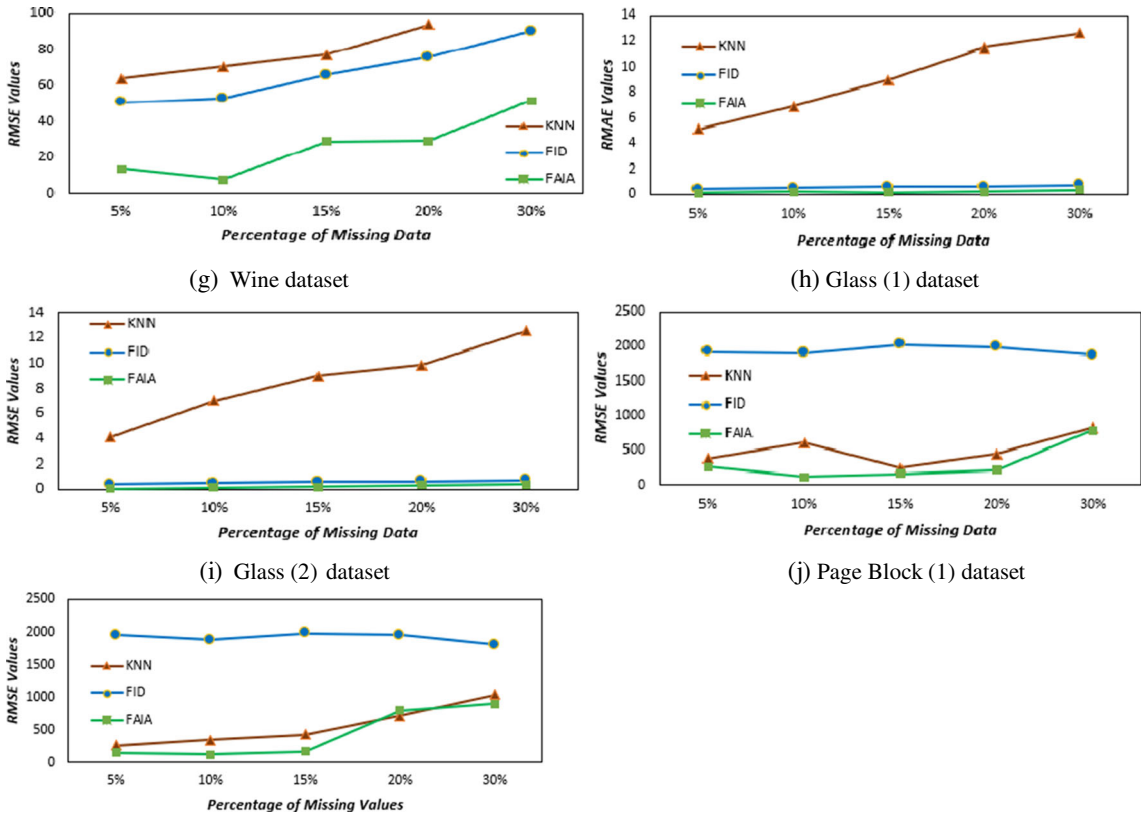



**Fig. 2** continued.

**Table 2** Database information for data stream with missing values

| Data Number | Data Name | #SIZE | #Attributes |
|---|---|---|---|
| 1 | SEA | 60,000 | 03 |
| 2 | Hyperplane | 200,000 | 10 |
| 3 | Weather | 18,159 | 08 |
| 4 | Electricity market | 45,312 | 08 |

B.2) **Performance Measure for the Imbalanced Data Stream:** The performances of imbalanced data stream methods can be measured in many ways, but all measuring models do not prove a favorable matric. In this case, three common performance measurement methods are used. They are area under the receiver operating characteristic (ROC) curve, AUC [30], F-measure [31] and Geometric mean (GM) [4].

AUC measures which method provides better classification performance [13] using the true positive rate (TP rate) and false positive rate (FP rate), and this is calculated from the confusion matrix [3].

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{15}$$

Here, $TP_{rate} = \frac{TP}{TP+FN}$, and $FP_{rate} = \frac{FP}{FP+TN}$

F-measure is calculated by the precision and recall [31–33]. The F-measure is computed by the following rule:

$$F\text{-}measure = \frac{(\beta + 1) \times precision \times recall}{\beta \times (precision + recall)}, \tag{16}$$

**Table 3** Confusion Matrix [33]

| Predicted Values<br>Actual Values | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | TP (number of True Positive) | FN (number of False Negative) |
| Actual Negative | FP (number of False Positive) | TN (number of True negative) |

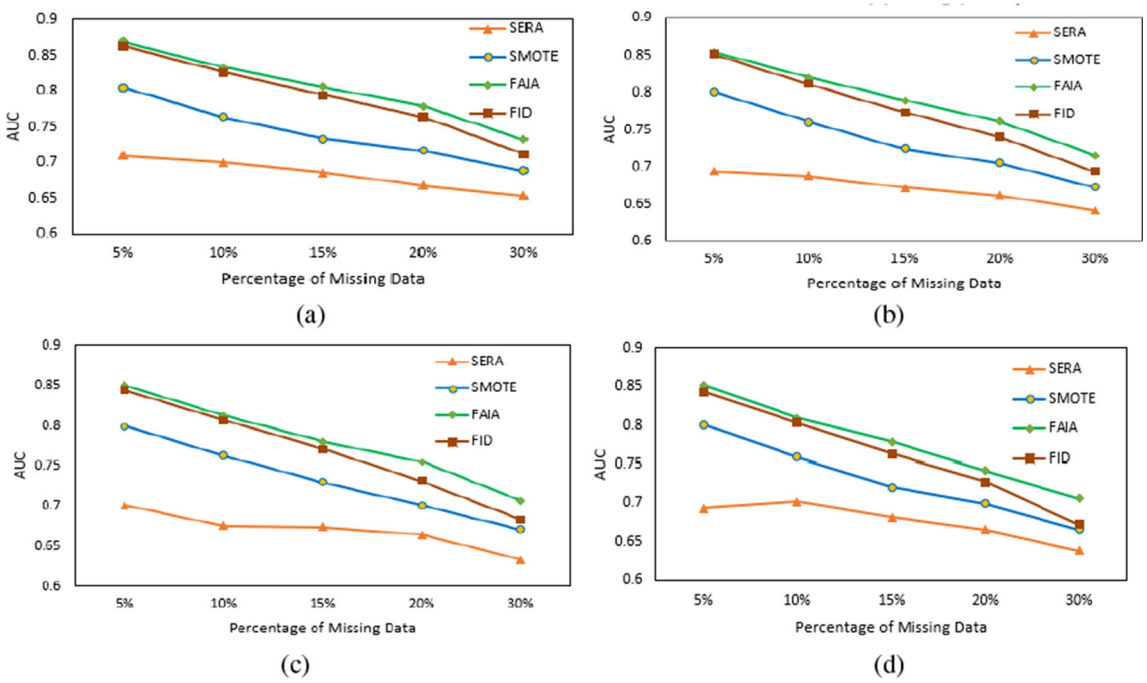



**Fig. 3** AUC values for Electricity market dataset with different rates of missing values: (a) AUC values for 500 data chunk, (b) AUC values for 1000 data chunk, (c) AUC values for 1500 data chunk, and (d) AUC values for 2000 data chunk

where $\beta$ is a constant. For this experiment, $\beta$ is considered as 1. Precision and recall are defined as follows in Table 3:

$precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$

G-mean evaluates the accuracy of the method based on the ration of positive accuracy and negative accuracy in the confusion matrix [4].

$$GM = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{17}$$

B.3) **Setting of Experiments:** Many methods are used to solved imbalanced data streaming problems.

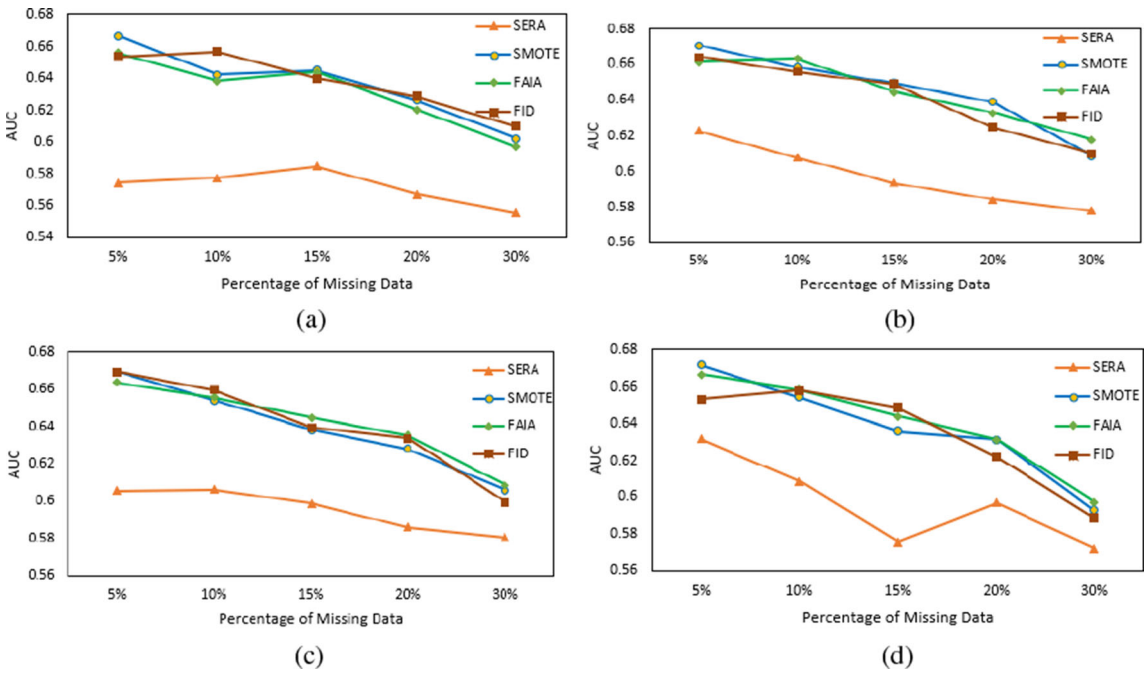



**Fig. 4** AUC values for Weather dataset with different rates of missing values: (a) AUC values for 500 data chunk, (b) AUC values for 1000 data chunk, (c) AUC values for 1500 data chunk, and (d) AUC values for 2000 data chunk

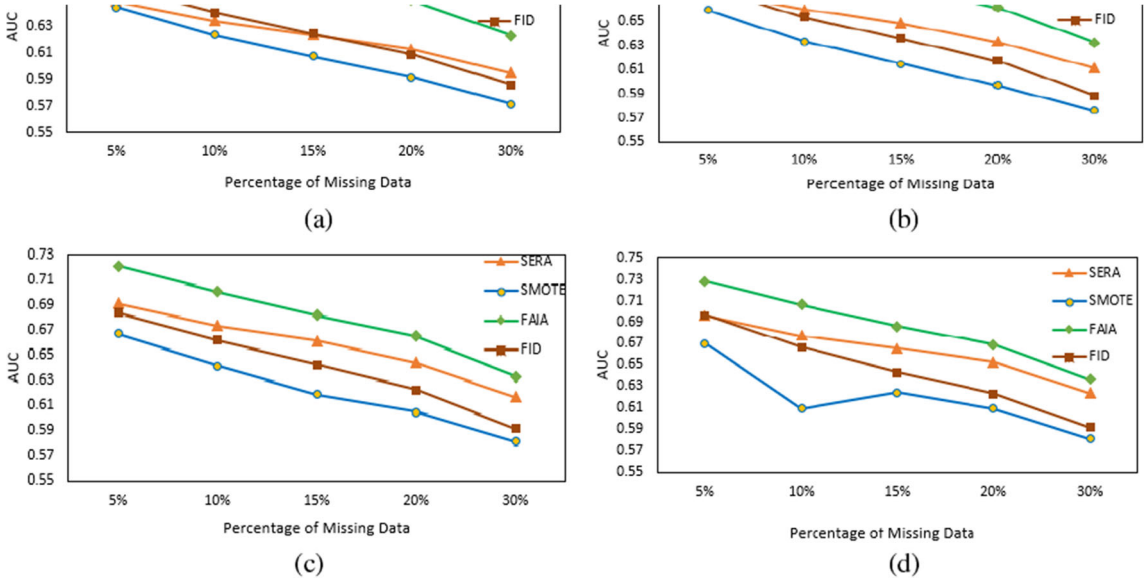



**Fig. 5** AUC values for Hyperplane dataset with different rates of missing values: (a) AUC values for 500 data chunk, (b) AUC values for 1000 data chunk, (c) AUC values for 1500 data chunk, and (d) AUC values for 2000 data chunk

However, these methods only focus on balancing the data and overlook missing information although it is a vital issue to solve imbalanced data. Therefore, the proposed FAIA concurrently focus on missing values and imbalanced dataset, and thus adaptive imputation is considered. For measuring the stability of the proposed FAIA, existing methods like SERA [25], SMOTE [2], and FID [3] are considered. Here, SMOTE [2] solves imbalance problem chunk by chunk. SMOTE creates synthetic minority class data to balance the chunk at each time. Although SMOTE [2] and FID [3] are offline-based imbalance data methods, these are considered to show how effective they are at data stream field. The chunk sizes are 500, 1000, 1500, and 2000. The chunk sizes are taken manually. Basically, there is no need

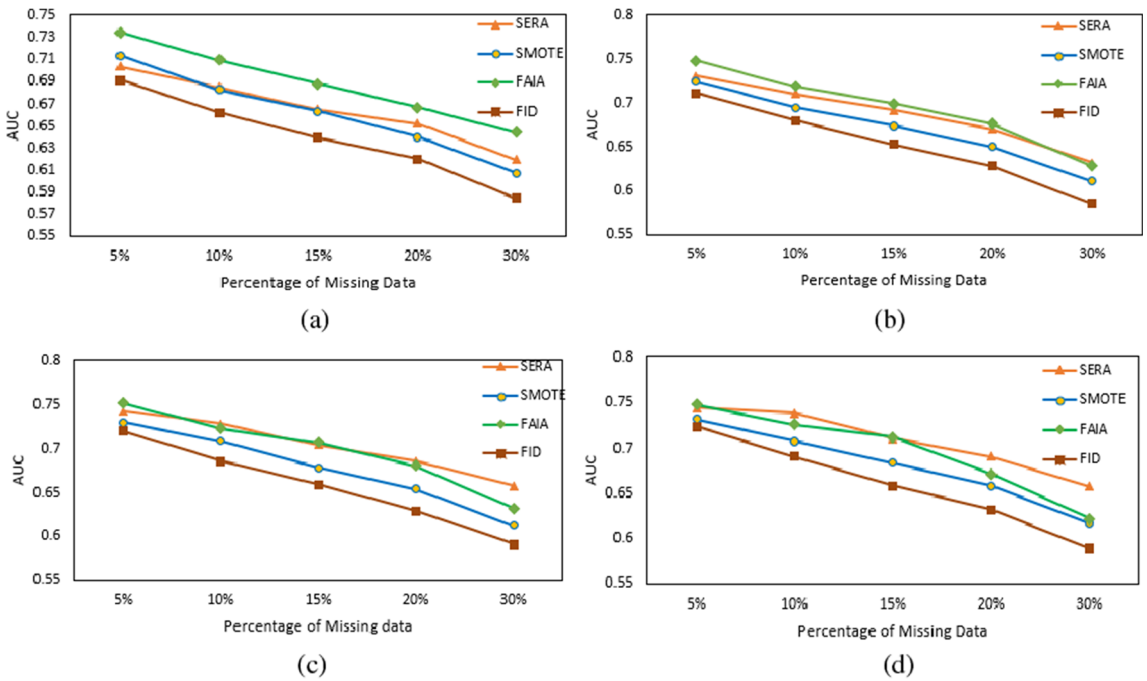



**Fig. 6** AUC values for SEA dataset with different rates of missing values: (a) AUC values for 500 data chunk, (b) AUC values for 1000 data chunk, (c) AUC values for 1500 data chunk, and (d) AUC values for 2000 data chunk

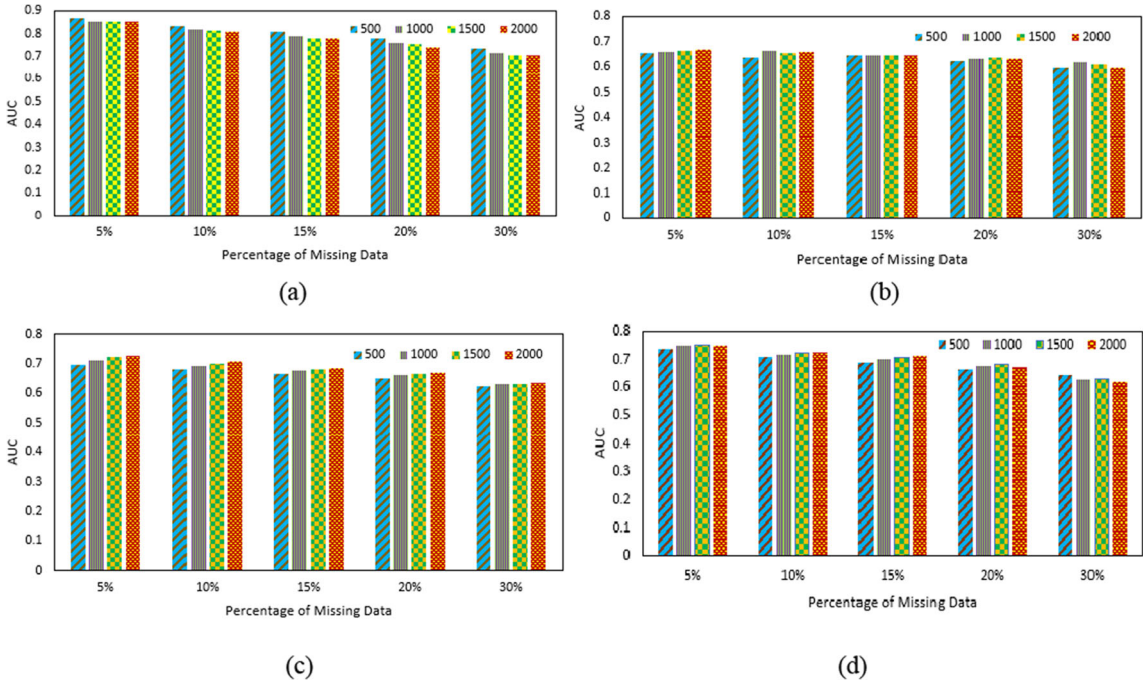



**Fig. 7** Effect of different chunks on the proposed FAIA: (a) AUC values for Electricity market dataset, (b) AUC values for Weather dataset, (c) AUC values for Hyperplane dataset, and (d) AUC values SEA dataset with different percentage of missing data

to determine appropriate chunk size because the chunk size, which is be discussed in the next section in depth, has no effect on the outcome. The oversampling rate used in the method is 500%, which mean 500% minority samples are added into the main dataset [3]. SERA [25] is an ensemble method for imbalanced data stream. It selects minority samples that are nearest to the current minority class to balance the recent chunk. The post-balance ratio $f$ is 0.3, $k = 10$ is the ensemble members for enriching the training set in the current chunk. For each chunk, 9.09% samples are considered as training data. Similar to the proposed FAIA, FID [3] takes into account both missing values and imbalance problem. As FAIA imputes missing values, the previous chunk's information is unnecessary. Therefore, FAIA imputes missing values and balances the current chunk distribution using Algorithms 1 and 2. All methods including FAIA are implemented in MatLab R2020. For classifying the samples, DT C4.5 is used. The results are computed using 500, 1000, 1500, and 2000 chunk size. Two-fold cross validation is used to train data for SMOTE [2], FID, [3] and the proposed FAIA.

B.4) **Result for Data Stream with Missing Data:** Fig. 3 shows the AUC values for Electricity market dataset with different missing values. The figure clear reveals how important measuring missing data is. Although SERA [25] and SMOTE [2] are traditional methods, these remain popular in this related field. The main motivation of the proposed method in to show the effectiveness of imputing missing data and balancing the data in data streams simultaneously. In recent methods, either only missing data or data imbalance problem is solved. These methods do not work on both problems simultaneously, so the proposed method may be new for this data stream field. The traditional methods are used for comparison to show how much effective FAIA is in this field.

Finally, all figures with different data chunks with different missing values show that FAIA achieves better results. Every time, SERA shows lower output because it considers previous data chunk with missing values. In addition, for each figure, when the rate of missing data is increasing, FAIA provides higher results than others because it considers the interrelationship, which is calculated using Eqs. (9), (10), (11), and (12). For Weather dataset in Fig. 4, when the chunk size is equal to 500, FID and SMOTE provide higher result than FAIA because the interrelationship among attribute data do not provide much vital details or some observed instances' attribute values create large distances using Eq. (10) for corresponding missing instance. These large distances are created because of some values of specific attributes in the same

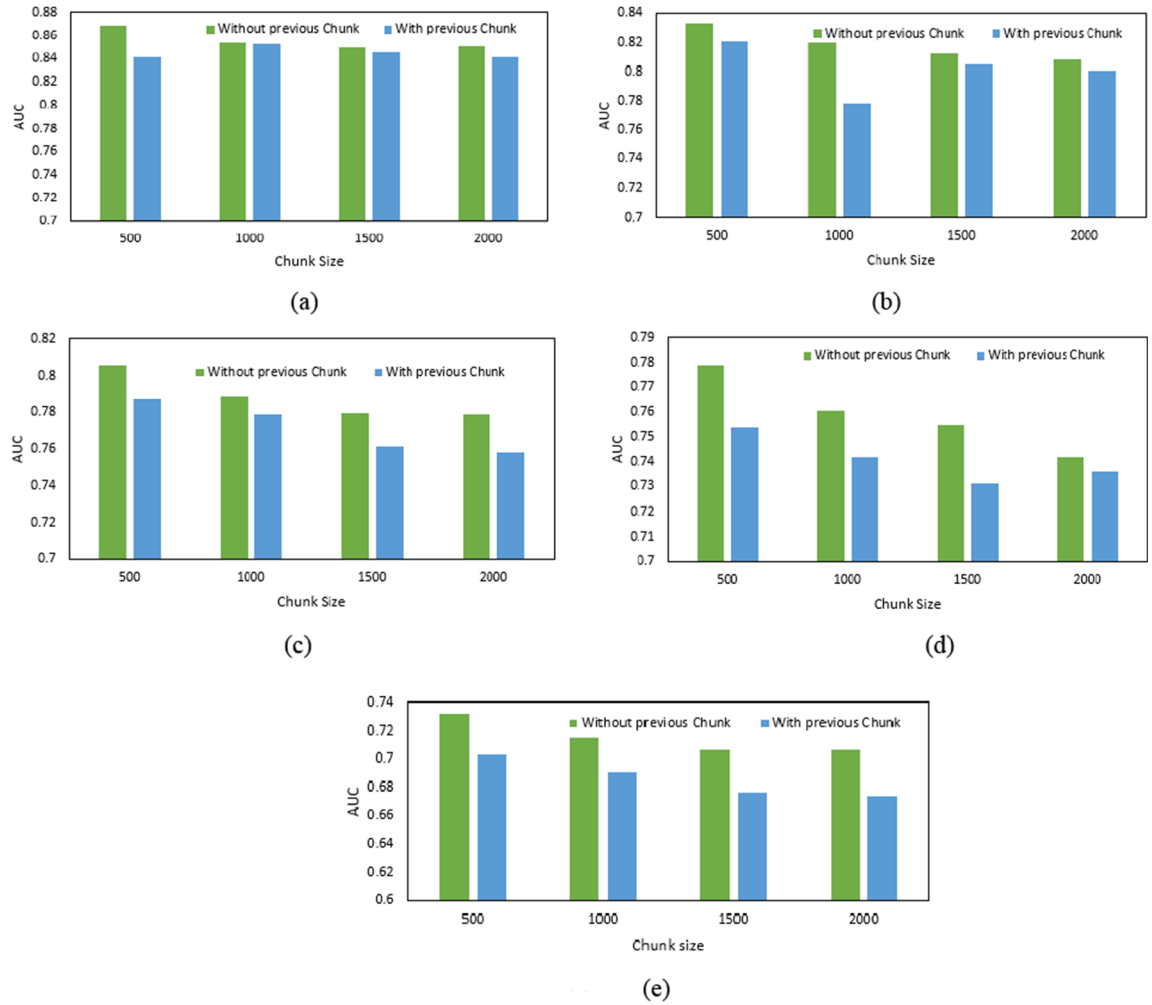



**Fig. 8** Previous chunk's effect on the proposed FAIA: (a) AUC values for Electricity market dataset with 5% missing values, (b) AUC values for Electricity market dataset with 10% missing values, (c) AUC values for Electricity market dataset with 15% missing values, (d) AUC values for Electricity market dataset with 20% missing values, and (e) AUC values for Electricity market dataset with 30% missing values

classes that are not related to the corresponding attributes of the missing instance. When the chunk sizes are 1000, 1500, and 2000 with lower rate of missing data, then initially FAIA provides lower or almost close results to SMOTE and FID (Fig. 4), but when the rate increases the AUC values improve with respect to SMOTE and FID. This happens because of the interrelationship of attributes, which provide detailed information about missing values. FAIA does not consider the previous chunk, so it is a one pass method that improves the time complexity and memory efficiency. Unlike the proposed FAIA, SERA in Fig. 4 always gives lower results than other methods because of the previous incomplete chunk data are ensembled.

Figure 5 shows that FAIA constantly obtains good results for all chunks when considering Hyperplane dataset. However, in Fig. 6, FAIA initially provides higher results than SERA, but when the rate of missing data increases, the AUC values decrease. Thus, for the SEA dataset, the proposed FAIA becomes overfitted if chunk size increases, and this happens because of the attribute numbers. The SEA dataset only has three attributes, and when the rate of missing data increases, FAIA fails to gain more information from the attributes for corresponding missing instances because the values of the third attribute in this dataset are considered as noise [4]. The distances measured using Eq. (10) misleads Eqs. (11) and (12) in choosing the correct $u_s$. As a result, when the rate of missing data is high and because noise is presented in the dataset, the proposed FAIA provide less AUC values. Nonetheless, this only happens when the rate of missing data is high.

Figure 7 shows the impact of different data chunks on the proposed FAIA with several rates of missing data. The results are almost similar for all of the data chunks. In all datasets, FAIA provides the closest or more similar values for 1500 and

**Table 4** F-Measure

| Data Name | Percentage of Missing Value | FAIA | | | | FID [3] | | | | SMOTE [2] | | | | SERA [25] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chunk Size | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 |
| Electricity market dataset | 5% | **0.85** | **0.82** | **0.82** | **0.83** | 0.84 | 0.826 | 0.817 | 0.82 | 0.76 | 0.75 | 0.75 | 0.75 | 0.65 | 0.64 | 0.63 | 0.63 |
| | 10% | **0.80** | **0.78** | **0.77** | **0.77** | 0.79 | 0.77 | 0.766 | 0.76 | 0.70 | 0.69 | 0.70 | 0.69 | 0.63 | 0.62 | 0.59 | 0.64 |
| | 15% | **0.76** | **0.74** | **0.73** | **0.73** | 0.75 | 0.72 | 0.71 | 0.70 | 0.64 | 0.63 | 0.64 | 0.62 | 0.62 | 0.62 | 0.59 | 0.62 |
| | 20% | **0.73** | **0.70** | **0.69** | **0.67** | 0.70 | 0.66 | 0.65 | 0.64 | 0.61 | 0.59 | 0.58 | 0.59 | 0.59 | 0.60 | 0.58 | 0.598 |
| | 30% | **0.65** | **0.62** | **0.61** | **0.61** | 0.61 | 0.58 | 0.56 | 0.53 | 0.56 | 0.53 | 0.53 | 0.52 | 0.57 | 0.57 | 0.53 | 0.55 |
| | Mean | 0.758 | 0.732 | 0.724 | 0.722 | 0.738 | 0.711 | 0.701 | 0.69 | 0.65 | 0.64 | 0.64 | 0.63 | 0.61 | 0.61 | 0.58 | 0.61 |
| | Standard Deviation | 0.075 | 0.077 | 0.08 | 0.086 | 0.088 | 0.096 | 0.1 | 0.111 | 0.08 | 0.09 | 0.09 | 0.09 | 0.03 | 0.03 | 0.04 | 0.04 |
| Weather dataset | 5% | **0.76** | **0.76** | **0.75** | 0.76 | 0.73 | 0.75 | 0.74 | 0.73 | 0.69 | 0.70 | 0.69 | 0.69 | 0.67 | 0.73 | 0.70 | **0.76** |
| | 10% | **0.72** | **0.73** | **0.73** | 0.73 | 0.71 | 0.71 | 0.71 | 0.71 | 0.64 | 0.66 | 0.64 | 0.64 | 0.69 | 0.72 | 0.71 | **0.74** |
| | 15% | 0.70 | 0.71 | 0.70 | 0.70 | 0.67 | 0.65 | 0.68 | 0.69 | 0.62 | 0.63 | 0.60 | 0.59 | **0.72** | **0.73** | **0.73** | **0.72** |
| | 20% | 0.67 | 0.67 | 0.68 | 0.66 | 0.64 | 0.64 | 0.64 | 0.64 | 0.60 | 0.60 | 0.58 | 0.58 | **0.71** | **0.74** | **0.70** | **0.73** |
| | 30% | 0.61 | 0.62 | 0.61 | 0.62 | 0.58 | 0.61 | 0.59 | 0.60 | 0.54 | 0.56 | 0.53 | 0.51 | **0.70** | **0.73** | 0.74 | **0.77** |
| | Mean | 0.69 | 0.7 | 0.69 | 0.69 | 0.67 | 0.67 | 0.67 | 0.67 | 0.62 | 0.63 | 0.61 | 0.60 | 0.7 | 0.73 | 0.72 | 0.74 |
| | Standard Deviation | 0.06 | 0.05 | 0.05 | 0.06 | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.05 | 0.06 | 0.07 | 0.02 | 0.01 | 0.02 | 0.02 |
| Hyperplane dataset | 5% | **0.67** | **0.69** | **0.70** | **0.71** | 0.61 | 0.64 | 0.64 | 0.66 | 0.56 | 0.57 | 0.58 | 0.59 | 0.64 | 0.67 | 0.69 | 0.69 |
| | 10% | **0.64** | **0.66** | 0.66 | 0.67 | 0.57 | 0.58 | 0.59 | 0.60 | 0.51 | 0.51 | 0.52 | 0.44 | 0.63 | 0.65 | **0.67** | **0.68** |
| | 15% | 0.61 | 0.62 | 0.63 | 0.63 | 0.52 | 0.54 | 0.54 | 0.54 | 0.46 | 0.47 | 0.47 | 0.48 | **0.62** | **0.64** | **0.66** | **0.66** |
| | 20% | 0.58 | 0.59 | 0.59 | 0.49 | 0.48 | 0.48 | 0.44 | 0.49 | 0.42 | 0.43 | 0.43 | 0.44 | **0.60** | **0.62** | **0.64** | 0.65 |
| | 30% | 0.51 | 0.52 | 0.51 | 0.52 | 0.40 | 0.40 | 0.40 | 0.40 | 0.36 | 0.36 | 0.37 | 0.37 | **0.58** | **0.69** | **0.61** | **0.62** |
| | Mean | 0.60 | 0.62 | 0..62 | 0.60 | 0.52 | 0.53 | 0.52 | 0.54 | 0.46 | 0.49 | 0.47 | 0.46 | 0.61 | 0.65 | 0.65 | 0.66 |
| | Standard Deviation | 0.06 | 0.07 | 0.07 | 0.1 | 0.08 | 0.09 | 0.1 | 0.1 | 0.08 | 0.08 | 0.08 | 0.08 | 0.02 | 0.03 | 0.03 | 0.03 |
| SEA dataset | 5% | **0.66** | 0.68 | **0.68** | **0.68** | 0.59 | 0.62 | 0.63 | 0.64 | 0.62 | 0.63 | 0.64 | 0.65 | 0.62 | 0.66 | 0.68 | 0.68 |
| | 10% | **0.62** | 0.64 | 0.64 | 0.65 | 0.54 | 0.57 | 0.58 | 0.59 | 0.56 | 0.58 | 0.60 | 0.60 | 0.59 | 0.63 | **0.65** | **0.67** |
| | 15% | **0.58** | 0.60 | 0.61 | 0.62 | 0.49 | 0.52 | 0.53 | 0.53 | 0.52 | 0.54 | 0.55 | 0.56 | 0.56 | 0.60 | **0.62** | **0.62** |
| | 20% | **0.54** | 0.56 | 0.56 | 0.55 | 0.45 | 0.46 | 0.46 | 0.47 | 0.47 | 0.49 | 0.49 | 0.50 | 0.53 | 0.57 | **0.59** | **0.60** |
| | 30% | **0.50** | 0.45 | 0.46 | 0.43 | 0.36 | 0.35 | 0.37 | 0.37 | 0.39 | 0.39 | 0.39 | 0.40 | 0.48 | 0.48 | **0.53** | **0.54** |
| | Mean | 0.58 | 0.59 | 0.59 | 0.59 | 0.49 | 0.50 | 0.51 | 0.52 | 0.51 | 0.53 | 0.53 | 0.54 | 0.56 | 0.59 | 0.61 | 0.62 |
| | Standard Deviation | 0.06 | 0.09 | 0.08 | 0.1 | 0.09 | 0.1 | 0.1 | 0.1 | 0.09 | 0.09 | 0.1 | 0.1 | 0.05 | 0.07 | 0.06 | 0.06 |

2000 data chunks. For 1000 and 500 data chunks, the results vary, but the difference is very negligible. Figure 7 implies that chunk size of chunk-based learning in data stream has no effect on FAIA. The effect of previous data chunk on the proposed FAIA is also considered in Fig. 8 for Electricity market dataset. For comparing the results by using the minority instances of previous chunk, Algorithm 3 is considered. Here, $K_p$ stores previous data chunk's minority instances. When the new chunk arrives, the previous chunk's minority class data are used to balance the current chunk at the step 4 of Algorithm 3. The process of using previous chunk is presented in Algorithm 3. Finally, Fig. 8 presents that when FAIA considers previous data chunk with different missing values, it always provides better result because of the missing values. The missing values of previous chunks mislead the current processing chunk, and this is the main reason that the previous chunk is not considered in FAIA. FAIA provides convincing output for AUC values because of imputing missing values in imbalanced information when considering data stream.

Table 4 shows the F-Measure, and its mean and standard deviation values with different chunk sizes and rates of missing data. For Electricity market dataset, FAIA continuously provides higher results than other existing methods. For Weather dataset, Hyperplane dataset, and SEA dataset, FAIA gives better or sometimes slightly lower result. For example, in Weather dataset where chunk size is 1500 and missing value is 20%, the best result is 0.70 for SERA; 0.68, FAIA. Furthermore, FAIA is a one pass algorithm and does

**Algorithm 3:** Fuzzy adaptive imputation approach using previous chunk

**Input:** $S$ = data stream, $B_m$ = current data chunk, $f$ = percentage of oversampling, $K_p$ = previous chunk's minority class data
**Output:** Predicted labels of testing sets of data in the dataset

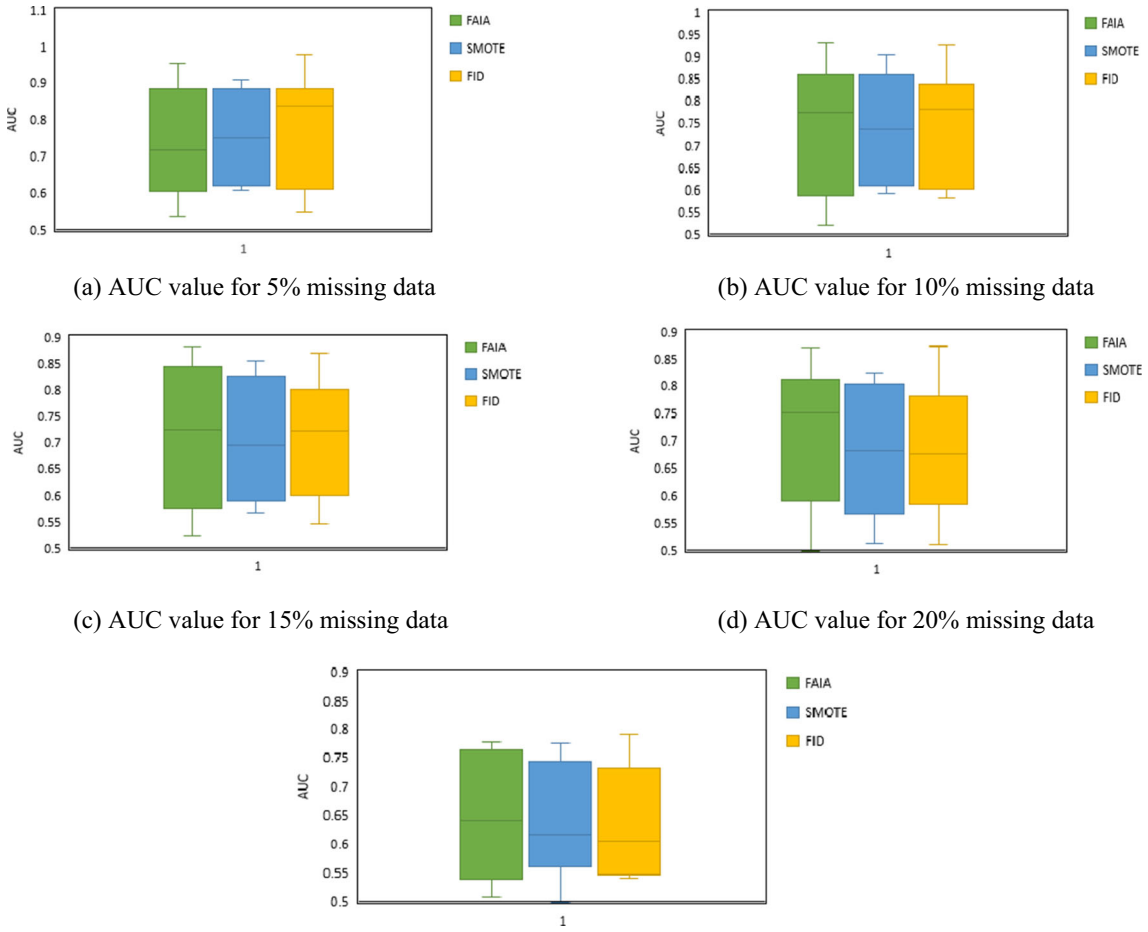
**Step 1:** ***for all current*** data chunk $B_m \in S$ ***do***
**Step 2:** split $B_m$ into $P_m$ *and* $N_m$ which are positive and negative data in the current chunk, respectively.
**Step 3:** find number of synthetic data needed to evaluate using $s_m = (|N_m| - |P_m|) \times f$
**Step 4:** add $K_p$ number of instances with $B_m$ to balance the chunk and compute the remaining number of required instances $r_m$
**Step 5:** assign $r_m$ number of missing values in the training data and call Algorithm 2 to impute missing data
**Step 6:** predict the label for each testing instance using C 4.5 classification algorithm.
**Step 7:** $K_p = N_m$
**Step 8:** ***end for***

not consider the previous information (see RMSE and AUC comparisons from Figs. 2, 3, 4, 5, and 6). Therefore, FAIA provides overall good result than existing methods with missing values.

Table 5 represents the G-mean with different rate of missing data. It also represents its mean and standard deviation values with different chunk sizes. As same as F-measure, for electricity market dataset, FAIA continuously provides better

**Table 5** G-mean

| Data Name | Percentage of Missing Value | FAIA | | | | FID [3] | | | | SMOTE [2] | | | | SERA [25] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Chunk Size | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 | 500 | 1000 | 1500 | 2000 |
| Electricity market dataset | 5% | **0.87** | **0.85** | **0.85** | **0.85** | 0.86 | 0.85 | 0.84 | 0.84 | 0.66 | 0.78 | 0.78 | 0.78 | 0.65 | 0.65 | 0.78 | 0.65 |
| | 10% | **0.83** | **0.81** | **0.80** | **0.80** | 0.82 | 0.80 | 0.80 | 0.79 | 0.74 | 0.73 | 0.74 | 0.73 | 0.70 | 0.64 | 0.64 | 0.67 |
| | 15% | **0.79** | **0.77** | **0.76** | **0.76** | 0.78 | 0.75 | 0.75 | 0.74 | 0.70 | 0.68 | 0.69 | 0.68 | 0.63 | 0.62 | 0.63 | 0.64 |
| | 20% | **0.76** | **0.74** | **0.73** | **0.71** | 0.74 | 0.71 | 0.70 | 0.69 | 0.67 | 0.65 | 0.65 | 0.65 | 0.60 | 0.62 | 0.62 | 0.61 |
| | 30% | **0.70** | **0.68** | **0.66** | **0.66** | 0.67 | 0.64 | 0.62 | 0.61 | 0.63 | 0.60 | 0.60 | 0.60 | 0.59 | 0.58 | 0.58 | 0.58 |
| | Mean | 0.79 | 0.77 | 0.76 | 0.76 | 0.77 | 0.75 | 0.74 | 0.73 | 0.68 | 0.69 | 0.69 | 0.69 | 0.63 | 0.62 | 0.65 | 0.63 |
| | Standard Deviation | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.08 | 0.09 | 0.09 | 0.04 | 0.07 | 0.07 | 0.07 | 0.04 | 0.03 | 0.08 | 0.04 |
| Weather dataset | 5% | 0.65 | 0.66 | 0.66 | 0.66 | 0.65 | 0.66 | **0.66** | 0.65 | **0.66** | **0.66** | 0.66 | **0.66** | 0.52 | 0.59 | 0.58 | 0.57 |
| | 10% | 0.63 | **0.66** | 0.65 | **0.66** | **0.65** | 0.65 | **0.66** | 0.65 | 0.63 | 0.65 | 0.64 | 0.64 | 0.52 | 0.58 | 0.58 | 0.57 |
| | 15% | **0.64** | 0.64 | **0.64** | 0.64 | 0.63 | **0.64** | 0.64 | **0.65** | 0.62 | 0.63 | 0.61 | 0.61 | 0.53 | 0.55 | 0.57 | 0.51 |
| | 20% | 0.62 | **0.63** | **0.63** | **0.62** | **0.62** | 0.62 | 0.62 | 0.61 | 0.60 | 0.61 | 0.60 | 0.60 | 0.48 | 0.54 | 0.55 | 0.54 |
| | 30% | 0.58 | **0.60** | **0.59** | **0.59** | **0.58** | 0.60 | 0.58 | 0.57 | 0.56 | 0.57 | 0.56 | 0.55 | 0.45 | 0.54 | 0.52 | 0.46 |
| | Mean | 0.62 | 0.64 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.61 | 0.62 | 0.61 | 0.61 | 0.62 | 0.5 | 0.56 | 0.56 | 0.53 |
| | Standard Deviation | 0.03 | 0.02 | 0.03 | 0.03 | 0.03 | 0.02 | 0.03 | 0.03 | 0.05 | 0.04 | 0.06 | 0.04 | 0.03 | 0.02 | 0.03 | 0.04 |
| Hyperplane dataset | 5% | **0.69** | **0.71** | **0.72** | **0.72** | 0.65 | 0.67 | 0.67 | 0.69 | 0.61 | 0.63 | 0.64 | 0.64 | 0.64 | 0.58 | 0.69 | 0.69 |
| | 10% | **0.67** | **0.68** | **0.69** | **0.70** | 0.62 | 0.63 | 0.64 | 0.64 | 0.58 | 0.58 | 0.59 | 0.53 | 0.62 | 0.62 | 0.67 | 0.67 |
| | 15% | **0.65** | **0.66** | **0.66** | **0.69** | 0.59 | 0.60 | 0.60 | 0.60 | 0.54 | 0.55 | 0.55 | 0.56 | 0.60 | 0.64 | 0.65 | 0.66 |
| | 20% | **0.63** | **0.64** | **0.64** | **0.64** | 0.55 | 0.56 | 0.57 | 0.56 | 0.51 | 0.52 | 0.53 | 0.53 | 0.59 | 0.65 | 0.63 | 0.64 |
| | 30% | **0.58** | **0.59** | 0.58 | 0.59 | 0.50 | 0.49 | 0.50 | 0.50 | 0.46 | 0.47 | 0.48 | 0.47 | 0.55 | 0.67 | **0.58** | **0.60** |
| | Mean | 0.64 | 0.66 | 0..66 | 0.67 | 0.58 | 0.59 | 0.60 | 0.54 | 0.54 | 0.55 | 0.56 | 0.55 | 0.60 | 0.63 | 0.64 | 0.65 |
| | Standard Deviation | 0.04 | 0.05 | 0.05 | 0.05 | 0.06 | 0.07 | 0.07 | 0.1 | 0.06 | 0.06 | 0.06 | 0.06 | 0.03 | 0.03 | 0.04 | 0.03 |
| SEA dataset | 5% | **0.72** | **0.74** | **0.74** | **0.74** | 0.67 | 0.69 | 0.70 | 0.71 | 0.68 | 0.69 | 0.70 | 0.70 | 0.68 | 0.71 | 0.74 | 0.74 |
| | 10% | **0.69** | **0.69** | 0.71 | 0.71 | 0.62 | 0.65 | 0.65 | 0.66 | 0.63 | 0.65 | 0.67 | 0.69 | 0.66 | 0.69 | **0.71** | **0.73** |
| | 15% | **0.66** | **0.67** | 0.68 | 0.69 | 0.58 | 0.60 | 0.61 | 0.61 | 0.60 | 0.62 | 0.62 | 0.63 | 0.63 | 0.67 | **0.69** | **0.69** |
| | 20% | **0.63** | 0.64 | 0.64 | 0.63 | 0.55 | 0.56 | 0.56 | 0.57 | 0.56 | 0.58 | 0.58 | 0.59 | 0.61 | **0.64** | **0.67** | **0.67** |
| | 30% | **0.59** | 0.55 | 0.55 | 0.54 | 0.47 | 0.47 | 0.48 | 0.48 | 0.50 | 0.50 | 0.50 | 0.51 | 0.56 | **0.57** | **0.62** | **0.62** |
| | Mean | 0.66 | 0.66 | 0.66 | 0.66 | 0.58 | 0.59 | 0.6 | 0.61 | 0.59 | 0.60 | 0.61 | 0.62 | 0.63 | 0.66 | 0.69 | 0.69 |
| | Standard Deviation | 0.05 | 0.07 | 0.07 | 0.08 | 0.08 | 0.09 | 0.08 | 0.09 | 0.07 | 0.07 | 0.08 | 0.08 | 0.05 | 0.05 | 0.05 | 0.05 |




(a) AUC value for 5% missing data

(b) AUC value for 10% missing data

(c) AUC value for 15% missing data

(d) AUC value for 20% missing data

(e) AUC value for 30% missing data

**Fig. 9** Comparison of the existing methods with the proposed FAIA using AUC values

result. For weather and hyperplane dataset, the same scenario can be noticed in Table 5. But for SEA dataset, the result is little changed when the chunk size and missing rate are increased because of the third attribute of the dataset is considered as noise [4]. These noises distract FAIA to compute correct missing values. Although FAIA not have much differences to impute correct missing values for SEA dataset, it proves better result for remaining datasets than other existing methods.

**C. offline data with missing values** This section measures the stability of the proposed FAIA for imputing missing values in imbalanced information when considering offline data.

C.1) **Datasets, Performance Measure, and Experimental Setting:** Datasets from Table 1 are considered for this experiment. For measuring performance, AUC, F-Measure and G-mean are used. FID [3] and SMOTE [2] are compared with the proposed FAIA. SMOTE [2] is considered because it is the most basic method and still popular in data balancing. In recent years, scant works consider missing data and imbalanced data problem simultaneously. DT C4.5 is considered for classifying data with five-fold cross validation. Similar to the existing methods, all outputs are computed from the average values of 10 times the individual measurements.

C.2) **Results for Imbalanced Data with Missing Values:** Figs. 9(a), (b), (c), (d), and (e) show the AUC values for 5%, 10%, 15%, 20%, and 30% of missing information, respectively. Eleven datasets (Table 1) are considered to represent the result. Three methods, namely, proposed FAIA, FID [3], and SMOTE [2] are compared with the different rates of missing information. Figure 9 (a) shows that the median values of the proposed FAIA, FID, and SMOTE are 0.7184, 0.8366, and 0.7506, respectively. FID produces 0.1182 times higher than FAIA. In Fig. 9(b) with 10% of missing information, the median value of FAIA is 0.7737, which is lower than that of FID (i.e., 0.007) and a bit higher than that of SMOTE (i.e., 0.037). Similarly, in Fig. 9(c) with 15% missing values, the result of FAIA is 0.0034 and 0.031 times better than those of FID and SMOTE,

**Table 6** F-Measure

| Data Name | Percentage of missing value | FAIA | FID [3] | SMOTE [2] |
|---|---|---|---|---|
| KC1 | 5% | **0.901661** | 0.893213 | 0.854078 |
| | 10% | **0.905302** | 0.897548 | 0.825437 |
| | 15% | **0.909697** | 0.896796 | 0.781785 |
| | 20% | **0.912369** | 0.897557 | 0.764599 |
| | 30% | **0.912899** | 0.905333 | 0.709162 |
| | Mean | 0.908386 | 0.898089 | 0.787012 |
| | Standard Deviation | 0.004814 | 0.00443 | 0.056046 |
| MC2 | 5% | **0.714476** | 0.680676 | 0.646911 |
| | 10% | **0.699066** | 0.666569 | 0.617164 |
| | 15% | 0.685902 | **0.706546** | 0.568212 |
| | 20% | **0.703901** | 0.648765 | 0.487064 |
| | 30% | 0.585715 | **0.589769** | 0.489871 |
| | Mean | 0.677812 | 0.658465 | 0.561844 |
| | Standard Deviation | 0.052497 | 0.043829 | 0.072645 |
| PC1 | 5% | **0.946866** | 0.948048 | 0.931073 |
| | 10% | 0.954358 | **0.955685** | 0.941155 |
| | 15% | **0.955183** | 0.950943 | 0.943069 |
| | 20% | **0.956036** | 0.952647 | 0.946415 |
| | 30% | **0.959287** | 0.953913 | 0.954249 |
| | Mean | 0.954346 | 0.952247 | 0.943192 |
| | Standard Deviation | 0.00458 | 0.002919 | 0.008421 |
| PC3 | 5% | 0.893016 | **0.920556** | 0.89656 |
| | 10% | 0.873522 | **0.919858** | 0.89685 |
| | 15% | 0.855824 | **0.925925** | 0.907204 |
| | 20% | 0.819034 | **0.930417** | 0.908337 |
| | 30% | 0.770987 | **0.93365** | 0.932276 |
| | Mean | 0.842477 | 0.926081 | 0.908245 |
| | Standard Deviation | 0.048365 | 0.006028 | 0.014534 |
| Sonar | 5% | **0.722246** | 0.682703 | 0.616333 |
| | 10% | 0.697351 | **0.703726** | 0.525323 |
| | 15% | **0.679476** | 0.666125 | 0.520357 |
| | 20% | **0.637971** | 0.592841 | 0.458558 |
| | 30% | **0.595566** | 0.53209 | 0.377083 |
| | Mean | 0.666522 | 0.635497 | 0.499531 |
| | Standard Deviation | 0.050183 | 0.071299 | 0.08862 |
| Vehicle | 5% | **0.919437** | 0.911649 | 0.88956 |
| | 10% | **0.883415** | 0.880158 | 0.842643 |
| | 15% | **0.845192** | **0.847141** | 0.806381 |
| | 20% | **0.791425** | **0.790189** | 0.746975 |
| | 30% | 0.730564 | **0.76896** | 0.695316 |
| | Mean | 0.834007 | 0.839619 | 0.796175 |
| | Standard Deviation | 0.074797 | 0.059842 | 0.076775 |
| Wine | 5% | **0.962672** | 0.845487 | 0.951243 |
| | 10% | **0.906128** | 0.845332 | 0.898916 |
| | 15% | **0.89253** | 0.850445 | 0.871764 |
| | 20% | **0.88167** | 0.807538 | 0.879386 |
| | 30% | 0.77102 | 0.647633 | **0.782744** |
| | Mean | 0.882804 | 0.799287 | 0.876811 |
| | Standard Deviation | 0.069847 | 0.086514 | 0.061052 |

**Table 6** (continued)

| Data Name | Percentage of missing value | FAIA | FID [3] | SMOTE [2] |
|---|---|---|---|---|
| Glass (1) | 5% | **0.926719** | 0.908813 | 0.925696 |
| | 10% | **0.934214** | 0.913227 | 0.914068 |
| | 15% | **0.932868** | 0.904347 | 0.92797 |
| | 20% | 0.931968 | **0.939177** | 0.928781 |
| | 30% | 0.929916 | **0.945505** | 0.929161 |
| | Mean | 0.931137 | 0.922214 | 0.925135 |
| | Standard Deviation | 0.002923 | 0.018774 | 0.006331 |
| Glass (2) | 5% | 0.968181 | 0.960946 | **0.972345** |
| | 10% | **0.972514** | 0.95271 | 0.969152 |
| | 15% | 0.974805 | 0.945187 | **0.975589** |
| | 20% | **0.968531** | 0.934911 | 0.964275 |
| | 30% | **0.962293** | 0.948423 | 0.952064 |
| | Mean | 0.969265 | 0.948435 | 0.966685 |
| | Standard Deviation | 0.004785 | 0.009595 | 0.009179 |
| *Page Block (1)* | 5% | **0.978795** | 0.966411 | 0.978266 |
| | 10% | 0.926197 | 0.957157 | **0.972727** |
| | 15% | 0.896985 | 0.954063 | **0.966943** |
| | 20% | 0.953022 | 0.958774 | **0.966159** |
| | 30% | 0.79736 | 0.948319 | **0.963722** |
| | Mean | 0.910472 | 0.956945 | 0.969563 |
| | Standard Deviation | 0.070182 | 0.006628 | 0.005881 |
| *Page Block (2)* | 5% | 0.95351 | 0.965067 | **0.977017** |
| | 10% | 0.923556 | 0.957125 | **0.973769** |
| | 15% | 0.900295 | 0.951785 | **0.971437** |
| | 20% | 0.864372 | 0.952521 | **0.966971** |
| | 30% | 0.802897 | 0.944632 | **0.959738** |
| | Mean | 0.888926 | 0.954226 | 0.969786 |
| | Standard Deviation | 0.058102 | 0.007531 | 0.006704 |

respectively. For Fig. 9(d) with 20% missing values, FAIA is 0.074 and 0.067 times better than FID and SMOTE, respectively. Finally, for Fig. 9(e) with 30% missing values, FAIA is 0.037 and 0.025 times better than FID and SMOTE, respectively. From this observation, FAIA provides lower AUC values with lesser missing data, and it performs continuously better when the rate of missing data increases in imbalanced data. In addition, the number of attributes affect FAIA. For example, for Sonar dataset with 60 attributes and 5% missing values, the AUC value of FAIA is 0.0305 times better than that of FID. Similarly, for MC2 with 39 attributes, the AUC value of FAIA is 0.0177 times better than that of FID; PC3 with 37 attributes, 0.01288. Similarly, for Vehicle dataset with 18 attributes, the AUC value of FAIA is 0.0031 times better than that of FID; wine dataset with 14 attributes, 0.02167. This survey summarizes that for FAIA, when the number of attributes increases, the possibility of AUC also increases for most of the datasets. However, FAIA provides large inter quartile range because it considers inter-relation among attributes which is not considered by other existing methods. All attributes are not important and there may have some noises. As a result, some attributes of some datasets mislead FAIA to find closer values for missing data. For this reason, some datasets which are considered in this paper provides little lower AUC results as compare to others. Therefore, FAIA shows the large inter quartile range. For data stream, the same variance is created for dataset SEA because of noise attribute [4]. Though FAIA provides large inter quartile range as compare to other existing methods, it gives higher median values when the missing rate is increased.

**Table 7** G-mean

| Data Name | Percentage of missing value | FAIA | FID [3] | SMOTE [2] |
|---|---|---|---|---|
| KC1 | 5% | 0.520995 | **0.64987** | 0.553865 |
| | 10% | 0.457171 | **0.635594** | 0.55941 |
| | 15% | 0.428984 | **0.643185** | 0.559197 |
| | 20% | 0.426554 | **0.615198** | 0.511429 |
| | 30% | 0.314589 | **0.592965** | 0.477294 |
| | Mean | 0.429659 | 0.627362 | 0.532239 |
| | Standard Deviation | 0.07475 | 0.023216 | 0.03669 |
| MC2 | 5% | 0.552558 | **0.598732** | 0.547646 |
| | 10% | 0.572484 | **0.603087** | 0.57062 |
| | 15% | 0.571521 | 0.548164 | **0.624973** |
| | 20% | **0.616786** | 0.514744 | 0.552272 |
| | 30% | 0.528152 | 0.523538 | **0.53366** |
| | Mean | 0.5683 | 0.557653 | 0.565834 |
| | Standard Deviation | 0.032552 | 0.041373 | 0.035603 |
| PC1 | 5% | 0.485819 | **0.520938** | 0.480918 |
| | 10% | 0.483472 | 0.492235 | **0.523708** |
| | 15% | 0.414816 | **0.419219** | 0.32646 |
| | 20% | **0.448403** | 0.361621 | 0.297424 |
| | 30% | **0.33552** | 0.269397 | 0.293483 |
| | Mean | 0.433606 | 0.412682 | 0.384399 |
| | Standard Deviation | 0.06206 | 0.101524 | 0.109442 |
| PC3 | 5% | **0.952815** | 0.851174 | 0.930438 |
| | 10% | **0.888814** | 0.844598 | 0.725817 |
| | 15% | **0.876939** | 0.845009 | 0.862951 |
| | 20% | **0.866662** | 0.810332 | 0.86722 |
| | 30% | 0.765556 | 0.677165 | **0.776138** |
| | Mean | 0.870157 | 0.805656 | 0.832513 |
| | Standard Deviation | 0.067417 | 0.073602 | 0.081059 |
| Sonar | 5% | **0.566084** | 0.541647 | 0.511021 |
| | 10% | **0.530187** | 0.51078 | 0.479096 |
| | 15% | **0.524687** | 0.448969 | 0.474558 |
| | 20% | **0.536496** | 0.458657 | 0.435151 |
| | 30% | **0.547373** | 0.24784 | 0.355132 |
| | Mean | 0.540965 | 0.441579 | 0.490992 |
| | Standard Deviation | 0.016379 | 0.114764 | 0.059977 |
| Vehicle | 5% | **0.878328** | 0.869622 | 0.87502 |
| | 10% | **0.845088** | 0.821513 | 0.833254 |
| | 15% | **0.809449** | 0.791788 | 0.800898 |
| | 20% | **0.757562** | 0.745789 | 0.741353 |
| | 30% | 0.71422 | 0.708258 | **0.730489** |
| | Mean | 0.800929 | 0.787394 | 0.796203 |
| | Standard Deviation | 0.065978 | 0.063124 | 0.061102 |
| Wine | 5% | **0.712736** | 0.643648 | 0.681449 |
| | 10% | 0.69638 | 0.578824 | **0.717594** |
| | 15% | **0.696771** | 0.580022 | 0.689847 |
| | 20% | **0.651675** | 0.532696 | 0.634674 |
| | 30% | **0.621771** | 0.467272 | 0.580919 |
| | Mean | 0.675867 | 0.560492 | 0.660897 |
| | Standard Deviation | 0.037844 | 0.065374 | 0.053757 |

**Table 7** (continued)

| Data Name | Percentage of missing value | FAIA | FID [3] | SMOTE [2] |
|---|---|---|---|---|
| Glass (1) | 5% | 0.243899 | **0.519502** | 0.266572 |
| | 10% | 0.154319 | **0.411196** | 0.359968 |
| | 15% | 0.175633 | **0.527062** | 0.270177 |
| | 20% | 0.097425 | **0.12075** | 0.016574 |
| | 30% | 0.134623 | 0.044202 | **0.259154** |
| | Mean | 0.16118 | 0.324542 | 0.234489 |
| | Standard Deviation | 0.054471 | 0.227296 | 0.128592 |
| Glass (2) | 5% | 0.615588 | **0.79565** | 0.76212 |
| | 10% | 0.626266 | **0.811098** | 0.613542 |
| | 15% | 0.633582 | **0.660986** | 0.622619 |
| | 20% | 0.623514 | **0.671275** | 0.514604 |
| | 30% | **0.451219** | 0.444006 | 0.297995 |
| | Mean | 0.590034 | 0.676603 | 0.562176 |
| | Standard Deviation | 0.077866 | 0.147169 | 0.171991 |
| *Page Block (1)* | 5% | 0.871834 | **0.906184** | 0.881816 |
| | 10% | **0.85877** | 0.854245 | 0.820031 |
| | 15% | **0.83501** | 0.810476 | 0.762257 |
| | 20% | **0.813455** | 0.791213 | 0.743625 |
| | 30% | **0.759722** | 0.728221 | 0.674319 |
| | Mean | 0.827758 | 0.818061 | 0.77641 |
| | Standard Deviation | 0.044132 | 0.066941 | 0.078599 |
| *Page Block (2)* | 5% | 0.885848 | **0.894278** | 0.868918 |
| | 10% | **0.857029** | 0.855693 | 0.825274 |
| | 15% | **0.844625** | 0.817668 | 0.780635 |
| | 20% | **0.792506** | 0.784831 | 0.761721 |
| | 30% | **0.752511** | 0.719208 | 0.66362 |
| | Mean | 0.826504 | 0.814336 | 0.780034 |
| | Standard Deviation | 0.053415 | 0.067142 | 0.077193 |

Table 6 tabulates F-measure values and its mean and standard deviation values. For most of the datasets, the proposed FAIA provides better or almost closer results. However, for three datasets (PC3, Page Block (1), and Page Block (2) datasets) in Table 6, FAIA provides lower F-measure than other methods. The two reason may be less interrelationship between attributes or imbalance ratio. FAIA considers the interrelationship between attributes, and adaptive measure is used to get a closer value for missing information, thus AUC results are convincing. Furthermore, imbalance ratio is basically the ratio of the number of minority class and majority class elements [3]. For the PC3, Page Block (1) and Page Block (2) datasets, the imbalance ratio is 0.114 because PC3 has 160 elements in minority class and 1403 elements in majority class. For Page Block (1), the elements in minority and majority class are 561 and 4913, respectively, which is the same for Page Block (2). Nevertheless, for the remaining datasets, FAIA provides better results than FID and SMOTE because the imbalance ratio is either greater or less but not equal to 0.114. For example, the imbalance ratio of Sonar dataset is 0.874 (minority class and majority class elements are 97 and 111, respectively), which is greater than 0.114 and that of PC1 dataset is 0.075 (minority class and majority class elements are 77 and 1032, respectively), which is less than 0.114. If imbalance ratio denoted as $\varepsilon$, then overfitted criteria $\psi$ is written as

$$\psi = \begin{cases} if\ \varepsilon = 11 \approx 12\%, then\ overfitted \\ otherwise\ not\ overfitted \end{cases} \tag{18}$$

Although for AUC, this imbalance ratio is not that much important as F-measure, FAIA presents better output when the rate of missing data increases.

Table 7 presents the G-mean values with different range of missing values. Its mean and standard deviation are also

considered in this Table 7. According to this Table 7, FAIA provides better or little lower results than other existing methods. If we consider the mean values then it can be noticed that FAIA provides better result except KC1, Glass (1) and Glass (2) datasets because of interrelationship among attributes of datasets (as discussed before). However, for the most of the datasets by considering F-measure and AUC values, FAIA provides higher result than other existing methods. Moreover, unlike G-mean of offline data (Table 7), the proposed FAIA is more stable in G-mean values to data streaming scenario (Table 5) for missing value imputation in imbalanced information.

## 5 Discussion

After a lengthy discussion about the proposed FAIA, FAIA can be concluded as an adaptive method due to its stability on both offline and data stream fields. It can also be used separately for imputing missing data because it constantly provides less RMSE value for all used datasets in section a. for offline imbalance data with missing values, although the average AUC value for 5% missing data of FAIA is less than those of FID [3] and SMOTE [2], the proposed method provides better AUC values for 10%, 15%, 20%, and 30% missing values. For F-measure values, FAIA provides overall better results for 8 out of 11 datasets. For offline imbalanced data with missing values, FAIA can be a better choice when the rate of missing data is high. For imbalanced data stream with missing data, the stability of the method can be shown as well as in the offline field. For AUC values, FAIA obtains better results than other methods except for the SEA dataset due to its noise. The main objective of the proposed method is to represent the effect of missing values in imbalanced data stream. This study aims to show how FAIA can handle the missing data problem in imbalanced data stream field. The results indicate that FAIA achieves this objective

## 6 Conclusion

Missing information largely affects pattern recognition. Imbalance problem also increases more with missing values. For data stream, this problem is more difficult because, for each chunk, it is difficult to classify imbalanced data with missing values. The proposed FAIA uses fuzzy decomposition method with adaptive imputation approach to determine the interrelationship among instances. FAIA improves the evaluation efficiency of missing values and the accuracy of data balancing with data stream (i.e., online data) and offline data. For measuring the capability of FAIA, performance measurement criteria, namely, AUC, F-measure, G-mean and RMSE are applied to 11 datasets for offline data and four datasets for data stream. The output of the proposed FAIA outperforms those of the existing methods. In addition, KNN is compared using RMSE measurement, and FAIA produces better result. However, FAIA is a numerical data imputation method. Without the interrelationship among attributes, it may not provide good result in some cases. Nevertheless, the FAIA provides good result for datasets in which interrelationship exists among attributes. Using this technique, most of the imbalanced information with missing data in data stream or offline data can lead to excellent classification performance.

Further improvements for the proposed FAIA can be explored. First, imputation of categorical missing values is not considered in the existing methods. Applying fuzzy system in categorical data is quite challenging, so another new categorical data imputation approach can be introduced to deal with it. Second, FAIA uses binary classification. Multi-classification can be used to compute more accurate AUC, G-mean and F-measure by individual measurement of each class with respect to majority class represented in dataset. Some multi-classification methods are available for imbalanced data, but these cannot solve the missing data problem. First measuring missing data for each class and then balancing information of all classes can be the research for multi-classification method.

**Acknowledgements** This research work was supported by Special Grant of ICT Division (Ministry of Posts, Telecommunications and Information Technology), Bangladesh, Grant No. 56.00.0000.028.20.004.20-333. The authors would like to acknowledge Ministry of Higher Education, Malaysia for their partial support through the Fundamental Research Grant Scheme (FRGS), under Grant 203/PELECT/6071398.

## References

1. Pan R, Yang T, Cao J, Lu K, Zhang Z (2015) Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. Appl Intell 43(3):614–632
2. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
3. Liu S, Zhang J, Xiang Y, Zhou W (Dec. 2017) Fuzzy-based information decomposition for incomplete and imbalanced data learning. *IEEE Trans Fuzzy Syst* 25(6):1476–1490
4. Ren S, Zhu W, Liao B, Li Z, Wang P, Li K, Chen M, Li Z (2019) Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning. Knowl-Based Syst 163:705–722
5. Lu Y, Cheung Y-M, Tang YY (2019) Adaptive Chunk-Based Dynamic Weighted Majority for Imbalanced Data Streams With Concept Drift." *IEEE Transactions on Neural Networks and Learning Systems*
6. Ng WWY, Zhang J, Lai CS, Pedrycz W, Lai LL, Wang X (2018) Cost-sensitive weighting and imbalance-reversed bagging for streaming imbalanced and concept drifting in electricity pricing classification. *IEEE Trans Ind Inf* 15(3):1588–1597

7. Zhu X, Zhang S, Jin Z, Zhang Z, Zhuoming X (2010) Missing value estimation for mixed-attribute data sets. IEEE Trans Knowl Data Eng 23(1):110–121
8. Little RJA, Rubin DB (2019) *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons
9. Lim P, Goh CK, Tan KC (2016) Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. IEEE Trans Cybern 47(9):2850–2861
10. Yoon J, Zame WR, van der Schaar M (May 2019) Estimating missing data in temporal data streams using multi-directional recurrent neural networks. IEEE Trans Biomed Eng 66(5):1477–1490
11. Zhang S (2012) Nearest neighbor selection for iteratively kNN imputation. J Syst Softw 85(11):2541–2552
12. García-Laencina PJ, Sancho-Gómez JL, Figueiras-Vidal AR, Verleysen M (2009) K nearest neighbours with mutual information for simultaneous classification and missing data imputation. Neurocomputing 72(7–9):1483–1493
13. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
14. Dataset Name: Iris dataset. https://archive.ics.uci.edu/ml/datasets/iris. Retrieved on January, 2021
15. Wang S, Minku LL, Yao X (2018) A systematic study of online class imbalance learning with concept drift. IEEE Trans Neural Netw Learn Syst 29(10):4802–4821
16. Folguera L, Zupan J, Cicerone D, Magallanes JF (2015) Self-organizing maps for imputation of missing data in incomplete data matrices. Chemom Intell Lab Syst 143:146–151
17. Cristianini N, Shawe-Taylor J (2004) Support vector machines and other kernel-based learning methods, Cambridge
18. Brás LP, Menezes JC (2007) Improving cluster-based missing value estimation of DNA microarray data. Biomol Eng 24(2):273–282
19. Brzezinski D, Stefanowski J, Susmaga R, Szczech I (Aug. 2020) On the dynamics of classification measures for imbalanced and streaming data. IEEE Trans Neural Netw Learn Syst 31(8):2868–2878
20. Han H, Wang W-Y, Mao B-H (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing. Springer, Berlin, Heidelberg
21. Barua S, Islam MM, Yao X, Murase K (2012) MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans Knowl Data Eng* 26(2):405–425
22. Zhang C, Bi J, Xu S, Ramentol E, Fan G, Qiao B, Fujita H (2019) Multi-imbalance: an open-source software for multi-class imbalance learning. Knowl-Based Syst 174:137–143
23. Sun J, Li H, Fujita H, Binbin F, Ai W (2020) Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting. Information Fusion 54:128–144
24. Gao J, Fan W, Han J, Yu PS (2007) A general framework for mining concept-drifting data streams with skewed distributions. In *Proceedings of the* 2007 *siam international conference on data mining*, pp. 3–14. Society for Industrial and Applied Mathematics
25. Chen S, He H (2009) Sera: selectively recursive approach towards nonstationary imbalanced stream data mining. 2009 *International Joint Conference on Neural Networks*. IEEE
26. Chen S, He H, Li K, Desai S (2010) Musera: Multiple selectively recursive approach towards imbalanced stream data mining. In *The* 2010 *international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE
27. Ren S, Liao B, Zhu W, Li Z, Liu W, Li K (2018) The gradual resampling ensemble for mining imbalanced data streams with concept drift. Neurocomputing 286:150–166
28. Chongfu H (1997) Principle of information diffusion. Fuzzy Sets Syst 91(1):69–90
29. Datasets for analyzing the data streaming: Electricity market dataset, Weather dataset, Hyperplane dataset, SEA dataset. https://github.com/vlosing/driftDatasets, Retrieved on July, 2020
30. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C (Applications and Reviews)* 42(4):463–484
31. Arabmakki E, Kantardzic M (2017) SOM-based partial labeling of imbalanced data stream. Neurocomputing 262:120–133
32. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2011) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, no. 4 463–484
33. Ditzler G, Polikar R (2012) Incremental learning of concept drift from streaming imbalanced data. *IEEE Trans Knowl Data Eng* 25.10: 2283–2301