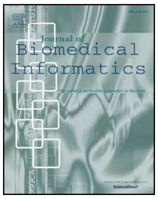
Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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



A hybrid of whale optimization and late acceptance hill climbing based imputation to enhance classification performance in electronic health records

Gayathri Nagarajan*, L.D. Dhinesh Babu

School of Information Technology and Engineering, VIT University, India

ARTICLE INFO

Keywords:
Missing data imputation
Whale optimization
Late acceptance hill climbing
Electronic health records
Machine learning

ABSTRACT

Electronic health records (EHR) are a major source of information in biomedical informatics. Yet, missing values are prominent characteristics of EHR. Prediction on dataset with missing values results in inaccurate inferences. Nearest neighbour imputation based on lazy learning approach is a proven technique for missing data imputation and is recognized as one among the top ten data mining algorithms due to its simplicity and understandability. But its performance is deteriorated due to the curse of dimensionality as unimportant features are likely to dominate. We address this problem by proposing a novel approach for feature weighting based on a hybrid of metaheuristic whale optimization algorithm (WOA) and local search late acceptance hill climbing algorithm (LAHCA) on nearest neighbour imputation method. Our proposed approach Metaheuristic and Local Search based Feature Weighted Nearest Neighbour Imputation (kNN + LAHCAWOA) also learns different k values for different test points. Our approach is tested on benchmark EHR datasets with three proven classifiers Support Vector Machines(SVM), Random forest(RF) and Deep neural networks(DNN). The results prove that kNN + LAHCAWOA is an effective imputation strategy and aids in improving the classification performance when compared with its competitor methods.

1. Introduction

Biomedical informatics is an interdisciplinary emerging field applied in the context of biomedicine [1]. It is the science of problem solving using biomedical data [2]. One of its applications is to derive inferences from huge biomedical data sets and take necessary actions in health care services. EHR is one among the major sources of biomedical data and is no exception to problems such as missing data, noise, etc. Elimination or proper handling of such information is a crucial step that directly impacts the accuracy of prediction. Hence there is a need to properly handle the missing data values in data preprocessing stage.

The reasons for missing data vary for different scenarios. It may be because of improper handling of data collection and samples, non-response, data entry error, etc [3]. It is always a big question for data scientists and analysts whether to handle the missing values or ignore them. This decision depends on the proportion of missingness and whether the missing variable is a dependent or independent variable [4]. It is generally accepted that dataset with missingness less than 5% does not require any treatment for missing values and complete case analysis may yield reasonable results. Missingness of 5–15% need to be handled with proper missing data handling mechanisms [5]. Missingness of 15–25% may be worked upon with suitable missing data handling mechanism whereas these methods may not always yield expected results for dataset with more than 25% missingness. Hence different techniques are adopted to treat missing values in different ways. Nearest neighbour imputation is one among the popular and efficient imputation techniques owing to its simplicity and understandability. Yet, its performance is deteriorated with an increase in the number of features. This happens because of the influence of unimportant features in identifying the neighbours. Our work (kNN + LAHCAWOA) proposes a novel method for obtaining feature weights based on a hybrid of metaheristic WOA and local search LAHCA and incorporates this weight on nearest neighbour imputation method. Although many approaches exist for missing data imputation, our approach is one among the few approaches that can work even when each observation or feature is having a missing value. Our approach also learns different number of neighbours for different test points and hence proves to improve the classification performance in comparison with its competitor approaches for well known classifiers including deep learning classification.

* Corresponding author.
E-mail address: gayunagarajan1083@gmail.com (G. Nagarajan).

https://doi.org/10.1016/j.jbi.2019.103190
Received 14 December 2018; Received in revised form 13 April 2019; Accepted 26 April 2019
Available online 02 May 2019

The rest of the paper is comprised of the following sections. Section 2 presents the related works, Section 3 discusses about the methods used in our approach, Section 4 explains our proposed method kNN + LAHCAWOA for imputation of missing data, Section 5 summarizes our experimentation framework, Section 6 winds up with conclusion.

2. Related work

There are several imputation techniques in use and no single imputation technique has been proven the best till now. Six different methods for missing data imputation – imputation by mean value, KNN algorithm, Fuzzy k means, Singular Value Decomposition (SVD), Bayesian principal component analysis (bPCA), Multiple imputation using chained equations (MICE) are compared based on different criteria – Root mean square error (RMSE), unsupervised classification error, supervised classification error, execution time in [3]. The paper concludes that there is no universal method applicable for all kinds of datasets. The classification techniques are categorized into three methods rule based learning, approximate models and lazy learning and tested few imputation methods with all the three categories of classifiers in [6]. This paper also concludes that there is no universal imputation method that performs well for all classifiers. Yet, certain imputation methods are found to cater well with specific predictive algorithms. Event covering imputation method yields better results for radial basis function networks [5]. A method is proposed in [7] that pairs the classifier with the imputation method based on the characteristics of missing values in the datasets.

Imputation methods like imputation with mean, imputation based on regression, etc. are based on statistics. The statistical based imputation methods suffer from their own limitations. Basic simple imputation technique involves the use of mean, median or mode to fill in the missing values. Several methods for imputation of missing values in electric vehicle charging data are compared and median and constant imputations are found to improve the predicted results in comparison with the other methods in this application [8]. Though such methods prove to be effective in certain cases the major limitation with these methods is that it ignores relationships [9]. Complete case analysis proves to be effective when the proportion of missingness is low but when the proportion of missingness is high, the method is found to be ineffective. Regression method preserves relationships but underestimates variability of missing values.

Multiple imputation technique is also widely used to fill in the missing values. It is used in matched case control studies [10]. Multiple imputation is used for prediction on the case study of children mental health initiative. Multiple imputation using MICE package in R is discussed in [11]. But multiple imputation is underutilized in health care sector because of its own limitations. Multiple imputation fails to provide proper theoretical justification in many instances. Inconsistencies might arise because of lack of proper joint distribution in fitting the condition models. Many interactions are required to preserve the associations in data [12]. Multiple imputation also suffers from computational complexity. These limitations of statistical techniques led to the application of machine learning algorithms like KNN, neural networks for imputation.

K-Nearest Neighbor algorithm is a commonly used machine learning technique that finds its use in imputation. There are few works carried out with variations in KNN and are found to perform well for imputation. Dependence induced by weighed KNN imputation is accounted for paired samples in colorectal cancer study in [13]. Patients demographic values and lifestyle are considered for weighed KNN imputation and the statistical dependence is accounted. This modified KNN performs better than the traditional KNN and the other multiple imputation techniques like Markov Chain Monte Carlo (MCMC) and Expectation Maximization (EM). An adaptive imputation method for incomplete pattern classification is proposed in [14]. The concept of credal classification is used along with the KNN and self-organizing map techniques. Belief function theory is used to model uncertainty and imprecision.

Imputation techniques are underutilized in health care datasets. A survey on reporting and handling of missing data in predictive research for type 2 diabetes mellitus states that if missing data is less than 5%, no imputation method is required. If the proportion of missing data is between 5% and 15%, single imputation performs better. If the proportion of missing data is more than 15%, multiple imputation proves to be effective. The paper explains about the inadequate handling of missing data in the study of diabetes mellitus. It concludes that guidelines explaining the missing data, reasons for missing data, etc. might help statisticians to decide on the imputation method [15].

EHR contains several observations and features. Machine learning techniques are applied on them for clinical decision making. Though there are several features in EHR, not all features contribute equally towards prediction. Inspite of the underlying fact that several techniques are used for imputation, very few imputation techniques consider the concept of feature weighting to overcome the negative impact of unimportant features in imputation that might affect the classification performance. We incorporated feature weighting inferred from a hybrid approach of WOA and LAHCA in nearest neighbour imputation technique to impute missing values in EHR. It has been observed that there is a decrease in the misclassification error rate when our proposed method of feature weighting is applied in nearest neighbour imputation. This in turn improves the classification performance.

3. Methods

3.1. Missing data patterns

Before selecting the appropriate imputation method for missing data, there is a need to identify the pattern of missingness in the dataset. There are three types of missingness patterns. Missing completely at random (MCAR), Missing at random (MAR) and Missing Not at Random (MNAR) [16].

Missing completely at random states that the probability of an observation being missing depends neither on the observed nor on the unobserved measurements. It can be represented as

$$P(Y|x_o, x_m) = P(Y) \tag{1}$$

where Y represents the missing value indicator which is 0 if the value of X is missing and 1 if it is observed. x_o represents the complete observations and x_m represents the missing observations. For example, an observation is missing because the laboratory sample is dropped. In this case, the missing value neither depends on other observations nor on the missing value itself. It is completely independent. Complete case analysis might yield better statistical inferences for MCAR though there is some loss of information.

Missing at random states that the probability of an observation being missing depends on the observed measurements and not on the unobserved measurements. It can be represented as

$$P(Y|x_o, x_m) = P(Y|x_o) \tag{2}$$

For example, patients' observations may be missing because they are dropping out as they find the treatment to be ineffective. This can be predicted by their observed values. Also, the patients who are having missing observations might have a similar value for their measurements like the other patients with the same characteristics. Methods based on likelihood might yield better statistical inferences for MAR.

Missing not at random states that the probability of an observation being missing depends on the unobserved measurements themselves. For example, a patient whose observation has to be taken on a particular day for cocaine level is not turning up because he had consumed the same and expects the level to be high. This results in missing data which can be classified as MNAR as the missing value depends on itself. A joint model of both X and Y is needed in this case.

3.2. Nearest neighbour imputation

As discussed in the earlier sections of this paper, there is no universal imputation method that works well for all kinds of predictive algorithms. If the distribution of the dataset is known, parametric approaches may be used but in reality we may not know the distribution of data clearly. In those cases non parametric approaches for imputation can be a better choice [17].

Nearest Neighbour is the best algorithm used when there is no prior knowledge on the data distribution. Nearest Neighbour method computes the k nearest neighbours for the observation with missing values and imputes the missing values from the values of its neighbours. Usually Euclidean distance is used as a metric to compute the neighbours. KNN is capable of imputing both categorical and continuous variables. The nominal attribute is replaced with the most commonly used value for that attribute from all its neighbours whereas the numerical attribute is replaced with the average of the values from its neighbours [5]. Since nearest neighbour imputation is a lazy learning approach, there is no explicit step required for training. Imputation using KNN proves to be effective in many scenarios. An example is [18] where KNN imputation is shown to be effective in comparison with the other imputation approaches. Different variations of KNN are proposed for imputation by optimizing parameters in various applications. Recent works carried out to optimize the value of k in KNN includes [19] where a method is proposed that optimizes k based on data driven approach, [20] which learns k by learning a correlation matrix that reconstructs test data point by training data points and [21] that constructs a kTree during training stage of KNN classification to output the value for k parameter based on the testing sample. The performance of KNN algorithm is improved by feature weighing in many scenarios. Few examples of recent works for feature weighting KNN includes [22] where feature weights for KNN is computed based on statistical measures, [23] that uses information gain to compute feature weight for KNN, [24] that uses evolutionary algorithm to compute the feature weights and weights of the neighbours simultaneously and [25] that uses swam intelligence to assign weights in KNN classification. [26] proposes Dudani measure to compute the distance and the missing values are imputed using KNN based on gray relational analysis for software quality datasets.

Weighted KNNI method is similar to KNN but different weights are assigned to different neighbours depending on their distance. The other steps are similar to the normal KNN imputation algorithm [5]. Weighed KNN proves to yield better results in terms of RMSE for imputation of missing values in wireless sensor networks. The algorithm considers spatial, temporal and other non- linear attributes and the concept of minimized similarity distortion when computing distance in weighed KNN [27].

3.3. Whale optimization algorithm

WOA is an evolutionary algorithm proposed by [28] based on the hunting behavior of the humpback whales. WOA is proved to be a competitive algorithm for optimization problems in comparison with the state of the art metaheuristic and conventional approaches. The exploitation and exploration ability of WOA helps to yield better results for optimization algorithms. Humpback whales are observed to go deep into the ocean and create bubbles in spiral shape around the prey. WOA works based on three steps – encircling the prey, bubble net attacking method and search for the prey. The last two steps correspond to the exploitation and exploration ability of the algorithm respectively. WOA starts with a random set of solutions and at each iteration, the search agents update their positions according to an arbitrary search agent or the best solution obtained till then based on the value of a parameter that is constantly decreased in order to ensure for the exploitation and exploration ability of the WOA. The mathematical model to encircle the prey (solution) is given by Eqs. (3), (4) [28].

$$D = |C\cdot \overrightarrow{S^{*}}(t) - \vec{S}\,(t)| \tag{3}$$

$$\vec{S}\,(t+1) = \overrightarrow{S^{*}}(t) - \overrightarrow{A}\cdot D \tag{4}$$

where t represents the current iteration, S^{*} represents the best solution obtained till then, S is the position vector. The coefficient vectors A and C are given by Eqs. (5) and (6) respectively.

$$\overrightarrow{A} = 2\vec{a}\cdot\vec{r} - \vec{a} \tag{5}$$

$$\vec{C} = 2\cdot\vec{r} \tag{6}$$

where a decreases from 2 to 0 over iterations and r is the random vector in $[0, 1]$. The search agents update their positions according to Eq. (4) depending on the position of the best solution. The shrinking encircling behavior is obtained by decreasing a in Eq. (5) by Eq. (7).

$$a = 2 - t\frac{2}{MI} \tag{7}$$

where t is the iteration number and MI is the maximum number of allowed iterations. The spiral path is simulated by calculating the distance between S and S^{*}. The position of the neighbour search agent is created by a spiral Eq. (8)

$$\vec{S}\,(t+1) = D'\cdot e^{\mathrm{bt}}\cdot cos(2\pi l) + \vec{S}^{*}(t) \tag{8}$$

where

$$D' = |\vec{S}^{*}(t) - \vec{S}\,(t)| \tag{9}$$

represents the distance of the *ith* search agent and the prey (best solution till then), b is the constant to determine the shape of logarithmic spiral, l is the random number in [-1,1]. A probability of 50% is assumed to be chosen between the shrinking encircling mechanism and spiral path during optimization process given by Eq. (10)

$$\vec{S}\left(t+1\right) = \begin{cases} Shrinking\ \ mechanism\ (eqn4)\ if\ \ (p < 0.5) \\ Spiral\ \ path\ \ eqn8)\ if\ \ (p > = 0.5) \end{cases} \tag{10}$$

where p is random in $[0, 1]$. To increase the exploration capability of WOA, the position of search agent is updated according to a random search agent rather than updating with the best solution obtained so far. Hence it moves away from the best known agent to explore new solutions. It is mathematically modelled by Eqs. (11) and (12).

$$\overrightarrow{D} = \left| \overrightarrow{C} \cdot \overrightarrow{Srand} - \overrightarrow{S} \right| \tag{11}$$

$$\overrightarrow{S}\left(t+1\right) = \overrightarrow{Srand} - \overrightarrow{A} \cdot \overrightarrow{D} \tag{12}$$

where $\overrightarrow{Srand}$ is the random search agent chosen from the current population.

The algorithm of WOA is shown in Algorithm 1. WOA is used in many optimization problems including breast cancer diagnosis [29], liver segmentation in MRI images [30], etc. WOA also finds its use in feature selection problems [31,32].

Algorithm 1. Pseudocode of WOA

```
Create initial population S_i=(i=1,2...np);
Calculate the fitness value for each solution;
$S^* \leftarrow$ best search agent;
Initialize a,A,C,l,and p;
while $t < Maxiterations$ do
    foreach solution do
        Update a,A,C,l, and p;
        if $p < 0.5$ then
            if $|A| < 1$ then
                Update the position of current
                 solution by eqn 4;
            if $|A| >= 1$ then
                Select a random search agent;
                Update the position of current
                 search agent by eqn 12;
        if $p >= 0.5$ then
            Update the position of current
             search by eqn 8;
    Calculate the fitness value for each
     solution;
    Update $S^*$ if there is a better solution;
    t = t + 1;
Return $S^*$;
```

3.4. Late Acceptance Hill Climbing Algorithm

Late Acceptance Hill Climbing Algorithm is proposed by [33]. It is a simple local search algorithm based on the idea of late acceptance strategy on hill climbing algorithm. Most of the local search algorithms like Simulated annealing, Threshold accepting employs a cooling schedule. Late acceptance strategy also employs a control parameter for acceptance condition that is derived from the history of the search. Hence LAHCA accepts a solution by comparing the current solution with the solution obtained few iterations before rather than with the immediate previous solution. The algorithm of LAHCA is shown in Algorithm 2. The LAHCA is used extensively in wide variety of applications including high school timetabling [34], combinatorial interaction testing problem [35], Google machine reassignment problem [36], etc.

Algorithm 2. Pseudocode of LAHCA

```
Create initial solution S;
Calculate the fitness value for solution S -
 O(S);
Initialize $L_h$;
for $k \in 0...L_{h-1}$ do
    $f_k \leftarrow O(S)$
while $(I > 100000)$ and $(I_{idle} > I * 0.02)$ do
    Construct a candidate solution $S^*$;
    Construct a candidate fitness function
     $O(S^*)$;
    if $(O(S^*) >= O(S))$ then
        Increment the idle iteration number
         $I_{idle} = I_{idle} + 1$;
    if $(O(S^*) < O(S))$ then
        $I_{idle} = 0$;
    Calculate the virtual beginning
     $v = ImodL_h$;
    if $(O(S^*) < f_v)$ or $(O(S^*) <= O(S))$
     then
        Accept the candidate $S = S^*$;
    else
        Reject the candidate $S = S$;
    if $(O(S) < f_v)$ then
        Update the fitness function array
         $f_v = O(S)$;
    Increment the iteration number
     $I = I + 1$;
```

4. Proposed approach: kNN + LAHCAWOA

One of the prominent characteristics of EHR datasets is that it contains many features but not every feature is equally important for prediction. For example, to identify whether the type of cancer in a patient is benign or malignant, his 'height' might have been recorded in EHR and this feature might not contribute to the classification. Another feature 'tumour size' which has significant effect on classification might have missing values for certain observations. Less important features like 'height' are also considered and given equal weightage during imputation of high important features like 'tumour size' when algorithms such as KNN imputation are used. This might distort the classification performance. Most of the existing imputation techniques do not consider the significance of the features in the imputation process. The relevance of a feature is found to affect the classification [37]. Our

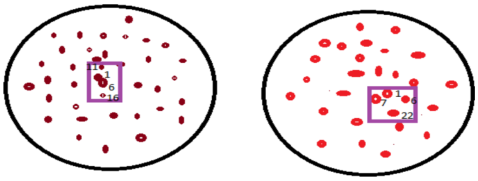


Fig. 1. Neighbours (a) KNNI and (b) KNN + LAHCAWOA.

proposed approach kNN + LAHCAWOA incorporates feature weights calculated by a hybrid of WOA and LAHCA in nearest neighbour imputation and developed an improved nearest neighbour imputation approach to impute missing values.

The major problem with the metaheuristic algorithms is its convergence. Most of the algorithms tend to converge slower or end up in local optima. Moreover, a balance between the exploitation and the exploration capability of the metaheuristic algorithms play a major role in its performance. A lot of metaheuristic algorithms exist and few popular among them include Particle Swam Optimization(PSO), Genetic algorithms(GA), Ant Colony Optimization(ACO), WOA, etc. Owing to the fact that WOA does both exploitation and exploration simultaneously by performing a smooth transit between them using the adaptive variation with a predetermined parameter, WOA is proved to yield better solutions in many applications when compared with other metaheuristic algorithms [38,39]. Besides, WOA has the lowest number of predetermined parameters and is able to avoid local optima so that it converges to a global solution quickly [40,41]. Hence, we have chosen WOA in our proposed method. Inspite of these advantages, WOA suffers from its own deficiencies. The major deficiency of WOA is its local search mechanism. [42]. Inorder to cope up with this deficiencies and considering the fact that local search algorithms are combined with metaheuristic algorithms to improve their exploitation capability and increase their convergence speed, a popular local search algorithm LAHCA is combined with WOA in our proposed approach. To enable the search agents to explore the search space in a much better way, EM-GMM clustering is also embedded with WOA. EM-GMM clustering is found to be more flexible than other popular clustering algorithms and can also entertain mixed membership. During the exploration phase, EM-GMM clustering is used to cluster the search agents and a random search agent from a different cluster is chosen to improve diversity. This aids WOA to explore the search space in a much better way without getting trap in local optima. Yet, in order to study the performance of our proposed approach, the experiments are also conducted by replacing WOA with GA and PSO in our proposed approach and the results discussed in Section 5.3. Consider an input data matrix X with m number of observations and n features. Missing ratio is given as

$$Missing\ ratio = \frac{Number\ of\ missing\ observations}{Total\ number\ of\ observations} \tag{13}$$

4.1. Hybrid of WOA and LAHCA

A hybrid of WOA and LAHCA is used to obtain feature weights in our proposed approach. WOA starts with a set of random solutions for feature weights. A sample Solution weight vector S takes the form

$$S = \{w^1, w^2, ..., w_n\} \tag{14}$$

where $w_1, w_2, \ldots, w_n$ represents the feature weights for the features $1, 2, \ldots, n$ respectively and take the values in the interval $[0, 1]$. The classification accuracy of SVM classifier is used as the fitness function. WOA algorithm is expected to find the feature weights that maximize this fitness value. The solutions are evaluated at each iteration and updated accordingly. Two different versions of hybrid approach is proposed in our work. In version I, LAHCA is embedded into WOA and used in the exploitation part of WOA. Rather than choosing the best solution in exploitation part, WOA in our proposed approach uses LAHCA to enhance the best solution by searching in its nearby region. Hence S^* in Eq. (4) is replaced with the best solution from LAHCA. This aids in improving the exploitation capability of WOA. In addition, the exploration capability of WOA is improved by using Expectation-Maximization Clustering using Gaussian Mixture Models (EM-GMM) to cluster the search agents and choosing the random search agent from a different cluster to improve diversity. Hence the random search agent in Eq. (12) is replaced with the random search agent obtained after clustering. Hence S^* in Eq. (4) is enhanced in our approach kNN + LAHCAWOA-I as

$$S^* = S_{best}\ from\ LAHCA\ with\ S^*\ as\ its\ initial\ solution \tag{15}$$

$\overrightarrow{Srand}$ in Eq. (12) is enhanced in kNN + LAHCAWOA-I as

$$\overrightarrow{Srand} = S_a\ from\ diverse\ cluster\ after\ EM\text{-}GMM. \tag{16}$$

In version II, the WOA is used to obtain the feature weights and the final solution is improved with LAHCA. The final feature weight vector is given as

$$S = \{w_{1u}, w_{2u}, ..., w_{nu}\} \tag{17}$$

where $w_{1u}, w_{2u}, \ldots, w_{nu}$ represents the final feature weights for the features $1, 2, \ldots, n$ respectively.

The pseudocode of our proposed approach (version I and version II) is shown in Algorithms 3 and 4 respectively.

Table 1
Dataset description.

D.No	Dataset	Features	Rows	Feature types	Classification	Samples proportion	Source
1	Breast cancer	10	699	Discrete	Binary	65%-35%	UCI
2	Breast tissue	9	108	Continuous	Multiclass(6)	21%-20%-13%-14%-15%-17%	UCI
3	Cancer	32	570	Continuous	Binary	37%-63%	Kaggle
4	Spine	12	310	Continuous	Binary	68%-32%	Kaggle
5	Diabetes	9	768	Mixed	Binary	65%-35%	Kaggle
6	Heart	13	270	Mixed	Binary	55%-45%	Kaggle
7	Liver	10	583	Mixed	Binary	71%-29%	UCI
8	Thoraric	16	470	Mixed	Binary	85%-15%	UCI
9	Parkinson‘s disease	754	756	Mixed	Binary	25%-75%	UCI
10	Colonoscopy	698	76	Mixed	Multiclass(3)	28%-20%-52%	UCI

Algorithm 3. Pseudocode for kNN + LAHCAWOA – Version I

```
$X[m][n] \leftarrow$ Read from input file;
$NormalizedX[i,j] = \frac{X[i,j] - min(X[][j])}{max(X[][j]) - min(X[][j])}$;
Create initial population $S_i (i = 1, 2...np)$;
Calculate the fitness value for each solution;
Find the best search agent;
Initialize a,A,C,l, and p;
while $t < Max\ iterations$ do
    foreach solution do
        Update a,A,C,l, and p;
        if To model shrinking encircling mechanism then
            if Exploitation phase then
                Apply LAHCA(2) to find a better solution using eqn 15;
                Update the position of current search agent by eqn 4;
            if Exploration phase then
                Use EM-GMM clustering to cluster the search agents;
                Select a random search agent from a different cluster by eqn 16;
                Update the position of current search agent by eqn 12;
        if To model spiral shaped path then
            Use LAHCA(2) to find a better solution using eqn 15;
            Update the position of current search by eqn 8;
    Calculate the fitness value for each solution;
    Update the best search agent if there is better solution;
    $t = t + 1$;
Return the best search agent;
Get the optimized feature weight vector S;
dist = Compute distance matrix() using S;
foreach test data point do
    Learn 'k' by CM-KNN method;
    for $n = 0$ to $k$ do
        if $NX[i][j] = null$ then
            $NX[i][j] = Impute(NX[0..k])$;
```

Algorithm 4. Pseudocode for kNN + LAHCAWOA – Version II

```
$X[m][n] \leftarrow$ Read from input file;
$NormalizedX[i,j] = \frac{X[i,j]-min(X[][j])}{max(X[][j])-min(X[][j])}$;
Create initial population $S_i (i = 1, 2...np)$;
Calculate the fitness value for each solution;
Find the best search agent;
Initialize a,A,C,l, and p;
while $t < Maxiterations$ do
    foreach solution do
        Update a,A,C,l, and p;
        if To model shrinking encircling mechanism then
            if Exploitation phase then
                Update the position of current solution by eqn 4;
            if Exploration phase then
                Update the position of current search agent by eqn 12;
        if To model spiral shaped path then
            Update the position of current search by eqn 8;
    Calculate the fitness value for each solution;
    Update the best search agent if there is a better solution;
    $t = t + 1$;
Identify the best search agent;
Use LAHCA(2) to find a better solution using eqn 15;
Return the best search agent;
Get the optimized feature weight vector S;
dist = Compute distance matrix() using S;
foreach each test data point do
    Learn 'k' by CM-KNN method;
    for $n = 0$ to $k$ do
        if $NX[i][j] = null$ then
            $NX[i][j] = Impute(NX[0..k])$;
```

4.2. Distance calculation

For each missing observation, distances are calculated with the other observations according to the feature weights from the hybrid approach of WOA and LAHCA. Three different distance metrics – Euclidean, Manhattan, Canberra are tried for both the versions of our proposed method kNN + LAHCAWOA. [5] states that Pearsons coefficient can also be used instead of Euclidean distance to identify the nearest neighbours. The euclidean distance metric is given by

$$Edist(x_i, x_j) = \sqrt{(x_i - x_j)^2} \tag{18}$$

The Manhattan distance metric is given by

$$Mdist\left(x_i, x_j\right) = \sum_{i,j=1}^{n} \left|x_i - x_j\right| \tag{19}$$

The Canberra distance metric is given by

$$Cdist\left(x_i, x_j\right) = \sum_{i,j=1}^{n} \frac{|x_i - x_j|}{|x_i| + |x_j|} \tag{20}$$

where x_i, x_j represents the feature values of two observations between which the distance has to be calculated and n represents the number of features. Distance between observations with missing features is calculated as follows.

$$dist\left(x_i, x_j\right) = total\ distance\ calculated\ \left(x_i, x_j\right) \times \sqrt{\frac{length(x_i)}{length(x_i[!missing])}} \tag{21}$$

where missing is given by

$$missing = ismissing(x_i)|ismissing(x_j) \tag{22}$$

where | represents an 'OR' operation. Since the features are weighted based on their importance on classification, features with high importance play major role in calculating the distance between the observations with missing features unlike the existing method that gives equal weightage to all the features. Hence the proposed method tends to neighbour the observations that are close to each other based on the feature importance thereby helping in the reduction of misclassification error. An example is shown in Fig. 1. The figure is simply an assumption of representing observations in a 2-D space to help the readers understand the concept better. (a) Depicts how the observations are distanced from each other when conventional KNNI is used. (b) Depicts how the observations are distanced from each other when our proposed approach kNN + LAHCAWOA is used where feature weights are considered. It is obvious that the neighbours of any observation differ in both the figures. The neighbours selected for observation 1 by conventional KNN are 6, 11 and 16 whereas the neighbours selected by kNN + LAHCAWOA are 6, 7 and 22. This implies that observation 1 is close to observations 6, 7 and 22 when feature weighting is considered. The similarity measure among the observations 1, 6, 7 and 22 with respect to classification behavior is high. Once the final distance vector is calculated for each missing observation, the average values from *k* nearest neighbour are used to impute the missing values. If the value that has to be imputed is missing in the nearest neighbour, the next nearest neighbour is considered.

4.3. Learning *k*

There are different approaches used to learn *k*, the major approaches being ‘rule of thumb’ and ‘cross-validation’. We use a different approach proposed in [20] to learn different *k* for different test points to improve the efficiency. This approach derives a correlation matrix between the training data points and the test data points and reconstructs the test data from the training data points. The *k* value is identified from the correlation matrix and hence depending on the correlation of different test points with different training data points, the value of *k* differs for different test points. The final optimization function in this approach is given as

$$min_{W}\|X^{T}W-Y\|_{F}^{2}+\rho 1R1(W)+\rho 2R2(W)+\rho 3R3(W) \tag{23}$$

where *X* represents the training data points, *W* represents correlation matrix or reconstruction weight matrix, *Y* represents the test data points, $\rho 1$, $\rho 2$ and $\rho 3$ are tuning parameters, *R1* represents the l2-norm regularization term, *R2* represents l2,1-norm regularization term, *R3* represents Locality Preserving Projection regularization term and are defined as

$$R1(W)=\|W\|_{1} \tag{24}$$

$$R2(W)=\|W\|_{2,1} \tag{25}$$

$$R3(W)=Tr(W^{T}XLX^{T}W) \tag{26}$$

respectively with T_r as the Trace operator, *L* as Laplacian matrix and *L* = *D-S* with Diagonal matrix *D* and similarity matrix *S* in real space. The Frobenius norm, l2-norm, l1-norm, and l2,1-norm of a matrix *X* are given by

$$\left\|X\right\|_{F}=\sqrt{\sum_{1}\left\|X_{i}\right\|_{2}^{2}} \tag{27}$$

$$\left\|X\right\|_{2}=\sqrt{\sum_{i}\sum_{j}\left|x_{ij}\right|^{2}} \tag{28}$$

$$\left\|X\right\|_{1}=\sum_{i}\sum_{j}\left|x_{ij}\right| \tag{29}$$

$$\left\|X\right\|_{2,1}=\sum_{i}\sqrt{\sum_{j}x_{ij}^{2}} \tag{30}$$

The optimization function is optimized using Iteratively Reweighted Least Square method and the global optimum *W* is obtained. Once the correlation matrix is obtained, the parameter *k* is learned for different test points and the values are imputed using Eq. (31). The tuning parameters $\rho 1$, $\rho 2$ and $\rho 3$ are tested with values {0.005, 0.05, 0.5, 5, 50} and different values are found to optimize the equation for different datasets.

$$Imputedvalue(x_{ij})=\frac{\sum_{n=0}^{k}x_{nj}}{k} \tag{31}$$

Where x_{ij} represents the observation x_i in which the value for feature *j* has to be imputed and x_{nj} represents the values of the feature *j* in the neighbouring observations x_n.

5. Experimental setting

We propose two versions – kNN + LAHCAWOA-I and kNN + LAHCAWOA-II based on the technique of hybridization as explained in previous section. Three different distance metrics are tried with both the versions – Euclidean NNI (ENNI), Canberra NNI (CNNI) and Manhattan NNI (MNNI). The results are compared with five conventional and state of the art Nearest Neighbour Imputation (NNI) methods – Conventional KNN, Feature weighted KNN based on PSO (KNN + PSO), Feature weighted KNN based on GA (KNN + GA), Feature weighted KNN based on Information gain (KNN + IG) and KNN based on Gray distance (GDKNN). KNN + PSO and KNN + GA uses PSO and GA instead of WOA in version 1 of our proposed approach. Our approach is also compared with few other state of the art and more popular imputation methods – Mean value imputation (MV), Multiple imputation (MI), SVM regression imputation (SVM), weighted random forest imputation (WRF). Experiments are conducted for three different values of missingness – 10%, 20% and 30% and on three different classifiers – Support Vector Machines (SVM), Random Forest (RF) and DNN. A 10-fold cross-validation is used to avoid overfitting. The experiments are also repeated 5 times to avoid bias if any during cross-validation and the Tables 2–7 represent the average results.

5.1. Dataset description

Ten biomedical classification data sets downloaded from Kaggle and UCI data repository are considered for our experimentation. Few datasets are examples for binary classification tasks and few for multiclass classification tasks. A description of the datasets used for our experimentation is provided in Table 1. Following is the description of columns of Table 1. Oversampling is done in imbalanced datasets(Liver,Thoracic,Parkinson‘s).

D.No – Represents the dataset number.
Dataset – The datasets used in our experimentation.
Features – Number of features in the corresponding dataset.
Rows – Number of observations in the corresponding dataset.
Feature types – The nature of features in the dataset. Few include only discrete features, few include only continuous features and few has mixed features that include continuous, discrete and categorical.
Classification – Nature of classification task. Binary or multiclass classification.
Samples proportion – The proportion of observations belonging to different classes.
Source – The source of the dataset.

5.2. Parameter setting

The classifiers used in our experimentation include SVM, Random forest which is an example of ensemble technique bagging and deep neural networks (DNN). A radial basis kernel with degree 3 is used in most of the cases whereas a polynomial kernel is also used in few datasets for SVM classification. 50 trees are grown in random forest classification. Deep learning classification is executed with keras on top of tensorflow. Adaboost is used to optimize the network weights and

Table 2
Misclassification error rate for different NNI methods and kNN + LAHCAWOA – Deep Neural networks (for 10%,20% and 30% missingness in each dataset).

Dataset	KNN	KNN + PSO	KNN + GA	KNN + IG	GDKNN	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	3.6	3.9	4.1	3.9	3.4	3.2	**3.18**
	4.9	4.8	4.9	4.1	4.2	**4.05**	4.08
	6.8	6.17	6.19	6.2	6.08	**5.9**	5.96
Dataset2	10.3	10.1	10.07	10.8	10.15	**10.03**	10.12
	11.98	11.9	11.1	11.3	11.5	**11.05**	11.80
	13.9	13.7	**13.2**	13.9	13.5	13.8	13.79
Dataset3	3.9	3.92	3.6	3.9	3.8	3.67	**3.5**
	4.5	4.7	4.9	4.8	4.9	**4.42**	4.48
	6.15	6.3	5.9	6.5	5.58	**5.25**	5.37
Dataset4	23.3	22.1	21.8	25.5	21.8	**21.3**	21.5
	28.7	28.4	27.9	29.5	29.1	**27.2**	27.28
	32.9	31	32	32.89	31.76	**30.7**	30.9
Dataset5	16.1	16.5	16.72	16.8	16.12	**16.03**	16.19
	21.6	21.05	21.8	23.63	24.9	**20.9**	21
	25.8	25.16	26.81	26.49	26.19	**25.3**	25.61
Dataset6	33.5	33	33.8	34.56	32.8	**32.76**	32.9
	35.8	**35.1**	35.9	35.37	35.20	35.12	35.58
	38.4	39	38.8	38.4	38.54	**38.25**	38.34
Dataset7	30.2	30	30.7	30.3	30.9	**29.98**	30
	32.3	32.19	32	32.65	32.32	32.49	**32.15**
	34.2	34	34.58	34.7	34.12	34.6	**33.83**
Dataset8	17.1	17.8	17.59	17.55	17.65	**17.02**	17.17
	19.23	19.97	19.9	19.5	20	**19.04**	19.06
	22.8	22.3	21.9	21.6	21.7	**20.9**	21
Dataset9	22.1	21.81	21.05	23.15	20.09	**20.02**	20.83
	25.13	24.89	24.1	23.15	24.10	23.01	**23**
	26.1	24.93	25.91	24.6	25.17	**25**	25.94
Dataset10	25.09	24.14	23.19	24.67	25.79	**23.09**	23.98
	26.93	25.91	26.16	25.01	24.11	**24.02**	24.98
	27.15	26.92	27.99	26.16	26.77	**26.01**	26.54

Bold values represent the least misclassification error rate among all the methods.

Table 3
Misclassification error rate for different NNI methods and kNN + LAHCAWOA – Random forests (for 10%,20% and 30% missingness in each dataset).

Dataset	KNN	KNN + PSO	KNN + GA	KNN + IG	GDKNN	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	3.03	3.08	3.05	3.1	3.09	**2.95**	2.99
	4.9	4.98	4.56	4.6	4.18	4.19	**4.15**
	5.2	5.92	5.64	5.53	5.16	5.1	**5.02**
Dataset2	7.2	7.36	7.51	7.95	7.8	7.02	**7**
	11.3	11.0	11.28	11.68	11.4	**10.90**	10.98
	16.32	16.7	16.45	16.9	16.81	16.25	**16.1**
Dataset3	4.5	4.7	4.7	4.2	4.2	4.15	**4.02**
	5.5	5.8	5.68	5.42	5.8	**5.1**	5.2
	6.3	6.54	6.16	6.8	6.23	**6.07**	6.10
Dataset4	20.9	20.8	21.3	21.37	21.89	20.76	**20.38**
	26.7	26.79	26.92	26.59	26.8	**26.41**	26.45
	29.4	29.2	29.68	29.32	29.7	**29.11**	29.5
Dataset5	16.12	**16**	16.7	16.15	16.28	16.02	16.09
	18.25	18.2	18.3	18.8	18.3	**18.1**	18.29
	24.18	24.84	25	25.3	24.8	**24.06**	24.2
Dataset6	26.9	25.9	27.7	26.19	27	**25.59**	25.92
	29.5	29.1	28.9	28.9	29.14	28.59	**28.10**
	32.4	32.2	32.58	32.67	32	**31.93**	31.95
Dataset7	12.19	12.5	12.29	12.8	12.23	**12.08**	12.14
	15	14.9	14.6	14.9	14.5	**14.39**	14.58
	18.2	18.0	18.27	18.9	18	**17.83**	17.96
Dataset8	14.56	14.9	14.5	14.85	14.93	14.7	**14.30**
	18.4	18.9	18.2	18.59	18.32	**18.09**	18.23
	22.7	22.8	22.9	22.2	21.17	21.1	**21.07**
Dataset9	20.91	20.8	21.15	22.98	21.07	**20.02**	20.10
	21.95	22.16	22.93	22.99	21.80	**21.11**	21.96
	23.18	23.38	23.11	23.67	24.8	**23.09**	23.1
Dataset10	23.10	23.18	24.26	25.10	24.00	**22.98**	23.10
	24.99	24.43	24.98	25.89	24.90	**23**	23.16
	26	25.67	26.09	25.11	26.08	**25.09**	25.1

Bold values represent the least misclassification error rate among all the methods.

Table 4
Misclassification error rate for different NNI methods and kNN + LAHCAWOA – SVM (for 10%,20% and 30% missingness in each dataset).

Dataset	KNN	KNN + PSO	KNN + GA	KNN + IG	GDKNN	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	3.7	3.24	3.29	3.89	3.1	**3.01**	3.09
	4.8	4.58	4.81	4.27	4.23	**4.08**	4.1
	5.2	5.36	5.29	5.19	5.67	**5.0**	5.06
Dataset2	12.16	12.05	12.57	12	12.3	**11.95**	11.13
	14.7	14.19	14.8	14.5	14.96	**14.9**	15.0
	16.95	17.08	16.4	16.89	16.9	**16.13**	16.19
Dataset3	2.6	2.5	3.2	2.67	2.91	**2.19**	2.30
	4.13	4.53	4.66	4.89	4.23	4.83	**4.01**
	6.5	6.76	6.39	6.17	6.13	6.15	**6**
Dataset4	21.2	21.11	21.67	21.68	21.4	**20.98**	21.15
	26.3	26.34	**26**	26.7	26.45	26.2	**26**
	29.9	28	28.7	25.2	27.8	**25**	25.1
Dataset5	21.7	20.97	21.09	21.47	21.12	21.02	**20.9**
	23.2	23.48	23.64	23.96	23.41	**23.09**	23.18
	26.95	26.81	26.31	26.28	26.3	**25.96**	26.08
Dataset6	29.7	29.18	29.64	29	29.91	29.01	**28.85**
	31.1	31.91	30.96	30.85	31.01	**30.7**	30.92
	32.5	32.95	32.17	32.91	32.71	32.09	**32.01**
Dataset7	29.8	29.78	29.9	29.74	29.68	**29.09**	29.06
	31.19	31.31	31.65	31.9	31.39	31.14	**31.09**
	34.2	34.54	34.7	34.97	34.9	**34.06**	32.18
Dataset8	17.6	17.26	17.31	17.9	17.0	**16.8**	16.9
	19	18.9	19.0	19.85	19.67	**18.09**	18.02
	21.7	21.5	21.6	22	22.44	21.09	**21.03**
Dataset9	19.11	20.2	20.9	20.18	19.07	**18.95**	19.01
	20.10	21.8	21.76	20.65	20.78	**19.01**	19.08
	22.95	22.26	22.54	21.79	22.10	**20.18**	20.65
Dataset10	20	21.8	20.19	20.98	19.92	**19.12**	19.9
	22.07	22.98	22.76	22.09	21.87	**20.94**	20.96
	24.97	24.12	24.88	23.56	22.90	**21.18**	21.65

Bold values represent the least misclassification error rate among all the methods.

Table 5
Comparison of different imputation methods in Deep Neural networks – for 10%,20% and 30% missingness in each dataset.

Dataset	MV	MI	SVM	WRF	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	4.4	4.36	4.51	4.8	3.2	**3.18**
	5.8	5.9	**3.92**	5.19	4.05	4.08
	6.7	6.6	6.75	6.9	**5.9**	5.96
Dataset2	10.1	11.19	11.8	10.69	**10.03**	10.12
	11.7	11.17	12	11.29	**11.05**	11.8
	13.9	**13.3**	14	14.04	13.8	13.79
Dataset3	4.7	4.5	4.21	4.17	**3.67**	3.5
	5.8	5.25	5.18	5.05	**4.42**	4.48
	6.9	6.67	6.95	6.25	**5.25**	5.37
Dataset4	21.9	22	22.29	21.81	**21.3**	21.5
	29.8	**27.02**	29.19	28	27.2	27.28
	31.7	31.8	31.78	31.89	**30.7**	30.9
Dataset5	17	17.2	16.12	16.89	**16.03**	16.19
	21.8	21.78	21.9	21.18	**20.9**	21
	26.5	25.91	26	26.18	**25.3**	25.61
Dataset6	33.1	33.12	32.85	33.8	**32.76**	32.9
	36.8	35.18	36.12	36.9	**35.12**	35.58
	38.9	**38.15**	38.95	39.06	38.25	38.34
Dataset7	30.8	30.9	30.2	30.19	**29.98**	30
	32.9	**32.05**	32.99	32.76	32.49	32.15
	35.6	35.8	35	34.8	34.6	**33.83**
Dataset8	17.8	17.9	18.1	18.13	**17.02**	17.7
	19.9	19.16	19.38	19.4	**19.04**	19.06
	21.3	21.5	21	21.4	**20.9**	21
Dataset9	21.18	20.9	22.19	20.99	**20.02**	20.83
	26	23.18	25.76	24.04	23.01	**23**
	27.87	25.15	26	25.94	**25**	25.94
Dataset10	24.08	24.19	24.99	24.89	**23.09**	23.98
	25.19	25.23	25.18	25.94	**24.02**	24.98
	26.41	26.89	27.67	26.82	**26.01**	26.34

Bold values represent the least misclassification error rate among all the methods.

Table 6
Comparison of different imputation methods in Random forests – for 10%,20% and 30% missingness in each dataset.

Dataset	MV	MI	SVM	WRF	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	3.8	3.19	3.2	3.43	**2.95**	2.99
	4.4	4.48	4.19	4.98	**4.19**	5.02
	5.19	5.1	5.30	5.15	5.1	**5.02**
Dataset2	7.1	7.12	7.94	7.19	7.02	**7**
	11.8	11.51	11.89	11.76	**10.9**	10.98
	16.58	16.53	16.12	16.17	16.25	**16.1**
Dataset3	5.05	4.98	4.69	4.74	4.15	**4.02**
	5.56	5.13	5.98	5.3	**5.1**	5.2
	6.7	6.62	6.8	6.8	**6.07**	6.1
Dataset4	21.2	21.8	21.09	21.13	20.76	**20.38**
	27.38	27.15	27	27.1	**26.41**	26.45
	29.5	30	29.9	29.3	**29.11**	29.5
Dataset5	16.7	16.27	17.19	16.7	**16.02**	16.09
	18.8	18.22	19.16	18.8	**18.1**	18.29
	24.6	24.6	24.8	24.7	**24.06**	24.2
Dataset6	26.1	26.19	26.28	25.9	**25.59**	25.92
	29.8	**28.01**	29.9	29.59	28.59	28.10
	32.16	32.9	31.97	32.7	**31.93**	31.95
Dataset7	13.8	13.4	13.6	13.1	**12.08**	12.14
	16.8	16.12	15.5	16.58	**14.39**	14.58
	18.87	18.6	18.07	18.4	**17.83**	17.96
Dataset8	14.8	14.78	**14.17**	14.85	14.7	14.3
	18.7	18.9	18.3	18.8	**18.09**	18.23
	21.8	22.1	21.98	21.8	21.1	**21.07**
Dataset9	20.78	21.23	20.78	21.98	**20.02**	20.10
	21.77	21.90	22.0	21.96	**21.11**	21.96
	24.8	23.9	23.97	24.67	**23.09**	23.1
Dataset10	24.15	24.12	23.98	23.29	**22.98**	23.10
	24.17	24.80	24.08	23.75	**23**	23.16
	25.98	25.41	26.19	25.65	**25.09**	25.1

Bold values represent the least misclassification error rate among all the methods.

Table 7
Comparison of different imputation methods in SVM – for 10%,20% and 30% missingness in each dataset.

Dataset	MV	MI	SVM	WRF	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
Dataset1	3.3	3.19	3.25	3.75	**3.01**	3.09
	4.9	4.1	4.2	4.7	**4.08**	4.1
	6.2	6.1	6.9	6.75	**5.0**	5.06
Dataset2	12.5	12.73	12.6	12.4	**11.95**	11.13
	15.6	15.0	15.8	15.2	**14.9**	15.0
	17.2	17.1	17.03	17.79	**16.13**	16.19
Dataset3	2.5	2.7	3.0	2.99	2.19	2.3
	5.7	5.3	5.7	4.9	4.83	**4.01**
	6.3	6.9	6.5	6.2	6.15	**6**
Dataset4	21.2	21.8	21.59	21.87	**20.98**	21.15
	26.4	26.2	26.8	27	26.2	**26**
	26.9	26.9	26.7	**24.97**	25	25.1
Dataset5	21.2	21.1	21.9	21.19	21.02	**20.9**
	24	23.8	23.74	23.8	**23.09**	23.18
	26.3	26.8	26.72	25.98	**25.96**	26.08
Dataset6	29.15	**29**	29.8	29.09	29.01	28.85
	30.8	**29.15**	31	31.6	30.7	30.92
	32.1	32.3	32.9	32.7	32.09	**32.01**
Dataset7	29.8	29.6	30	29.53	29.09	**29.06**
	32	31.8	31.6	31.8	31.14	**31.09**
	34.9	34.8	34.46	34.51	**34.06**	32.18
Dataset8	16.98	16.9	17.6	17.24	**16.8**	16.9
	18.4	19.0	18.9	18.9	18.09	**18.02**
	21.4	21.94	22.0	21.5	21.09	**21.03**
Dataset9	19.87	18.99	19.11	18.99	**18.95**	19.01
	19.4	19.19	20.86	21.65	**19.01**	19.08
	22.14	22.09	21.08	21.94	**20.18**	20.65
Dataset10	21.13	20.65	19.98	19.96	**19.12**	19.9
	22.19	22.78	21.08	21.72	**20.93**	21.01
	24.21	24.89	23.54	23.17	**21**	21.18

Bold values represent the least misclassification error rate among all the methods.

Table 8
Misclassification error rate for different NNI methods and kNN + LAHCAWOA – Dataset3(MAR-30%).

Classifier	KNN	KNN + PSO	KNN + GA	KNN + IG	GDKNN	kNN + LAHCAWOA-I	kNN + LAHCAWOA-II
DNN	6.06	6.18	6.02	6.15	5.49	**5.11**	5.39
RF	6.35	6.44	6.17	6.14	6.25	**6.01**	6.10
SVM	6.14	6.79	6.90	6.2	6.13	**6.09**	6.11

Bold values represent the least misclassification error rate among all the methods.

rectifier is used as the activation function. Cross validation is used in all the classifiers. The tuning parameters $\rho1$, $\rho2$ and $\rho3$ in Eq. (23) are tested with values {0.005, 0.05, 0.5, 5, 50} and different values are found to do best in different datasets. With respect to the algorithms used for comparison, the parameters are set the same as their original corresponding papers. k is determined by 'rule of thumb' in the conventional kNN used for comparison. The population size is set between 60~100 in GA and PSO used along with our proposed approach for comparison. The acceleration factors of PSO is set to 2.025 and the inertia weight is set to 0.625. The crossover and mutation ratio are set to 0.9 and 0.1 in GA. The value of a is decreased from 2 to 0 over iterations in WOA. k in our proposed approach is determined by the correlation matrix between training and testing data as discussed in the earlier section. Five imputed datasets are generated with multiple imputation.

5.3. Results and discussion

The ten clinical datasets we have chosen for our experiment are examples for supervised machine learning. Most of the existing imputation techniques are defined for MCAR and MAR methods. MCAR can be considered as a special case of MAR in which the missing data values are a simple random sample of all the data values. Moreover, MAR is a less restrictive version of MCAR [13]. Hence, like few other missing data imputation works that assumes the missing data pattern as MCAR [3,5,7,43,44], our proposed approach also simulates the missing values under the assumption that the missing values are missing completely at random. The results are displayed only for the commonly used Euclidean distance metric in this paper due to space constraints. The experiments conducted using the other two distance metrics also yielded similar results. The simulated missing values are imputed in the datasets using our proposed approach and also using other methods used in our experimentation for comparison. The metric chosen for evaluation is misclassification error rate. Misclassification error is the proportion of misclassified observations that includes both false positive and false negative. The dataset with the imputed values from kNN + LAHCAWOA is used as input for the respective supervised machine learning models and the misclassification error rate is computed for each of the ten datasets. Another experiment is conducted using the same datasets with the same supervised machine learning models but the missing values are imputed with other state of the art NNI methods. The results of each dataset for the three different supervised learning models are shown in Tables 2–4 respectively. It is evident that the misclassification error rate for all the three models is less when missing values are imputed with kNN + LAHCAWOA. There are variations in the misclassification error rate among all the three models. This is because different supervised learning models perform well for different datasets depending on many other parameters such as the characteristics of the datasets, parameters set for the supervised learning models, nature of missingness, etc. Though there may be slight variations, it is observed that our imputation method reduces the misclassification error rate in all the three models for all the ten EHR datasets used for experimentation.

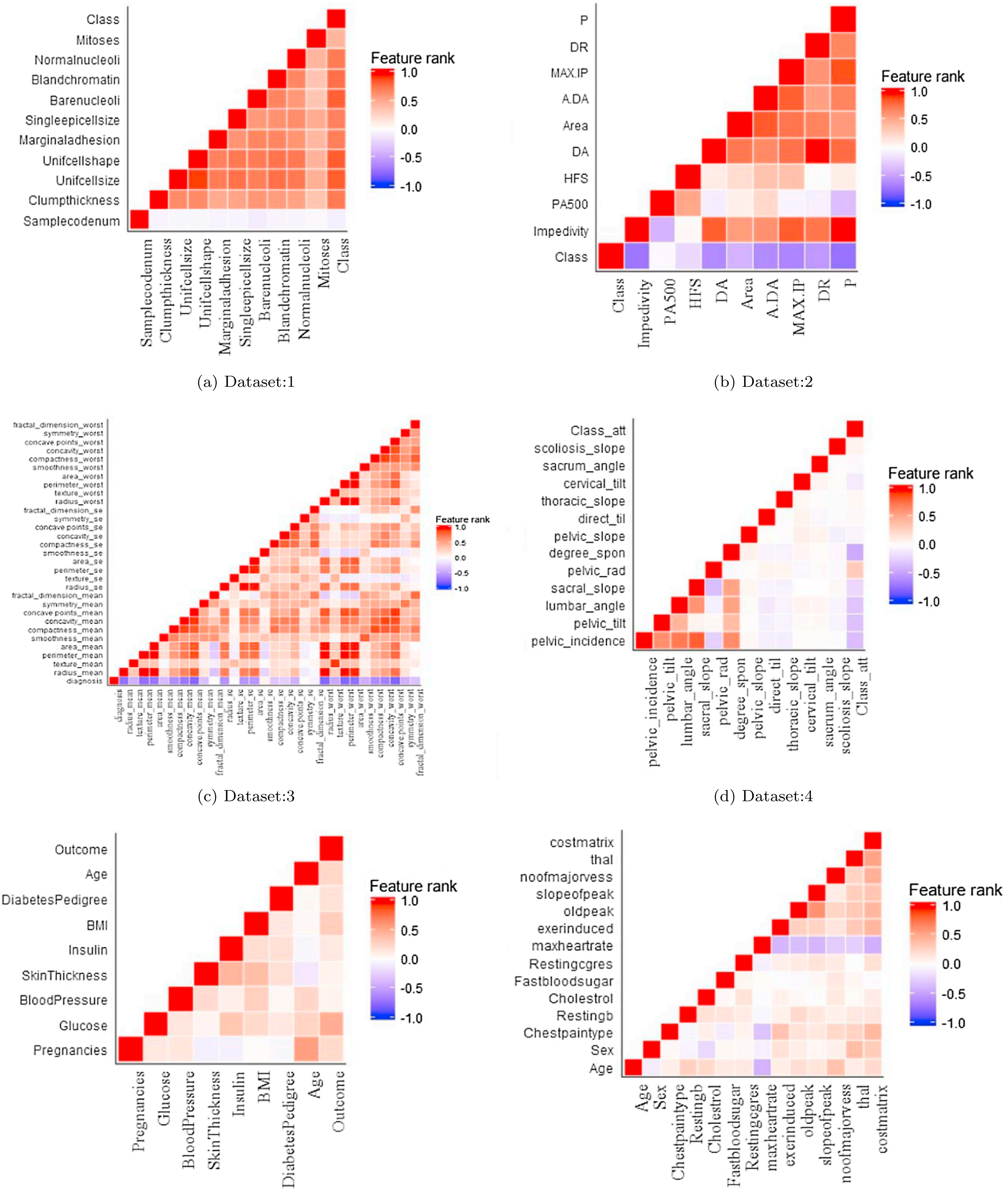
kNN + LAHCAWOA is also compared with other state of the art imputation methods and the results are shown in Tables 5–7 respectively. It is evident that our approach performs equally well or even superior in certain scenarios. kNN + LAHCAWOA outperforms other methods in more than half of the test cases. To compare the results statistically, a paired t-test is performed at 95% confidence interval. The results confirmed that both the versions of our proposed approach outperforms the other methods considered for comparison. But the results are statistically not significant between the two versions of our proposed approach.

It is observed that our proposed method outperforms all the other methods for datasets with more number of features like dataset 3, dataset 9 and dataset 10. But there are few cases in smaller datasets where other methods outperform our proposed method and such cases are found especially in DNN classifier where multiple imputation is found to cater well with DNN just like event covering method with radial basis networks [5]. Yet, DNN classifier is not the best classifier in such datasets. The least misclassification error rate is given by RF or SVM in such datasets and our proposed method outperforms other methods in these classifiers (Ex:-Dataset 7). Moreover, even in such cases, our proposed method is found to be superior with DNN for few other proportion of missingness. The optimization algorithms GA, PSO and WOA differ in terms of the quality of solutions, performance, convergence, etc. They differ in the way exploitation and exploration is done. GA does the exploration and exploitation with selection, crossover and mutation operators. PSO updates the position of the particles based on two values, pbest and gbest. WOA updates the position of search agents based on the best search agent. Since WOA does more exploration in its initial iterations, it does well with larger datasets. In smaller datasets, PSO and GA are found to converge faster and yield better solution in few cases (Ex:-Dataset 6) probably due to lesser number of feature combinations. Though such cases exist, the difference in misclassification error rate between the identified superior method and our proposed method is not significant. Hence our proposed method is found to be a stable method and more specifically when the number of features is high.

Yet, as suggested by the reviewers, it is quite interesting to know the performance of our proposed approach with non-MCAR pattern. Hence, we violated the MCAR assumption and simulated the MAR pattern in our proposed approach to perform a small experimentation on dataset 3 with 30% missingness and study the performance. The result in Table 8 shows that our proposed approach is found to be effective with MAR pattern also.

Fig. 2 'a' shows the heat map visualization for 1st dataset – breast cancer. It is obvious that features such as normalnucleoli, blandchromatin, etc. has significantly higher impact on the classification behavior than the features like mitoses, barenucleoli, etc. Amidst the features normalnucleoli and blandchromatin, the impact of the latter is higher than the former determined by the intensity of the shade. Fig. 3 shows the Root Mean Square Error (RMSE) between the actual and the predicted values by different approaches. It is clear that the RMSE between the actual and imputed values is less when kNN + LAHCAWOA is used.

Our experimental set up tested kNN + LAHCAWOA with different proportions of missingness 10% missingness, 20% missingness and 30% missingness. The results prove that kNN + LAHCAWOA reduces misclassification error rate in comparison with the state of the art approaches. There can be variations depending on the characteristics of the datasets, proportion of missingness and certain other parameters. Yet, our approach proves to be effective for all the datasets considered for experimentation.



(a) Dataset:1

(b) Dataset:2

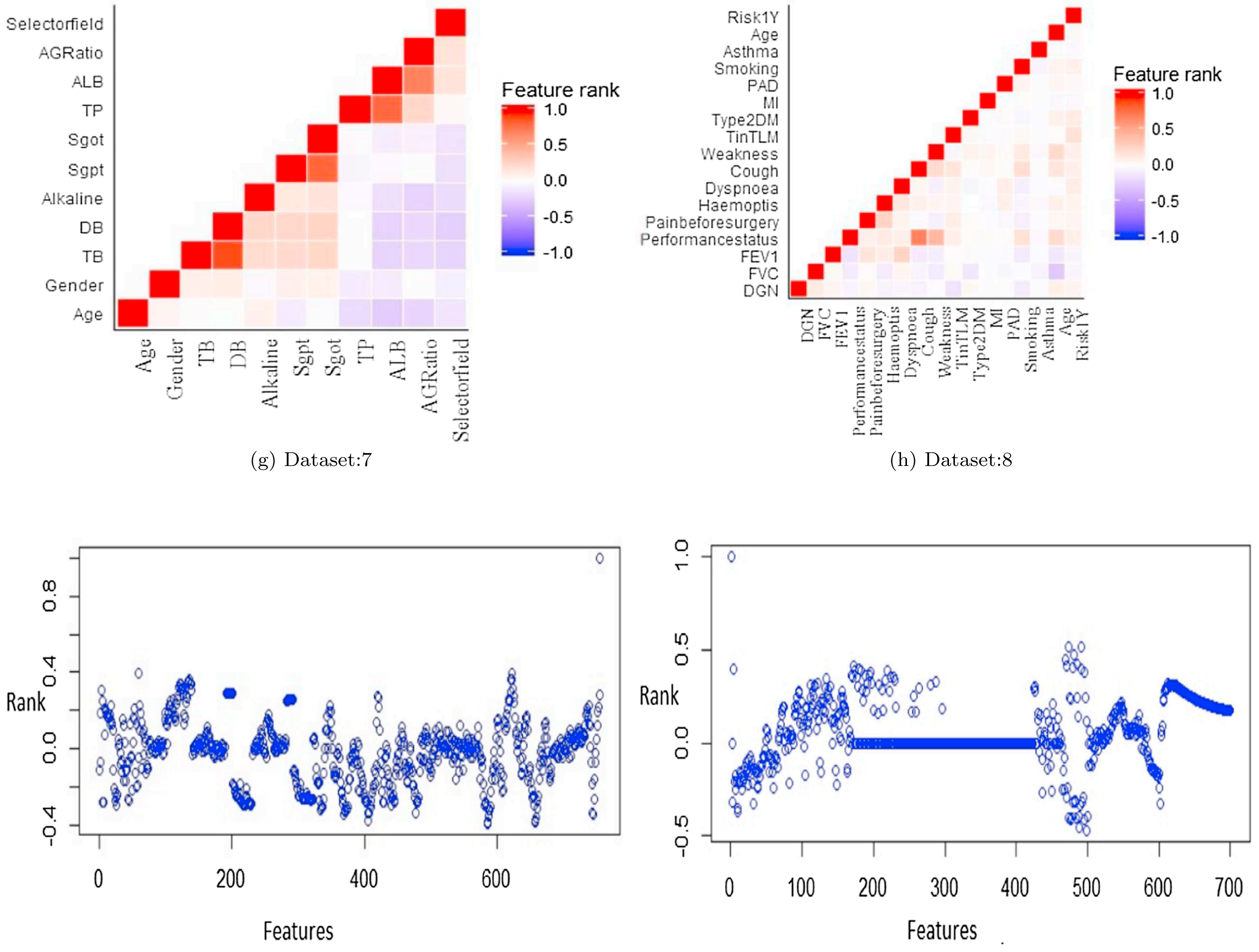
(c) Dataset:3

(d) Dataset:4

(e) Dataset:5

(f) Dataset:6

Fig. 2. Feature ranking shown as heat map for Datasets 1–8 and as plot for datasets 9, 10.



(g) Dataset:7

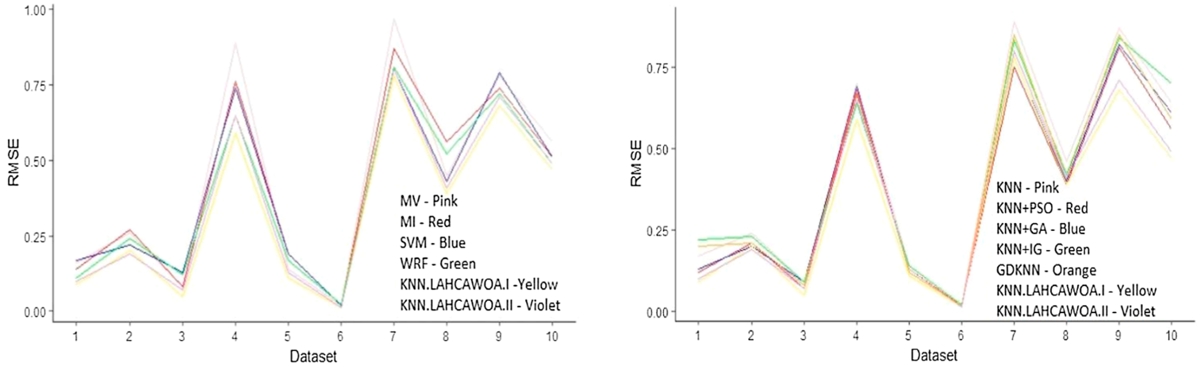
(h) Dataset:8

(i) Dataset:9

(j) Dataset:10

Fig. 2. (*continued*)

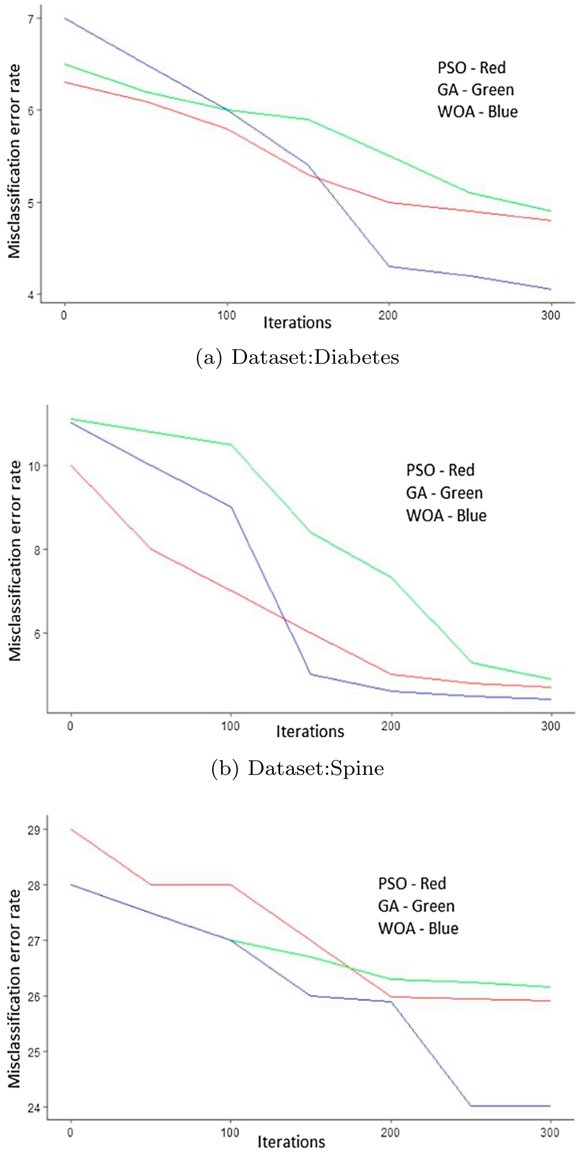
Hence, the proposed method is a generalized missing data imputation method to different kinds of classification problems with health datasets (Ex:- prediction of the type of a disease like cancer whether it is benign or malignant, identify if a patient is abnormal or normal using collected data, diagnose if a patient has a particular type of a disease like diabetes or not, etc.). The major characteristics of our proposed method includes its capability to work with mixed feature types (Ex:- categorical, discrete, continuous) and also with different kinds of classifiers. Our method has also proved its performance with different proportions of missingness. Moreover, determining *k* is also not an issue since we use different *k* for different test points by learning the correlation matrix as discussed in earlier section. Use of WOA algorithm with an enhanced exploitation and exploration capability yields faster convergence rate when compared with GA and PSO. The sample convergence of the curves of WOA, PSO and GA for three datasets is shown in Fig. 4a–c respectively. It is obvious that WOA converges to a better



(a) Among Other Imputation methods

(b) Among kNN based methods

Fig. 3. Comparison of RMSE between the actual and predicted values.



(a) Dataset:Diabetes

(b) Dataset:Spine

(c) Dataset:Cancer

Fig. 4. Convergence of the curves of WOA, PSO and GA for three datasets – 20% Missingness with DNN.

solution in final iterations and avoids local optima. WOA does more exploration in the initial iterations but the algorithm converges to a better solution during final iterations especially in the last 50 iterations. Hence it performs better than GA and PSO. Nevertheless, there is more scope for future work in our proposed method. Our proposed method can be modified or enhanced to handle MNAR pattern of missingness. The computational time of our proposed method is little high with large datasets owing to many feature combinations in metaheuristic algorithm. A hybrid of our proposed method with feature selection methods or big data tools can help reduce the computational time.

6. Conclusion

Though it is proved that there is no universal imputation method that caters well with all the datasets and with different predictive algorithms, there are always new techniques and approaches emerging from which the best approach and technique can be chosen based on the nature of the dataset, predictive algorithm used, etc. We addressed the issue of performance deterioration in nearest neighbour imputation due to curse of dimensionality by identifying the feature weights using our proposed hybrid method based on WOA and LAHCA. Since our approach also learns different *k* for different test points using the correlation matrix between the training and test data points for imputation, the missing values are also predicted more accurately. The experiments are conducted with ten different EHR datasets on three supervised machine learning models and the features are weighted and imputed as explained in this paper. The results demonstrate the effectiveness of our proposed approach kNN + LAHCAWOA in comparison with the existing state of the art nearest neighbour and other imputation methods with respect to classification accuracy.

Declaration of interests

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] E.V. Bernstam, J.W. Smith, T.R. Johnson, What is biomedical informatics? J. Biomed. Inform. 43 (2010) 104–110.

[2] USFhealth, What is biomedical informatics?, 2017. https://www.usfhealthonline.com/resources/key-concepts/biomedical-informatics.

[3] P. Schmitt, J. Mandel, M. Guedj, A comparison of six methods for missing data imputation, J. Biometrics Biostat. (2015).

[4] D.S. Central, How to treat missing values in our data, 2017. http://www.datasciencecentral.com/profiles/blogs/how-to-treat-missing-values-in-your-data.

[5] J. Luengo, S. Garcia, F. Herrera, A study on the use of imputation methods for experimentation with radial basis function network classifiers handling missing attribute values: The good synergy between rbfns and eventcovering method, Neural Netw. 23 (2010) 406–418.

[6] J. Luengo, S. García, F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowl. Inform. Syst. 32 (2012) 77–108.

[7] J. Sim, O. Kwon, K.C. Lee, Adaptive pairing of classifier and imputation methods based on the characteristics of missing values in data sets, Expert Syst. Appl. 46 (2016) 485–493.

[8] M. Majidpour, P. Chu, R. Gadh, H.R. Pota, Incomplete data in smart grid: treatment of missing values in electric vehicle charging data, 2014 International Conference on Connected Vehicles and Expo (ICCVE), IEEE, 2014, pp. 1041–1042.

[9] Z. Zhang, Missing data imputation: focusing on single imputation, Ann. Translational Med. 4 (2016).

[10] S.R. Seaman, R.H. Keogh, Handling missing data in matched case-control studies using multiple imputation, Biometrics 71 (2015) 1150–1159.

[11] Z. Zhang, Multiple imputation with multivariate imputation by chained equation (mice) package, Ann. Translational Med. 4 (2016).

[12] E.A. Stuart, M. Azur, C. Frangakis, P. Leaf, Multiple imputation with large data sets: a case study of the children's mental health initiative, Am. J. Epidemiol. (2009) kwp026.

[13] A. Suyundikov, J.R. Stevens, C. Corcoran, J. Herrick, R.K. Wolff, M.L. Slattery, Accounting for dependence induced by weighted knn imputation in paired samples, motivated by a colorectal cancer study, PloS One 10 (2015) e0119876.

[14] Z.-G. Liu, Q. Pan, J. Dezert, A. Martin, Adaptive imputation of missing values for incomplete pattern classification, Pattern Recogn. 52 (2016) 85–95.

[15] K.L. Masconi, T.E. Matsha, J.B. Echouffo-Tcheugui, R.T. Erasmus, A.P. Kengne, Reporting and handling of missing data in predictive research for prevalent undiagnosed type 2 diabetes mellitus: a systematic review, EPMA J. 6 (2015) 1.

[16] L. school of hygience, tropical medicine, Missing data, 2017. http://missingdata.lshtm.ac.uk/.

[17] S. Zhang, X. Wu, M. Zhu, Efficient missing data imputation for supervised learning, 9th IEEE International Conference on Cognitive Informatics (ICCI), 2010, IEEE, 2010, pp. 672–679.

[18] C.J. Carmona, J. Luengo, P. Gonzalez, M. del Jesus, A preliminary study on missing data imputation in evolutionary fuzzy systems of subgroup discovery, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2012, IEEE, 2012, pp. 1–7.

[19] S. Zhang, D. Cheng, Z. Deng, M. Zong, X. Deng, A novel knn algorithm with data-driven k parameter computation, Pattern Recogn. Lett. 109 (2018) 44–54 (Special Issue on Pattern Discovery from Multi-Source Data (PDMSD).

[20] S. Zhang, X. Li, M. Zong, X. Zhu, D. Cheng, Learning k for knn classification, ACM Trans. Intell. Syst. Technol. 8 (2017) 43-1–43-19.

[21] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient knn classification with different numbers of nearest neighbors, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 1774–1785.

[22] G. Bhattacharya, K. Ghosh, A.S. Chowdhury, Granger causality driven ahp for feature weighted knn, Pattern Recogn. 66 (2017) 425–436.

[23] Y. Chen, Y. Hao, A feature weighted support vector machine and k-nearest neighbor algorithm for stock market indices prediction, Expert Syst. Appl. 80 (2017) 340–355.

[24] D. Mateos-García, J. García-Gutiérrez, J.C. Riquelme-Santos, On the evolutionary weighting of neighbours and features in the k-nearest neighbour rule, Neurocomputing (2017).

[25] H.L. Qinghua Wu, X. Yan, Multi-label classification algorithm research based on swarm intelligence, Cluster Comput. (2016).

[26] J. Huang, H. Sun, Grey relational analysis based k nearest neighbor missing data imputation for software quality datasets, IEEE International Conference on Software Quality, Reliability and Security (QRS), 2016, IEEE, 2016, pp. 86–91.

[27] K. Niu, F. Zhao, X. Qiao, A missing data imputation algorithm in wireless sensor network based on minimized similarity distortion, Sixth International Symposium on Computational Intelligence and Design (ISCID), 2013, vol. 2, IEEE, 2013, pp. 235–238.

[28] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[29] G.I. Sayed, A. Darwish, A.E. Hassanien, J.S. Pan, Breast cancer diagnosis approach based on meta-heuristic optimization algorithm inspired by the bubble-net hunting strategy of whales, International Conference on Genetic and Evolutionary Computing, 2017, Springer, 2016, pp. 306–313.

[30] A. Mostafa, A.E. Hassanien, M. Houseni, H. Hefny, Liver segmentation in mri images based on whale optimization algorithm, Multimedia Tools Appl. 76 (2017) 24931–24954.

[31] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.

[32] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453.

[33] E.K. Burke, Y. Bykov, The late acceptance hill-climbing heuristic, Eur. J. Oper. Res. 258 (2017) 70–78.

[34] G.H.G. Fonseca, H.G. Santos, E.G. Carrano, Late acceptance hill-climbing for high school timetabling, J. Sched. 19 (2016) 453–465.

[35] M. Bazargani, J.H. Drake, E.K. Burke, Late acceptance hill climbing for constrained covering arrays, in: EvoApplications 2018: Applications of Evolutionary Computation, 2018, pp. 778–793.

[36] A. Turkyi, N.R. Sabar, A. Sattar, A. Song, Parallel late acceptance hill-climbing algorithm for the google machine reassignment problem, in: AI 2016: Advances in Artificial Intelligence, 2016, pp. 163–174.

[37] M.M. Giraldo, J.S. Sanchez, V.J. Traver, A comparison of techniques for handling incomplete input data with a focus on attribute relevance influence, Ninth International Conference on Machine Learning and Applications (ICMLA), 2010, IEEE, 2010, pp. 819–822.

[38] M. Sharawi, H.M. Zawbaa, E. Emary, H.M. Zawbaa, E. Emary, Feature selection approach based on whale optimization algorithm, in: 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 2017, pp. 163–168.

[39] A.S. Ahmed, M.A. Attia, N.M. Hamed, A.Y. Abdelaziz, Comparison between genetic algorithm and whale optimization algorithm in fault location estimation in power systems, in: 2017 Nineteenth International Middle East Power Systems Conference (MEPCON), 2017, pp. 631–637.

[40] J. Nasiri, F.M. Khiyabani, A whale optimization algorithm (woa) approach for clustering, Cogent Math. Stat. 5 (2018) 1483565.

[41] A. Mukherjee, N. Chakraborty, B.K. Das, Whale optimization algorithm: an implementation to design low-pass fir filter, in: 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, pp. 1–5.

[42] W.Z. Sun, J.S. Wang, X. Wei, An improved whale optimization algorithm based on different searching paths and perceptual disturbance, Symmetry 10 (2018).

[43] G. Tutz, S. Ramzan, Improved methods for the imputation of missing data by nearest neighbor methods, Comput. Stat. Data Anal. 90 (2015) 84–99.

[44] M. Amiri, R. Jensen, Missing data imputation using fuzzy-rough methods, Neurocomputing 205 (2016) 152–164.