An improved K-nearest neighbour with grasshopper optimization algorithm for imputation of missing data

Nadzurah Zainal Abidin [a,1,*], Amelia Ritahani Ismail [a,2]

[a] Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[1] nadzurah.zabidin@gmail.com; [2] amelia@iium.edu.my
* corresponding author

ARTICLE INFO

Article history
Received September 7, 2021
Revised October19, 2021
Accepted November 19, 2021
Available online November 30, 3021

Keywords
Grasshopper
KNN
Imputation accuracy
GOA
Missing data

ABSTRACT

K-nearest neighbors (KNN) has been extensively used as imputation algorithm to substitute missing data with plausible values. One of the successes of KNN imputation is the ability to measure the missing data simulated from its nearest neighbors robustly. However, despite the favorable points, KNN still imposes undesirable circumstances. KNN suffers from high time complexity, choosing the right k, and different functions. Thus, this paper proposes a novel method for imputation of missing data, named KNNGOA, which optimized the KNN imputation technique based on the grasshopper optimization algorithm. Our GOA is designed to find the best value of k and optimize the imputed value from KNN that maximizes the imputation accuracy. Experimental evaluation for different types of datasets collected from UCI, with various rates of missing values ranging from 10%, 30%, and 50%. Our proposed algorithm has achieved promising results from the experiment conducted, which outperformed other methods, especially in terms of accuracy.

1. Introduction

Data is an essential asset for any discipline of work to efficiently analyze in making better decisions. Data is accessible at every edge of life, which provides different insights. The first step in data mining, concerning collecting data, is that a researcher must confront common problems that any data are prone to. Practically, data collected that inclined to noise, incomplete, inconsistence, and redundant are the major source of poor data quality. Besides, more than 40% of datasets embedded in the UCI Machine Learning Repository were missing, extensively used to make an empirical analysis [1]. Missing data can significantly influence the efficacy of the result, which could lead to biased estimates of parameters, loss of information, decreased statistical power, increased standard errors, and weakened generalizability of findings [2]. Missing data is commonly described as a significant issue in most scientific research domains that may originate from mishandling samples, low signal-to-noise ratio, measurement error, non-response, or deleted aberrant value [3]. There are many possible reasons the dataset tolerates missing data, especially when the respondents do not respond due to stress, fatigue, or inadequacy of knowledge. Some of the questions are sensitive and lack option answers [4].

Treatment of missing data has become increasingly significant. Improper handling of missing data could reduce the validity of the conclusion drawn [5]. Therefore, it is crucial to develop a sophisticated algorithm to replace the missing values. Several theories have proposed many solutions to deal with missing data, which can be classified into three categories: (1) case deletion, (2) parameter estimation, and (3) imputation [6][7]. Case deletion is the easiest and commonly known default option in most

statistical analyses. At the same time, parameter estimation implies maximum likelihood techniques to estimate a parameter's value that is most likely to have resulted in the observed data. This method does not impute any data, rather uses each case available to compute maximum likelihood estimates [8]. Although the parameter estimation approach is generally superior to case deletion, these two methods still suffer from high degree complexity, high sensitivity to outliers, and massive lost information. The third category, imputation, replaces the missing values with plausible estimates nearly to the actual values to make the data complete [9]. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Imputation preserves all cases by replacing the missing value with an estimated value based on other available information. Imputation theory is constantly developing, which has caught the attention of statistical and machine learning techniques. A well-known attempt to tackle missing value using statistical techniques is mean imputation. Mean imputation (sometimes called by substitution) replaces missing values by calculating a mean for the variable based on all cases that have data for that variable [10][11]. This technique can lead to bias and underestimates of standard errors. Despite this, machine learning techniques proposed many algorithms to investigate the efficacy of algorithms when dealing with missing data. Machine learning has gained increasing attention to universally solve missing data imputation issues.

The typical imputation strategy regarding K-nearest neighbors (KNN) has been extensively applied to solve the ubiquitous issues in incomplete data. The fundamental idea of KNN can be expressed as a straightforward, robustness, highly efficient, and powerful algorithm that is useful in matching a point with its closest neighbors for all data types, such as continuous, discrete, ordinal, and categorical. KNN imputation has always been known as the lazy and instance-based estimation method [12][13]. The main benefits of KNN imputation are the ability to predict both qualitative and quantitative attributes, easily treat instances with multiple missing values, and consider the correlation structure of the data [6]. Moreover, the success of the KNN imputation algorithm relies on the excellent option of value k. The k in KNN represents the number of nearest neighbors. However, one of the well-known drawbacks of this approach is its inability to deal with high-dimensional and sparse data, which leads to the objective of this paper [14][15]. To overcome the limitation, we proposed to develop an optimization of KNN imputation based on one of the optimization algorithms, the Grasshopper Optimization Algorithm (GOA). A grasshopper optimization algorithm is recent population-based metaheuristics which have shown improved results and efficiencies in tackling issues with missing data [16]. The performances of the proposed algorithm will be compared with other optimization algorithms (Particle Swarm Optimization, Genetic Algorithm, Dragonfly Optimization) in terms of imputation accuracy.

The accuracy obtained from the state-of-art KNN imputation algorithm is not necessarily sufficient until it's proven to handle more versatile KNN with better accuracy. Therefore, this paper proposes a KNN based approach, with an additional optimization algorithm developed to improve the overall performance.

2. Method

2.1. K-nearest neighbors (KNN) Algorithm

K-nearest neighbors (KNN) are universally recognized as one of the most powerful learning algorithms and used for a wide range of real-world applications. The efficacy of the KNN algorithm and its performances mainly depends on the distances or similarity measures and appropriate value for the parameter k [17]–[19].

KNN is the most straightforward algorithm in imputing missing values [20]. This algorithm has been used to solve many predictive problems. In order to impute a value of a variable, KNN defines a set of nearest neighbors for a sample and substitutes the missing data by calculating the average of non-missing values to its neighbors [21]–[23]. There are many merits and demerits of KNN for imputation. However, despite the good points, KNN still imposes undesirable circumstances. KNN suffers from high time complexity, choosing the right k, and different functions.

Many articles, [1][24]–[26], have presented a novel method based on KNN to impute missing data. Most of the experimental work found that KNN efficiently and consecutively shows an accurate imputation on datasets better than any state-of-art algorithms. Besides, an extensive combination of the KNN approach with other ensemble approaches produced the highest robustness and accuracy [27]. Batista and Monard [8] analyzed one preferred standpoint of KNN that is independent of missing data treatment, which makes the algorithm the most suitable imputation for any circumstances.

2.2. Grasshopper Optimization Algorithm (GOA)

Grasshopper Optimization Algorithm (GOA) is a recent swarm intelligence developed by Mirjalili *et al.* [28] and Luo *et al.* [29] that mimics the behavior of grasshopper swarms in nature. The grasshopper is an insect that can be considered a pest due to its nature damaging crop production and agriculture. These creatures are commonly found to be seen individually. However, they often join the swarm as one of the largest swarms of all creatures [30][31]. The swarm of grasshoppers is a nightmare for the farmers as the size can be of continental-scale [32][33]. The grasshopper's life cycle passes through three main stages: egg, nymph, and adult (Fig. 1). Another unique quality of the grasshopper swarm is the swarming behavior found in both nymph and adulthood [34]. The nymph grasshopper does not have wings; thus, they slowly eat all vegetation on their path [35]. However, after a period of time, the grasshopper will become an adult with wings to form a swarm in the air and move fast to a large-scale region [30].

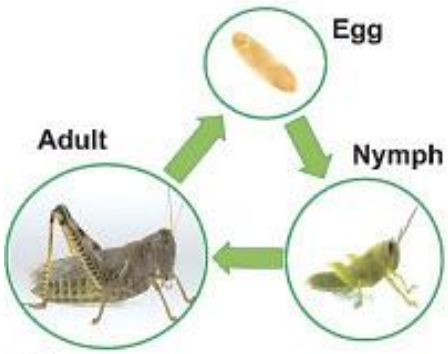

Fig. 1. The life cycle of a grasshopper

The inspiration of GOA comes from the attacking strategy of a grasshopper on corps in the form of swarms. Although they are herbivores, they feed on grasses, leaves, and stems of plants, but when a swarm of grasshoppers infests farms or garden areas, they can cause extensive plant damage and loss. They manage to survive according to the gravitational and wind force so that these factors become helpful for them to attack crops of their target [36] [37]. A grasshopper can easily be at a 'gregarious' state when an increase in the chemical serotonin in certain parts of the nervous system (which boosts mood in humans) initiates the swarming behavior. Besides, as claimed by Melina [38], a solitary grasshopper could be made gregarious within 2 hours simply by tickling their hind legs to simulate the jostling they experience in the wild. Grasshopper optimization algorithm could be visualized as seen in Fig. 2.

According to the US Department of Agriculture (USDA), a swarm of grasshoppers is punctual despite their structured formation. They strictly swarm to migrate in search of food between 10 am and 6 pm. There are clear skies, and the temperature has risen to at least 75 degrees Fahrenheit (24 degrees Celsius). Moreover, a grasshopper is reported as a very structured swarm as a way it joins the formation and flies in an organized way as a member of the swarm when approached by a dense group of flying grasshoppers, although a single grasshopper merely flying follows its random path [39].

For this study, GOA favors KNN imputation methods by surviving to avoid local optima and finding the global space in the given space. Nevertheless, GOA beneficially balances exploration and exploitation to drive grasshoppers towards the global optimum. A fundamental assumption of GOA that may improve the processes of KNN imputation can be found in the way GOA finds its optimum solution. KNN estimates a value from its nearest neighbors while GOA has a high avoidance to find a solution between a set of neighborhoods and provides a solution among all possible solutions. Besides, one of the

limitations of KNN imputation is that the algorithm searches through all datasets for estimating most similar instances, which takes a great deal of time. GOA favors KNN imputation in the sense of time complexity, where one of the main characteristics of grasshopper in the adulthood phase is long-range and abrupt movement.

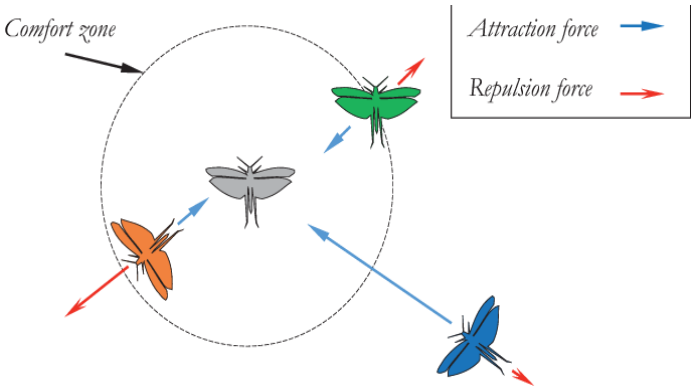

Fig. 2. Grasshopper Optimization Algorithm

Three forces influence the position of each swarm grasshopper. The three forces are social interaction between an individual grasshopper and another grasshopper, S_i; the gravity force on grasshopper, G_i; and Ai's wind advection. The mathematical model of the three forces and simulated grasshopper behaviors are presented as follows:

$$X_i = S_i + G_i + A_i \tag{1}$$

Note that to provide random behavior the equation can be written in $X_i = r_1 S_i + r_2 G_i + r_3 A_i$ where r_1, r_2, and r_3 are random number in [0,1].

$$S_i = \sum_{j=1}^{N} S(d_{ij}) \hat{d}_{ij} \tag{2}$$

Where d_{ij} is the distance between the i-th and the j-th grasshopper, calculated as $d_{ij}=|x_j - x_i|$, s is a function to define the strength of social forces in equation 3, and $(d_{ij})\hat{} = (x_j - x_i) / d_{ij}$ is a unit vector from the i-th grasshopper to the j-th grasshopper.

The s function, which defines the strength between two social forces, attraction and repulsion between grasshoppers are calculated as follows:

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} \tag{3}$$

Where f, l are the intensity of the attraction and the attractive length scale. Social behavior is affected by changing the parameters f, l.

The second affected force on the position of grasshopper is the gravity force which is calculated as follows:

$$G_i = -g\widehat{e_g} \tag{4}$$

Where g is the gravitational constant and $(e_g)\hat{}$ is a unity vector towards the center of the Earth. The A component in equation 1 is calculated as follows:

$$A_i = u\widehat{e_w} \tag{5}$$

Where u is constant drift and $(e_w)\hat{}$ is a wind direction unity vector. The nymph grasshopper movements is highly correlated with wind direction because they have no wings. The main process is to impute the dataset with KNN by calculating its nearest neighbors' distance between each missing

data. Then, the imputed value will be optimized with GOA according to the information of the missing dataset.

2.3. Experiments Design

In this section, the nine datasets used for this research are described. Data were acquired from public access websites such as *data.world*, UC Irvine Machine Learning Repository, *Kaggle.com* and Public Library of Science (PLOS One). The description of selected datasets is shown in the table below, including the domain, sources, number of instances, number of attributes, data types, and percentage of missing values.

Table 1. Experimental Data

Dataset	Domain	Sources	Num of instances	Num of attributes	% of missing
Chronic Kidney Disease (KD1)	Medical	UCI ML Repository	35	6	7.23
US. Chronic Disease Indicators (KD2)	Medical	Data world – data.gov	400	26	10.26
HCC Survival	Medical	UCI ML Repository	165	49	10.22
AKI	Medical	PLOS|One	84	24	17.857
EHP	Medical	Data.world	2663	70	38.73
ECG	Medical	Kaggle	132	13	7.69
Blood test analysis	Medical	Kaggle	576	5	20
Automobile	Transportation	UCI ML Repository	204	26	20.59
Air Quality	Engineering	UCI ML Repository	9357	14	27.9

The data used in this paper are from the medical, engineering, and transformation domain. These three domains are claimed to be classified among the most beneficiaries in missing data subject. The nature of imputation was evaluated by comparing the imputed values against original values. The experiments will be computed regarding the accuracy, time complexity, and sensitivity of each imputation method. The parameters to evaluate the performance and measure the error differences between values are by employing Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These parameters are negatively oriented, which implies lower values are better. The three criteria significantly a meaningful representation that computes an error between two numeric vectors. An alternative for the corresponding significance tests is supported with Vargha – Delaney A test. The A test helps to assess the difference between two populations concerning a variable. Upon testing, each swarm optimization algorithm shall be compared to determine which results are greater or smaller from the KNN-imputed values [40].

3. Results and Discussion

The following table shows the analysis done to examine the performance of four machine learning algorithms performance: Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Bayesian Network. Imputation with machine learning algorithm performs better than any statistical tools considering that machine learning is more flexible with better predictive accuracy. Nonetheless, a standard range of machine learning imputation algorithms will still introduce vague analysis results [41].

3.1. Statistical Correlation

In performing the visualization of all datasets between actual and predicted, a scatterplot is chosen to help illustrate a relationship between two variables. In a scatterplot, the points can discern a clear trend in the data. All the scatterplot figures in this subsection will visualize the differences between the actual and the imputed values for all seven medical datasets explained in the previous section.

A good scatterplot is best defined as the closer the data points forming a straight line from the origin out to high y-values. Besides, the best fit for this description is a strong, linear, and positive association between the two variables. Fig. 3 to Fig. 11 illustrates of seven scatterplot correlations for all nine

datasets, which are KD1 (Fig. 3), KD2 (Fig. 4), HCC survival (Fig. 5), AKI (Fig. 6), EHP phthalates (Fig. 7), ECG (Fig. 8), Blood test (Fig. 9), Automobile (Fig. 10), and Air Quality (Fig. 11).

The results in Fig. 3 demonstrate two things. First, there is a positive linear association between two variables for all subfigures. Second, for Fig. 3(b) and (d), the association looks weaker compared to Fig. 3(a) and (c). This result concludes that only the GOA algorithm presents a higher correlation for actual and imputed values after being optimized by a conventional KNN imputation algorithm.

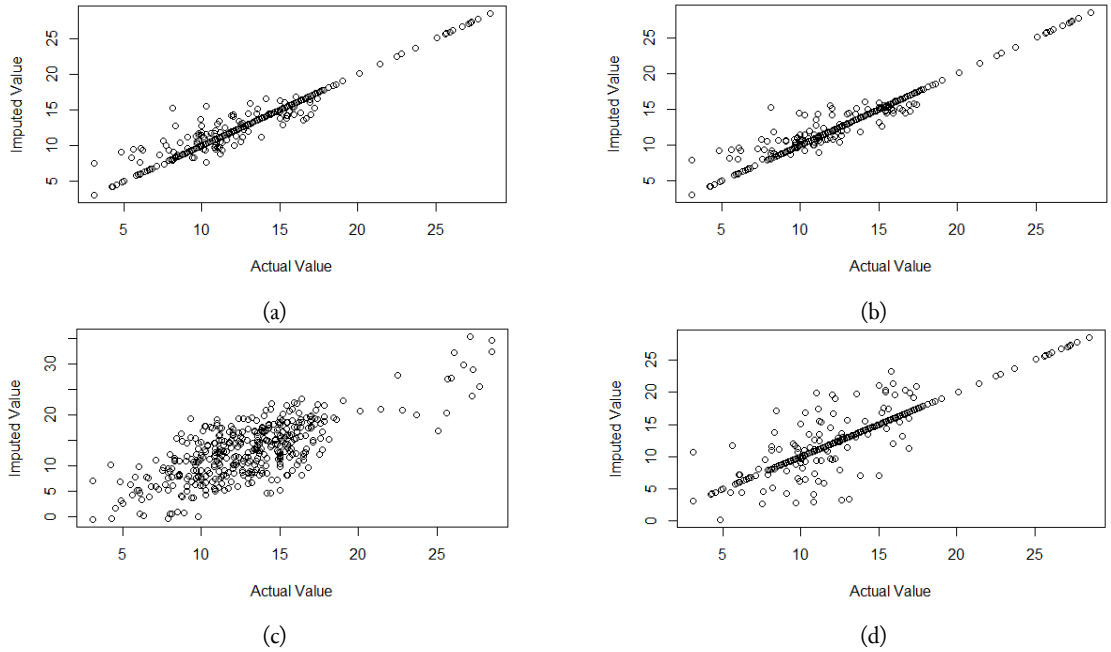

Fig. 3. Scatterplot for KD1 Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA

The result in Fig. 4 shows that the pattern is significantly identical between the actual value and imputed values for all metaheuristics algorithms.

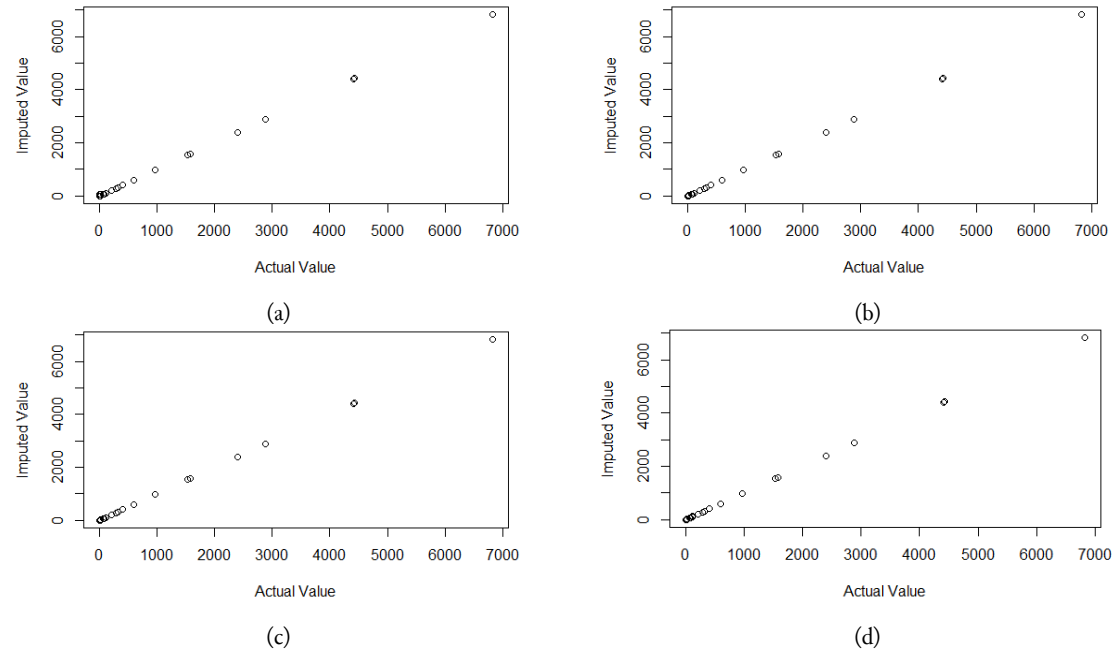

Fig. 4. Scatterplot for KD2 Datasets; (a) scatterplot for KNN, (b) scatterplot for PSO, (c) scatterplot for GOA, (d) scatterplot for DA

Fig. 5 highlights the trend of positive linear association but weak strength correlation between two variables. Among all subfigures in Fig. 5, only GOA shows a positive, linear, and strong relationship between actual and imputed values. It signifies that both variables move in the same direction and are correlated for GOA.

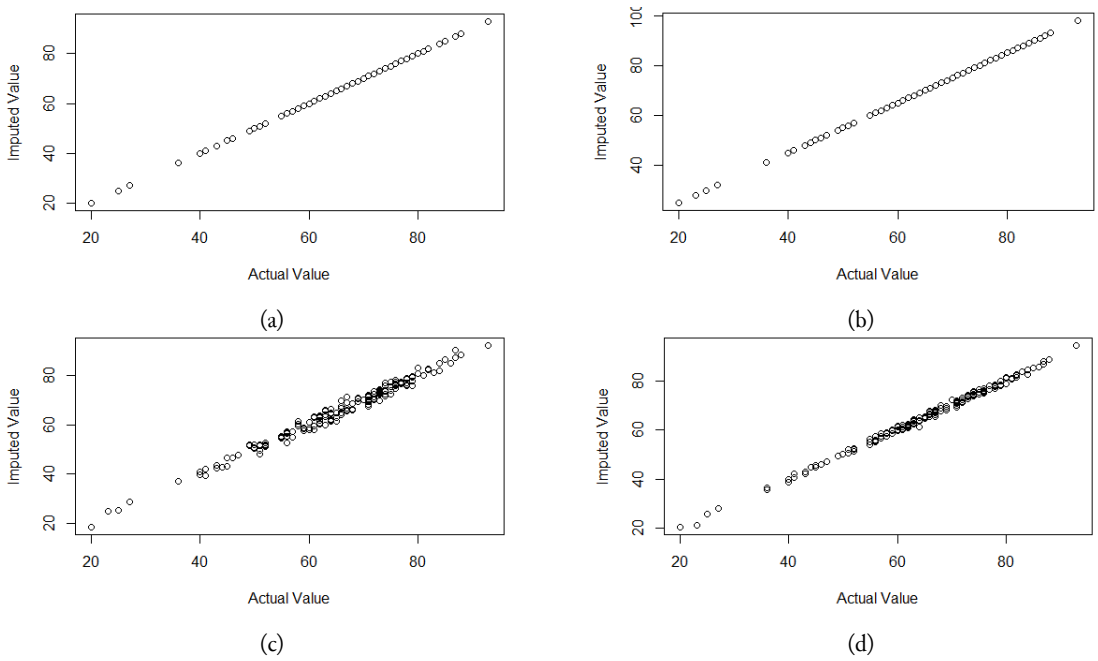

Fig. 5. Scatterplot for HCC Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA.

The result in Fig. 6 demonstrates a moderate, positive, and linear relationship between actual and imputed values for the GOA algorithm. Unlike GOA, the scatterplot for all other metaheuristics algorithms displays a diverse form of correlations which can be statistically considered as no relationship measured.

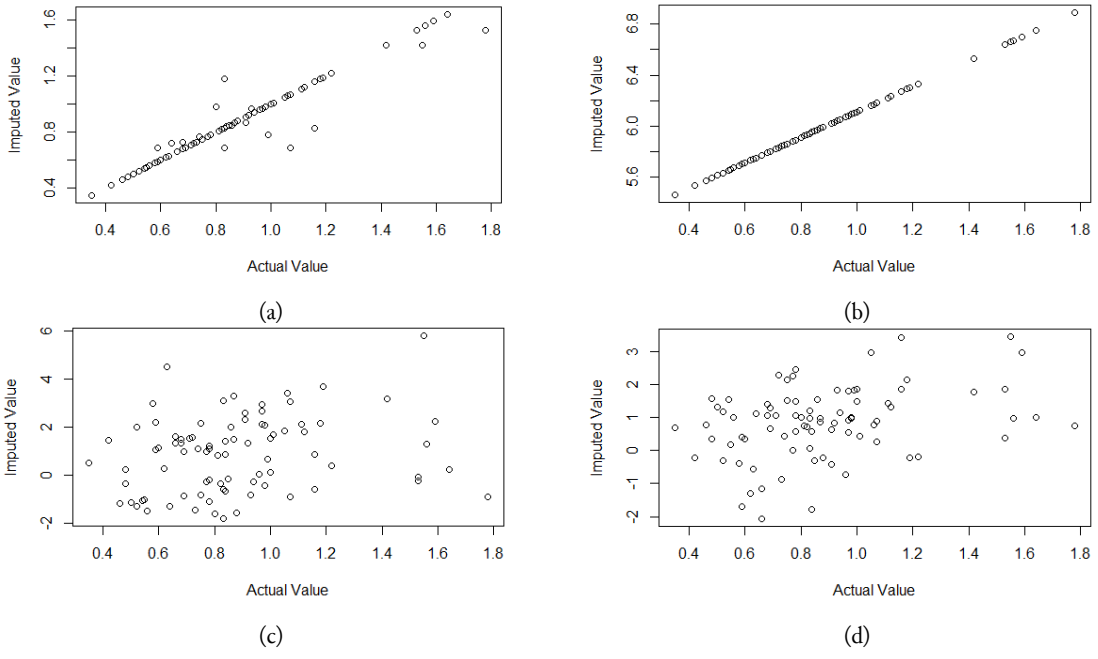

Fig. 6. Scatterplot for AKI Datasets; (a) scatterplot for KNN, (b) scatterplot for PSO, (c) scatterplot for GOA, (d) scatterplot for DA.

Fig. 7 shows one similar pattern for all eight metaheuristics algorithms where specifically, the data has a general look of a line going uphill. The finding best describes that it shows a positive linear association between two variables, actual and imputed values. Besides, to assess the relationship between the variables, Fig. 7(b), (c), and (d), shows a stronger relationship, compared to Fig. 7(a), which means higher correlation.

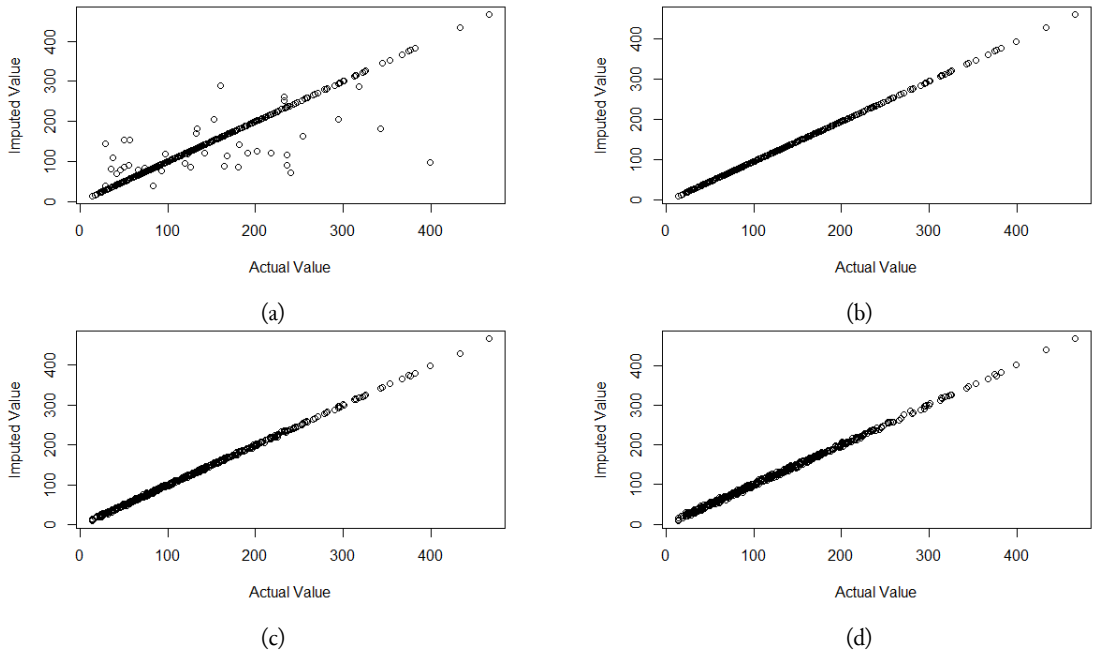

Fig. 7. Scatterplot for EHP Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA

Fig. 8 illustrates various relationships between the two variables for all metaheuristics algorithms. All points in the scatterplot Fig. 8 are far remotely to a straight line. However, Fig. 8(b), and (c) display a weak positive correlation which indicates that they tend to go up in response to one another for both variables, but the relationship is not very strong.

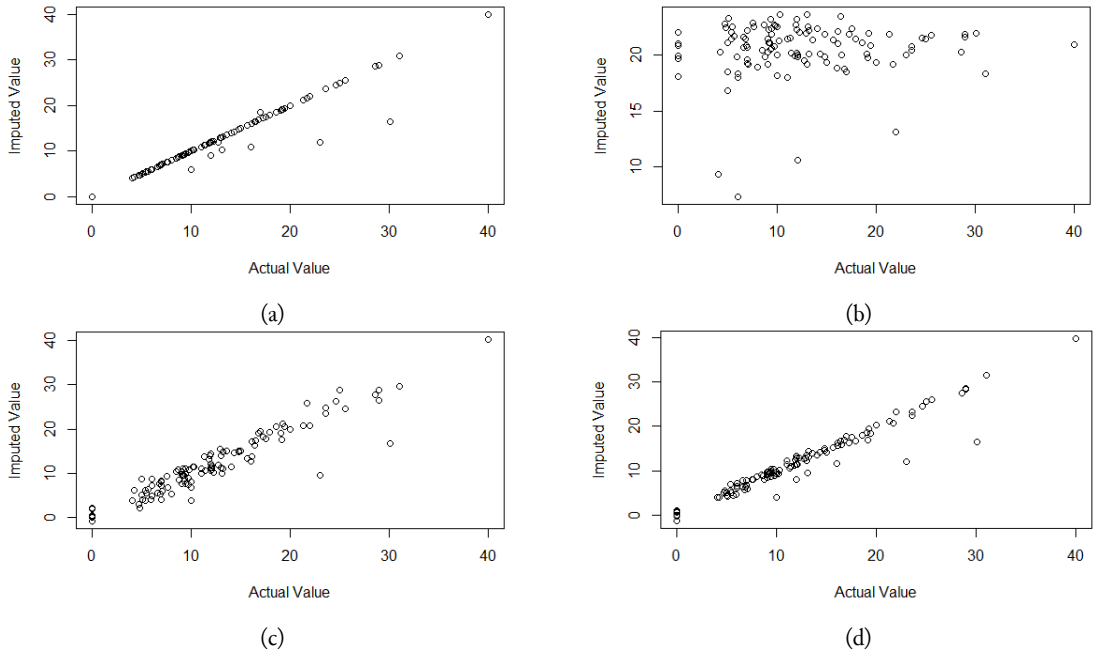

Fig. 8. Scatterplot for ECG Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA.

Among all the subfigures in Fig. 9, GOA has demonstrated the perfect positive, linear, and strong relationship for both variables. However, Fig. 9(b) and (c) illustrate a positive and moderate correlations.

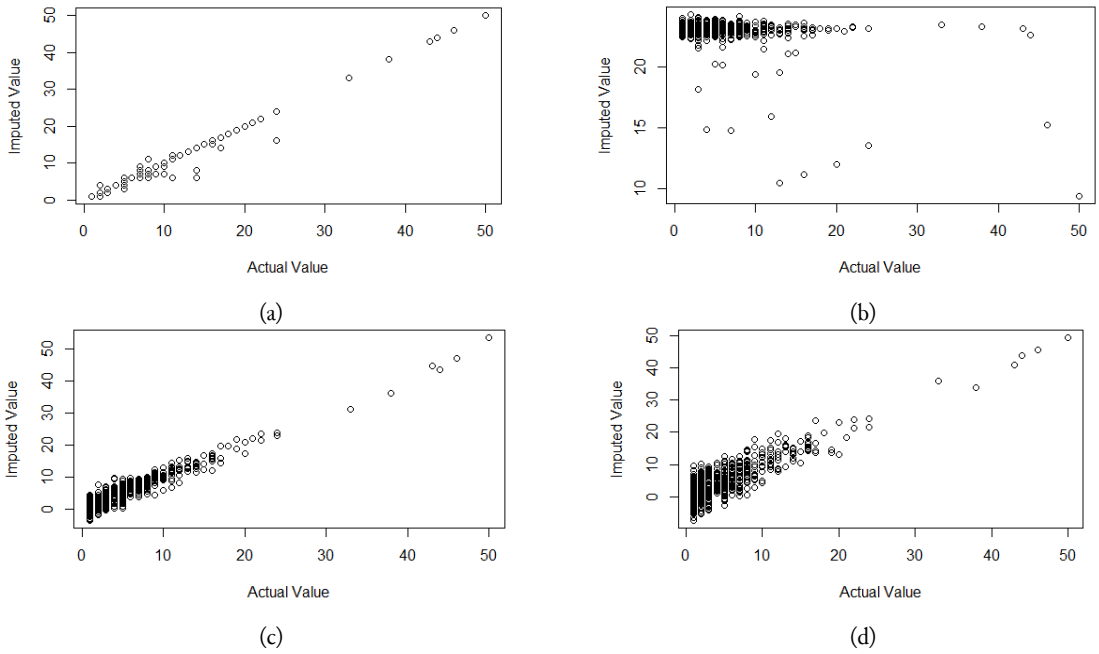

Fig. 9. Scatterplot for Blood test Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA.

Fig. 10 clearly illustrates that only Fig. 10(b) demonstrated a different trend, which is weakly correlated. As shown in Fig. 10(a), (c), and (d), the graph describes a strong correlated positive linear relationship between the two variables.

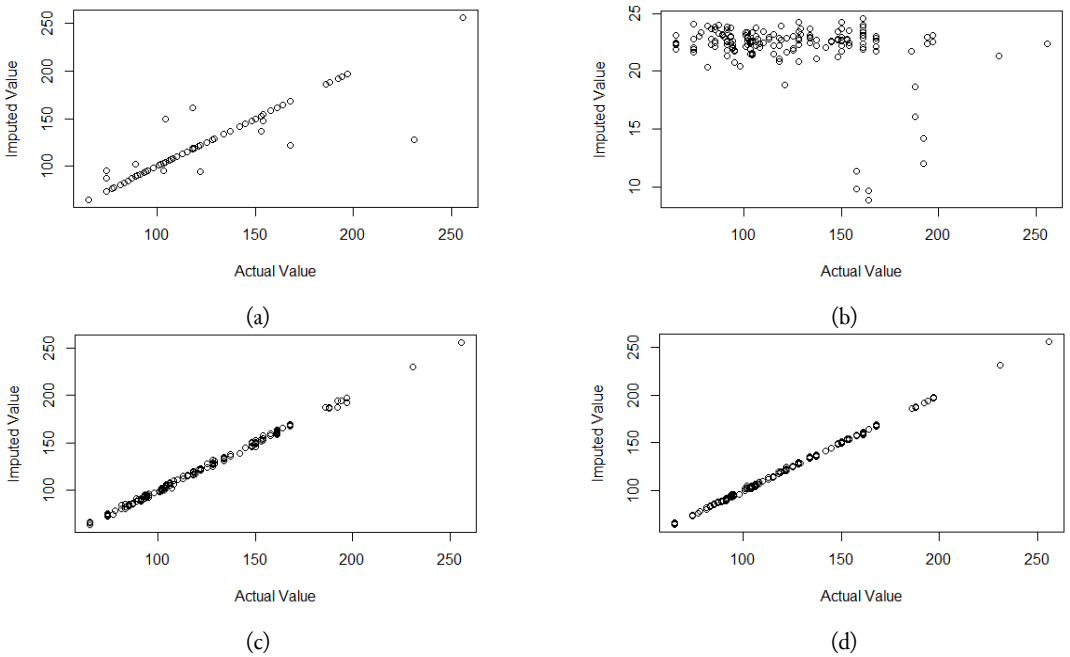

Fig. 10. Scatterplot for Automobile Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA.

Fig. 11 shows a close trend for all eight algorithms. Fig. 11 (c) and (d) points out an identical strongly correlated, positive, and linear relationship. The evidence is by the much cleaner line formed by the data points.

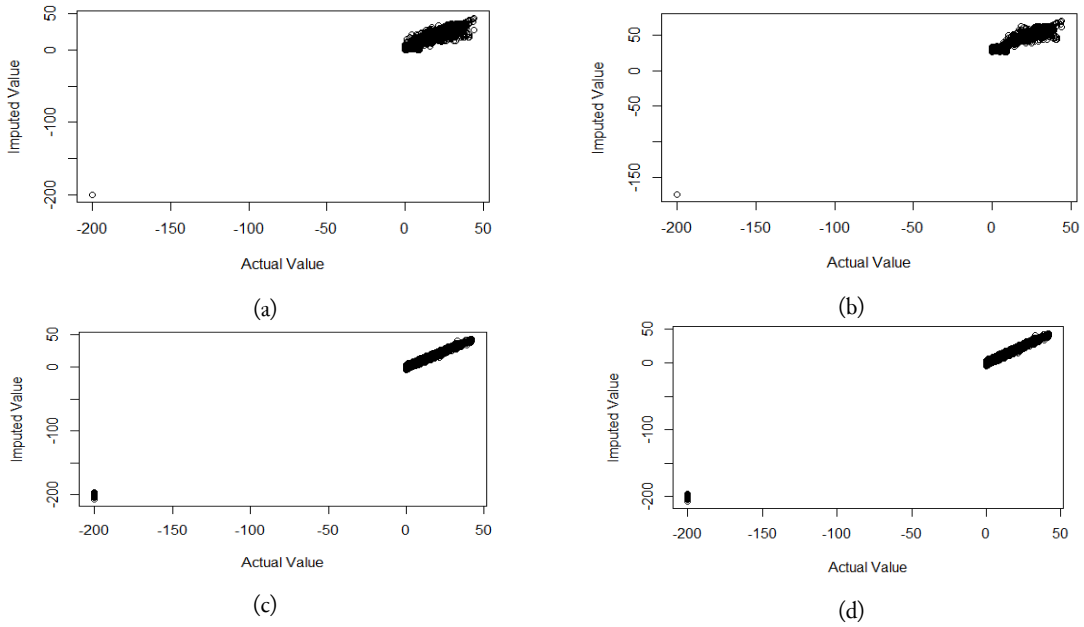

Fig. 11. Scatterplot for Air Quality Datasets; (a) scatterplot for KNN, (b) scatterplot for KNNGA, (c) scatterplot for KNNPSO, (d) scatterplot for KNNGOA

To conclude, relying the interpretation on scatterplot only is individually biased. Therefore, extensive experiments are carried out to support the discussion made in the following section.

3.2. Error Accuracy

In general, the results highlighted for error accuracy is that the most promising finding was an optimization of KNN with Grasshopper Optimization Algorithm (GOA). KNNGOA showed the lowest error accuracy for all nine datasets regarding the size of datasets and missing value rates, except for the ECG heartbeat dataset.

Table 4 describes all four relative error parameters. KNNGOA consistently demonstrated impressive performance, providing a low error accuracy from the KNN imputation algorithm. According to Table 4, ECG heartbeat and Air Quality datasets display an unusual result for MAPE. For all algorithms, KNN and eight metaheuristics-based KNN algorithms, the results imply that the function will return –Inf, Inf, or NaN if actual is instability at or near zero.

Table 2. Error accuracy for all datasets

Dataset	ML Algorithm	MAE	MSE	RMSE	MAPE
KD1	KNN	1.6561	4.4834	2.1174	0.1402
	KNN GA	12.054	164.29	12.817	1.1355
	KNNPSO	3.1511	14.682	3.8318	0.2714
	KNNGOA	0.8947	1.5695	1.2523	0.0751
KD2	KNN	11.914	831.28	28.832	4.1541
	KNNGA	7.75E+2	2.99E+6	1.73E+3	4.9291
	KNNPSO	10.974	14.338	9.2211	8.4302
	KNNGOA	1.3426	2.8493	1.6880	0.2437
HCC Survival	KNN	1.7878	32.491	5.7000	0.0355
	KNNGA	5.1198	26.213	5.1198	0.0841
	KNNPSO	1.3284	2.6266	1.6207	0.0216
	KNNGOA	0.6595	0.6621	0.8136	0.0109

Table 2. (Continued)

Table 2. (Continued)

Dataset	ML Algorithm	MAE	MSE	RMSE	MAPE
AKI	KNN	2.7619	0.6838	8.2693	2.8554
	KNNGA	5.1111	26.124	5.1111	6.4715
	KNNPSO	1.2897	2.3314	1.5269	1.5847
	KNNGOA	0.7773	1.0139	1.0069	0.9703
EHP	KNN	5.2444	661.00	25.709	0.0386
	KNNGA	5.1151	26.163	5.1151	0.0637
	KNNPSO	2.3977	9.9731	3.7243	0.0524
	KNNGOA	2.3719	8.2633	2.8746	0.0303
ECG	KNN	7.3614	5.1679	9.7798	NaN
	KNNGA	9.9721	127.52	11.293	Inf
	KNNPSO	0.9266	3.8344	1.9581	Inf
	KNNGOA	1.5718	5.9489	2.4390	Inf
Blood Test Analysis	KNN	0.1093	0.4427	0.6653	0.0150
	KNNGA	18.133	349.33	18.690	8.2347
	KNNPSO	1.3389	2.8845	1.6983	0.5356
	KNNGOA	0.0259	0.0105	0.0322	0.0102
Automobile	KNN	2.1595	114.56	10.703	0.0163
	KNNGA	1.00E+2	1.13E+4	1.06E+2	8.02E-1
	KNNPSO	1.2505	2.5725	1.6039	0.0109
	KNNGOA	0.6461	0.6751	0.8221	0.0057
Air Quality	KNN	6.7390	784.88	28.015	NaN
	KNNGA	26.169	689.78	26.263	Inf
	KNNPSO	1.4398	3.5005	1.8709	Inf
	KNNGOA	1.3865	2.7723	1.7932	Inf

3.3. Computation Time

Another investigation that governs the efficiency of an algorithm is by measuring the computation time. The tradeoff between error accuracy and time complexity is considered by comparing the results. Table 3 shows that 4 out of 9 datasets have the fastest time using GOA. Time computation tradeoff refers to slow execution time in exchange for the lowest error accuracy. Although only four datasets show GOA achieved as the fastest computation time, GOA still appeared and achieved as the high accuracy.

Table 3. Computation time for all dataset

Dataset	KNNGA	KNNPSO	KNNGOA
KD1	10.38	11.16	40.19
KD2	22.34	18.82	17.9
HCC Survival	12.3	12.05	**7.54**
AKI	1.05	1.68	**1.19 min**
EHP	53.34	2.23 min	30.39
ECG Heartbeat	12.43	11.84	**11.59**
Blood test analysis	24.73	25.90	23.45
Automobile	15.89	17.45	**13.96**
Air Quality	3.14 hrs	3.58 hrs	43.59 min

4. Conclusion

In this paper, we present a novel method that improves imputation performance based on K-nearest neighbors by using the Grasshopper Optimization Algorithm (GOA). The hybrid model KNNGOA is applied to optimize the imputation algorithm and missing value problems. It is essential because any analysis can draw an inaccurate inference due to the missing value. Experiments are conducted to evaluate the imputation accuracy of the proposed KNNGOA on the five real-world datasets from all public websites. According to three different evaluation criteria, error accuracy, statistical test, and time computing, the proposed KNNGOA constantly outperforms and performs better than other algorithms. Currently, the proposed solution is time-consuming because the training procedure for GOA is repeated many times to find the optimal solution and attribute weights for big datasets. Therefore, some modifications are needed as a tradeoff, thus reducing the computational time. In future work, we attempt to tailor the model for big datasets by concurrently applying a solution of speeding up the training time of KNN by using some methods to reduce the size of datasets.

Acknowledgment

This research is financially supported by a grant from the Ministry of Higher Education Malaysia: Fundamental Research Grant Scheme (FRGS/1/2018/ICT02/UIAM/02/1).

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. This research is financially supported by a grant from the Ministry of Higher Education Malaysia: Fundamental Research Grant Scheme (FRGS/1/2018/ICT02/UIAM/02/1).
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] R. Pan, T. Yang, J. Cao, K. Lu, and Z. Zhang, "Missing data imputation by K nearest neighbours based on grey relational structure and mutual information," *J. Appl. Intell.*, vol. 43, no. 3, pp. 614–632, 2015. doi: 10.1007/s10489-015-0666-x.

[2] A. Ngueilbaye, H. Wang, D. A. Mahamat, and S. B. Junaidu, "Modulo 9 model-based learning for missing data imputation," *Appl. Soft Comput.*, vol. 103, p. 107167, 2021, doi: 10.1016/j.asoc.2021.107167.

[3] V. Agarwal, "Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis," *Int. J. Comput. Appl.*, vol. 131, no. 4, pp. 30–36, 2015. doi: 10.5120/ijca2015907309.

[4] N. Z. Zainal Abidin, A. R. Ismail, and N. A. Emran, "Performance Analysis of Machine Learning Algorithms for Missing Value Imputation," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, 2018. doi: 10.14569/IJACSA.2018.090660.

[5] B. J. Wells *et al.*, "Strategies for Handling Missing Data in Electronic Health Record Derived Data," *EDM Forum Community*, vol. 1, pp. 12–17, 2013. doi: 10.13063/2327-9214.1035.

[6] P. Rani, R. Kumar, and A. Jain, "Multistage Model for Accurate Prediction of Missing Values Using Imputation Methods in Heart Disease Dataset," in *Innovative Data Communication Technologies and Application*, 2021, pp. 637–653. doi: 10.1007/978-981-15-9651-3_53.

[7] N. A. M. Pauzi, Y. B. Wah, S. M. Deni, S. K. N. A. Rahim, and Suhartono, "Comparison of single and mice imputation methods for missing values: A simulation study," *Pertanika J. Sci. Technol.*, vol. 29, no. 2, pp. 979–998, 2021. doi: 10.47836/pjst.29.2.15.

[8] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, no. 5–6, pp. 519–533, 2003. doi: 10.1080/713827181.

[9] S. Faisal and G. Tutz, "Multiple imputation using nearest neighbor methods," *Inf. Sci. (Ny).*, vol. 570, pp. 500–516, 2021. doi: 10.1016/j.ins.2021.04.009.

[10] H. M. Dodeen, "Effectiveness of Valid Mean Substitution in Treating Missing Data in Attitude Assessment," *Assess. Eval. High. Educ.*, Vol. 28, No. 5, p. 505-513, 2003. doi: 10.1080/02602930301674.

[11] J. W. Graham, "Analysis of Missing Data," in *Missing Data: Analysis and Design*, Statistics for Social and Behavioral Sciences, Ed. New York: Springer Science + Business Media, 2012, pp. 47–68. doi: 10.1007/978-1-4614-4018-5_2.

[12] A. Lamba and D. Kumar, "Survey on KNN and Its Variants," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, no. 5, 2016. doi: 10.17148/IJARCCE.2016.55101.

[13] S. Zhang, "Nearest neighbor selection for iteratively k NN imputation," *J. Syst. Softw.*, vol. 85, no. 11, pp. 2541–2552, 2012. doi: 10.1016/j.jss.2012.05.073.

[14] O. Kramer, "Dimensionality Reduction by Unsupervised K-Nearest Neighbor Regression," *Int. Conf. Mach. Learn. Appl.*, pp. 2–5, 2011. doi: 10.1109/ICMLA.2011.55.

[15] N. A. B. Kamisan, M. H. Lee, A. G. Hussin, and Y. Z. Zubairi, "Imputation techniques for incomplete load data based on seasonality and orientation of the missing values," *Sains Malaysiana*, vol. 49, no. 5, pp. 1165–1174, 2020. doi: 10.17576/jsm-2020-4905-22.

[16] H. Singh and B. Singh, "A Comparison of Optimization Algorithms for Standard Benchmark Functions," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 7, pp. 1249–1254, 2017. doi: 10.26483/ijarcs.v8i7.4581.

[17] S. Jiang, G. Pang, M. Wu, and L. Kuang, " An improved K-nearest-neighbor algorithm for text categorization," *Expert Syst. Appl.,* Vol. 39, No. 1, p. 1503-1509, 2012. doi: 10.1016/j.eswa.2011.08.040.

[18] H. A. E Alfeilat, A. B. A. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. E. Salman, and V. B. S. Prasath " Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review," *Big Data,* Vol. 7, No. 4, pp. 1–50, 2019. doi: 10.1089/big.2018.0175.

[19] C. D. Yu and B. Xiao, "Performance Optimization for the K Nearest-Neighbor Kernel on x86 Architectures," *Proceeding Int. Conf. High Perform. Comput. Networking, Storage Anal.*, 2015. doi: 10.1145/2807591.2807601.

[20] M. Shahjaman, M. R. Rahman, T. Islam, M. R. Auwul, M. A. Moni, and M. N. H. Mollah, “rMisbeta: A robust missing value imputation approach in transcriptomics and metabolomics data,” *Comput. Biol. Med.*, vol. 138, p. 104911, 2021. doi: 10.1016/j.compbiomed.2021.104911.

[21] S. P. Mandel J, "A Comparison of Six Methods for Missing Data Imputation," *J. Biom. Biostat.*, vol. 6, no. 1, pp. 1–6, 2015. doi: 10.4172/2155-6180.1000224.

[22] S. I. Khan and A. S. M. L. Hoque, "SICE: an improved missing data imputation technique," *J. Big Data*, vol. 7, no. 1, 2020. doi: 10.1186/s40537-020-00313-w.

[23] K. K. Sharma and A. Seal, “Spectral embedded generalized mean based k-nearest neighbors clustering with S-distance,” *Expert Syst. Appl.*, vol. 169, p. 114326, 2021. doi: 10.1016/j.eswa.2020.114326.

[24] P. Jönsson and C. Wohlin, "An evaluation of k-nearest neighbour imputation using lIkert data," in *Proceedings - International Software Metrics Symposium*, 2004, pp. 108–118. doi: 10.1109/METRIC.2004.1357895.

[25] S. Oehmcke, O. Zielinski, and O. Kramer, "kNN ensembles with penalized DTW for multivariate time series imputation," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016–Octob, pp. 2774–2781, 2016. doi: 10.1109/IJCNN.2016.7727549.

[26] M. Tabassian, M. Alessandrini, R. Jasaityte, L. De Marchi, G. Masetti, and J. D'Hooge, "Handling missing strain (rate) curves using K-nearest neighbor imputation," *IEEE Int. Ultrason. Symp. IUS*, vol. 2016–Novem, pp. 1–4, 2016. doi: 10.1109/ULTSYM.2016.7728809.

[27] M. Askarian, G. Escudero, M. Graells, R. Zarghami, F. Jalali-Farahani, and N. Mostoufi, "Fault diagnosis of chemical processes with incomplete observations: A comparative study," *Comput. Chem. Eng.*, vol. 84, pp. 104–116, 2016. doi: 10.1016/j.compchemeng.2015.08.018.

[28] S. Mirjalili, P. Jangir, and S. Saremi, "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," *Appl. Intell.*, vol. 46, no. 1, pp. 79–95, Jan. 2017. doi: 10.1007/s10489-016-0825-8.

[29] J. Luo, H. Chen, Y. Xu, H. Huang, and X. Zhao, "An Improved Grasshopper Optimization Algorithm with Application to Financial Stress Prediction," *Appl. Math. Model.*, 2018. doi: 10.1016/j.apm.2018.07.044.

[30] A. G. Neve, G. M. Kakandikar, and O. Kulkarni, "Application of Grasshopper Optimization Algorithm for Constrained and Unconstrained Test Functions," *Int. J. Swarm Intell. Evol. Comput.*, vol. 6, no. 3, 2017. doi: 10.4172/2090-4908.1000165.

[31] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm : Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017. doi: 10.1016/j.advengsoft.2017.01.004.

[32] M. J. Zeynali and A. Shahidi, "Performance Assessment of Grasshopper Optimization Algorithm for Optimizing Coefficients of Sediment Rating Curve," *AUT J. Civ. Eng.*, vol. 2, no. 1, pp. 39–48, 2018. doi: 10.22060/ajce.2018.14511.5480.

[33] L. Abualigah and A. Diabat, "A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications," *Neural Comput. Appl.*, vol. 32, no. 19, pp. 15533–15556, 2020. doi: 10.1007/s00521-020-04789-8.

[34] S. M. Rogers, T. Matheson, E. Despland, T. Dodgson, M. Burrows, and S. J. Simpson, "Mechanosensory-induced behavioural gregarization in the desert locust Schistocerca gregaria," *J. Exp. Biol.*, vol. 206, no. 22, pp. 3991–4002, 2003. doi: 10.1242/jeb.00648.

[35] M. Mafarja *et al.*, "Evolutionary Population Dynamics and Grasshopper Optimization Approaches for Feature Selection Problems," *Knowledge-Based Syst.*, no. December, 2017. doi: 10.1016/j.knosys.2017.12.037.

[36] M. Singh, V. M. Srivastava, K. Gaurav, and P. K. Gupta, "Automatic test data generation based on multi-objective ant lion optimization algorithm," in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, 2017, pp. 168–174. doi: 10.1109/RoboMech.2017.8261142.

[37] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 4385–4405, 2019. doi: 10.1007/s00521-018-3343-2.

[38] R. Melina, "What makes grasshoppers swarm?," *Live Science*, 2010. [Online]. Available: https://www.livescience.com/32609-what-makes-grasshoppers-swarm.html.

[39] S. Lukasik, P. A. Kowalski, M. Charytanowicz, and P. Kulczycki, "Data clustering with grasshopper optimization algorithm," *Proc. 2017 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2017*, vol. 11, pp. 71–74, 2017. doi: 10.15439/2017F340.

[40] H. D. Delaney and A. Vargha, "A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong," *J. Educ. Behav. Stat.*, vol. 25, no. 2, pp. 101–132, 2000. doi: 10.3102/10769986025002101.

[41] M. Najib and N. A. Samat, "FCMPSO : An Imputation for Missing Data Features in Heart Disease Classification," *IOP Conf. Ser. Mater. Sci. Eng.*, 2017. doi: 10.1088/1757-899X/226/1/012102.