A particle swarm optimization levy flight algorithm for imputation of missing creatinine dataset

Amelia Ritahani Ismail [a,1,*], Normaziah Abdul Aziz [a,2], Azrina Md Ralib [b,3], Nadzurah Zainal Abidin [a,4], Samar Salem Bashath [a,5]

[a] Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[b] Department of Anaesthesiology, Kulliyyah of Medicine, International Islamic University Malaysia, Kuantan Pahang, Malaysia
[1] amelia@iium.edu.my; [2] naa@iium.edu.my; [3] azrinar@iium.edu.my; [4] nadzurah.abidin@gmail.com; [5] samarsalim7076@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received May 11, 2021
Revised June 8, 2021
Accepted July 30, 2021
Available online July 31, 2021

Keywords
Baseline creatinine
Imputation
Missing data
Particle swarm optimization
Levy flight

ABSTRACT

Clinicians could intervene during what may be a crucial stage for preventing permanent kidney injury if patients with incipient Acute Kidney Injury (AKI) and those at high risk of developing AKI could be identified. This paper proposes an improved mechanism to machine learning imputation algorithms by introducing the Particle Swarm Levy Flight algorithm. We improve the algorithms by modifying the Particle Swarm Optimization Algorithm (PSO), by enhancing the algorithm with levy flight (PSOLF). The creatinine dataset that we collected, including AKI diagnosis and staging, mortality at hospital discharge, and renal recovery, are tested and compared with other machine learning algorithms such as Genetic Algorithm and traditional PSO. The proposed algorithms' performances are validated with a statistical significance test. The results show that SVMPSOLF has better performance than the other method. This research could be useful as an important tool of prognostic capabilities for determining which patients are likely to suffer from AKI, potentially allowing clinicians to intervene before kidney damage manifests.

1. Introduction

Acute Kidney Injury (AKI) is a sudden episode of abrupt loss of kidney failure within a few hours and or a few days. A well-described definition by Kidney Disease Improving Global Outcomes (KDIGO) on AKI is a syndrome of diverse etiology that is characterized by a rapid decline in the glomerular filtration rate (GFR) [1], [2]. AKI is a common disease in hospitalized patients and has a high mortality due to the severity of injury associated with poor outcomes [3]. Besides, AKI has been recognized as a global public health problem, with roughly over 50 percent of AKI mortality occurred in intensive care unit (ICU) settings [4]. Despite being adequately associated with higher mortality, a high incidence of AKI may also be attributed to sepsis, about 60 percent of patients reported in Malaysia. However, the burden of AKI in hospitalized patients is vastly underestimated, especially in developing countries. Even though data generation was detailed and reliable, the underestimation was obvious, let alone the diagnosis rate by disease code.

Sunday Star (2017) reported that between two and three million Malaysians currently suffer from chronic kidney disease (CKD) and are expected to rise. Besides, the health ministry also adds that almost 20,000 patients with kidney failure are on a waiting list for treatment and dialysis. According to a study

entitled "Forecasting the Incidence and Prevalence of Patients with End-Stage Renal Disease in Malaysia up to the year 2040", Malaysia is ranked as the top seventh-highest dialysis treatment rate globally.

Acute kidney failure usually happens when kidneys lose the ability to eliminate excess salts, fluids, and waste materials from the blood. This eliminations process is the core of the kidney's primary function. AKI is a kidney disease that can lead to stroke, heart attacks, and other serious diseases. This kidney disorder is defined by an abrupt decrease in kidney function at Kidney Disease: Improving Global Outcomes (KDIGO) AKI Guideline [1]. This disorder is a sudden event of kidney failure that may happen within a few hours or even a few days. Patients with AKI need special attention and care, especially with their records such as creatinine and urine values. Accordingly, AKI stages are defined by the maximum serum creatinine or urine output [5]. Missing those (creatinine and urine values) is a common obstacle to access AKI [6]. This common obstacle imposes surrogate estimates, leads to poor estimation of kidney function, misclassifies AKI, and adversely affects the study of associated outcomes [7].

Machine learning algorithms and optimization algorithms are successful approaches employed in the recent decade to treat missing baseline creatinine [8]. These approaches allow estimating of missing data for statistical analysis. Therefore, this research proposed improved machine learning with particle swarm optimization techniques enhanced with a Levi flight.

2. Method

Creatinine and urine values are frequently missing in AKI studies. Therefore, this paper aimed to identify the best machine-learning algorithm to impute for missing creatinine and urine values.

For the methodology, the first step concerns exposing new issues and challenges, and it is instructive to have a variety of problems when considering supervised learning methods (Fig. 1). This phase also identifies different techniques for developing the rules and classification to concentrate on the information needed, such as creatinine and urine values. The estimation process of the dataset is applied to real data stored in the International Islamic University of Malaysia Hospital (HUIAM).

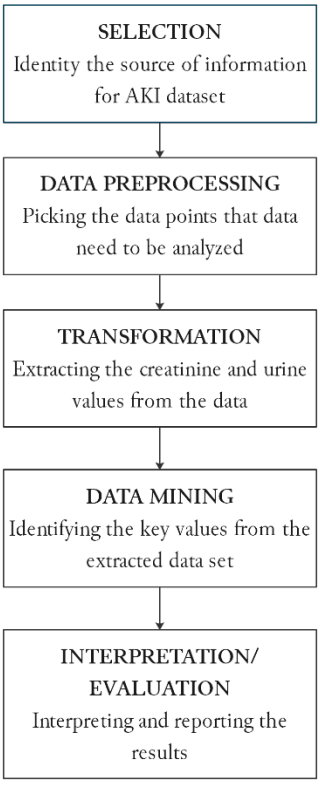

Fig. 1. Methodology for Diagnosis Missing Creatinine

The second step helps to identify the data that need to be analyzed. The Bayesian approach relies on data collection then calculates the probability that data is significantly related to the extracted information. The dataset should be extracted and identified during this phase and turn the information and structure into a result. The second and third steps cover the role of implementing processes and decision-making that generate ultimate results. The next phase covers the identification of relevant values and information, substituting missing values with valid estimations. Besides, this phase should define the appropriate approach of imputing missing values for the AKI dataset. The performance of each approach is compared, and results are presented.

The last phase involves resolving the information into a more understandable model, qualifiable values to choose the best methodology. The data extracted in earlier stages will be compiled into the final result.

2.1. Data Collection

We collected data on demographic characteristics, past medical history, laboratory results, the severity of illness, and care processes from IIUM Medical Centre (IIUMMC). The data is collected according to the code of ethics ref. number NMRR-13-1631-18970 from Kementerian Kesihatan Malaysia. We also retrieved SCr levels for each patient for up to one year before hospital admission from the dataset. Outcomes included AKI diagnosis and staging, mortality at hospital discharge, and renal recovery at least three months after hospital discharge.

2.2. Study Designs and Participants

We performed a retrospective study of critically ill adult patients admitted to our tertiary care academic center between January 1 and December 31, 2012. In this study, we assessed the performance of four surrogate methods: 1) first SCr level at hospital admission; 2) minimal SCr level within two weeks after intensive care unit (ICU) admission; 3) SCr computed from the MDRD formula for an eGFR of 75 ml/min per 1.73 m2 and 4) SCr computed from the Chronic Kidney Disease Epidemiology Collaboration (CKD-EPI) formula for an eGFR of 75 ml/min per 1.73 m2. We performed a multilinear regression model to identify patients' characteristics that best predict preadmission SCr. We then performed imputation strategies using calculated SCr values from the multilinear regression models to assess AKI diagnosis.

We included randomly selected critically ill patients aged 18 or more and excluded readmissions, patients on chronic dialysis, those having a kidney transplant, or those who stayed in the ICU less than 24 h. We followed the STrengthening the Reporting of OBservational studies in Epidemiology (STROBE) guidelines for observational studies.

2.3. Evaluation Methods

The nature of imputation is evaluated by comparing the imputed values against original values. The evaluation of the optimized KNN algorithm with GOA and other optimization algorithms involves two performance metrics such as error accuracy, running time, and statistical significance test.

The most powerful parameters to evaluate the performance and measures the error differences between values are by employing Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These parameters are negatively oriented, which implies the better of lower values. The three criteria significantly a meaningful representation that computes an error between two numeric vectors. MAE measures the average squared difference in a set of predictions, where absolute differences between prediction and actual observation are calculated. All the individual differences are weighted equally in the average as in (1).

$$MAE = 1/n \, \Sigma^{n}_{j=1} \, | y_j - \hat{y}_i | \quad (1)$$

MSE is an estimator that measures the average squared difference of its error between the predicted values and actual values as shown in (2).

$$MSE = 1/n \, \Sigma^{n}_{j=1} \, (y_j - \hat{y}_i) \quad (2)$$

RMSE is a quadratic scoring rule that measures the average magnitude of error, a square root of the average squared differences between actual and prediction observation. The following is an equation of RMSE:

$$RMSE = \sqrt{1/n \, \Sigma^{n}_{j=1} (y_j - \hat{y}_i)^2} \quad (3)$$

Apart from assessing the algorithm efficiency through accuracy and performance, an algorithm is also measured by calculating the running time. Running time is a measure of the amount of time for an algorithm to execute. A statistical significance test is interested in assessing the performance of optimizing the KNN imputation algorithm with GOA against other optimization algorithms. The purpose of statistical significance testing is to help gather evidence of the extent to which the results returned by an evaluation metric represent the general behavior of the proposed algorithm. Vargha-Delaney A Test is another non-parametric test used to evaluate the performance of the optimized KNN imputation algorithm with GOA [9]. The comparison of the algorithm's actual value and the predicted value is taken and compared whether there is a significant difference between the two results.

2.4. Machine Learning Imputation Algorithm

Imputation is a common way to deal with missing values where the missing value's substitutes are discovered through statistical or machine learning approaches. Even though the statistical approach has been adopted for decades, machine learning-based data imputation techniques are becoming popular in handling missing values, especially in large data sets [10].

Many machine learning-based imputation methods have been introduced to resolve the missing data problem [11]. These methods work by using machine learning techniques to find rules from the input data to estimate the possible value of the missing data. These methods have several advantages [12].

The machine learning approach has revolutionized the world with various algorithms to aid data analysis. Recent studies on imputation indicate that four popular machine learning classifiers are K-nearest neighbors (KNN), Decision Tree, Naïve Bayes, and Support Vector Machine (SVM), as shown in Fig. 2. Hence, the focus of this thesis is the machine learning that has been proposed in data imputation.

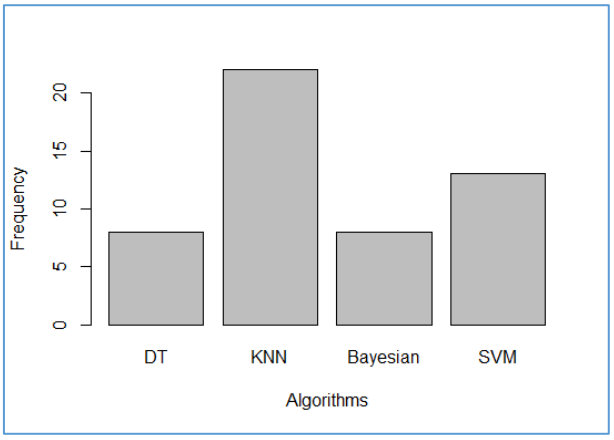

Fig. 2. Frequency of algorithms from literature studies

2.5. Particle Swarm Optimization (PSO)

Particle swarm optimization is a simple method introduced by Kennedy and Eberhart in 1995 based on the swarm of bird communication inspiration. PSO consider one of the most efficient optimization algorithm used to solve optimization problems [13]–[15]. Because the simplicity and robust performance, PSO attract researcher and engineer [16]. PSO has been widely applied for solving real-world optimization problems, including feature selection [17], Control System in an Internet of Things (IoT) Environment [18], tracking 3D objects in RGB-D image [19], Path Planning For Mobile Robot [20], Face recognition [21], trained recurrent neural network [22], Network Security [23], Gene

selection [24], digital image watermarking [25], design digital A proportional–integral–derivative controller (PID), and in various science and engineering problems [16].

PSO depends on the movement and intelligence of the swarm [26], [27]. The swarm consists of the number of particles tending to move toward a better solution [28]. The particles in the search space present the solutions. PSO relies on two formulas belonging to every single particle: position and. The new velocity particle is updated using equation (4) [29].

$$V_i\,(t+1) = w * V_i\,(t) + c_1 * r_1 * (P_{best} - X_i\,(t)) + c_2 * r_2 * (G_{best} - X_i\,(t)) \quad (4)$$

where i is the particle index; t is the number of iteration; $V_i(t)$ is the current velocity of the particle; w is inertia Weight; $V_i(t)$ is the current position of the particle; P_{best} represents the best previous position of particle i; G_{best} represents the best position among all particles; r_1, r_2 random numbers with values between (0,1); c_1, and c_2 are positive numbers called acceleration coefficients guide the particle toward the particle best and swarm best positions. PSO use equation (4) to update the position of the particles [29].

$$X_i\,(t+1) = X_i\,(t) + V_i\,(t+1) \quad (5)$$

where $X_i(t)$ is the previous position of the particle; $V_i\,(t+1)$ is the particle's current velocity. The number of studies has been improved PSO. One of these studies is levy PSO [30]; the algorithm shows high performance compared to PSO. Algorithm 1 presents the Pseudocode of PSO (Fig. 3).

Algorithm 1: Particle Swarm Optimization (PSO)

```
Initialization of the parameter NP, w, C1, and C2, maximum iteration
Initialize particle velocity and position
Evaluate the fitness value
If (the current fitness value <  the particle best value)
        Assign the current value to Pbest
End if
Set particle with the best fitness value to Gbest
Iteration = 1
While stopping criteria  is not reached Do
    For i =1 to NP
      Update particle velocity according to equation 4
      Update particle position according to equation 5
    End for
End while
Output the best solution
```

Fig. 3. The Pseudocode of PSO

2.6. Particle Swarm Optimization Levy Flight (PSO-LF)

The Particle Swarm Optimization (PSO) is classified as one of the meta-heuristics, which mimics the behavior of swarm intelligence of schools of fish and flock of birds. There are two important control parameters of PSO: inertia and acceleration [31]. The inertia coefficient, in particular, govern the convergence property. Various control methods for the inertia coefficient are proposed to improve the performance of the solution searchability.

The inertia coefficient determines the speed of convergence. As the inertia parameter increases, the convergence speed slows. Furthermore, if these parameters exceed certain thresholds, the system does not converge. The inertia coefficient controls the phase transition. As a result, the inertia coefficient is critical to the PSO's dynamics. A large inertia coefficient is highly associated with slow convergence, keeping the system searching for the optimum solution. Although this approach helps to improve the search performance, however, if the inertia coefficient is less than 1, the system cannot escape the local minimum. Nevertheless, if the inertia coefficient is larger than 1, this can lead to divergence. The divergence property results in the ability to escape from the local minimum. If the divergence property

is tamed, the ability to find the solutions can improve. Therefore, we propose a novel PSO with Levy flight to the inertia coefficient to control the divergence property.

A Levy flight is a random walk in which the step size is according to a heavy-tailed distribution that is drawn from Levy distribution [31], [32]. Fig. 4 depicts two-dimensional Levy flight and random walk examples. As shown in Figure 1, Levy flight has a broad step size on occasion.

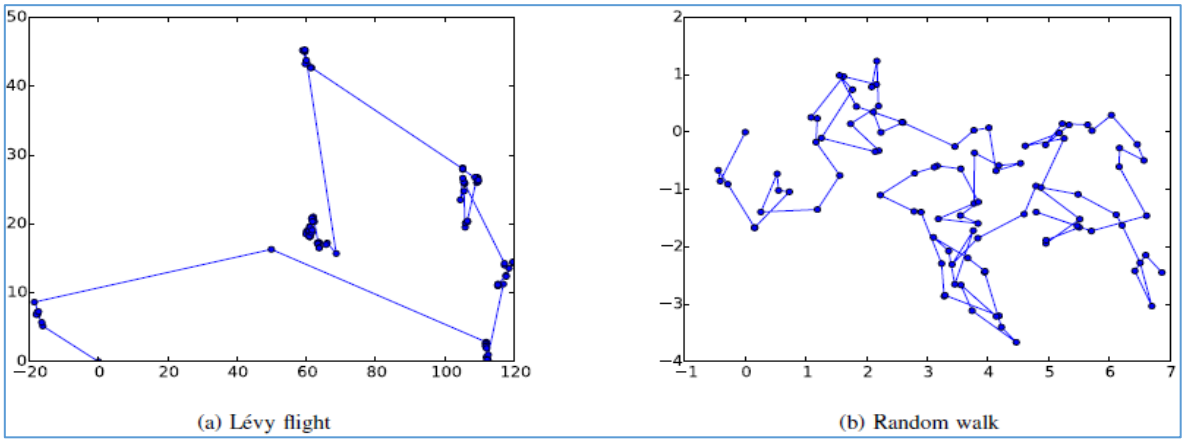

Fig. 4. Example of two-dimensional motion

The proposed algorithm Levy flight method solves premature convergence and enables PSO to produce more efficient results (Fig. 5). This approach ensures that PSO, which cannot perform global search well, can perform global search more efficiently and avoid being stuck in local minima [33], [34].

Algorithm 2: The proposed PSO-LF algorithm

Initialization the parameter *NP, D,* C_1, and C_2, *max_iter,* V_{min}, V_{max}
Initialize particle velocity and position
Trial (keeps the limit value for each particle) = 0;
Evaluate the fitness value
Set X to be P_{best}
Set the particle with best fitness to be G_{best}
While iter > max_iter do
    For i = 1: NP
        If trial(i) < limit (if current particle is not exceeded limit value)
            Update the velocity V_i of particle using equation 4
            Update the position of particle using equation 5
        Else (if current particle is exceeded limit value)
            Trial (i) = 0
            Update the particle positions using Levy flight method
            Positions value exceeding the boundaries in the search space (X_{min}, X_{max})
        End if
        Evaluate fitness value for new particle X_i
        If X_i is better than P_{best}
            Trial(i) = 0;
            Set X_i to be P_{best};
        Else
            Trial(i) = trial(i) +1
        End if
        If X_i is better than G_{best}
            Set X_i to be P_{best}
        End if
    End for
Iter = iter + 1
End while

Fig. 5. The proposed PSO-LF algorithm

The Levy flight method is studied to seek updated velocity to increase the PSO algorithm's performance. Similar to the state-of-art PSO, particles are first distributed randomly in the search space, fitness values of all particles are assessed, and particles P_{best} and swarm G_{best} are obtained. The velocity and position of each particle are then updated based on the random probability. The particles' velocity and position are updated with probability greater than or equal to 0.5, just like in the traditional PSO by equations (4) and (5). If the random value is less than 0.5, the particle velocity is updated, and the particle's velocity becomes its position. Using the levy flight technique to update the particle's velocity, the particle takes a lengthy hop towards its P_{best} and G_{best}, increasing the diversity of the swarm and allowing the algorithm to do global exploration over the search space.

Occasionally, the PSO-LF inertia coefficient, ω, reaches a high value. As a result, the particle of PSO-LF can escape the local minimum and continue to seek the best solution in the global domain [33]. PSO-LF, in particular, combines the capacity to search locally and globally. However, there is a chance that the obtained moving distance will be excessively long. Based on Algorithm 2, NP is the number of particles, D is the dimension of benchmark function, C1 and C2 are the acceleration coefficients, max_iter refers to the maximum number of iterations, $V_{min,}$ and V_{max} represent the maximum and minimum limit of the velocity increase to be made.

In the proposed PSO-LF method (Fig. 5), two changes are made compared to the traditional PSO method. First, the limit value is assigned for each particle, where the limit value is increased by 1 in case the particles are unable to enhance their self-solutions for each subsequent iteration. Second, particles that surpass the limit value are redistributed using the Levy flight method in the search space. The loss of diversity is prevented by employing the random phenomena of levy flight while updating the velocity. As the efficiency of the PSO algorithm is improved by introducing the benefits of random walk into the PSO, particle's positions in each iteration due to increased exploration and exploitation of the search space.

In the levy flight technique, ß parameters have a significant impact on distribution. The random distribution is altered by changing the value by using a different value for ß [30]. The distribution is frequently expressed as equation (6), where ß parameter is an index in the range (0,2] [35].

$$L(s) \sim |s|^{-1-\beta} \tag{6}$$

For random walk, the step length S is derived by Mantegna's algorithm as:

$$S = u / |v|^{1/\beta} \tag{7}$$

Where ß is referred to as Levy index, and where u and v are drawn from normal distribution as follows:

$$u \sim N(0, \sigma_u^2),\ v \sim N(0, \sigma_v^2) \tag{8}$$

$$\sigma_u = \{\Gamma(1+\beta)\sin(\pi\beta/2) / \Gamma[1+\beta/2]\,\beta\,2^{(\beta-1)/2}\}^{1/\beta},\ \sigma_v = 1 \tag{9}$$

where Γ is standard Gamma function. Then, step size is calculated by:

$$Stepsize = scale \times S \tag{10}$$

Here, the step size represents the step size in the search space, and the dimension of the desired problem determines the factor S. Otherwise, Levy flight may exhibit very aggressive behavior, resulting in the generation of new solutions outside the design space [36].

A nontrivial approach for producing step size S samples is explained in-depth, and it can be summarized as below [32].

$$S = random(size(D)) \oplus Levy(\beta) \sim 0.01\,(u / |v|^{1/\beta})\,(X_j^t - G_{best}^t) \tag{11}$$

The S value with D dimension derived from equation (11) is added to update the position Xi particles determines the position values of the new particle. Then, the fitness value for this new particle is assessed, if the particle achieves a better result than its P_{best}, the P_{best} value is updated, and the trial value of this particle is set to zero; otherwise, the trial value is increased by one. The algorithm is then repeated until the stopping criterion is met.

3. Results and Discussion

The evaluation of the proposed algorithm is investigated concerning the error accuracy and running time applied to the creatinine dataset. This performance metric compares three optimization algorithms (Genetic Algorithm, Particle Swarm Optimization, and Particle Swarm Optimization Levy Flight) with four well-established machine learning imputation algorithms (K-nearest neighbors, Decision Tree, Naïve Bayes, and Support Vector Machine).

Generally, the result also highlights that the most promising finding is an optimization of machine learning imputation algorithm with Particle Swarm Optimization Levy Flight (PSOLV). Table 1 describes four machine learning optimized with PSOLV consistently demonstrating impressive performance for all three relative error parameters, which provides a low error accuracy against the traditional GA and PSO algorithm. Among four optimized machine learning imputation algorithms, SVMPSOLF shows the lowest error accuracy for all three error accuracy parameters.

Table 1. Error Accuracy.

Machine Learning	Meta Algorithm	MAE	MSE	RMSE
K-nearest neighbors (KNN)	GA	7.9e+02	4.2e+06	2.1e+03
	PSO	7.6e+02	4.2e+06	2.1e+03
	PSOLF	**1.087**	**5.4e+01**	**0.0435**
Naïve bayes (NB)	GA	8.1e+03	9.5e+07	9.8e+03
	PSO	1.146	2.023	1.422
	PSOLF	**0.62+01**	**4.56+02**	**9.938**
Decision Tree (DT)	GA	8.1e+03	9.5e+07	9.7e+03
	PSO	1.1737	2.138	1.4625
	PSOLF	**0.97+01**	**6.49+04**	**1.76+01**
Support Vector Machine (SVM)	GA	8.1e+03	9.5e+07	9.7e+03
	PSO	1.1602	2.0693	1.4385
	PSOLF	**0.53+01**	**1.43+02**	**1.94+01**

Another investigation that governs the efficiency of an algorithm is by measuring the running time. Table 2 shows that all optimized machine learning with PSOLF executes as the fastest running time. Table 2 also illustrates that the fastest imputation algorithm for imputing missing baseline creatinine uses SVMPSOLF among four machine learning imputation algorithms.

Table 2. Running Time.

Machine Learning	GA	PSO	PSOLF
K-nearest neighbors (KNN)	0.78	1.91	**0.59**
Naïve Bayes (NB)	2.08	1.09	**0.92**
Decision Tree (DT)	1.95	1.35	**1.05**
Support Vector Machine (SVM)	2.15	1.58	**0.43**

The result demonstrates that error accuracy and running time are insufficient to evaluate the difference in performances between all algorithms fully. Precisely, the result needs to be verified whether the differences in performances are statistically significant and not merely coincidental. To intensely analyze the performance of each optimization algorithm, a statistical significance test is compared

between actual values and imputed values. In the majority of statistical analyses, an alpha of 0.05 is used as the cutoff for significance. Vargha and Delaney suggested a threshold for interpreting the effect size where 0.5 means no difference at all; up to 0.56 indicates a small difference; up to 0.64 indicates medium, and anything over 0.71 is large.

The result in Table 3 illustrates an optimized machine learning imputation with PSOLF constantly displayed the closest difference between actual and imputed values. This result displays a statistical significance which implies that the differences can be negligible. Among the statistical significance test, SVMPSOLF is the most accepted, with the closest significance to 0.5. In order to assess the consistency performance of each optimized machine learning imputation method, a hypothesis is formulated for the comparison of performance and efficiency. For a missing dataset that achieves the lowest error accuracy and fastest time, the hypothesis is that the differences between the actual value and imputed value must be zero.

The hypothesis is $H1_0$: *There is no statistical difference between actual and imputed values for optimized machine learning with the PSOLF algorithm.*

Table 3. Statistical Significance Test

Machine Learning	Meta Algorithm	Vargha Delaney Test	Significant
K-nearest neighbors (KNN)	GA	0.465102	Small different
	PSO	0.474693	No different
	PSOLF	**0.499975**	No different
Naïve bayes (NB)	GA	1	Large different
	PSO	0.499591	No different
	PSOLF	**0.499872**	No different
Decision Tree (DT)	GA	1	Large different
	PSO	0.499795	No different
	PSOLF	**0.5021**	No different
Support Vector Machine (SVM)	GA	1	Large different
	PSO	0.499183	No different
	PSOLF	**0.50071**	No different

From Table 4, PSOLF has a perfect p-value closest to 0.5, consistent with their error accuracy and running time. Therefore, $H1_0$ is accepted. All optimized machine learning algorithms with PSOLF have almost the same p-value, with the lowest error accuracy and fastest running time compared to traditional GA and PSO optimization. Hence, all PSOLF based on four machine learning imputation algorithms is accepted. A final analysis can be deduced from all results that PSOLF performs well for optimizing all machine learning imputation algorithms.

Table 4. Statistical differences for all datasets by error accuracy and running time

Algorithm	Error Accuracy	Running Time	P-Value	Annotation
KNNPSOLF	1.087	**0.59**	0.4999	Accept
NBPSOLF	0.62E+1	**0.92**	0.4998	Accept
DTPSOLF	8.1E+3	**1.05**	0.5021	Accept
SVMPSOLF	0.53E+1	**0.43**	0.5007	Accept

4. Conclusion

Missing baseline creatinine value can be a root cause of a poor estimation of kidney function and misclassify AKI biased estimations. In this regard, four popular machine learning imputation methods (K-nearest neighbors, Decision tree, Naïve Bayes, and Support Vector Machine) are employed to analyze the optimization of each machine learning with an optimization approach. This paper demonstrates the application of the PSO algorithm based on machine learning to treat missing baseline creatinine values. The results show that the PSOLF imputation algorithm is reliable as the performance of PSOLF

constantly outperformed regarding error accuracy and running time, which verified with statistical significance test. The results also show that SVMPSOLF is superior to other proposed algorithms as the lowest error accuracy and fastest running time. The proposed algorithm, SVMPSOLF, is recommended to be executed in other medical applications to determine if the SVMPSOLF imputation algorithm can treat missing values. Further comparison of SVMPSOLF with other optimization algorithms is also recommended to judge the performance of SVMPSOLF.

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] J. a Kellum *et al.*, "KDIGO Clinical Practice Guideline for Acute Kidney Injury," *Kidney Int. Suppl.*, vol. 2, no. 1, pp. 1–138, 2012. Available at: Google Scholar.

[2] H. E. Wang, G. Jain, R. J. Glassock, and D. G. Warnock, "Comparison of absolute serum creatinine changes versus Kidney Disease : Improving Global Outcomes consensus definitions for characterizing stages of acute kidney injury," *Nephrol. Dial. Transplant.*, no. January, pp. 1447–1454, 2013. doi: 10.1093/ndt/gfs533.

[3] G. M. Chertow, E. Burdick, M. Honour, J. V Bonventre, and D. W. Bates, "Acute Kidney Injury, Mortality, Length of Stay, and Costs in Hospitalized Patients," *J. Am. Soc. Nephrol.*, 2005. doi: 10.1681/ASN.2004090740.

[4] S. A. Hamid, W. W. Adnan, N. N. Naing, and A. S. Adnan, "Acute Kidney Injury in Intensive Care Unit , Hospital Universiti Sains Malaysia : A Descriptive Study," *Saudi J. Kidney Dis. Transplant.*, vol. 29, no. 5, pp. 1109–1114, 2018. doi: 10.4103/1319-2442.243961.

[5] M. Ostermann and M. Joannidis, "Acute kidney injury 2016 : diagnosis and diagnostic workup," *Crit. Care*, vol. 20, no. 299, pp. 1–13, 2016. doi: 10.1186/s13054-016-1478-z.

[6] A. Bernier-Jean *et al.*, "Diagnosis and outcomes of acute kidney injury using surrogate and imputation methods for missing preadmission creatinine values," *BMC Nephrol.*, vol. 18, no. 1, pp. 1–9, 2017. doi: 10.1186/s12882-017-0552-3.

[7] E. D. Siew, J. F. Peterson, S. K. Eden, K. G. Moons, T. A. Ikizler, and M. E. Matheny, "Use of Multiple Imputation Method to Improve Estimation of Missing Baseline Serum Creatinine in Acute Kidney Injury Research," *Clin. J. Am. Soc. Nephrol.*, vol. 8, 2013. doi: 10.2215/CJN.00200112.

[8] W. Y. Lai, K. K. Kuok, S. Gato-trinidad, and K. X. Ling, "A Study on Sequential K-Nearest Neighbor (SKNN) Imputation for Treating Missing Rainfall Data," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 3, pp. 363–368, 2019. doi: 10.30534/ijatcse/2019/05832019.

[9] 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.

[10] 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.

[11] 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.

[12] G. Wang, Z. Deng, and K.-S. Choi, "Tackling missing data in community health studies using additive LS-SVM classifier," *IEEE J. Biomed. Heal. Informatics*, vol. 22, no. 2, pp. 1–1, 2016. doi: 10.1109/JBHI.2016.2634587.

[13] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence, 2001. Available at: Google Books.

[14] Lili-Li and Xingshi-He, "Gaussion mutation Particle Swarm Optimization with dynamic adaptation inertia weight," *World Congr. Softw. Eng.*, no. 1, pp. 454–459, 2009. doi: 10.1109/WCSE.2009.24.

[15] S. Bashath and A. R. Ismail, "Comparison of Swarm Intelligence Algorithms for High Dimensional Optimization Problems," *Indones. J. Electr. Eng. Comput. Sci.*, no. July, pp. 300–307, 2018. doi: 10.11591/ijeecs.v11.i1.pp300-307.

[16] Q. Cui *et al.*, "Globally-optimal Prediction-based Adaptive Mutation Particle Swarm Optimization," *Inf. Sci. (Ny).*, 2017. doi: 10.1016/j.ins.2017.07.038.

[17] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *J. Comput. Sci.*, no. October, 2017. doi: 10.1016/j.jocs.2017.07.018.

[18] A. L. Sangeetha, N. Bharathi, A. B. Ganesh, and T. K. Radhakrishnan, "Particle Swarm Optimization Tuned Cascade Control System in an Internet of Things (IoT) Environment," *Measurement*, 2017. doi: 10.1016/j.measurement.2017.12.014.

[19] D. S. Junior, J. G., do M. Lima, and J. P. S., "Particle Swarm Optimization for 3D object tracking in RGB-D images," *Comput. Graph.*, 2018. Available at: Google Scholar.

[20] H. S. Dewang, P. K. Mohanty, and S. Kundu, "A Robust Path Planning For Mobile Robot Using Smart Particle Swarm Optimization," *Procedia Comput. Sci.*, vol. 133, pp. 290–297, 2018. doi: 10.1016/j.procs.2018.07.036.

[21] A. Sajid, A. Khan, M. Ishtiaq, and M. Shaheen, "Face Recognition under varying Expressions and Illumination using particle swarm optimization," *J. Comput. Sci.*, 2018. Available at: Google Scholar.

[22] A. M. Ibrahim and N. H. El-amary, "Particle Swarm Optimization trained recurrent neural network for voltage instability prediction," *J. Electr. Syst. Inf. Technol.*, 2017. doi: 10.1016/j.jesit.2017.05.001.

[23] D. Zhao and J. Liu, "Study on network security situation awareness based on particle swarm optimization algorithm," *Comput. Ind. Eng.*, vol. 125, 2018. doi: 10.1016/j.cie.2018.01.006.

[24] N. Pashaei, E., Pashaei, E., & Aydin, "Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization," *Genomics*, vol. 1, no. 1, pp. 33–57, 2017. Available at: Google Scholar.

[25] Z. Zheng, N. Saxena, K. K. Mishra, and A. K. Sangaiah, "Guided Dynamic Particle Swarm Optimization for Optimizing Digital Image Watermarking in Industry Applications," *Futur. Gener. Comput. Syst.*, 2018. doi: 10.1016/j.future.2018.05.027.

[26] M. R. Bonyadi and Z. Michalewicz, "Particle Swarm Optimization for Single Objective Continuous Space Problems : A Review," *Evol. Comput.*, no. xx, pp. 1–54, 2016. doi: 10.1162/EVCO_r_00180.

[27] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm : an overview," *Soft Comput.*, vol. 22, no. 2, pp. 387–408, 2018. doi: 10.1007/s00500-016-2474-6.

[28] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization An overview," *Swarm Intell.*, pp. 33–57, 2007. doi: 10.1007/s11721-007-0002-0.

[29] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE Int. Conf.*, vol. 4, pp. 1942–1948, 1995. doi: 10.1109/ICNN.1995.488968.

[30] R. Jensi and G. W. Jiji, "An Enhanced Particle Swarm Optimization with Levy Flight for Global Optimization," *Appl. Soft Comput. J.*, 2016. doi: 10.1016/j.asoc.2016.02.018.

[31] Y. Hariya, T. Kurihara, T. Shindo, and K. Jin'No, "Lévy flight PSO," *2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc.*, no. May 2015, pp. 2678–2684, 2015. doi: 10.1109/CEC.2015.7257220.

[32] H. Hakli and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput. J.*, vol. 23, pp. 333–345, 2014. doi: 10.1016/j.asoc.2014.06.034.

[33] C. Tholen, T. A. El-Mihoub, and L. Nolle, "On a novel search strategy based on a combination of particle swarm optimisation and levy-flight," *Proc. - Eur. Counc. Model. Simulation, ECMS*, pp. 190–194, 2018. doi: 10.7148/2018-0190.

[34] N. D. Jana and J. Sil, "Particle Swarm Optimization with Lévy Flight and Adaptive Polynomial Mutation in gbest Particle," *Recent Adv. Intell. Informatics*, pp. 275–276, 2014. doi: 10.1007/978-3-319-01778-5_28.

[35] T. Guan, F. Han, and H. Han, "A Modified Multi-Objective Particle Swarm Optimization Based on Levy Flight and Double-Archive Mechanism," *IEEE Access*, vol. 7, pp. 183444–183467, 2019. doi: 10.1109/ACCESS.2019.2960472.

[36] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Appl. Soft Comput. J.*, vol. 73, pp. 697–726, 2018. doi: 10.1016/j.asoc.2018.09.019.