



A MISSING DATA IMPUTATION METHOD BASED ON GREY WOLF ALGORITHM FOR DIABETES DISEASE

Anas AHMED^{1*}, Timur İNAN²

¹Department of Information Technology, Altınbaş University, İstanbul, Türkiye
anas87sa@gmail.com ( <https://orcid.org/0000-0001-5324-3442>)

²Department of Software Engineering, Altınbaş University, İstanbul, Türkiye
timur.inan@altinbas.edu.tr ( <https://orcid.org/0000-0002-6647-3025>)

Received: 09.11.2022

Research Article

Accepted: 03.01.2023

*Corresponding author

Abstract

The bulk of medical databases contain coverage gaps due in large part to the expensive expense of some tests or human error in documenting these tests. Due to the absence of values for some features, the performance of the machine learning models is significantly impacted. Consequently, a specific category of techniques is necessary for the aim of imputing missing data. In this study, the Grey Wolf Algorithm (GWA) is used to generate and impute the missing values in the Pima Indian Diabetes Disease (PIDD) dataset. The proposed method is known as the Pima Indian Diabetes Disease (PIDD) Algorithm (IGW). The obtained results demonstrated that the classification performance of three distinct classifiers, namely the Support Vector Machine (SVM), the K-Nearest Neighbor (KNN), and the Naive Bayesian Classifier (NBC), was enhanced in comparison to the dataset prior to the application of the proposed method. In addition, the results indicated that IGW performed better than statistical imputation procedures such as removing samples with missing values, replacing them with zeros, mean, or random values.

Keywords: Missing Values, Grey Wolf Algorithm, Diabetes Disease. Classification.

DİYABET HASTALIĞI İÇİN GRİ KURT ALGORİTMASINA DAYALI EKSİK BİR VERİ TAHMİN YÖNTEMİ

Öz

Tıbbi veritabanlarının büyük kısmı, büyük ölçüde bazı testlerin pahalı masraflarından veya bu testlerin belgelenmesindeki insan hatasından dolayı kapsam boşlukları içermektedir. Bazı özellikler için değerlerin olmaması nedeniyle, makine öğrenimi modellerinin performansı önemli ölçüde etkilenir. Sonuç olarak, eksik verileri atamak amacıyla belirli bir teknik kategorisi gereklidir. Bu çalışmada, Pima Indian Diabetes Disease (PIDD) veri setindeki eksik değerleri oluşturmak ve hesaplamak için Gray Wolf Algoritması (GWA) kullanılmıştır. Önerilen yöntem Pima Hint Diyabet Hastalığı (PIDD) Algoritması (IGW) olarak bilinir. Elde edilen sonuçlar, Destek Vektör Makinesi (SVM), K-En Yakın Komşu (KNN) ve Naive Bayes Sınıflandırıcısı (NBC) olmak üzere üç farklı sınıflandırıcının sınıflandırma performansının önceki veri kümesine kıyasla arttığını göstermiştir. Önerilen yöntemin uygulanması. Ek olarak, sonuçlar IGW'nin istatistiksel olarak daha iyi performans gösterdiğini göstermiştir. eksik değerlere sahip örneklerin çıkarılması, sıfırlar, ortalama veya rastgele değerler ile değiştirilmesi gibi atama prosedürleri.

Anahtar Kelimeler: Eksik Değerler, Gri Kurt Algoritması, Diyabet Hastalığı. Sınıflandırma.

1. Introduction

The quality of the outcome of Data Mining (DM) operations is determined by the quality of the considered data; therefore, data pre-processing is a crucial stage in obtaining clean and high-quality data and influences the success of the mining process (Soofi and Awan 2017; Tao et al 2022). Pre-processing the data is the most crucial step in KDD since it streamlines the information and sets the stage for better results from the subsequent analysis. With the aid of data preprocessing, the nature of the data may be better comprehended, paving the way for more precise and effective data analysis. The data themselves comprise the subsequent essential step in the KDD procedure. Each DM action requires unique input data, and it's important that it be prepared in a form and structure that are optimal for

that task. There is no expectation that unprocessed, unfiltered data would be perfect. Given that clean, organized data is usually a prerequisite for effective DM models, it's important to take the time to clean and organize your data thoroughly. Since missing data is a major issue throughout DM processes, especially when it occurs in significant volumes, it is imperative that the values be correct and consistent; nevertheless, it is not possible to remove all characteristics (instances) from the sample that have missing values due to their abundance (Sowniya and Suneetha, 2017).

When dealing with missing data, one of the first things that should be done is to investigate the circumstances that led to the loss of the data points in the first place. There are three categorical types of missing data as per described by (Donders et al, 2006; Pigott 2009). First, if a subset of data is missing and it is not recorded in a random order, then the subset is "Missing Completely at Random" (or "MCAR"). The possibility that certain data points in a collection do not have their associated observations recorded at random is what is meant by the phrase "Missing at Random," which is abbreviated as "MAR." The third category, known as the Non-Ignorable scenario, is intended to convey the idea that the missing data are not the result of a random process but rather are dependent on the values that have been omitted (Sowniya and Suneetha, 2017; Garcia Laencina, Sancho-Gómez and Figueiras-Vidal, 2010). One of the straightforward approaches to address MVs is to delete the characteristics/features from the data set that contain those variables. If, on the other hand, working with data that contains a sizeable number of records that have MVs, this method would result in bias of data classification, thus limiting how broadly the outcomes of the research can be applied in the real world (Donders et al, 2006; Choudhury and Pal, 2019).

Insomuch as addressing the issue of missing data is time-consuming in most applications, it is often overlooked during the decision-making process. As a result, approaches that are quick, notwithstanding the likelihood that they are inefficient, are required to manage data-related problems. This gives rise to problems that can be solved computationally as well as conceptually, hence increasing the requirement for resources such as techniques and theoretical frameworks (De Silva and Perera, 2016; Alhroob et al, 2020). Most of the time, inadequate strategies are utilized to handle missing data since there is little time to identify more effective solutions before they are observed. This results in the deployment of inefficient approaches, such as case deletions, because there is insufficient time to develop more efficient strategies for dealing with missing data. Unfortunately, many of the often employed procedures are not only inefficient, but also detrimental, as they frequently provide skewed and unreliable results. The remainder of the study is structured as follows: section two provides background information on missing values and summarizes the most significant relevant works on machine learning models for diabetic disease. The recommended strategy is described in section three. The PIDD dataset is given and evaluated in section four. In addition, the proposed approach for imputation is examined. Finally, section five summarizes the consequences of the proposed method and provides suggestions for future research.

2. Grey Wolf Algorithm

Nature-inspired metaheuristics may aid with modern optimization challenges. NP-hard optimization tasks like the traveling salesman problem. Grey Wolf Optimizer considers searching behavior, social structure, and hunting technique (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014). The GWO method is easy to implement and converges quickly due to reduced unpredictability and varying numbers of individuals in global and local searching (Long et

al., 2020; Mohammadzadeh et al., 2021). Its effectiveness is superior than the PSO algorithm and other bionic algorithms, according to the evidence. This algorithm's better performance has increased its popularity. Alpha, Beta, Delta, and Omega make comprise the wolf pack's hierarchy. And are designated alpha, beta, and delta for the finest, second-best, and third-best foxes, respectively. Within the GWO, alpha, beta, and delta lead optimization or hunting. They lead the group to the best hunting grounds. The alpha, beta, and delta wolves assess whether they've located prey while they search (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014; Tawhid and Ali, 2017; Zhang et al., 2021).

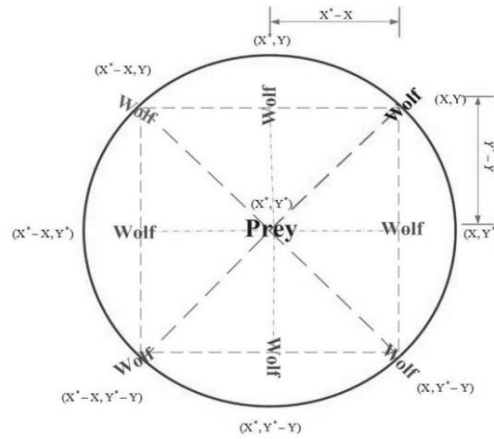


Figure 1. Wolf positions in GWO (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

Wolves have a strict hierarchy among themselves. By putting the wolves into four different groups—alpha, beta, delta, and omega—it is possible to re-create the wolves' internal leadership structure (see Figure 1). The alpha, beta, and delta wolves of the GWO are in charge of leading the other wolves (W) to the best places to hunt. This gives the alpha, beta, and delta wolves a chance to improve their way of hunting. During an iterative search, it's up to the three wolves to figure out where the prey could possibly be (alpha, beta, & delta). During the process of optimizing the system, equations 1 and 2 are used to change where the wolves are (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (1)$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

where t is the t th iteration while \vec{A} and \vec{C} are coefficient vectors at each iteration. The position vector of the prey is denoted by \vec{X}_p while \vec{X} is the position of the wolf. Vectors \vec{A} and \vec{C} are expressed as follows (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

$$\vec{A} = 2a \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

where \vec{a} is a coefficient that has a negative slope in the range $[2, 0]$ and gets progressively smaller as the number of iterations gets higher. \vec{r}_1 and \vec{r}_2 are random vectors generated within the range of 0 to 1. The rules for updating position were displayed in Figure 1 and can be found defined in Equations 1 and 2. The figure showed that the wolf at position (X, Y) can move to any position around the prey using the updating formulas presented above. despite the fact that there are only seven different spots depicted in Figure 1 that the wolf might theoretically go to. By adjusting C and A , it is feasible to move the wolf to any location inside space that is close to the prey (the random parameters).

In the GWO, it is generally agreed that the positions of the alpha, beta, and delta wolves are always the best (the position of the prey). During an iteration, the notations alpha, beta, and delta are utilized in order to keep track of which individuals are now thought of as being the best, second best, and third best respectively. While the remaining wolves (omega) make adjustments to their placements based on the whereabouts of the alpha, beta, and delta wolves, those three wolves continue to explore the territory. To update the positions of the omega wolves, the following relations are used (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (5)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (6)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (7)$$

\vec{X}_α , \vec{X}_β , and \vec{X}_δ are the respective position vectors of alpha, beta, and delta wolves. Vectors denoted by \vec{C}_1 , \vec{C}_2 , and \vec{C}_3 are randomly generated. Calculating the distance between the current location of an individual and the location of the alpha, beta, and delta wolf packs requires the employment of equations 5, 6, and 7 respectively. For this reason, the calculation of the current individual's final position vectors is carried out as follows (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (8)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (9)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (10)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (11)$$

where \vec{A}_1 , \vec{A}_2 , and \vec{A}_3 are vectors that are created at random and t is the number of times the process is iterated. The position of a region within a plane is determined by three points; hence, the best three wolves can determine the scope of where the prey is located. Due to the fact that the target solution of the GWO is evaluated by three different solutions, there is a low chance that it will entrap the local optimal solution. It is clear, based on equations 5–7, that the step-size of omega is moving in the direction of alpha, beta, and delta. Equations 8–11 form the basis for the definition of the ultimate positions of the omega wolves (Mirjalili, S, Mirjalili, S. M. and Lewis, 2014)

3. The Proposed Methodolgy

The imputation algorithm that has been proposed and is based on the Grey Wolf Algorithm is presented in this section. The section is broken down into two sub-sections: the first of which discusses the suggested imputation method in a more general sense, while the second of which provides a more in-depth explanation of the GWO algorithm.

3.1. Imputation-Grey Wolf Optimizer (IGWO)

The proposed algorithm consists of three main phases as follows (See Figure 2):

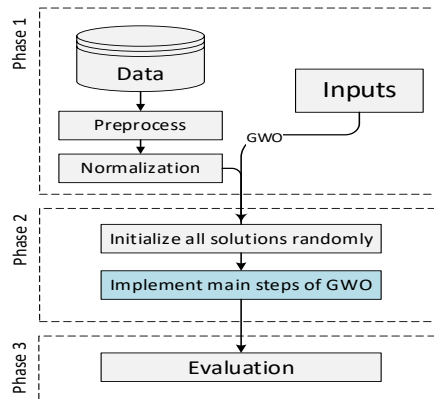


Figure 2. The block diagram of the proposed algorithm

First Phase: This phase is comprised of multiple steps in which the dataset is prepared and the algorithm's primary parameters are initialized. In the suggested procedure, the first step is the data preparation process. It entails reading the dataset and preparing it in two basic steps: I Get the Dataset, and ii) Preprocess and normalize the dataset within the range [0,1] using the *MinMax* method, which is formulated as follows:

$$N_v = \frac{X_v - Min}{Max - Min} \quad (12)$$

Where N_v represents the normalized value, while X_v represents the original value. *Min* and *Max* denote the maximum and minimum values of a specific feature respectively. The locations of the missing values in the dataset are recorded with the intention of using IGWO to populate these empty locations with appropriate values during the

subsequent phase. Then, the computational parameters such as the number of the wolves – or the swarm size, the maximum number of optimization iterations, and other GWO regulating variables are entered.

Phase 2: In this phase, the steps of GWO are executed. First, all candidate solutions are randomly initialized in the search space, then the determination of the best solutions (*Alpha, Beeta* and *Delta*). Each solution is uniquely represented, with the number of dimensions equal to the total number of missing values in the dataset. Following the rest of GWO algorithm steps, the optimal replacement of each of which is determined.

Phase 3: In this phase, the best solution obtained using GWO is evaluated in terms of classification accuracy, error rate, sensitivity and specificity.

3.2 GWO for Missing Values Estimation

The following procedures must be performed for the GWO algorithm to work efficiently. These procedures steps are not strictly required, but they do aid in effectively applying the algorithm. The major GWO steps are depicted in Figure 3.

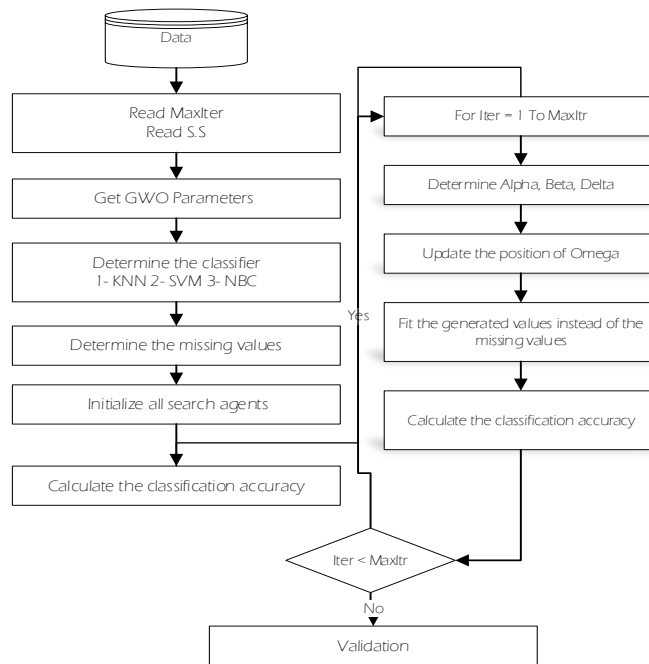


Figure 3. GWO algorithm for predicting the missing values

1. Initialize the $[S.S \ MaxIter]$ parameter vector with the desired values. Two limiting parameters, known as the Upper bound (UB) and Lower bound (LB), are used to restrict the search space. The case study informs the initialization of UB and LB, while the various scenarios inform the initialization of Swarm Size (S.S) and the maximum number of iterations.

2. Initialization: Create a uniformly distributed random position for each search agent or wolf in the swarm as follows:

$$X_i = (UB - LB) \times Rand(0,1) + LB \quad (13)$$

Where *Rand* is a randomization method that generates a random value in the range [0,1].

3. FitnessFunction: In order to assess the quality of each generated or estimated solution in terms of its classification precision (A). The accuracy is determined using a method called K-Fold Cross Validation, and the value of K is 5. In this investigation, we make use of three distinct classification models: the K-Nearest Neighbor (KNN), the Support Vector Machine (SVM), and the Naive Bayesian Classifier (NBC).
4. Determination of W_{Alpha} , W_{Beta} , and W_{Delta} . In this phase, the best wolves should be identified by ascendingly sorting the solution. If there is a new solution with superior fitness, maintain it as W_{Alpha} ; otherwise, retain the original W_{Alpha} .
5. Position Updating: Each wolf's (and the omegas') location is then updated. The fitness of each search agent is then re-evaluated to see if a better solution has emerged, and if so, the new alpha should be used.
6. Check the Boundaries Limits: Next, it must be determined if the values received at the search agent's new position fall within the search space.

$$F_i.Position = \begin{cases} LB & \text{If } X_i.Position < LB \\ UB & \text{If } X_i.Position > UB \end{cases} \quad (14)$$

Then, F_i is evaluated using the fitness function explained in Step 3.

7. Stop Condition:

The first two steps are only carried out once, while the remaining three steps are repeated t times. This means the algorithm moves on to step 4 only if t is still lower than the $MaxItr$ value determined in the first step. If not, return the most recent F_{Best} and leave the loop.

The pseudo-code of the proposed IGW is summarized in the algorithm below.

Imputation GWO Algorithm (IGWA)

1. Set Initial values for all parameters ($N, MaxItr$)
2. Determine the positions of Missing Values (MVs)
3. Determine the classifier (1. KNN, 2. SVM, 3. NBC)
4. Initialize all search agents in the swarm via eq. 3.13
5. Evaluate each search agent using the fitness function (i.e., classification accuracy)

5. While ($Itr \leq MaxItr$)
6. Determine the best solutions ($Alpha, Beta, Delta$)
7. For $i = 1$ To N
8. Update the position of F_i via eq. 3.5 - 3.11
9. Check the boundaries via eq. 3.14
10. Full the dataset and update the Fitness value of X_i
11. Next i
12. Rank the swarm and determine F_{Best}
13. Loop ($Itr + 1$)
14. Return $Alpha$

3.3 PIDD Dataset

Originally, the "National Institute of Diabetes and Digestive and Kidney Diseases" compiled the data set. The investigations adhered to the guidelines established by the WHO. This research involved women of PIMA Indian heritage who were at least 21 years old. In this work, we chose to use this dataset since it has been used by other researchers in the past to create classification systems, which would make it easier to compare our results to those of other studies that have tackled the challenge of PID diagnosis. Each of the 768 cases in this dataset is comprised by a total of 8 features or attributes.

Table1 showed all the features in this dataset and their numerical values while the ranges, mean, median, and standard deviation for each attribute in the datasets are presented in Table 2.

Table 1. The features set in the dataset

N	Name of the Attribute	Type
1	No. of times pregnant	Numeric
2	Plasma Glucose Concentration	Numeric
3	Diastolic Blood Pressure	Numeric ($mmHg$)
4	Triceps skin fold thickness	Numeric (mm)
5	2 Hours Serum insulin	Numeric ($\mu U/ml$)
6	Body mass index	Numeric (kg/m^2)
7	Diabetes pedigree function	Numeric
8	Age	Numeric (years)

Table 1. Statistical Information

F	Min	Max	Mean	Median	Std.dev
1	0	17	3.845	3	3.370
2	0	199	120.673	117	32.282
3	0	122	69.105	72	19.356
4	0	99	2.536	23	15.952
5	0	846	79.788	30.5	115.236
6	0	82.7	32.058	32.1	8.100
7	0.078	2.42	0.474	0.375	0.332
8	21	81	33.241	29	11.760

In order to finish the process of classification, the final value was first converted to a binary representation, and then it was split into two separate groups: "Class Zero (Non-diabetic) and Class One (Diabetic). During model training, the initial eight features were used as input, and the final value was used as the classification's truth. 268 individuals out of a total of 552, or 34.9%, were diagnosed with diabetes, whereas the remaining 348 patients, or 65.1%, did not have diabetes (500 cases). There are two key reasons why the overwhelming majority of medical case studies suffer from insufficient data. As a starting point, there are certain individuals who cannot pay for critical medical testing due to financial constraints. Second, due to the limited amount of time available, tests were not always precisely recorded. Depending on the nature of the omitted data, inaccurate categorization results are conceivable. Examining Table 3 reveals that the PIMA dataset lacks crucial information in a considerable number of categories. With the exception of the very first feature, which is completely devoid of any missing values, all of the other qualities contain missing values.

Table 3. Information about missing values in the dataset

N	Name of the Attribute	Missing Values
1	No. of times pregnant	-
2	Plasma Glucose Concentration	5
3	Diastolic Blood Pressure	35
4	Triceps skin fold thickness	227
5	2 Hours Serum insulin	374
6	Body mass index	11
7	Diabetes pedigree function	1
8	Age	63

4. Results and Discussion

4.1 Experimental Settings

To assess the performance of the suggested imputation algorithm, a series of experiments must be conducted. Each experiment in the review process uses a unique combination of criteria for evaluation. The imputation procedure was developed in MATLAB (version 2018b), and it has been tested on Windows 10 (64-bit) using a 2.6 GHz Intel Core i7 processor and 8 gigabytes of RAM. On the other hand, the settings of the experiments depend mainly on the structural parameters, which are: number of iterations (ITR), and the number of solutions in the swarm (N). In order to validate the effect of these two parameters on the performance of the algorithm, several values of each one are implemented.

First, the efficiency of nature optimization process can be affected by varying the number of solutions considered. In some cases, a larger value of swarm size improves performance, albeit doing so can slow down the procedure. Therefore, multiple tests are conducted, with $N = \{10,15,20,30\}$, to discover the best N feasible. While the second parameters is the number of iterations, (Itr), it affects an optimization algorithm's efficiency is the number of iterations used. Multiple trials are conducted with $ITR = \{25, 50, 100, 200\}$ to find the optimal ITR . The last parameter is the classifier itself which is utilized for measuring the classification accuracy, i.e., the fitness function. Three separate classifiers are required for the fitness function of the proposed imputation algorithm, as was mentioned previously. The three variants of the proposed imputation algorithm are the Imputation algorithm with K-Nearest Neighbors (IGWO-KNN), the Imputation algorithm with Support Vector Machine (IGWO-SVM), and the Naive Bayesian Algorithm (IGWO-NBC).

The results of the tests may be summarized by saying that the settings reflect the accuracy that was accomplished via the utilization of a variety of classifiers and the improved dataset. The improved dataset was partitioned into a training set that comprised 65% of the data and a testing set that comprised the remaining 35% of the data. There was a total of ten run of each test, and the outcomes of each and every one of these simulations are presented in Table 4, these results have been calculated and compared based on several accuracy calculation scenarios. Firstly, Original Accuracy (Acc), which reflects the accuracy that was obtained based on the initial dataset that had values that were missing. Secondly, K-Fold Cross validation (CV), which reflects the accuracy that was achieved by applying the imputation algorithm that was proposed. While the third and final method are Training-Testing Accuracy ($OR_c. Acc$) and Optimized Training-Testing Accuracy ($OP_c. Acc$), which indicates the accuracy that was acquired using various classifiers with the initial and enhanced dataset respectively, when the dataset is split into a training set consisting of 70% of the data and a testing set consisting of 60% of the data. Table 4 below contains the experimental data obtained for all.

Table 4. Tests Settings

Test	<i>N</i>	<i>ITR</i>
1	5	50
2	5	100
3	5	200
4	5	300
5	10	50
6	10	100
7	10	200
8	10	300
9	15	50
10	15	100
11	15	200
12	15	300
13	20	50
14	20	100
15	20	200
16	20	300

Each test was run through ten iterations, and the results of those iterations are displayed here.

4.2 Discussion

A) IGWO Integrated with SVM (IGWO-SVM)

In order to evaluate the many different solutions produced by the swarm, a support vector machine (SVM) classification model is being used for the purposes of this experiment. The validity of the experiments has been established through the application of the test set that is detailed in Table 4. This procedure has been repeated ten times, and the results of these iterations, as well as their averages, are shown in the two figures that are shown below.

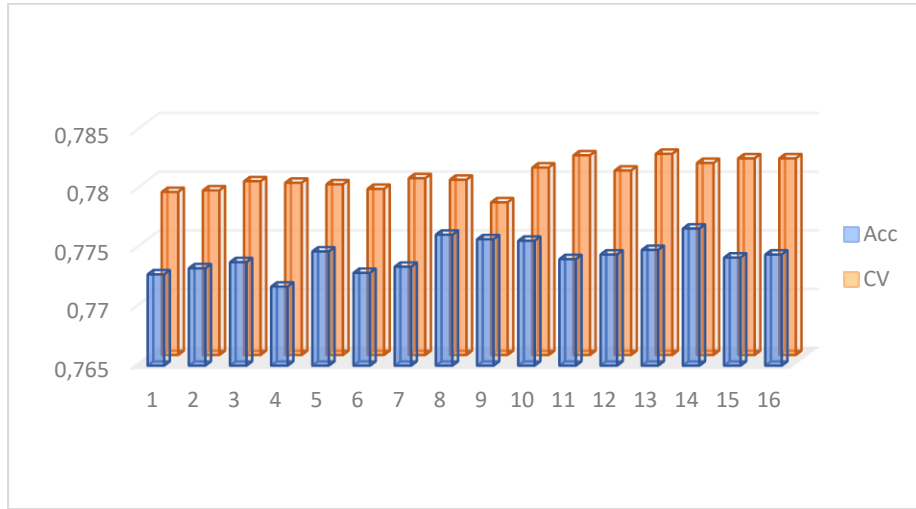


Figure 3. Comparison between average results results of the accuracies using cross-validation

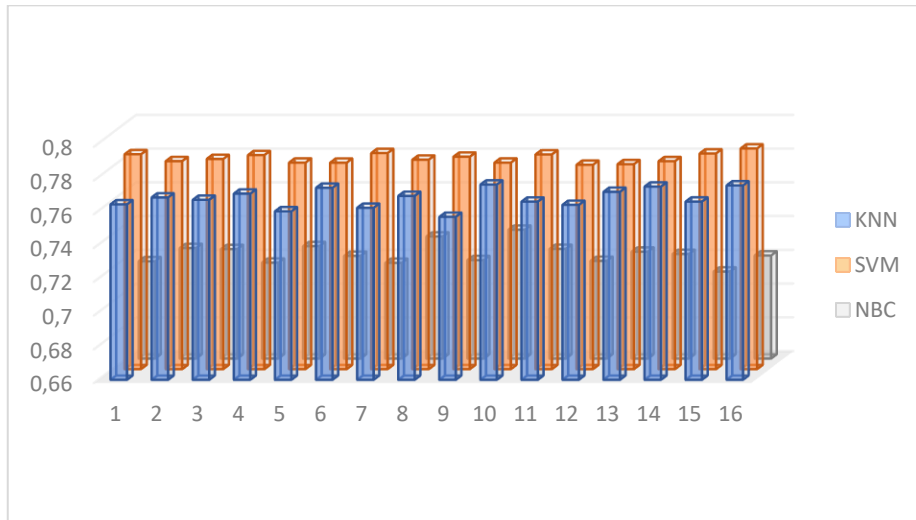


Figure 4. Comparison between the average accurizes using Testing-Training Accuracy

The data that were shown before demonstrated dramatically different outcomes when contrasted with the findings of the initial experiment. When compared to the results of KNN and NBC, the SVM results shown in Figure 3 indicated an outstanding level of performance. In contrast, the findings obtained in this experiment's dataset were noticeably superior when compared to those gained utilizing the original dataset with missing values.

B) IGWO Integrated with NBC (IGWO-NBC)

The NBC classification model is used for the purpose of evaluating the many different solutions that the swarm has come up with for the purpose of this experiment. The results of the experiments have been validated through the application of the analysis that is detailed in Table 4. This procedure has been repeated ten times, and the results of these iterations, as well as their averages, are shown in

the two figures that are shown below.

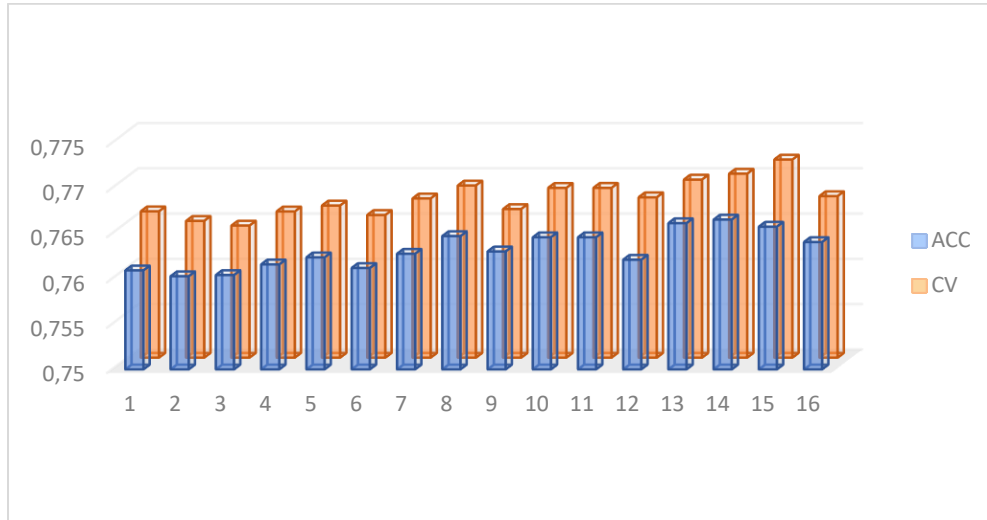


Figure 5. Comparison between average results of the accuracies using cross-validation

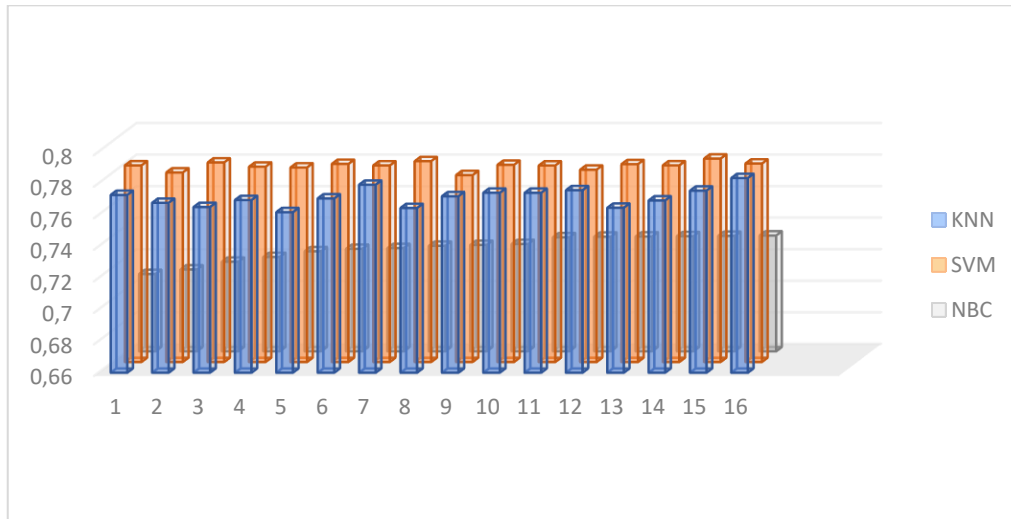


Figure 6. Comparison between the average accurizes using Testing-Training Accuracy

According to the findings that were shown in Figure 5 and 6 compared to the other three classifiers, NBC had the worst performance. Additionally, the accuracy obtained based on the dataset filled using the recommended imputation approach was superior than the accuracy gained based on the original dataset in every single test that was carried out. This was the case regardless of the type of test that was carried out. As a consequence, NBC classifiers increased the overall performance of the proposed algorithm, albeit to a lower level and with results that are less accurate than those provided by the various other classifiers.

C) IGWO integrated with KNN (IGWO-KNN)

In this section, the KNN classification model is used to the problem of determining how effective each solution or search agent in the swarm is.

Figure 7 illustrates an average of the findings obtained through performing cross validation on both the original dataset and the enhanced dataset. While Figure 8 presents the findings of a comparison that was carried out using

the holdout results obtained from three different classifiers, and the third picture illustrates how these findings were attained.

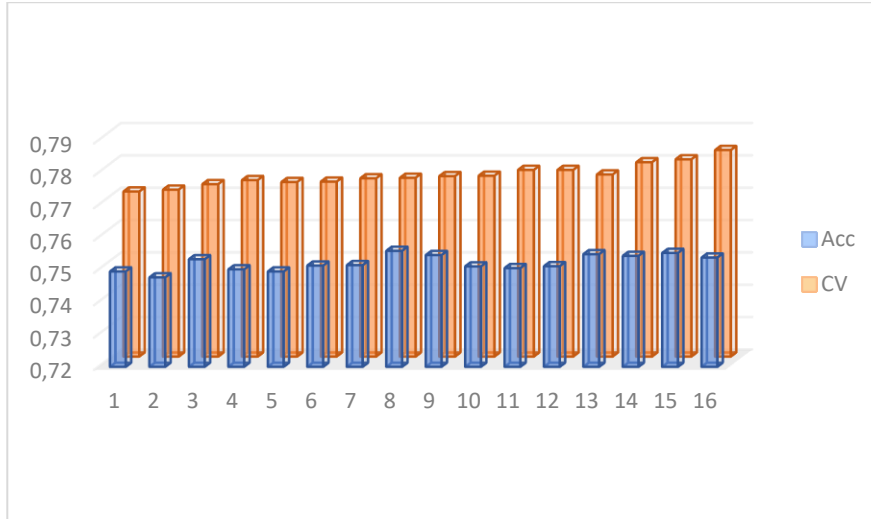


Figure 7. Comparison between average results of the accuracies using cross-validation

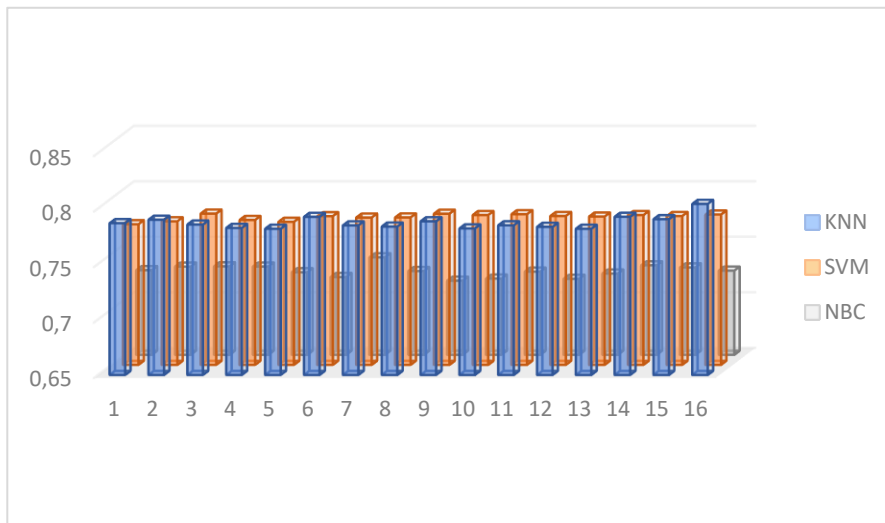


Figure 8. Comparison between the average accurizes using holdout

The Two illustrated figures clearly showed that the proposed imputation technique has improved the classification outcomes. In summary, the suggested algorithm was applied to the aforementioned missing values in the PIDD dataset, and the gaps in the dataset were afterwards filled with values that were more suitable for the process of prediction and classification. In addition, as seen in the second image, the KNN model performed the best when the newly built dataset was validated. This conclusion was reached by comparing its outcomes to those of other available models. This is a considerable difference when compared to the other two classifiers that were employed. In contrast, SVM's performance is comparable to that of KNN, and NBC's performance was the lowest of the three.

4.3 Discussion

As can be seen, the proposed GWO imputation technique, which is based on all classifiers, successfully dealt with the issue of missing values in the PIDD dataset. In contrast to other tests developed from the original dataset, the NBC classifier's poorest score was still higher than that of any other. What's more, a brief summary of the next three considerations follows.

First, when using KNN as a fitness function, the KNN classifier trained on the 35% testing set outperformed the other classifiers by a large margin, as shown by the results of the holdout validation trials. However, when support vector machines (SVMs) or neural network-based approaches (NBCs) were utilized as fitness functions, KNN ranked lower. Sequential Minimum Optimization (SMO) was used to tune the parameters C and γ in the RBF kernel function, which significantly improved the Support Vector Machine's performance (SVM). When using SVM or KNN, the results were consistently over 77%, while when using NBC, the results were in the range [70%-78%]. Experiments involving cross-validation revealed that increasing the number of solutions, also known as the size of the solution search space (i.e., Tests 1 – 16), improved the quality of the results. The solution search space grows in proportion to the number of solutions, which is another way of putting it. That is to say, there is a clear correlation between the total number of solutions and GWO's effectiveness when searching the database. While number of repetitions ITR does have some bearing on GWO, it is not as significant as other factors.

Several classifiers were used in the preceding sections to evaluate the proposed GWO imputation method. A total of sixteen tests were used in conjunction with the Cross Validation and Holdout validation methods to reach this conclusion. In this part, we compare the proposed imputation approach to four standard imputation methods using the PIDD dataset, and draw conclusions on their relative merits. These approaches are, firstly, IMP_1 : Removing the entire row that does not have the required information. The classification process could be affected by the fact that less training data is available when using this approach. Secondly, IMP_2 : In some cases, it could be helpful to simply replace missing data with zeros. However, the value of zero could potentially affect categorization when the classification model is trained with new data. Thirdly, IMP_3 : the missing value(s) are replaced by the average (or mean) of the remaining values of the characteristic. This approach is better than previous ones in the great majority of cases. This is because the preceding values of the same attribute greatly influence the values that are created for the attribute. The last approach is IMP_4 : Using a random number generator within the range of [0,1] to fill in the missing values. This approach may have value-related effects on the categorization models. Alternatively stated, this may cause the numbers to be noisy or cause a change in the sample distribution.

A total of ten distinct runs have been executed, during which the aforementioned techniques have been combined with the three classifiers that have been used over the course of this inquiry. After that, the maximum score, the mean, and the standard deviation were calculated for each classifier. In the following table, which can be found below, comparisons are made between the IGW-KNN, IGW-SVM, and IGW-NBC algorithms and the four alternative approaches.

Table 5. Different imputation approach vs. IGWO

Approach	Classifier	Accuracy	Std. Dev
<i>Without IMP</i>		0.75008	0
<i>IMP₁</i>		0.73641	0.24782
<i>IMP₂</i>		0.75421	0.21412
<i>IMP₃</i>	KNN	0.76822	0.19321
<i>IMP₄</i>		0.76025	0.20411
IGWO		0.790634	0.18695
<i>Without IMP</i>		0.77935	0
<i>IMP₁</i>		0.75982	0.22782
<i>IMP₂</i>		0.76724	0.21842
<i>IMP₃</i>	SVM	0.77942	0.20142
<i>IMP₄</i>		0.77834	0.19782
IGWO		0.79527	0.002744
<i>Without IMP</i>		0.70414	0
<i>IMP₁</i>		0.69842	0.25413
<i>IMP₂</i>		0.69624	0.24821
<i>IMP₃</i>	NBC	0.70128	0.20421
<i>IMP₄</i>		0.70431	0.20142
IGWO		0.74278	0.00754

The suggested imputation algorithm outperformed the others. A_1 method deleted many samples from the dataset, reducing the training set and causing the poor performance with all classifiers. The second approach, A_2 , generated nearly identical results as the first, but they were slightly better because it substituted zero for missing data. A_3 and A_4 were better than A_1 and A_2 because they filled in missing data with mean or random values. This improved their accuracy over the first two methods. This improved A_3 and A_4 's results. Because these procedures filled in the data gaps, it's better to use the generated values than zero or the missing data sample. Using zero or removing the sample with incomplete data doesn't fill in the gaps.

The IGWO-KNN algorithm delivered the best results overall; however, the IGWO-SVM approach produced superior outcomes on average. Moreover, the IGWO-SVM method generated the best results overall. It was

determined through the use of the standard deviation that IGWO-SVM and IGWO-NBC are more reliable than IGWO-KNN. IGWO-KNN was the one with the least consistent results.

4. Conclusion

In the majority of medical datasets, the absence of data or values is problematic. It occurred for two primary reasons: a) the expense of the medical testing, and b) the error in documenting all the characteristics due to time limits or human error. For this reason, a procedure known as "Imputation" has been developed to fill in the gaps left by the missing data. As an imputation technique, Firefly Algorithm (FA) is utilized in this study. The resulting missing values are evaluated using three distinct classifiers: K Nearest Neighbors (KNN), Support Vector Machine (SVM), and Nave Bayesian Classifier (NBC). The proposed imputation algorithm was tested based on the results of two primary experiments. First, by utilizing cross validation with 5 folds, and then in the second experiment, the algorithm was tested by using the holdout validation method, where the generated dataset was split into training set (70%) and testing set (30%), respectively. The findings demonstrated that the suggested imputation approach was able to estimate the missing values in PIDD and improved the classification accuracy of all classifiers. The SVM show ranked highest, while the NBC show ranked lowest.

Referances

- Soofi, A. A., & Awan, A. (2017). Classification techniques in machine learning: applications and issues. *J. Basic Appl. Sci*, 13, 459-465.
- Tao, H., Awadh, S. M., Salih, S. Q., Shafik, S. S., & Yaseen, Z. M. (2022). Integration of extreme gradient boosting feature selection approach with machine learning models: application of weather relative humidity prediction. *Neural Computing and Applications*, 34(1), 515-533.
- Tao, H., Salih, S., Oudah, A. Y., Abba, S. I., Ameen, A. M. S., Awadh, S. M., ... & Yaseen, Z. M. (2022). Development of new computational machine learning models for longitudinal dispersion coefficient determination: Case study of natural streams, United States. *Environmental Science and Pollution Research*, 29(24), 35841-35861.
- Sowmya, R., & Suneetha, K. R. (2017, January). Data mining with big data. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)* (pp. 246-250). IEEE.
- Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., & Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10), 1087-1091.
- García-Laencina, P. J., Sancho-Gómez, J. L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: a review. *Neural Computing and Applications*, 19, 263-282.
- Howell, D. C. (2007). The treatment of missing data. *The Sage handbook of social science methodology*, 208, 224.

- Pigott, T. D. "Handling Missing Data," in Using Propensity Scores in Quasi-Experimental Designs, 1 Oliver's Yard, 55 City Road London EC1Y 1SP: SAGE Publications, Ltd, 2009, pp. 245–271. doi: 10.4135/9781452270098.n11.
- Choudhury, S. J., & Pal, N. R. (2019). Imputation of missing data with neural networks for classification. *Knowledge-Based Systems*, 182, 104838.
- De Silva, H., & Perera, A. S. (2016, September). Missing data imputation using Evolutionary k-Nearest neighbor algorithm for gene expression data. In 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 141-146). IEEE.
- Alhroob, A., Alzyadat, W., Almukahel, I., & Altarawneh, H. (2020). Missing data prediction using correlation genetic algorithm and SVM approach. *International Journal of Advanced Computer Science and Applications*, 11(2).
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- Long, W., Cai, S., Jiao, J., & Tang, M. (2020). An efficient and robust grey wolf optimizer algorithm for large-scale numerical optimization. *Soft Computing*, 24(2), 997-1026.
- Mohammadzadeh, A., Masdari, M., Gharehchopogh, F. S., & Jafarian, A. (2021). Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. *Evolutionary Intelligence*, 14, 1997-2025.
- Tawhid, M. A., & Ali, A. F. (2017). A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. *Memetic Computing*, 9, 347-359.
- Zhang, X., Lin, Q., Mao, W., Liu, S., Dou, Z., & Liu, G. (2021). Hybrid Particle Swarm and Grey Wolf Optimizer and its application to clustering optimization. *Applied Soft Computing*, 101, 107061.