**A feature selection approach based on cuttlefish algorithm for diagnosing type 2 diabetes**

**Marwan Al-Tawil\*, Basel A. Mahafzah, Arar Al Tawil, Ibrahim Aljarah, ...**

King Abdullah II School of Information Technology

The University of Jordan

Amman – Jordan

\*Corresponding author: m.altawil@ju.edu.jo

b.mahafzah@ju.edu.jo

[arartawil@gmail.com](mailto:arartawil@gmail.com)

Ibrahim Aljarah

…

**ORCID**

*Marwan Al-Tawil* https://orcid.org/0000-0002-5547-9691

*Basel A. Mahafzah* <https://orcid.org/0000-0003-3979-1076>

*Arar Al Tawil* <https://orcid.org/0000-0002-3194-1407>

*Ibrahim Aljarah*

...

### Abstract

Type 2 diabetes is a common life-changing disease that has been growing rapidly in recent years. According to the World Health Organization, around 90% of patients with diabetes around the world suffer from type 2 diabetes. Although there is no permanent cure for type 2 diabetes, however, this disease needs to be diagnosed at an early stage to help provide prognosis support to allied health professionals and develop an effective prevention plan. This could be accomplished through analyzing medical datasets using data mining and machine learning techniques. The main goal of this work is to improve the performance of existing approaches for predicting Type 2 Diabetes. For this, we utilize a bio-inspired algorithm called the cuttlefish for selecting essential features in the medical data pre-processing stage. To validate the proposed approach, we compare the performance of the proposed approach to a well-known bio-inspired feature selection algorithm called genetic algorithm. The selected features from the cuttlefish and genetic algorithms are utilized with different classifiers. The implementation of the algorithms is applied over two datasets, the Pima Indian diabetes dataset and the hospital Frankfurt diabetes dataset. The results show that the cuttlefish algorithm has better accuracy rates, especially with the increase in the number of instances in the dataset.

**Keywords** Diabetes; Cuttlefish algorithm; Feature selection; Genetic algorithm

# Introduction

Chronic diseases are the largest cause of death in the world [1]. Chronic diseases are a group of illnesses that do not transmit infection from one person to another, as they have nothing to do with viruses or bacteria. Such diseases are slow to occur, and patients do not show any signs of illness. Chronic diseases are closely related to the level of our lifestyle and healthy behavior, such as the type of food we eat, our motor behavior (daily exercises), or some bad habits such as smoking. One of the most common chronic diseases is diabetes. According to the World Health Organization (WHO), the majority of people with diabetes have type 2 diabetes[[1]](#footnote-1). This type of diabetes is largely caused by excess body weight and physical inactivity. Recent statistics by the International Diabetes Federation (IDF) showed that 1 in 11 adults have diabetes (around 463 million worldwide) and type 2 diabetes counts for 90% of all diabetes cases[[2]](#footnote-2). The IDF reported that the current cases of diabetes will rise by the year 2045 to around 700 million cases. This is worrying, especially since type 2 diabetes has some scary side effects such as malfunctioning and permanent damage to body organs. In the long run, type 2 diabetes may result in several life-threatening conditions such as coma, destruction of pancreatic beta cells, joint failure, and many other conditions. Injection of adequate insulin is the best remedy to treat type 2 diabetes. Although, there is no long-term cure, however, type 2 diabetes can be controlled if diagnosed at an early stage [2]. One way for predicting type 2 diabetes is through predictive modeling and analysis of medical datasets.

Researchers have been using data mining and machine learning techniques to analyze medical datasets to find the best ways for increasing the accuracy and efficiency in predicting type 2 diabetes [3]. Data mining methods are used to preprocess the dataset to discover hidden patterns and select the most relevant dataset of features [4]. This will enable machine learning algorithms to train faster in diagnosing type 2 diabetes [5]. However, analyzing medical datasets is not a trivial task as medical datasets are often massive in dimensions and have complex features, leading to data noise and dependency among features. Therefore, it is vital to remove irrelevant and redundant features before analyzing datasets to increase prediction accuracy and improve result comprehensibility. Feature selection is a complex process, requiring artificial intelligence methods to solve it [6]. The success of the feature selection process depends on reducing the number of attributes and increasing accuracy rates. There have been several works on detecting type 2 diabetes using ordinary feature selection algorithms (a recent review can be found in [7]).

However, limited research focused on using bio-inspired metaheuristic feature selection algorithms for diagnosing type 2 diabetes [8]. The efficiency of metaheuristic algorithms can be attributed to the fact that they imitate the best features in nature [9]. Among several bio-inspired feature selection algorithms, the Genetic Algorithm (GA) has proven to be one of the most effective evolutionary techniques for solving a wide range of global optimization problems [10]. To the best of our knowledge, the GA is the only metaheuristic algorithm that has been utilized in diagnosing type 2 diabetes. The work in [2] combined the GA with Multiple Objective Evolutionary Fuzzy Classifier (MOEFC) to predict the type 2 diabetes in Pima Indian Diabetes (PID) dataset. While the GA has successfully handled feature selection in detecting type 2 diabetes with an accuracy rate of (83.0435%), however, this work aims to enhance the accuracy rates by utilizing a bio-inspired feature selection algorithm called the Cuttlefish Algorithm (CFA). The essential subset of features selected by the CFA is evaluated with six popular classifiers namely, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR).

The implementation of the algorithms is applied over two datasets: the Pima Indian Diabetes (PID) dataset that is extracted from the UCI repository and the hospital Frankfurt diabetes (HFD) dataset. The PID and HFD datasets are two of the most widely used medical datasets for predicting type 2 diabetes [7]. Both datasets are publicly available at Kaggle.com[[3]](#footnote-3). The PID contains information about 768 instances (i.e., patients), whereas the HFD dataset contains 2000 instances. Both datasets share the same features (eight features) such as insulin level, Body Mass Index (BMI), and blood pressure (the full list of features is in Table 2). The evaluation showed that accuracy rates for the CFA are better than the GA, especially with the increase in the number of instances in the dataset. Thus, the main contributions of this work are:

1. Apply the CFA and GA on PID and HFD datasets for feature selection.
2. Combine the CFA and GA with several classification algorithms to predict type 2 diabetes.
3. Analyze the performance of the CFA and the GA over the PID and HFD datasets.

The rest of the paper is structured as follows. Section 2 describes related work. Section 3 presents the preliminaries. The proposed approach is presented in Sect. 4. Section 5 describes the results and Sect. 6 provides discussion and concludes the paper.

# Related Work

In this section, we will review feature selection algorithms over medical datasets to justify the main contributions of this paper. In [2], the authors proposed a feature selection approach that applied the GA to the PID dataset. The best four features in the dataset were identified. Several classification approaches were applied such as Naive Bayes (NB), Decision Tree (DT), and MOEFC. The results showed that the MOEFC provided the best accuracy rate at 83.04%. The work in [11] developed a diabetes prediction method where feature election was applied to two datasets: the PID dataset and the Diabetes 130-US hospitals dataset. After the feature selection process, the authors used the Support Vector Machine (SVM) classification algorithm. Standard features were used to combine the datasets, and then feature selection using the ranker and wrapper algorithms were applied. 10-fold cross-validation was used. The results showed that the ranker algorithm had the best accuracy at 72.49%, whereas the wrapper algorithm had 71.11% accuracy. In [12] the authors predicted type 2 diabetes using several classification algorithms, namely, logistic, K-Nearest Neighbor (KNN), SVM, NB, DT, and Random Forest (RF). The results showed that the RF algorithm had the highest accuracy at 77.4%. To diagnose Parkinson's, the CFA was applied in [13] as a way of detecting this illness at an early stage. This approach had a 94% accuracy rate with KNN for classification. The work in [14] proposed an enhanced moth flame optimization (MFO) feature selection algorithm. The approach was applied to 23 medical datasets that were taken from UCI and Kaggle repositories and the results showed that the proposed approach outperforms other methods in most of the selected datasets. In [15] the authors used GA which is based on hierarchical feature selection to optimize handwritten word images features. The proposed method was applied to 12K words and results showed that word recognition was enhanced by 1.28% compared to recognition obtained with the unreduced feature set. An approach for encoding gene data based on unsupervised deep learning clustering with GA has been proposed in [16]. Three classifiers were used namely, SVM, KNN, and RF. The proposed approach was applied to six cancer datasets. Accuracy results ranged between (0.66 for the RF and 0.99 for the SVM). A hybrid deep learning model that is based on Laplacian Score-Convolutional Neural Network was proposed in [17] for gene selection and classification of given cancer data. Ten datasets were used to test the performance of the proposed model. The results showed that the proposed model outperformed other algorithms. An approach for classifying and detecting Diabetes Mellitus has been proposed in [18]. The authors used F-Score feature selection to identify the valuable features in the dataset. Fuzzy SVM was then used to train the dataset and generate fuzzy rules. The evaluation results showed an accuracy of 89.02%. Table 1 summarizes related work.

Metaheuristic algorithms have become powerful optimization tools for complex problems [19]. Among several purposes of metaheuristic algorithms, we focus on the feature selection problem. Earlier work has shown that metaheuristic algorithms are highly efficient compared to ordinary features selection algorithms [20]. However, there is limited research on utilizing metaheuristic algorithms for detecting type 2 diabetes. To the best of our knowledge, the GA is the only natural bio-inspired metaheuristic algorithm that has been utilized for detecting type 2 diabetes [7]. In this work, we utilize another nature-inspired metaheuristic algorithm called the CFA as a search strategy to ascertain the optimal subset of features for detecting type 2 diabetes, with the MOEFC as a judgment on the selected features. To evaluate the proposed approach, we applied the GA and CFA to the PID datasets.

**Table 1** Summary of related work

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ref.** | **Feature selection algorithm** | **Classifier algorithm** | **Dataset** | **Result** | **Domain** |
| [2] | GA + CFS | Fuzzy | PID | 0.83 | Diabetes |
| [11] | Ranker and wrapper | SVM | Merge (Pima dataset and 130-US hospital) | Ranker → 0.72 Wrapper → 0.71 | Diabetes |
| [12] | All data | RF | PID | 0.77 | Diabetes |
| [18] | Features extraction (PCA) | SVM | PID | 0.89 | Diabetes |
| [13] | CFA | KNN | Handwriting dataset | 0.94 | Parkinson’s disease |
| [15] | GA-based Hierarchical Feature Selection (HFS) | - | Word handwriting dataset | Improve by 0.12 | Improve features selection |
| [16] | Deep learning-based clustering with GA | KNN, SVM,  RF | Cancer datasets | 0.98, 0.85 and 0.94 | Cancer |
| [14] | Improved MFO | KNN | Medical dataset UCI and Kaggle | Outperforms other methods in the literature across 75% of the data set | Medical diagnosis |
| [17] | LS-CNN | LS-CNN | Cancer datasets | 0.93 | Cancer |

# Preliminaries

In this section, we formally describe the main algorithms; namely, the cuttlefish algorithm and genetic algorithm, which are used throughout this paper to better understand the research context.

## **3.1 Cuttlefish algorithm**

The Cuttlefish Algorithm (CFA) was proposed in [21] as a bio-inspired optimization algorithm that mimics the color-changing behavior of a marine animal called cuttlefish. Compared to other bio-inspired algorithms such as GA and the Bees algorithm, the CFA algorithm has shown its effectiveness in solving optimization problems [22]. Reflection and visibility are two important processes that are responsible for changing the color of a cuttlefish. In the reflection process, the CFA simulates the light reflection mechanism by chromatophores, iridophores, and leucophores' skin layers (see Fig. 1), and the visibility process simulates the visibility of matching patterns of the cuttlefish. Overall, six cases are included in the CFA (see Fig. 2). Cases 1 and 2 (reflected color or light) are produced by interacting cells of chromatophores and iridophores' skin layers. Muscles of the chromatophore cells are stretched or shrank, and iridophores cells (light-reflecting cells) reflect the light that is coming from chromatophores cells causing the cells to penetrate. In cases 3 and 4, the iridophores cells reflect light (with specific color) from the outside environment. In case 5, the light is coming through chromatophores cells with a specific color and the color of reflected light is very similar to the incoming light. The incoming color is assumed to be the best solution (Best), and the reflected color represents any value around the Best solution. In case 6, the leucophores cells will reflect the incoming light. These cells work as a reflecting mirror of the predominant wavelength of light in the environment, reflecting white color (of white light) and brown color (of brown color). Accordingly, a cuttlefish blends itself based on the surrounding environment. This case works as initialization and is used to find new solutions.

Algorithm I describes the CFA. Like other meta-heuristic optimization algorithms, the CFA starts with random solutions for initializing the population. After that, the six cases shown in Fig. 2 are applied until a stop condition is met, in our case it is the number of maximum iterations. The main steps of the CFA algorithm are summarized as follows. The algorithm takes the maximum number of iterations and four random values as an input and identifies the best four features as an output (Lines 1 and 2). The algorithm in Line 3 initializes the population and the number of features to be selected. Then, the algorithm evaluates the population using a fitness function (Line 4) and stores the best solution (Line 5). The population is divided into four independent groups (Line 6). The first two groups (G1 and G2) are used for global search, while the other two groups (G3 and G4) are used for local search. The combination of the visibility and reflection processes provides six different cases and provides a new possible solution using Eq. 1.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

|  |  |
| --- | --- |
| **Fig. 1** Diagram of cuttlefish skin detailing the  three main skin structures [21] | **Fig. 2** Reorder of the six cases in Fig. 1 [21] |

|  |  |
| --- | --- |
| **Algorithm I** The cuttlefish algorithm | |
| 1 | **Input:** *Max Iteration, v*1*, v*2*, r*1*, r*2*, Upper, Lower* | |
| 2 | **Output:** *Find the best* 4 *features* | |
| 3 | *Initialize the number of populations with dimensions* | |
| 4 | *Evaluate the fitness of the population* | |
| 5 | *Store the best solution* | |
| 6 | *Divide cells into four groups G*1*, G*2*, G*3*, and G*4 | |
| 7 | ***while*** *l* <= *Max Iteration* ***do*** | |
| 8 | *Calculate the average value of the best solution, and store it in the best* | |
| 9 | ***for*** *each cell in G*1***do // Case 1 and Case 2*** | |
| 10 | *Generate a new solution using Eqs.* 1, 2, and 3 | |
| 11 | *Reflection* = *random*(*r*1*, r*2)× *G*1[*i*]*.Points*[*j*] | |
| 12 | *Visibility* = *random*(*v*1*, v*2)×(*Best.Point*[*j*]) – *G*1[*i*]*.Points*[*j*] | |
| 13 | *Calculate the fitness for the new solution* | |
| 14 | ***if*** (*fitness* > *best subset*)***then*** *current* = *new solution* | |
| 15 | ***end if*** | |
| 16 | ***end for*** | |
| 17 | ***for*** *each cell in G*2***do // Case 3 and Case 4*** | |
| 18 | *Generate a new solution using Eqs.* 1 and 3 | |
| 19 | *Reflection* = *Best.Point*[*j*] | |
| 20 | *Visibility* = *ranodm*(*v*1*, v*2)*×*( *Best.Points*[*j*] – *G*2[*i*]*.Points*[*j*]) | |
| 21 | *Calculate the fitness for the new solution* | |
| 22 | ***if*** (*fitness* > *best subset*)***then*** *current* = *new solution* | |
| 23 | ***end if*** | |
| 24 | ***end for*** | |
| 25 | ***for*** *each cell in G3* ***do // Case 5*** | |
| 26 | *Generate a new solution using Eqs.* 1 and 7 | |
| 27 | *Reflection* = *Best.Point*[*j*] | |
| 28 | *Visibility* = *random*(*v*1*, v*2)×(*Best.Points*[*j*] *– AVbest*) | |
| 29 | *Calculate the fitness for the new solution* | |
| 30 | ***if*** (*fitness* > *best subset*)***then*** *current* = *new solution* | |
| 31 | ***end if*** | |
| 32 | ***end for*** | |
| 33 | ***for*** *each cell in G*4***do // Case 6*** | |
| 34 | *Generate a random solution using Eq.* 1 | |
| 35 | *P*[*i*]*.points*[*j*]= *random* ×(*Upper – Lower*)+ *Lower* | |
| 36 | *Calculate the fitness for the new solution* | |
| 37 | ***if*** (*fitness* > *best subset*)***then*** *current* = *new solution* | |
| 38 | ***end if*** | |
| 39 | ***end for*** | |
| 40 | *l* = *l* + 1; | |
| 41  42 | ***end while***  *Output the best result* (*best 4 features*) | |

The interaction operator, between chromatophores (i.e. stretch and shrink processes) and iridophores cells in cases 1 and 2 (: Lines 9–16) use reflection and the visibility of the pattern to produce a new solution through Eq. 2, where is a group of cells with and represent the cell and point of the cell in , respectively. Then, Eq. 3 formulates the visibility of the matching background, where *Best.Points* represent the best solution points; s a parameter that is used to find the stretch or shrink interval of the saccule and is the pattern’s visibility degree. Eqs. 4 and 5 are used to find the values of both, and , respectively ( and are two constants).

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | (3) |
|  |  | |  |
|  | | (4) | | |
|  | |  | | |
|  | | (5) | | |
|  | |  | | |

In cases 3 and 4 (Lines 17–24), a new solution is calculated based on the reflected light coming from the best solution and the matching pattern visibility (i.e. local search). Equation 6 (having *R* = 1) produces an interval around the best solution as a new search area.

|  |  |
| --- | --- |
|  | (6) |
| Similar to Eq. 6, the algorithm in Eq. 7 (case 5 - Lines 25–32), uses the leucophores cells to provide a new solution by reflecting light from the area around the best solution and visibility of the pattern, where *AVbest* is the average value of the *Best.Points*. Finally, a random solution is produced using the leucophores cells operator in case 6 (represents Lines 33–39).   |  |  |  | | --- | --- | --- | |  |  | (7) | | |

## **3.2 Genetic algorithm**

The Genetic Algorithm (GA) was proposed by John Holland in the 1970s as an important searching technique to find the best option for a set of available solutions [23]. The GA was applied to the PID dataset to reduce the number of features using a fitness function (Eq. 8) [2]. The algorithm initializes the population in the dataset and performs selection, crossover, mutation, and termination operators. The selection process is based on the concept of survival of the fittest. The experiment in [23] was simulated using a multi-object fuzzy classifier.

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

# Experiments and results

In this section, the experimental environment including datasets attributes, hardware, software, and tools specifications in addition to the feature selection and classification rate are presented. The performance metrics including accuracy, Kappa statistic, and mean absolute error are defined. The results are presented and discussed. However, Fig. 3 illustrates the main steps of the proposed approach followed in this research.

**Step-4**

**Evaluation**

**Step-3**

**Classification rate**

**Step-2**

**Feature   
selection**

**Step-1**

**Dataset**

**Fig. 3** Steps for predicting type 2 diabetes

## **4.1 Datasets**

Two popular datasets have been used to evaluate the proposed feature selection algorithm: Pima Indian Diabetes (PID)[[4]](#footnote-4) and Hospital Frankfurt Diabetes (HFD)[[5]](#footnote-5) datasets. The PID dataset is collected from the UCI machine learning repository which originated from the national institute of diabetes and digestive and kidney diseases and is used to predict whether a patient has diabetes, based on diagnostic measurements. The PID dataset contains information about 768 females and eight features. The HFD dataset is taken from the Hospital Frankfurt, Germany. The HFD dataset has 2000 instances and eight features. Both datasets share the same features (described in Table 2).

**Table 2** Attributes of the PID and the HFD datasets

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Attribute Name** | **Description** | **Data Type** |
| 1 | Pregnancies | Number of times pregnant | Numeric |
| 2 | Glucose | Plasma glucose concentration 2 hours in an oral glucose tolerance test | Numeric |
| 3 | Blood Pressure | Diastolic blood pressure (mm Hg) | Numeric |
| 4 | Skin Thickness | Triceps skinfold thickness (mm) | Numeric |
| 5 | Insulin | 2-Hour serum insulin (mu U/ml) | Numeric |
| 6 | BMI | Body Mass Index (weight in kg/ (height in m)^2) | Numeric |
| 7 | Age | Age in year | Numeric |
| 8 | Diabetes Pedigree Function | Diabetes diagnostic history of the person's relatives | Numeric |

## **4.2 Feature selection**

The abundant increase of medical data is incorporating varied attributes and features. Most attributes do not contribute to the result in the predictive applications, leading to increasing computation time and resources. Hence, selecting a subset of features to achieve high accuracy rates, is required. In this research, the CFA and the GA are implemented on the PID and HFD datasets to select the best subset of features. Features in the dataset are reduced by applying the cost function for the logistic regression for the CFA (Eq. 9) and the correlation-based feature selection for the GA (Eq. 8) [13], where *m* is the number of examples, *h* is the hypothesis function, and *Y* are output values. Table 3 lists the subset of features that were selected from CFA and GA.

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

**Table 3** PID Features selected by CFA and GA

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Algorithm** | **Selected Features** |
| PID | CFA | Glucose, Skin thickness, BMI, and Insulin |
|  | GA | Glucose, BMI, Diabetes pedigree function, and Age |
| HFD | CFA | Diabetes pedigree function, Age, Glucose, and BMI |
|  | GA | Pregnancies, Glucose, Insulin, and Age |

## **4.3 Classification rate**

Since we have successfully identified the most appropriate features in the PID and HFD datasets and cleaned up all potential noisy data based on the feature selection process in the previous step, the next step is to start the classification process, where the set of features is trained using different classifiers. We experimented on different parameters in CFA and GA such as the number of iterations (repetition of a process to generate an outcome) and population (populations are created randomly to find the best population size depending on the problem). We tested with 10, 20 to 70 iterations and 10, 20 to 100 populations.

## **4.4 Performance metrics**

In this section, we describe the metrics that are used to evaluate the performance of the feature selection algorithms. These performance metrics are accuracy, Kappa statistic, and mean absolute error.

* + 1. **Accuracy**

Accuracy is a measure of statistical bias that represents the proportion of success rate of a given test, where low accuracy values indicate a difference between a result set and true values. Accuracy (Eq. 10) uses four test measures, as shown in Table 4, which represent the classification of the possible result of a recommendation of an item to a user.

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

**Table 4** Classification of the possible result of a recommendation of an item to a user

|  |  |  |
| --- | --- | --- |
|  | **Recommended** | **Not recommended** |
| Preferred | True Positive (TP) | False Negative (FN) |
| Not preferred | False Positive (FP) | Ture Negative (TN) |

* + 1. **Kappa statistic**

Kappa statistic (*K*) is a metric that is used to examine classifiers by comparing observed and expected accuracy [24]. According to [25], it is advantageous to use the Kappa coefficient for comparison of the accuracy of classification algorithms. Values of Kappa statistic vary between 0 (agreement equivalent) and 1 )perfect agreement). Equation 11 represents the metric for *K*, where is the proportion of trials agreed by judges and is the proportion of trials in which agreement would be expected by chance. Interpretation of agreement strength [26] is indicated in Table 5.

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

**Table 5** Interpretation of Cohen’s Kappa statistic

|  |  |
| --- | --- |
| Kappa | Strength of agreement |
| <0.00 | Poor |
| 0.00-0.2 | Slight |
| 0.21-0.40 | Fair |
| 0.41-0.60 | Moderate |
| 0.61-0.80 | Substational |
| 0.81-1.00 | Almost perfect |

* + 1. **Mean absolute error**

The Mean Absolute Error (MAE) is used to test the mean absolute values of individual prediction errors over all instances in the test set. Equation 12 shows the MAE metric, where *yi*represents the prediction value, *xi* represents the true value, and *n* is the number of instances.

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

## **4.5 Results and discussion**

In this section, the CFA is evaluated against the GA using two datasets: PID and HFD. To our knowledge, the GA is the only metaheuristic algorithm that has been utilized for detecting type 2 diabetes. Both algorithms, the CFA and GA, are evaluated using six classifiers: K-NN, RF, DT, LR, SVM, and NB. The algorithms are trained on the same dataset with the same methodology to ensure the fairness of the results. The classifiers are available in the scikit-learn library in Python. Python provides built-in libraries that were used to implement the feature selection algorithms. Table 6 presents the hardware and software specifications used in the implementation.

**Table 6** Hardware, software, and tools used in the implementation

|  |  |
| --- | --- |
| **Hardware** | **Description** |
| Processor | Intel(R) Core (TM) i7-6700HQ CPU @ 2.60GHz |
| Memory (RAM) | 16.0 GB |
| Hard Disk Storage | 2 terabytes |
| Cache Memory | 6 MB Cache |
| **Software** | **Description** |
| System type | 64-bit Operating System |
| Operating System | Windows 10 |
| Programming language | Python |
| Environment | Anaconda |
| IDE | Pycharm |

The implementation of the CFA uses several input parameters (line 1 in Algorithm I). Table 7 describes the input parameters used in the implementation and their values. This value got them by experiments.

**Table 7** Input parameters and their values used in the CFA

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value** |
| *Dimension* | Number of features | 4 |
| *Upper* | Maximum limit to initialize population | 8 |
| *Lower* | Minimum limit to initialize population | 1 |
| *r1* | Maximum limit to find reflection | 1.5 |
| *r2* | Minimum limit to find reflection | -1.5 |
| *v1* | Maximum limit to find visibility | 2.5 |
| *v2* | Minimum limit to find visibility | -2.5 |

Table 8 presents the average accuracy of 30 runs for the CFA on the PID and HFD datasets using the logistic classifier. Both datasets (PID and HFD) are split into 70% for training and 30% for testing. The algorithms were examined using different iterations (10, 20 to 70) and populations (10, 20 to 100). The results showed that the CFA and GA provided better accuracy results with 50 iterations and 100 populations since…. Accordingly, all reported results from now on are based on 50 iterations and 100 populations.

**Table 8** Average accuracy rates for CFA and GA on PID and HFD datasets using different iterations and populations

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Population**  **Iteration** | **10** | **20** | **30** | **40** | **50** | **60** | **70** | **80** | **90** | **100** |
| **PID** | **10** | 0.77 | 0.78 | 0.77 | 0.77 | 0.79 | 0.78 | 0.77 | 0.80 | 0.79 | **0.79** |
| **20** | 0.75 | 0.75 | 0.75 | 0.76 | 0.76 | 0.77 | 0.76 | 0.79 | 0.79 | **0.80** |
| **30** | 0.75 | 0.76 | 0.77 | 0.78 | 0.79 | 0.79 | 0.80 | 0.79 | 0.80 | **0.80** |
| **40** | 0.74 | 0.75 | 0.75 | 0.76 | 0.77 | 0.77 | 0.79 | 0.79 | 0.80 | **0.79** |
| **50** | **0.74** | **0.75** | **0.77** | **0.77** | **0.76** | **0.78** | **0.79** | **0.80** | **0.80** | **0.80** |
| **60** | 0.75 | 0.76 | 0.77 | 0.77 | 0.77 | 0.78 | 0.79 | 0.79 | 0.80 | **0.80** |
| **70** | 0.76 | 0.77 | 0.77 | 0.78 | 0.77 | 0.78 | 0.78 | 0.80 | 0.80 | **0.80** |
| **HFD** | **10** | 0.76 | 0.77 | 0.76 | 0.75 | 0.76 | 0.76 | 0.76 | 0.74 | 0.75 | **0.73** |
| **20** | 0.76 | 0.74 | 0.75 | 0.74 | 0.73 | 0.75 | 0.77 | 0.74 | 0.76 | **0.74** |
| **30** | 0.75 | 0.75 | 0.76 | 0.76 | 0.75 | 0.76 | 0.77 | 0.75 | 0.75 | **0.75** |
| **40** | 0.74 | 0.76 | 0.76 | 0.76 | 0.77 | 0.77 | 0.78 | 0.73 | 0.74 | **0.75** |
| **50** | **0.75** | **0.76** | **0.76** | **0.77** | **0.77** | **0.77** | **0.78** | **0.75** | **0.75** | **0.76** |
| **60** | 0.74 | 0.77 | 0.77 | 0.77 | 0.76 | 0.77 | 0.77 | 0.74 | 0.75 | **0.76** |
| **70** | 0.74 | 0.75 | 0.76 | 0.76 | 0.76 | 0.77 | 0.77 | 0.74 | 0.76 | **0.77** |

Tables 9 and 10 present the evaluation results for the CFA and GA on PID and HFD datasets, respectively, in terms of accuracy, Kappa, and MAE. In terms of accuracy, it is clear from the results that CFA has better results compared to GA on the HFD dataset, and the highest accuracy values were achieved with the RF classifier (maximum accuracy = 0.97) since…. However, this was not the case in the PID dataset. The accuracy results for the CFA and GA on PID varied through different classifiers. The results showed that the CFA provided better accuracy results with LR, SVM, and NB classifiers. This shows that the CFA works well with larger datasets, in particular with the RF and DT classifiers. This is because DT is a series of sequential decisions made to reach a specific result regarding the importance of features and the sequence of attributes to be checked is decided based on criteria like Gini Impurity Index or Information Gain. Whereas RF leverages the power of multiple decision trees for making decisions (i.e., a forest of trees).

As a more conservative measure compared to accuracy, Kappa results have been measured and observed for the CFA and GA. The results showed that the CFA in general provided better performance compared to the GA, especially with the HFD dataset, because... All classifiers provided better results, except RF with only (0.01) less than GA since... Whereas the performance of classification algorithms using Kappa varied in the PID with all values less than 0.50. This shows that the Kappa coefficient aligns well with accuracy results, especially with larger datasets. Furthermore, classification algorithms should possess reduced amounts of MAE rates to prove it has better performance. The MAE results showed that the proposed CFA outperforms the GA in all cases on the PID and HFD datasets since …. This suggests applying MAE as a performance metric for evaluating classification algorithms on datasets of various sizes.

**Table 9** Comparison between CFA and GA using PID by different classification algorithms in terms of accuracy, Kappa, and MAE.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classifier | Algorithm | Accuracy ± STD | Accuracy Maximin | Accuracy Minimum | Kappa | MAE |
| LR | CFA | 0.80 ± 0.03 | 0.82 | 0.70 | 0.49 | 0.2 |
| GA | 0.78 ± 0.04 | 0.80 | 0.70 | 0.4 | 0.24 |
| RF | CFA | 0.77 ± 0.04 | 0.77 | 0.73 | 0.3 | 0.23 |
| GA | 0.78 ± 0.03 | 0.79 | 0.72 | 0.39 | 0.25 |
| K-NN | CFA | 0.72± 0.02 | 0.73 | 0.69 | 0.30 | 0.29 |
| GA | 0.74± 0.02 | 0.75 | 0.71 | 0.38 | 0.25 |
| SVM | CFA | 0.80 ± 0.03 | 0.81 | 0.70 | 0.48 | 0.21 |
| GA | 0.76± 0.03 | 0.77 | 0.73 | 0.4 | 0.25 |
| NB | CFA | 0.76± 0.02 | 0.77 | 0.69 | 0.4 | 0.24 |
| GA | 0.75 ± 0.03 | 0.76 | 0.73 | 0.34 | 0.26 |
| DT | CFA | 0.69± 0.02 | 0.70 | 0.64 | 0.35 | 0.29 |
| GA | 0.72 ± 0.03 | 0.75 | 0.67 | 0.28 | 0.31 |

**Table 10** Comparison between CFA and GA using HFD by different classification algorithms in terms of accuracy, Kappa, and MAE.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Classifier | Algorithm | Accuracy ± STD | Accuracy Maximin | Accuracy Minimum | Kappa | MAE |
| Logistic | CFA | 0.79 ± 0.02 | 0.78 | 0.69 | 0.46 | 0.22 |
| GA | 0.73 ± 0.02 | 0.73 | 0.69 | 0.37 | 0.26 |
| Random Forest | CFA | 0.97 ± 0.01 | 0.97 | 0.90 | 0.91 | 0.03 |
| GA | 0.96 ± 0.03 | 0.97 | 0.89 | 0.92 | 0.03 |
| KNN | CFA | 0.77± 0.04 | 0.82 | 0.72 | 0.53 | 0.19 |
| GA | 0.76 ± 0.03 | 0.78 | 0.74 | 0.52 | 0.21 |
| SVM | CFA | 0.75± 0.02 | 0.78 | 0.69 | 0.45 | 0.22 |
| GA | 0.73 ± 0.03 | 0.74 | 0.70 | 0.4 | 0.26 |
| NB | CFA | 0.75 ± 0.02 | 0.77 | 0.69 | 0.46 | 0.22 |
| GA | 0.72 ± 0.04 | 0.73 | 0.69 | 0.36 | 0.28 |
| DT | CFA | 0.95± 0.04 | 0.97 | 0.76 | 0.89 | 0.04 |
| GA | 0.93± 0.01 | 0.96 | 0.94 | 0.86 | 0.06 |

Figures 4 and 5 show Kappa results of six classifiers for the PID and HFD datasets respectively, at different training set levels (0.5 to 0.9). The results in Fig. 4 show the features selected by the CFA provided better results than the GA when LR, SVM, and NB classifiers were used since these classifiers…. Whereas the results in Fig. 5 show that CFA provided better results than GA on the HFD, especially with RF and DT since these classifiers... This suggests using RF and DT classifiers with larger datasets. This shows that the performance of classifiers varies through datasets with different sizes. The RF and DT outperformed other classifiers when applied to the larger dataset (HFD), whereas LR, SVM, and NB provided better results for CFA when applied to the PID dataset.

|  |
| --- |
| **Fig. 4** Kappa values on the PID dataset at different levels of the training set  **Fig. 5** Kappa values on the HFD dataset at different levels of the training set  Figures 6 and 7 show the MAE values of six classifiers for the PID and HFD datasets, respectively, at different training set levels (0.5 to 0.9) with 50 iterations and 100 as population size. The results in Fig. 6 show that each time the number of training samples in the HFD dataset is increased, the features selected by the CFA provided better results than the GA using all classifiers, except KNN. Similar to accuracy and Kappa results, the results in Fig. 7 show that CFA provided better results than GA, especially with RF and DT. The main reasons behind this is that …  **Fig. 6** MAE values on the PID datasets at different levels of the training set  **Fig. 7** MAE values on the PID datasets at different levels of the training set |

**Fig. 8.** Execution time for the CFA and GA on PID and HFD datasets

By examining the execution time of the CFA and GA (see Fig. 8), it was noticed that the CFA outperforms the GA in terms of execution time. Since the algorithmic design for the GA is different from the CFA and the solutions are ranked based on the fitness values. The GA usually will cluster around good solutions in the population. This is based on the observation where the selection of parents in GA is based on probabilities that favor fitness individuals. Solutions are more likely to be similar to the parents as crossover operation produces offspring with parent’s parts. The diversification aspect of GA is accomplished through the mutation operation that injects some difference in the solutions from time to time. The solution time of GA also increases non-linearly as the population size increases, whereas the CFA aims to find the optimal solution based on the color-changing behavior. The patterns and colors seen in cuttlefish are produced by reflected light from three layers (described in Section 3.1). The simulation of light reflecting, and visibility of the matching patterns used are formulated.

# Conclusion and future work

Medical data analysis is a critical research field where every decision matters. However, medical datasets are often massive in dimensions with complex redundant features which increases the possibility of noise and dependency among the features. Therefore, identifying a proper feature selection approach is important in data pre-processing stages to reduce the redundancy and irrelevance among features, which positively induces the speed of performance and prediction accuracy. In this paper, a bio-inspired algorithm called the cuttlefish has been adapted for feature selection classification. This algorithm is inspired based on the color-changing behavior of cuttlefish to find the optimal solution. Earlier research has proven the effectiveness of the cuttlefish algorithm compared to other bio-inspired algorithm such as the genetic algorithm for solving various optimization problems. We applied the cuttlefish and the genetic algorithm to two datasets: the Pima Indian diabetes dataset and the hospital Frankfort dataset, and the results were observed. The results show that the cuttlefish algorithm works well in predicting type 2 diabetes and it enjoys better performance and execution time compared to the genetic algorithm. The classification results showed that RF and DT classifiers outperformed other classifiers when a larger dataset was used. Furthermore, the results suggested using LR, SVM, and NB classifiers with small-scale datasets.

Overall, as future work, the CFA algorithm can be used in the medical domain for predicting how to choose diabetes treatment, predicting cancer diseases, and classifying diabetic retinopathy caused by high blood sugar levels damaging the back of the eye (retina). Also, predicting diabetes can be done using various heuristic and metaheuristic algorithms, such as A\* heuristic search algorithm [27], iterative deepening A\* (IDA\*) algorithm [28], 2-opt local search algorithm [29], nearest neighbor search algorithm [30], harmony search algorithm [31], chemical reaction optimization [32], grey wolf optimizer [33], and most valuable player algorithm [34], for different large high-dimensional datasets.

**Funding**

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Declarations**

**Conflict of interest** The authors declare no known potential conflicts of interest concerning financial interests or the research, authorship, and publication of this article

**References**

[1] D. Yach, C. Hawkes, C.L. Gould, and K.J. Hofman, The Global Burden of Chronic DiseasesOvercoming Impediments to Prevention and Control, *JAMA*. **291** (2004) 2616–2622. doi:10.1001/jama.291.21.2616.

[2] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri, Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset, in: 2017 Int. Conf. Comput. Netw. Informatics, 2017: pp. 1–5. doi:10.1109/ICCNI.2017.8123815.

[3] J.J. Khanam, and S.Y. Foo, A comparison of machine learning algorithms for diabetes prediction, *ICT Express*. (2021). doi:https://doi.org/10.1016/j.icte.2021.02.004.

[4] S. Khalid, T. Khalil, and S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in: 2014 Sci. Inf. Conf., 2014: pp. 372–378. doi:10.1109/SAI.2014.6918213.

[5] S. G., V. R., and S. K.P., Diabetes detection using deep learning algorithms, *ICT Express*. **4** (2018) 243–246. doi:https://doi.org/10.1016/j.icte.2018.10.005.

[6] L. Yu, and H. Liu, Efficient feature selection via analysis of relevance and redundancy, *J. Mach. Learn. Res.* **5** (2004) 1205–1224.

[7] L. Ismail, H. Materwala, M. Tayefi, P. Ngo, and A.P. Karduck, Type 2 Diabetes with Artificial Intelligence Machine Learning: Methods and Evaluation, *Arch. Comput. Methods Eng.* (2021) 1–21.

[8] S.C. Yusta, Different metaheuristic strategies to solve the feature selection problem, *Pattern Recognit. Lett.* **30** (2009) 525–534.

[9] A.H. Gandomi, X.-S. Yang, and A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.* **29** (2013) 17–35.

[10] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, 2010.

[11] A. Negi, and V. Jaiswal, A first attempt to develop a diabetes prediction method based on different global datasets, in: 2016 Fourth Int. Conf. Parallel, Distrib. Grid Comput., 2016: pp. 237–241. doi:10.1109/PDGC.2016.7913152.

[12] N.P. Tigga, and S. Garg, Prediction of Type 2 Diabetes using Machine Learning Classification Methods, *Procedia Comput. Sci.* **167** (2020) 706–716. doi:https://doi.org/10.1016/j.procs.2020.03.336.

[13] D. Gupta, A. Julka, S. Jain, T. Aggarwal, A. Khanna, N. Arunkumar, and V.H.C. de Albuquerque, Optimized cuttlefish algorithm for diagnosis of Parkinson’s disease, *Cogn. Syst. Res.* **52** (2018) 36–48. doi:https://doi.org/10.1016/j.cogsys.2018.06.006.

[14] R. Abu Khurmaa, I. Aljarah, and A. Sharieh, An intelligent feature selection approach based on moth flame optimization for medical diagnosis, *Neural Comput. Appl.* (2020). doi:10.1007/s00521-020-05483-5.

[15] S. Malakar, M. Ghosh, S. Bhowmik, R. Sarkar, and M. Nasipuri, A GA based hierarchical feature selection approach for handwritten word recognition, *Neural Comput. Appl.* **32** (2020) 2533–2552. doi:10.1007/s00521-018-3937-8.

[16] Uzma, F. Al-Obeidat, A. Tubaishat, B. Shah, and Z. Halim, Gene encoder: a feature selection technique through unsupervised deep learning-based clustering for large gene expression data, *Neural Comput. Appl.* (2020). doi:10.1007/s00521-020-05101-4.

[17] S.H. Shah, M.J. Iqbal, I. Ahmad, S. Khan, and J.J.P.C. Rodrigues, Optimized gene selection and classification of cancer from microarray gene expression data using deep learning, *Neural Comput. Appl.* (2020). doi:10.1007/s00521-020-05367-8.

[18] R.B. Lukmanto, Suharjito, A. Nugroho, and H. Akbar, Early Detection of Diabetes Mellitus using Feature Selection and Fuzzy Support Vector Machine, *Procedia Comput. Sci.* **157** (2019) 46–54. doi:https://doi.org/10.1016/j.procs.2019.08.140.

[19] A.H. Gandomi, X.-S. Yang, S. Talatahari, and A.H. Alavi, Metaheuristic algorithms in modeling and optimization, *Metaheuristic Appl. Struct. Infrastructures*. (2013) 1–24.

[20] A. Almomani, M. Alweshah, and S. Al, Metaheuristic algorithms-based feature selection approach for intrusion detection, *Mach. Learn. Comput. Cyber Secur. Princ. Algorithms, Pract.* (2019).

[21] A.S. Eesa, A.M.A. Brifcani, and Z. Orman, Cuttlefish algorithm-a novel bio-inspired optimization algorithm, *Int. J. Sci. \& Eng. Res.* **4** (2013) 1978–1986.

[22] A.S. Eesa, A.M.A. Brifcani, and Z. Orman, A new tool for global optimization problems-cuttlefish algorithm, *Int. J. Math. Comput. Nat. Phys. Eng.* **8** (2014) 1208–1211.

[23] J.H. Holland, and others, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, 1992.

[24] A. Ben-David, Comparison of classification accuracy using Cohen's Weighted Kappa, *Expert Syst. Appl.* **34** (2008) 825–832.

[25] S.M. Vieira, U. Kaymak, and J.M.C. Sousa, Cohen’s Kappa coefficient as a performance measure for feature selection, in: Int. Conf. Fuzzy Syst., 2010: pp. 1–8.

[26] J.R. Landis, and G.G. Koch, The measurement of observer agreement for categorical data, *Biometrics*. (1977) 159–174.

[27] B.A. Mahafzah, Performance evaluation of parallel multithreaded A\* heuristic search algorithm, *J. Inf. Sci.* **40** (2014) 363–375.

[28] B.A. Mahafzah, Parallel multithreaded IDA\* heuristic search: algorithm design and performance evaluation, *Int. J. Parallel, Emergent Distrib. Syst.* **26** (2011) 61–82.

[29] A. Al-Adwan, A. Sharieh, and B.A. Mahafzah, Parallel heuristic local search algorithm on OTIS hyper hexa-cell and OTIS mesh of trees optoelectronic architectures, *Appl. Intell.* **49** (2019) 661–688.

[30] A. Al-Adwan, B.A. Mahafzah, and A. Sharieh, Solving traveling salesman problem using parallel repetitive nearest neighbor algorithm on OTIS-Hypercube and OTIS-Mesh optoelectronic architectures, *J. Supercomput.* **74** (2018) 1–36.

[31] B.A. Mahafzah, M. Alshraideh, and others, Hybrid harmony search algorithm for social network contact tracing of COVID-19, *Soft Comput.* (2021) 1–23.

[32] B.A. Mahafzah, R. Jabri, and O. Murad, Multithreaded scheduling for program segments based on chemical reaction optimizer, *Soft Comput.* **25** (2021) 2741–2766.

[33] A. Al-Shaikh, B.A. Mahafzah, and M. Alshraideh, Metaheuristic approach using grey wolf optimizer for finding strongly connected components in digraphs, *J Theor Appl Inf Technol*. **97** (2019) 4439–4452.

[34] H. Khattab, A. Sharieh, and B.A. Mahafzah, Most valuable player algorithm for solving minimum vertex cover problem, *Int J Adv Comput Sci Appl*. **10** (2019) 159–167.

1. https://www.who.int/news-room/fact-sheets/detail/diabetes [↑](#footnote-ref-1)
2. https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html [↑](#footnote-ref-2)
3. https://www.kaggle.com/uciml/pima-indians-diabetes-database [↑](#footnote-ref-3)
4. https://www.kaggle.com/uciml/pima-indians-diabetes-database [↑](#footnote-ref-4)
5. https://www.kaggle.com/johndasilva/diabetes/version/1 [↑](#footnote-ref-5)