Journal of King Saud University – Computer and Information Sciences 35 (2023) 101757

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Role of transfer functions in PSO to select diagnostic attributes for chronic disease prediction: An experimental study

Samir Malakar [a,b], Swaraj Sen [c], Sergei Romanov [d], Dmitrii Kaplun [e,d,*], Ram Sarkar [c]

[a] Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway
[b] Department of Computer Science, Asutosh College, Kolkata, India
[c] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
[d] Department of Automation and Control Processes, Saint Petersburg Electrotechnical University "LETI", Saint Petersburg, Russia
[e] Artificial Intelligence Research Institute, China University of Mining and Technology, Xuzhou, China

ARTICLE INFO

Article history:
Received 21 May 2023
Revised 20 August 2023
Accepted 11 September 2023
Available online 22 September 2023

Keywords:
PSO
Transfer function
Feature selection
Disease prediction
Diagnosis report
Study

ABSTRACT

Particle Swarm Optimization (PSO) is a classic and popularly used meta-heuristic algorithm in many real-life optimization problems due to its less computational complexity and simplicity. The binary version of PSO, known as BPSO, is used to solve binary optimization problems, such as feature selection. Like other meta-heuristic optimization techniques designed on the continuous search space, PSO uses the transfer functions (TFs) to map the candidate solutions to the discrete search space in BPSO, and these TFs play a vital role to get the desired results. Over the years, many forms of TFs have been introduced in the literature, most of which fall under one of the five families - Linear, S-shaped, V-shaped, U-shaped, and Time-varying Mirrored S-shaped TFs. The goal of this study is to determine an appropriate setup constituting a TF and a classifier for feature selection from different types of clinical data. In this study, the impacts of the five TF families have been investigated, considering one from each family for the selection of attributes/features, while predicting disease using diagnosis or medical reports. The classification tasks are carried out using four standard classifiers: Support Vector Machine, Decision Tree, K-Nearest Neighbors, and Gaussian Naive Bayes. For experimental purposes, we have used four publicly available datasets namely, the UCI Heart Disease dataset, Wisconsin Breast Cancer dataset, UCI Chronic Kidney Disease dataset, and PIMA Indians Diabetes dataset. After an exhaustive set of experiments, we have obtained 96.72%, 99.82%, 100.00%, and 84.41% disease prediction scores in the best case for Heart disease, Breast Cancer, Chronic Kidney disease, and Diabetes, respectively. The obtained results are comparable to several state-of-the-art methods considered here for comparison. The present study helps in selecting a suitable BPSO setup (i.e., a TF and a classifier) to select important diagnostic attributes useful to design a computer-aided decision support system for the said diseases.

1. Introduction

A disease can be considered a particular abnormal condition that affects the functionality or structure of all or part of an organism. Chronic disease is a type of disease that lasts for three months or more. Generally, chronic diseases cannot be prevented by vaccines or cured by medication. Chronic diseases such as Cancer, Stroke, Heart disease, and Diabetes are the leading causes of disability and death in human beings. However, early detection of some chronic diseases becomes a pressing need to save human lives. But early detection of such diseases is not always easy for specialist physicians since they rely on reviewing different symptoms of the subjects, which sometimes takes time. Even so, some human error is still present in the diagnosis process. Today, the advent of technologies helps to store diagnostic reports along with the status of the subjects in digital form, leading to a new generation of computer-assisted decision support systems. In this study, we have considered the prediction of four chronic diseases, namely Heart disease, Chronic Kidney Disease (CKD), Breast Cancer, and Diabetes, using such diagnostic reports.

According to a report published by the World Health Organization (WHO) in 2020, the number of deaths caused by Heart disease

* Corresponding author.
E-mail addresses: s.malakar@uit.no (S. Malakar), saromanov@etu.ru (S. Romanov), dikaplun@etu.ru (D. Kaplun).

https://doi.org/10.1016/j.jksuci.2023.101757
1319-1578/

each year is approximately 17.9 million worldwide, making this disease the deadliest[1]. Heart disease is the most common in low- and middle-income countries. Even developed nations are also experiencing a large number of deaths (more than 650 thousand deaths in America in the year 2019 (Virani et al., 2021)) each year due to this disease. Breast Cancer, a kind of cancer that attacks women primarily, has become a real concern in world-health as the number of identified Breast Cancer diseases (more than 2.3 million cases) in the female body has exceeded the number of lung cancer cases in 2020 (Sung et al., 2021). According to the researchers, approximately 700,000 people died from Breast Cancer in 2020. Diabetes may not be as deadly as other chronic diseases such as cancer; however, it causes many health-related problems in the human body. According to the WHO, between 2000 and 2016 premature deaths increased by 5% due to Diabetes, and in 2016, 1.6 million people died from other diseases, but most of them were due to diabetes. According to Colagiuri et al. (2019) from 9.30% in 2019, the estimated number of diabetic patients will increase by approximately 10.2% in 2030, and 10.90% in 2045 around the world. Diabetes with hypertension is the primary reason for CKD. The kidneys of affected patients begin to malfunction and thus do not filter the waste from the blood, eventually leading to renal failure. Around 100,000 patients are diagnosed with end-stage kidney disease every year in India. All these facts motivate us to choose the said chronic diseases in the present case study, which investigate the effectiveness of transfer functions to transform meta-heuristic aided optimization techniques from continuous domain to discrete domain.

In recent times, machine learning (ML) strategies have taken over traditional rule-based decision support systems when screening diseases using diagnostic reports. As a result, several powerful ML-based methods can be found in the literature to diagnose chronic diseases. For example, Amirgaliyev et al. (2018) used the Support Vector Machine (SVM) with a linear kernel to predict CKD from diagnostic attributes. Sisodia and Sisodia (2018) studied the effect of different ML classifiers such as Naive Bayes (NB), SVM, and Decision Tree (DT) on predictive attributes based on the diabetes diagnosis report. Huang et al. (2017) applied the SVM with varying kernels in the ensemble techniques used (bagging and boosting) for the prediction of Breast Cancer. In another work, Mohan et al. (2019) used a hybrid random forest with a linear model (HRFLM) to predict heart disease. Recently, Malakar et al. (2022) discussed more such ML-aided computer-assisted diagnostic approaches for such diseases that have been introduced in the last two decades. However, these researchers mostly used their methods on the entire set of diagnostic attributes without understanding these attributes' importance in the classification task.

Generally, when there is an informative set of features (here diagnostic attributes), which represent the dataset better, ML-based methods perform better (Ghosh et al., 2019; Ghosh et al., 2021). However, in many real-life datasets, it may be found that some features are not very informative, but redundant. Such irrelevant and noisy features mislead the underlying classification algorithm while generating the class separating hyperplanes (Shaw et al., 2021). Therefore, classical ML classifiers may not be able to produce the desired results. Additionally, such features not only make the training process time-consuming but also lead the classifier to encounter problems like overfitting or underfitting and the "curse of dimensionality". Thus, these redundant features decrease the generalizability of a classification model (Chandrashekar and Sahin 2014; Liu and Yu 2005). These facts lead to a new set of methods called feature selection (also known as variable or attribute selection). These methods can also handle the over-/under-fitting problems of classifiers (Guyon and Elisseeff 2003; Liu and Yu 2005). In addition, they help improve classification performance, produce more cost-effective models, and improve understanding of the data under consideration. Feature selection algorithms choose a subset of important features from the original datasets using some criteria (Guyon and Elisseeff 2003; Mandal et al., 2021). All these boost the use of feature selection in the diagnostic report-based decision support system for chronic disease detection like other ML applications (Malakar et al., 2022).

The easiest way to obtain the most suitable feature subset is to consider each possible subset of the entire feature set and evaluate its performance to decide the best one. However, this solution is computationally expensive, since we need to evaluate $2^N - 1$ possible feature subsets for a feature set containing N features. This makes the feature selection problem an NP-hard problem (Ghosh et al., 2020; Begum et al., 2015). Therefore, researchers came up with an alternative solution paradigm that aims to extract near-optimal solutions instead of the best. These types of solutions follow one of the three categories of methods - filter (Ghosh et al., 2021; Ghosh et al., 2019), wrapper (Ghosh et al., 2017; Malakar et al., 2020b), or embedded (Guha et al., 2020). Of these approaches, the wrapper feature selection methods first select a set of feature subsets, known as candidate solutions, from the original feature set and then evaluate the merit of the selected feature subsets using an objective function (Ghosh et al., 2019; Chakraborty et al., 2021). Although this category of feature selection is computationally expensive compared to filter methods, in general, it outperforms filter methods (Ghosh et al., 2019; Chakraborty et al., 2021). Wrapper-based feature selection approaches are more beneficial than the other two approaches when selecting attributes or features (Shaw et al., 2021; Malakar et al., 2020a). It is also found in the literature that meta-heuristic-based feature selection methods perform better than traditional wrapper methods such as forward feature selection, backward feature elimination, and exhaustive feature selection (Banerjee et al., 2021; Dey et al., 2022; Mukhopadhyay et al., 2022). A meta-heuristic algorithm is a high-level framework that provides a set of strategies or recommendations to develop optimization algorithms in a problem-independent way.

As a result, several methods could be found in the literature that performed feature/variable selection using a meta-heuristic-based wrapper method. For example, Begum et al. (2021) utilized active learning with symmetric uncertainty-based feature selection to identify biomarkers for the identification of cancer. Mohiuddin et al. (2023) designed a hierarchical feature selection strategy for selecting effective features to isolate deepfake videos from real ones. Also, meta-heuristic-based algorithms like advanced Grey Wolf optimizer (Nadimi-Shahraki et al., 2021b) and multi-trial vector-based differential evolution algorithm (Nadimi-Shahraki et al., 2020) were designed to solve several engineering applications that require optimization. In another set of works, researchers (Dey et al., 2021; Mandal et al., 2022; Mandal et al., 2021) used a meta-heuristics-based approach for selecting diagnostic attributes while designing a decision support system for chronic disease. Dey et al. (2021) developed an improved grasshopper optimization algorithm with the help of learning automata, Mandal et al. (2022) designed an ensemble of filter methods, Mandal et al. (2021) introduced a three-stage feature selection method to handle such a problem. We could also find the use of the binary version of the Aquila optimization algorithm (Nadimi-Shahraki et al., 2022) and moth-flame optimization algorithm (Nadimi-Shahraki et al., 2021a) for selecting features from extracted features from the disease data. Dey et al. (2022) used meta-heuristic algorithms for deciding parameters of Sugeno fuzzy integral-based classifier ensemble technique for detecting Tuberculosis

[1] https://www.who.int/health-topics/cardiovascular-diseases.

from chest X-ray images. Many more such applications can be found in the surveys (Nssibi et al., 2023; Malakar et al., 2022) that described different applications of meta-heuristic algorithms used for disease prediction. On contrary, recently Kaur et al. (2023) discussed the open issues and different challenges associated with the use of meta-heuristics-based feature selection strategies for designing disease diagnostic systems.

Considering the usefulness of meta-heuristic-based feature selection algorithms, we have preferred to use such an algorithm for feature selection to be applied to medical report-based datasets related to chronic diseases. It is noteworthy to mention here that according to Wolpert and Macready (1997), Kumar and Minz (2014) there is no best method for feature selection, and as a result, a new feature selection technique is evolving now and then to solve some specific problems. Thus, in this study, we have restricted ourselves to the binary version of the Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995) (i.e., the Binary PSO (BPSO) (Kennedy and Eberhart 1997)) based wrapper feature selection method. The primary reason behind this choice over the other methods is its applicability to various complex optimization problems found in the literature (Jain et al., 2022). PSO is a nature-inspired meta-heuristic optimization algorithm that has been used in the past to solve many real-life problems. For example, Ghosh et al. (2020) designed a binary Genetic Swarm Optimization (BGSO) algorithm in which the Genetic algorithm was fused with PSO and used as a feature selector. Recently, Das et al. (2023) utilized BPSO to improve the performance of breast cancer prediction from histological images.

Several variants of PSO and BPSO exist in the literature and have performed well in various optimization problems or feature selection applications. These variants can be categorized into two border groups: one that transforms PSO from a continuous domain to a binary domain (i.e., from PSO to BPSO) and the other that tries to improve its functionality by changing the algorithmic procedure. The variants of the first group focused on designing different transfer functions (TFs) that transform the PSO from the continuous domain to the binary domain. Such a conversion is required when PSO variants are used for feature selection purposes, as feature selection is a binary optimization problem. It should be mentioned that these TFs do not only use to transform PSO to BPSO but can be used to transform any meta-heuristics algorithm to their corresponding binary version (Taghian et al., 2018). Some of the well-known TFs found in the literature are S-shaped TF (Kennedy and Eberhart 1997), V-shaped TF (Rashedi et al., 2010), variants of S- and V-shaped TFs (Mirjalili and Lewis 2013) and U-shaped TF (Mirjalili et al., 2020). The functional nature of a TF determines the transformation of a real value into a binary value ('0' or '1').

The second variant of PSO and BPSO focuses on improving the intrinsic properties of the PSO algorithm. For example, Wei et al. (2017) introduced a memory renewal strategy that helps particles overstep the local extremum and a mutation schema to overcome premature convergence of the particles. Sarkar et al. (2018) proposed the modified BPSO algorithm by incorporating modifications suggested by Wei et al. (2017) to perform digit recognition with optimally selected DAISY features proposed by Chatterjee et al. (2018). In another work, Jain et al. (2018) proposed an improved BPSO (iBPSO) that avoids early convergence of particles towards a local optimum and used this iBPSO for Breast Cancer detection. Recently, Li et al. (2021) proposed an improved sticky BPSO (ISBPSO) algorithm by introducing two new strategies for initialization and gradual reduction of search space while Nguyen et al. (2021) proposed a dynamic sticky BPSO (DSPSO), where the concepts of velocity and momentum were reformulated in a binary domain instead of a continuous domain.

In the literature, we find some research efforts made by Wei et al. (2017), Nguyen et al. (2021), and Li et al. (2021) that modified the intrinsic working principle of the PSO algorithm, while the methods by Daliri (2012), and Herliana et al. (2018) investigated many such PSO variants fall into the second category as mentioned above to perform disease prediction utilizing diagnosis reports. There are also a number of research attempts proposed by Islam et al. (2017), Mirjalili et al. (2020), Guo et al. (2020), and Beheshti (2020) that designed several TFs to improve PSO-based discrete optimization problems such as feature selection. A time-varying TF (Islam et al., 2017), U-shaped TF (Mirjalili et al., 2020), Z-shaped TF (Guo et al., 2020), and time-varying mirrored S-shaped TF (Beheshti, 2020) were introduced with the objective to have better performance when solving a binary optimization problem. Apart from these, Mirjalili and Lewis (2013) made a study to investigate the effectiveness of U-shaped and V-shaped TFs while used in PSO to convert into BPSO while Taghian et al. (2018) made a comparative study of these two TFs (i.e., U-shaped and V-shaped) while employed with different meta-heuristics algorithms like bat algorithm (BA), gravitational search algorithm (GSA) and grey wolf optimization algorithm (GWO) for feature selection purpose. However, there is no such work in the literature that has investigated the impact of different available TFs on the end outcome of PSO while selecting the near-optimal diagnostic attributes for the prediction of diseases from patient diagnosis reports on varying datasets. Furthermore, how the classifier used in the selection process impacts the end performance is not also analyzed. This has become our primary motivation for our current study. This study also aims to find a near-optimal set of diagnostic attributes that contribute more to the said disease predictions from the diagnostic reports. For performance investigation, we have considered four publicly available datasets of chronic diseases: Heart disease, CKD, and Breast Cancer datasets from the UCI data repository, and the PIMA Indians Diabetes dataset along with four popularly used classifiers like Gaussian Naive Bayes (GNB), K-Nearest Neighbors (K-Nearest Neighbors (KNN), SVM, and DT.

In summary, the highlights of this study are as follows:

- The feature selection capability of BPSO with five different TFs is investigated.
- For evaluation purposes, four most common chronic diseases datasets namely, Heart disease, Breast Cancer, CKD, and Diabetes are considered.
- Exhaustive experiments are performed to select the most appropriate diagnostic attributes/features.
- The results obtained are comparable with many state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 describes some state-of-the-art methods proposed for the four mentioned diseases. Sections 3 and 4 describe the working procedure of PSO and BPSO, respectively. In Section 5, we have described the different TFs used in this work. Section 6 describes how BPSO selects a subset of features that are near-optimal. Details of the four datasets and the experimental results are reported in Section 7. Section 8 and Section 9 discuss the performance of the present study on high-dimensional datasets and the advantages and limitations of the present study. Finally, we conclude the paper in Section 10 by highlighting some future research scopes.

2. Related study

In this work, we have investigated the performance of different decision support systems employing different BPSO-based feature selection techniques on four publicly available chronic disease datasets: UCI Heart disease dataset, Wisconsin Breast Cancer dataset, UCI CKD dataset, and PIMA Indians Diabetes dataset (see the

Section 7.1 for further information about these datasets). Different BPSO-based feature selection algorithms are built by employing five different TFs (please refer to Section 5 for more details) that convert the continuous PSO to its binary version. We have discussed feature selection-aided decision support systems and how they performed on these datasets. These datasets are widely used for evaluating the performance of various chronic disease detection methods, as well as the performance of newly designed feature selection techniques.

A two-stage kernel F-score feature selection (KFFS) method was designed by Polat and Güneş (2009). In the first stage, features were transformed into kernel space using either the Linear kernel function or the Basis Function (RBF) kernel. In the second phase, the F-score values of all the features present in the transformed domain were first calculated, and then features with a higher F-score value (higher than the mean of all the F-score values) were selected. With the help of the Least Square Support Vector Machine (LS-SVM) as the classifier, the authors obtained a 3.70% improvement in performance when using KFFS for feature selection. They also showed that feature selection using the actual F-score measure failed to achieve the accuracy obtained by using all features when the top six features were selected. In another work, Peter and Somasundaram (2012) conducted a series of experiments to establish the usefulness of feature selection. The authors first used a Correlation-based Feature Selection (CBFS) algorithm to select a feature subset in this work, and then these selected features were fed into the NB classifier for further feature reduction. NB selects attributes using a conditional probability score. The authors were able to reduce the feature dimension to 3 while improving the classification accuracy of KNN from 75.18% to 85.55%. In another work, Khemphila and Boonjing (2011) used information gain (IG) as the feature selection method and a neural network (NN) as the classifier. The authors selected the 8 top-ranked features and obtained an accuracy of 80.90%. Recently, Gokulnath and Shantharajah (2019) used the Genetic Algorithm (GA) as a feature selection method for the said purpose. They selected 7 features using the 'elitism' criteria of GA, and in the best case, they obtained 88.30% test accuracy using the SVM classifier.

Lavanya and Rani (2011) used a wrapper feature selection technique called the forward feature selection method. In this work, they obtained 94.84% accuracy with the help of a Classification And Regression Tree (CART) classifier and selected 9 features. In another work, Dhanya et al. (2019) conducted a comparative study using different feature selection schemes and classifiers. In the best case, they were able to select 8 features and obtained an accuracy of 96.42% using CBFS-based feature selection and a Random Forest (RF) classifier. Aličković and Subasi (2017) used GA as the feature selector and observed a certain amount of increased classification accuracy for classifiers such as SVM, DT, Linear Regression (LR), RF, Multilayer Perceptron (MLP), and Rotation Forest. RF produced the best accuracy (99.48%) among all the experiments conducted by the research team. Kumar and Singh (2021) developed an enhanced GWO based feature selection technique by amalgamating a fitness-sharing strategy with the actual GWO algorithm (Mirjalili et al., 2014). The researchers were able to select only 6 features, and with the help of the SVM classifier, they obtained 98.24% classification accuracy. In another work, Guha et al. (2020) proposed the Embedded Chaotic Whale Survival Algorithm (ECWSA) for feature selection purposes. In their work, they used a filter method to refine the selected features and also introduced the concept of chaos to better select the feature subset. By doing so, they obtained 95.00% classification accuracy. Recently, Yaghoubzadeh et al. (2021) used a nature-inspired binary version of the Bat Algorithm to select the near-optimal feature set. With the help of SVM, the researchers achieved 99.28% classification precision with only six selected features.

Chetty et al. (2015) used the best-fit feature selection technique with three classifiers: KNN, NB, and Sequential Minimal Optimization (SMO). This method selected 7, 6, and 12 diagnostic attributes using KNN, NB, and SMO, respectively. Their method produced the best result when the KNN classifier was used on top of the selected feature subset, which had 12 features. KNN provided 100% accuracy on selected features but only 95.75% accuracy when using the entire set of features. Furthermore, the authors reported a reasonable increase in accuracy when classifying using NB and SMO. The accuracy was increased to 99.00% from 95.00% and 98.25% from 97.75% for NB and SMO respectively. In another work, Polat et al. (2017) utilized feature subset selection with the best-fit feature selection approach from all combinations of features, and only 13 features were selected with the help of SVM as a classifier. This method improved the classification accuracy from 97.75% to 98.50%. Wibawa et al. (2017) used CBFS and selected 17 features. The researchers obtained 98.10% accuracy with the help of the AdaBoost classifier. In another work, Almansour et al. (2019) used CBFS and selected 12 features. In this work, a comparative study was carried out using ANN and SVM as CBFS classifiers, and it was observed that ANN (99.75%) produced better results than SVM (97.75%). Shrivas et al. (2018) introduced a new feature selection technique called the union-based feature selection technique (UBFST) where, the researchers took the union of selected features by employing four filter-based feature selection techniques: IG, gain ratio, Chi-squared, and symmetric uncertainty. They obtained 99.25% accuracy for the prediction of CKD. Recently, Senan et al. (2021) used the Recursive Feature Elimination (RFE) method, a wrapper-based feature selection. The authors achieved 100% disease prediction accuracy in this work using the RF classifier. However, the accuracy obtained using SVM (96.67%), KNN (98.33%), and DT (99.17%) classifiers are lower than that obtained using the RF classifier.

Balakrishnan and Narayanaswamy (2009) used a fast CBFS method and selected only 4 features from the Indian PIMA dataset. Using SVM with RBF kernel, the system could produce an accuracy of 77.99%. In another work, Choubey et al. (2017) used GA-based feature selection to obtain a near-optimal feature subset. Here also, the researchers identified four features as the near-optimal feature subset, and with the NB classifier, the authors obtained 78.69% test accuracy. Kewat et al. (2020) employed BPSO and GA-based feature selection techniques with KNN as the classifier. In their experiments, they first extracted the selected features from the dataset and then employed different classifiers to perform the classification task. In the best case, the researchers obtained 76.79% accuracy with the SVM classifier. Recently, de Lima et al. (2020) proposed an F-score-based novel feature selection technique where features are ranked by taking the mean of the ranks obtained from positive and negative samples. In their work, they used the twin-bounded SVM classifier and obtained 77.21% accuracy. In a recent work, Rathi and Acharjya (2021) performed a comparative study to test the performance of GA-based feature selection on different datasets. In their work, they obtained 76.30% accuracy with the help of 6 selected features in the best case.

2.1. Critical analysis of the discussed methods and motivation

From the above discussions, it is clear that researchers have been using different feature selection techniques over the year on the datasets used in this work for experimentation. Therefore, it can be said that feature selection plays an important role in improving the disease prediction capability of the classifiers used. However, in most of the works, filter-based feature selection schemes like KFFS in Polat and Güneş (2009), CBFS in Peter and Somasundaram (2012), Dhanya et al. (2019), Balakrishnan and

Narayanaswamy (2009), Wibawa et al. (2017), IG in Khemphila and Boonjing (2011), best fit Chetty et al. (2015, 2017), and UBFST Shrivas et al. (2018) were used in by the researchers. Filter-based methods are generally better than wrapper-based feature selection methods in terms of computational cost, execution time, and generalization to high-dimensional features. They do not use any classifiers to decide which features to include in the final feature set. However, the non-inclusion of the classifier in the selection process might lead to poor selection of features, and obtaining good performance is always a big concern. Moreover, they normally rank features, and researchers need to manually select an optimal number of top-ranked features that would provide satisfactory results. For example, Chetty et al. (2015) selected the top 6, 7, and 12 features to decide the optimal number of features to be used as the final set of selected features. The inclusion of such manual intervention to decide the final feature set could be a reason for dipping its usage in recent times. Furthermore, we can also observe the use of wrapper-based feature selection methods (e.g., GA in Gokulnath and Shantharajah (2019), Choubey et al. (2017), Aličković and Subasi (2017), forward feature selection in Lavanya and Rani (2011), ECWSA in Guha et al. (2020), enhanced GWO in Kumar and Singh (2021) and the performance of these methods is better than its counterparts, i.e., filter-based method. If we analyze the performances of wrapper-based feature selection methods, then the performances of meta-heuristics based methods like GA, WOA hybridized with ANOVA, ECWSA, and enhanced GWO are comparatively better than the performances of simple wrapper-based feature selection methods like forward feature selection, SFWS, SBS, and REE. These facts inspired us to select a meta-heuristic-based feature selection technique, which is PSO here. Apart from these, the existing feature selection techniques (for example, Gokulnath and Shantharajah (2019); Choubey et al. (2017); Aličković and Subasi (2017); Kumar and Singh (2021); Rathi and Acharjya (2021)), employed on the datasets used in the present work have used S-shaped TF. But theoretically more advanced TFs like V-shaped, U-shaped, and Time-Varying Mirrored S-shaped (TVMS) are present in the literature. Therefore, in this work, we have also made several experiments to study the effect of these TF alternatives on the said datasets.

3. Particle swarm optimization

PSO is a popular meta-heuristic algorithm used to solve various continuous optimization problems. It was introduced by Kennedy and Eberhart (1995) in 1995. This algorithm is inspired by the works of Reynolds (1987) and Heppner (1990). Reynolds (1987) provided an idea of the choreography of the birds' flock, and Heppner (1990) showed interest in scattering, direction change, and regrouping of the flock of birds. They provided an idea about how the inter-individual distance is maintained in a flock. They also established a mathematical formulation to maintain the optimum distance and the change in the distance of birds in a flock. PSO follows the social behavior of the swarm of birds or the flock of fish, where they try to achieve optimal results. They follow certain rules to reach the food source following an optimal path. They can achieve the feat even if the location of the food source (result) is not initially known. Following the group search approach, they eventually reach the location where food can be found. The fishes or birds synchronously move towards the near-optimal solution, and after some time, all swarms converge to that solution.

According to Reynolds (1987), there are three main rules: separation, alignment, and cohesion for a bird swarm, where a swarm moves toward the solution in the search space. In separation, particles move apart from each other to avoid crowding. During alignment, the particles head toward their neighbors, and the cohesion indicates a position update according to the neighbors.

Inspired by the social behavior of a swarm of particles, Kennedy and Eberhart (1995) introduced the PSO algorithm to effectively solve different optimization problems. Generally, PSO converges very fast and has very few parameters to adjust, and as a result, the computational time of this algorithm is also less. Many particles try to find a solution, and hence the chance of getting stuck to an optimal local solution is less. Apart from these, it is derivative-free and has a very efficient global search mechanism. In PSO, each particle in the swarm or flock traverses in a multidimensional search space to find the near-optimal solution. To start the search process, the particles (i.e. candidate solutions) are randomly generated in the search space, and then the velocity and fitness values (in general, the weighted mean of classification accuracy and the number of features present in the feature subset) are computed for each particle. This is done to update the velocity and direction of their paths after the first iteration, and these processes are continued until it reaches the stopping criterion.

Let, x_{ik}^t, x_{ik}^{t+1}, v_{ik}^t and v_{ik}^{t+1} represent the positions and velocities of the i^{th} particle after t^{th} and $(t+1)^{th}$ iterations respectively in the D-dimensional search space where, $k \in \{1, 2, \ldots, D\}$ represents the component in the k^{th} direction of the D-dimensional space. The i^{th} particle (i.e., any particle in the swarm) considers its current position (i.e., x_{ik}^t) and the next velocity (i.e., v_{ik}^{t+1}) when calculating its position at $(t+1)^{th}$ iteration (i.e., x_{ik}^{t+1}). The velocity at $(t+1)^{th}$ iteration in k^{th}-direction for i^{th} particle (i.e., v_{ik}^{t+1}) of each particle is calculated using current velocity (i.e., v_{ik}^t), the current position (i.e., x_{ik}^t), the personal best position (say, pb_{ik}^t) and the global best position (say, gb_k^t) achieved till current iteration. Shi and Eberhart (1998) defined v_{ik}^{t+1} and x_{ik}^{t+1} using Eq. (1) and Eq. (2), respectively. It should be mentioned that Shi and Eberhart (1998) improved the velocity equation of the original PSO as proposed by Kennedy and Eberhart (1995) by introducing an inertia component which is 'w' in Eq. (1).

$$v_{ik}^{t+1} = w * v_{ik}^t + c1 * r1 * (pb_{ik}^t - x_{ik}^t) + c2 * r2 * (gb_k^t - x_{ik}^t) \tag{1}$$

$$x_{ik}^{t+1} = x_{ik}^t + v_{ik}^{t+1} \tag{2}$$

In Eq. (1), 'w', '$c1$', and '$c2$' are parameters of PSO algorithm while '$r1 \in [0, 1]$' and '$r2 \in [0, 1]$' are random values. Eq. (1) consists of three parts and each part can be described from a sociological perspective. The first part (that is, $w * v_{ik}^t$) describes the influence of the i^{th} particle's velocity at t^{th} (i.e., v_{ik}^t) on v_{ik}^{t+1}. In other words, each particle has the possibility of staying in its current position, which is controlled by the parameter 'w' and thus this parameter is called *inertia coefficient*. In the second part, the term $(pb_{ik}^t - x_{ik}^t)$ represents the distance between the i^{th} particle's current position (i.e., x_{ik}^t) and its best position (that is, pb_{ik}^t). This item indicates that the particle's new position is guided by its own experience, i.e., its thinking, and thus is called the cognitive item. The cognitive item is controlled by the PSO constant '$c1$' and thereby this constant is called *cognitive coefficient* or sometimes *cognitive acceleration factor*. The item $(gb_k^t - x_i^t)$, in the last part, measures the distance between the i^{th} particle's current position (i.e., x_{ik}^t) and the best global position (i.e., gb_i^t) achieved by all particles in the swarm. This term guides the movement of the i^{th} particle towards the best position in the swarm, and thus this term is called the social item. Since the PSO coefficient '$c2$' controls this, it is called *social coefficient* or *social acceleration factor*.

From Eq. (1), it can be observed that a higher value of 'w' helps the particle stay in its position, while the values of '$c1$' and '$c2$' help the particle move to the best position personal or the best position achieved by all the particles in the swarm. Thus, the parameters 'w', $c1$', and '$c2$' control the exploration and exploitation capabilities of the particles in the search space. All of these make PSO very effective in searching for the near-optimal solution in a multidimensional search space because the particles' new positions and velocities are not only dependent on their position but also consider the position of the best particle in the swarm. The working principle of PSO is pictorially shown in Fig. 1.

4. Binary PSO

Optimization problems like feature selection need to encode the solutions as a binary vector (Rashedi et al., 2010). Even problems defined over a real number space can be transformed into a binary space, as real numbers can be encoded as binary numbers. The PSO defined in a continuous search space cannot be applied directly to these problems (Kennedy and Eberhart 1997; Mirjalili and Lewis 2013). A binary search space is a hypercube, where particles move around by flipping various bits (Kennedy and Eberhart 1997). If a position of the particle faces a 'zero' bit flip from t^{th} iteration to $(t+1)^{th}$ iteration, then the particle does not move at all, while if the 'all' bits are flipped, then the particle moves to the farthest point from its current position.

It should be noted that particles can move in the continuous search space using Eq. (2). But this is not true for binary search space, as in this space a particle moves from one place to another just by flipping various bits of its position vector. That is, we cannot directly use Eq. (2) to update the position of any particle in the binary search space. Here comes the need for a scheme that can be utilized to change the position of a particle by switching '0' to '1' or vice versa. In other words, a mapping between velocity and position must be established so that particles in the binary space can update their position using Eq. (1). From Eq. (1) one can see that x_{ik}^{t} can be represented using '0' and '1' but since v_{ik}^{t+1} generates real value and therefore x_{ik}^{t+1} cannot always be represented using '0' and '1' only. To solve this problem, Kennedy and Eberhart (1997) used a mechanism to map real-valued velocities to $\{0,1\}$ i.e., a TF that maps v_i^{t+1} from $\mathbb{R}$ to $\{0,1\}$, i.e., TF:$\mathbb{R} \rightarrow \{0,1\}$. Kennedy and Eberhart (1997) used a sigmoid function (see Eq. (3)) and since this

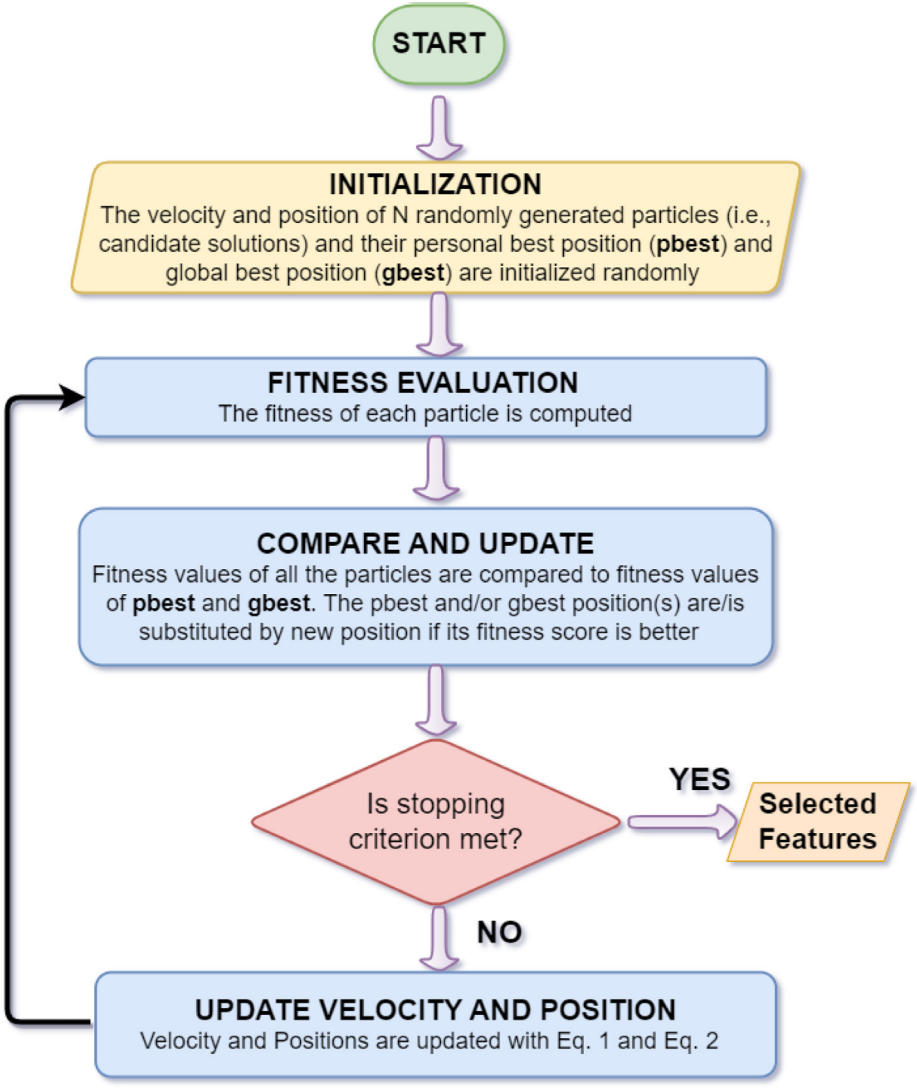


Fig. 1. Working principle of Particle Swarm Optimization.

TF looks like "S" graphically, it is also called as S-shaped TF. A sigmoid function maps any real number, which is the velocity of any particles, to the interval $[0, 1]$.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

In Eq. (3), x denotes a real value, which is here v_{ik}^{t+1}. For a D-dimensional space, say, the components of velocity and position vector in the k^{th} direction of i^{th} particle at iteration t are v_{ik}^{t} and x_{ik}^{t}. Thus, Eq. (3) can be written using Eq. (4).

$$f(v_{ik}^{t}) = \frac{1}{1 + e^{-v_{ik}^{t}}} \tag{4}$$

Eq. (4) converts all components of a velocity vector into probability values as $f(v_i^t(k)) \in [0, 1], \forall k \in \{1, 2, \ldots, D\}$. Now, the position vector of the particle i^{th} at iteration t in the D-dimensional real space can be assigned to the D-dimensional binary space using Eq. (5).

$$x_{ik}^{t+1} = \begin{cases} 0 & \text{if } rand < f(v_{ik}^{t+1}) \\ 1 & \text{if } rand \geqslant f(v_{ik}^{t+1}) \end{cases} \tag{5}$$

In Eq. (5), $rand \in [0, 1]$ is a real number generated randomly and $f()$ is the S-shaped TF defined in Eq. (3). All these make the BPSO as simple as the basic PSO. However, in the literature, there is a parallel line of research works (e.g., Mirjalili and Lewis (2013); Beheshti (2020); Sarkar et al. (2018); Wang et al. (2008); Wei et al. (2017); Nguyen et al. (2021); Li et al. (2021); Shen et al. (2004); Lee et al. (2008)) looking at the local optimum problem either by modifying the intrinsic nature of PSO (e.g., Nguyen et al. (2021); Li et al. (2021, 2018, 2004, 2008)) or using modified TFs (e.g., Mirjalili and Lewis (2013); Beheshti (2020); Wei et al. (2017)). But the problem is not fully resolved to date and hence, we confine ourselves to the more standard approach i.e., we have updated the position of a particle using Eq. (1) and Eq. (5) while employing different TFs as $f()$ mentioned in Eq. (5).

5. Different TFs used in BPSO

In the BPSO algorithm, a TF plays an important role in deciding the position of a particle in the binary search space. In other words, TFs are used in the BPSO algorithm to determine the probability score of each bit (i.e., x_{ik}^{t+1}), which depends on the corresponding velocity (i.e., v_{ik}^{t+1}). It is also worth mentioning here that, in general, in the early stages, an optimization algorithm should focus more on exploration to diminish the chances of getting trapped in local optima, while in the later stages, it should emphasize more exploitation capability to refine the solution quality. Keeping these facts in mind, many researchers (Mirjalili and Lewis 2013; Rashedi et al., 2010; Islam et al., 2017) suggested that the following set of rules should be satisfied by an ideal TF so that BPSO can benefit from higher exploration and exploitation.

1. A TF must map velocity values in the range $[0, 1]$.
2. A TF should return a high probability score ($\geqslant 0.5$) if the absolute velocity of a particle is high, and a low probability value ($\leqslant 0.5$) if the absolute velocity of a particle is low.
3. A TF should behave in the following way with an increased iteration count
   (a) In the early stages, the TF in used should return a high bit flipping probability score (i.e., score to decide to flip x_{ik}^{t+1}) for any value of v_{ik}^{t+1}, so that the BPSO can benefit from a stronger exploration capability.
   (b) In the intermediate stages, since the BPSO should start switching from exploration to exploitation, the used TF should return a decreased bit flipping probability score with an increased iteration count.
   (c) In the final stages, the TF should provide a low bit flipping probability score for any value of v_{ik}^{t+1} to allow the BPSO to exploit the data more effectively.

According to these rules, a TF that returns a large bit-flipping probability even for a small value of velocity (close to '0') is preferable at the early stages of search for a better exploration of the search space. On the other hand, a TF that generates a large bit flipping probability only when the value of velocity is large is preferable at the final stages of the search process to obtain better exploitation of the search space. In this work, we have considered five popularly used TFs to investigate their effects on diagnosis report-based chronic disease prediction systems when BPSO is used to select optimal diagnosis attributes. TFs that we have considered in this work are mentioned below.

- S-shaped or Sigmoid TF.
- Linear TF.
- V-shaped TF.
- U-shaped TF.
- Time-Varying Mirrored S-shaped (TVMS) TF.

In the following subsections, first, we have described all these five TFs and then analyzed them in view of the above-mentioned rule set.

5.1. S-shaped TF

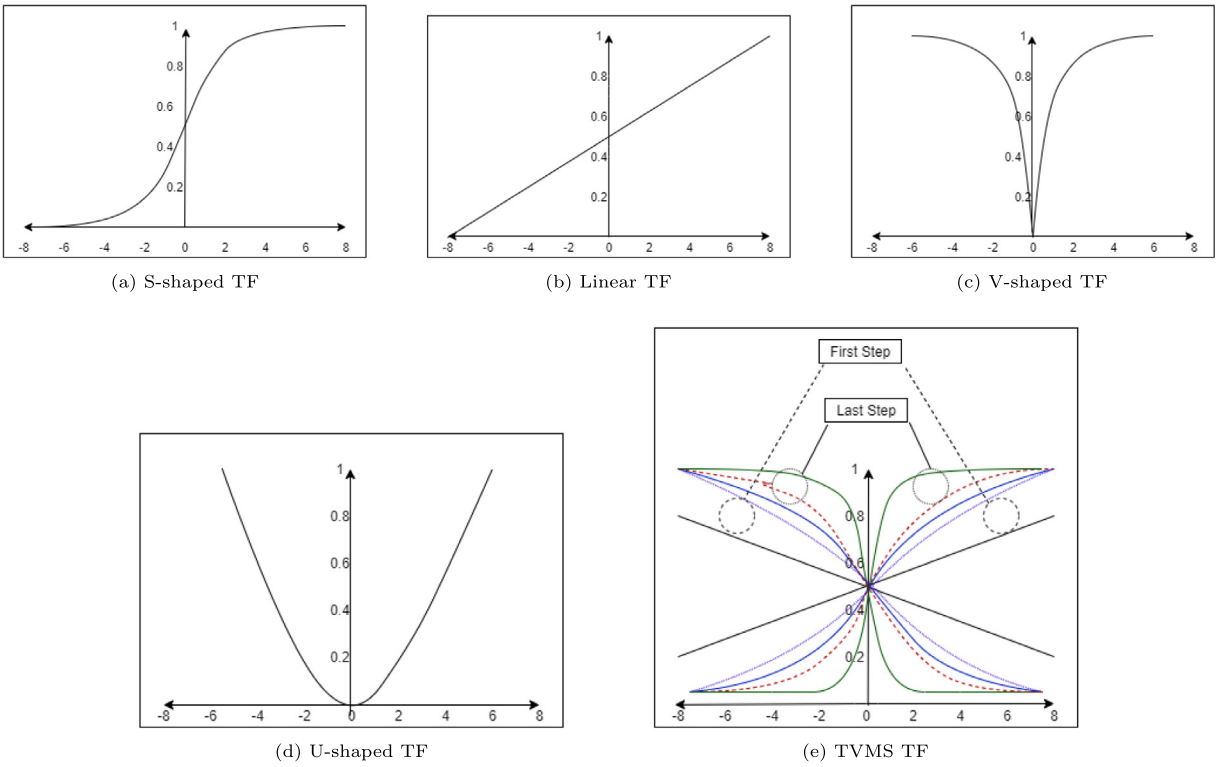
The use of S-shaped TF (see Eq. (3)) is the oldest in the context of BPSO. Kennedy and Eberhart (1997) used S-shaped TF in their work, where they introduced the concept of BPSO and this BPSO is termed as basic BPSO in many works in the literature. The graphical representation of the S-shaped TF has been shown in Fig. 2a and according to this figure, the S-shaped TF returns a high bit flipping probability score for a large value of velocity in the positive direction and negative direction, which are almost '1' and '0' respectively, and thus a large velocity in the negative direction causes no change in its position. Moreover, if the velocity is close to zero, then the value of the corresponding bit becomes uncertain. These circumstances can sometimes prevent a particle from reaching its global optimum (Mirjalili and Lewis 2013). To address this issue, Shen et al. (2004) and Lee et al. (2008) attempted various methods to update the position vector formula (i.e., Eq. (5) while Mirjalili and Lewis (2013) introduced three new variants of the actual S-shaped TF. But the problem of sticking to the local optimum is not fully resolved, and hence in our work we have considered the standard S-shaped TF as defined in Eq. (3).

5.2. Linear TF

Liner TF was introduced by Wang et al. (2008). It was probably the first initiative to solve the major problems of S-shaped TF with the differently shaped designed TF. This is very simple and tries to improve the exploration capability of S-shaped TF. The TF is defined using Eq. (6).

$$f(x_{ik}^{t+1}) = \frac{x_{ik}^{t+1} - R_{min}}{R_{max} - R_{min}} \tag{6}$$

In Eq. (6), $[R_{min}, R_{max}]$ is the predefined range of particles' position. The authors suggested that this TF works better with the value $R_{min} = -50, R_{max} = +50$ but this cannot be true always. There



(a) S-shaped TF (b) Linear TF (c) V-shaped TF

(d) U-shaped TF (e) TVMS TF

Fig. 2. Graphical representation of different TF used here.

may be some situations where $f(x_{ik}^{t+1})$ may attain a value that is larger than '1' or smaller than '0' since x_{ik}^{t+1} is controlled by Eq. (2) and $[R_{min}, R_{max}]$ posses preset value. This scenario violates the precondition that f() is a TF, which states that a TF's value must be in the range $[0, 1]$. It should be noted that x_i^{t+1} is a real vector rather than a binary vector like BPSO. To keep x_{ik}^{t+1} as a real number in each iteration, x_{ik}^{t+1} is calculated with Eq. (2), which means they used the PSO algorithm's actual velocity and position update formula. Now, by combining Eq. (6) and Eq. (2) we can write the linear shaped TF as Eq. (7)

$$f(x_{ik}^{t+1}) = \frac{x_{ik}^{t} + v_{ik}^{t+1} - R_{min}}{R_{max} - R_{min}} \tag{7}$$

The interesting fact about this technique is that the authors used different variables to keep track of the updated position of the particles in the binary search space. The updated position (say, px_{ik}^{t+1}) of the particles is decided using the rule mentioned in Eq. (8).

$$px_{ik}^{t+1} = \begin{cases} 1, & \text{if } rand < f(x_{ik}^{t+1}) \\ 0, & \text{if } rand \geqslant f(x_{ik}^{t+1}) \end{cases} \tag{8}$$

The simplicity of the TF makes the process a bit computationally efficient, but the BPSO with this TF should experience the same difficulties as the BPSO with an S-shaped TF, except that it possesses a better exploration capability (Islam et al., 2017). The graphical representation of linear TF is shown in Fig. 2b. From this, it is clear that the slope of the TF depends on the values of R_{min} and R_{max} and the y-intercept value is 0.5 when $R_{min} = -R_{min}$ i.e., zero velocity will generate a bit flipping probability score having a value of 0.5. Apart from this, the shape of the TF infers that it returns a larger bit-flipping probability score for a certain range of velocity than that of the S-shaped TF. In fact, this shows its better exploration capability. However, this TF cannot guarantee good exploitation capability at the end stages of searching, and thus may trap local optima like an S-shaped TF. Apart from these, it is also noticeable from Eq. (5) and Eq. (8) that the bit position of a particle may not change even if the TF returns a larger score than rand() returns when it is already '1'. This scenario may lead to lower exploration capacity.

5.3. V-shaped TF

Rashedi et al. (2010) introduced the V-shaped TF (defined in Eq. (9)) that uses a new position update strategy (defined in Eq. (10)). It was used in the binary gravitational search algorithm then, and later was used along with PSO with some improvement (Mirjalili and Lewis 2013; Liu et al., 2011).

$$f(v_{ik}^{t+1}) = |tanh(v_{ik}^{t+1})| \tag{9}$$

$$x_{ik}^{t+1} = \begin{cases} \neq gx_{ik}^{t} & \text{if } rand < f(v_{ik}^{t+1}) \\ x_{ik}^{t} & \text{if } rand \geqslant f(v_{ik}^{t+1}) \end{cases} \tag{10}$$

Rashedi et al. (2010) set an upper limit for the velocity in their implementation, and this forces them to clip a larger velocity to a lower one. This clipping not only negotiates the particles to attain their height velocity but also may increase the gradient of the V-shape resulting in a lower velocity to attain a higher bit flipping probability score and lowering the exploration capability of the BPSO (Mirjalili et al., 2020). This situation is clear from the shape of this TF shown in Fig. 2c. It is also clear from this figure that a large velocity in either direction (i.e., positive or negative) gets a probability score close to '1' while a smaller velocity gets a probability score close to '0'. That means a high velocity will increase the chance of a position change for the corresponding particle. Not only that, the Eq. (10) uses the term $\neq gx_{ik}^{t}$ to ensure the bit (i.e., x_{ik}^{t}) is flipped when the condition is true. This is also an improvement over the BPSO techniques that use S-shaped or linear TF. It

is also clear from Fig. 2c that it is highly unlikely that the particle will change its position when the particle is close to the local optima value and the velocity is close to '0'. As a result, the particle may become trapped in that local optimum solution and experience poor convergence, as seen in S-shaped and linear TF. This TF along with modifications suggested in Mirjalili and Lewis (2013), Liu et al. (2011) do not consider changing their shape with increasing iteration number, i.e. they maintain steady exploration capability, and thus once a particle traps into a local optimum it becomes hard to come out of this situation. This is because these V-shaped TFs return a small value (close to '0'), and thus the bit-flipping probability becomes low. All these restrict this TF from keeping a good balance between exploration and exploitation of the search space. Thus, in our study, we use the basic form of a V-shaped TFs.

5.4. U-shaped TF

Mirjalili et al. (2020) introduced a U-shaped TF that can solve the main problem of the S-shaped TF. The TF is controlled by two parameters, namely α and β, which allow particles in a swarm to explore different areas of the search space. The authors defined the U-shaped TF and the new position of a particle using Eq. (11), and Eq. (12) respectively and the shape of an instance of U-shaped TF is shown in Fig. 2d. It is to be noted here that the position update formula remains the same as the V-shaped TF-based BPSO due to its benefits over basic BPSO techniques.

$$f(v_{ik}^{t+1}) = \alpha * |(v_{ik}^{t+1})^{\beta}| \tag{11}$$

$$x_{ik}^{t+1} = \begin{cases} \neq gx_{ik}^{t} & \text{if } rand < f(v_{ik}^{t+1}) \\ x_{ik}^{t} & \text{if } rand \geqslant f(v_{ik}^{t+1}) \end{cases} \tag{12}$$

In Eq. (11), α and β control the slope and basin's width of the U-shaped TF respectively. The higher value of α means the TF reaches its saturation point faster. In other words, a larger α value ensures that the U-shaped TF returns a larger bit flipping probability, and thus, this parameter accelerates the exploration capability of a BPSO algorithm. The other parameter (i.e. β) controls the width of the basin of this TF. It should be noted that a wider U-shaped TF lowers the exploration capacity of the BPSO. This is because, as Fig. 2d suggests, a wider U-shaped TF slows down, generating a higher bit-flipping probability score. Therefore, we can say that the values of α and β can control the exploration capability of a particle in a swarm, but the associated BPSO fails to maintain a good balance between exploration and exploitation, a good characteristic for any optimization technique, like other TF-based BPSOs discussed earlier.

5.5. TVMS

In the recent past, Beheshti (2020) recently introduced the TVMS TF, which attempts to overcome the issues of S-, linear-, V-, and U-shaped TF and meets all of the suggested prerequisites for a TF to be used with PSO. As shown in Fig. 2e, this TF has two S-shaped functions: $f()$ defined in Eq. (13) and $f'()$ defined in Eq. (14) -one is a of the other. S Such use of the S-shaped TF and its mirror helps this TF return closer to the '1' value when a particle attains a higher velocity in either the positive or negative direction, similar to the V-shaped or U-shaped TF but different from the S-shaped or Linear TF. This modification helps the BPSO achieve better exploration over S-shaped TF.

$$f(v_{ik}^{t+1}) = \frac{1}{1 + e^{-\sigma * v_{ik}^{t+1}}} \tag{13}$$

$$f\prime(v_{ik}^{t+1}) = \frac{1}{1 + e^{\sigma * v_{ik}^{t+1}}} \tag{14}$$

In Eq. (13) and Eq. (14), σ is a value that varies over time, which is here iteration. This value decreases gradually from σ_{max} to σ_{min} as the iteration increases to switch smoothly from exploration to exploitation. Such modification helps the BPSO attain maximum exploration and exploitation at the initial and final stages, respectively, and thus has a higher chance of avoiding trapping at the local maxima. The value of σ is controlled by Eq. (15).

$$\sigma = (\sigma_{max} - \sigma_{min})(\frac{\text{current iteration number}}{\text{maximum iteration number}}) + \sigma_{min} \tag{15}$$

The author showed that $\sigma_{max} = 1$ and $\sigma_{min} = 0.1$ performs the best for the problem they have considered. Next, the author defined binary positions concentrating on each S-shaped function, say P and $P\prime$ using Eq. (16) and Eq. (17), respectively.

$$P_{ik}^{t+1} = \begin{cases} 1, & \text{if } rand < f(v_{ik}^{t+1}) \\ 0, & \text{if } rand \geqslant f(v_{ik}^{t+1}) \end{cases} \tag{16}$$

$$P_{ik}^{\prime t+1} = \begin{cases} 1, & \text{if } rand > f'(v_{ik}^{t+1}) \\ 0, & \text{if } rand \leqslant f'(v_{ik}^{t+1}) \end{cases} \tag{17}$$

After calculating the P and $P\prime$ values, the best position of a particle is confirmed using a greedy objective function defined in Eq. (18). The disadvantage of using a position update formula like the BPSO with an S-shaped or Linear TF is that it can lead to trapping at local optima if the parameters are not properly tuned.

$$x_{ik}^{t+1} = \begin{cases} P_{ik}^{t+1}, & \text{if } f(P_{ik}^{t+1}) \geqslant f(P_{ik}^{\prime t+1}) \\ P_{ik}^{\prime t+1}, & \text{if } f(P_{ik}^{\prime t+1}) \geqslant f(P_{ik}^{t+1}) \end{cases} \tag{18}$$

From the literature on TFs and discussions made in this section, it is clear that previous researchers Wang et al. (2008), Mirjalili and Lewis (2013), Liu et al. (2011), Mirjalili et al. (2020), Islam et al. (2017), Beheshti (2020) proposed different TFs to improve the basic BPSO proposed by Kennedy and Eberhart (1997) but with a larger set of parameters to control the shape of the TFs and aim at introducing better exploration and exploitation in BPSO. But tuning these parameters needs another round of expertise and careful observations. They also need the careful selection of a classifier and analysis of the datasets under consideration. These studies are outside the scope of the present study, and thus, in our experiments, we have considered the basic form of the TFs under consideration. From a theoretical standpoint, the TVMS TF appears to have the best exploration and exploitation capabilities, as it satisfies all the prerequisites, discussed earlier, for a TF to be used in any optimization technique.

6. Feature selection using BPSO

We have discussed PSO in Section 3, BPSO in Section 4, and the family of TFs used to transfer PSO from the continuous domain to the binary domain in Section 5. However, how BPSO extracts an efficient feature subset from the entire feature set has not been discussed, and thus, we describe the same here.

Let us assume that the number of particles (alternatively known as candidate solutions) in a population (say, P) used in PSO is M and the number of features present in a dataset is D. This means that each of the M particles $\in P$ has D columns and thus a particle of the population P can be represented as a $[1 \times D]$ dimensional vector in the D dimensional search space having only binary values- '0' (represents the corresponding feature is discarded) and '1'

(represents the corresponding feature is included in the selected feature set). A population containing M such particles creates a 2D array of size $[M \times D]$ in any iteration and P_i, where $i = 1, 2, \ldots, M$ represents the position of a particle in the population.

In the beginning, P particles are placed in the search space randomly. Similarly, M number of $[1 \times D]$ dimensional arrays is initialized with a zero value, representing the velocity of particles present in P. Additionally, we also have M number of $[1 \times D]$ dimensional vectors each representing the best position of the corresponding particle (i.e., pb_{ik}^t) in the population and one $[1 \times D]$ dimensional vector to represent the best global position of all particles (i.e., gb_k^t) in the population. Here, we have calculated the fitness score of the candidate solution by its classification score on the validation dataset and the number of features it contains (see Eq. (19)). In this equation, *fit* is the fitness score for a feature subset, while c and nf represent the classification accuracy and number of features in the candidate solution, respectively. $w_c \in [0, 1]$ and $w_{nf} \in [0, 1]$ (with $w_c + w_{nf} = 1$) represent weights for c and nf respectively, which we set here as $w_c = 0.7$ and $w_{nf} = 0.3$ empirically. A higher fitness score indicates a better candidate solution.

$$fit = w_c * c + w_{nf} * nf \tag{19}$$

Next, in each iteration, the particles' velocities and new positions are calculated using Eq. (1) and Eq. (2). From these equations, it is clear that the velocity and thus the new position vector of any particle is represented by real numbers. To convert these real numbers to '0' or '1', one can use any of the TFs and the corresponding position update formula described in Section 4. The pb_{ik}^t and gb_k^t in t^{th} iteration get updated whenever a better position is obtained in terms of the fitness score. When PSO is used as a feature selector, every particle (i.e., candidate solution) of the swarm represents a subset containing selected features. Therefore, after receiving multiple subsets of such features, each passes through the fitness calculation process. The candidate solution with the best fitness score is then compared with the best global best, and if it is better, it is set as the best global solution, which is finally considered *Leader Particle* at the end of iterations. In that *Leader Particle*, the positions (that is, the indices in the original vector of dimensional features D) having the value = 1, are considered the final selected features.

7. Experimental results

In this work, we have shown the performance of the BPSO algorithm in selecting optimal features while using different TFs and shallow classifiers. In this study, we have considered four chronic disease datasets: Heart disease, Breast Cancer, CKD, and Diabetes. Classification is done using four popular classifiers: GNB, KNN, SVM, and DT. We have also used five different TFs as described in Section 5 to apply the PSO to the binary domain. In the following subsections, we first describe the four disease datasets considered here for experimentation and then report and analyze the different experimental results.

7.1. Description of datasets

Here, we have presented the description of the four datasets of chronic diseases used in this work. It is noteworthy to mention that all the datasets are publicly available. The researchers, who made these datasets public, collected and processed the original samples and provided the processed data that are to be used by the research community.

7.1.1. Heart disease dataset

UCI Heart disease dataset[2] is one of the most commonly used datasets in the literature to validate the performance of any diagnostic report-based decision support system. The original samples were processed by Detrano et al. (1989) and made public as the diagnostic dataset for further research work. The original dataset contains 76 attributes (75 diagnostic attributes and 1 class attribute). However, most of the researchers used 14 attributes available in the publicly available dataset. These 14 attributes constitute of 13 diagnostic attributes: age, sex, chest pain (CP), resting blood pressure (trestbps), serum cholesterol in mg/dl (chol), fasting blood sugar (FBS), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), slope of the peak ST segment of exercise (slope), number of major vessels colored by fluoroscopy (ca) and, thal and 1 class attribute indicating whether the subject is affected by the disease or not. This dataset contains 303 observations where 165 cases have a heart attack and 138 cases are without a heart attack.

7.1.2. Breast cancer dataset

Wisconsin Breast Cancer dataset[3] was prepared and made public by Street et al. (1993). This dataset contains 699 observations. Each observation of this dataset consists of 9 diagnostic attributes, namely Sample code number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses and 1 class attribute (2 for benign and 4 for malignant). In our experiments, we have removed 16 incomplete instances. We have excluded such observations since the present study does not include any missing value substitution method as a wrongly chosen missing value substitution method might lead to a wrong analysis of the performance of TFs under consideration. As a result, the filtered dataset contains 684 observations, of which 444 cases are marked as benign and 239 cases as malignant.

7.1.3. CKD dataset

For experimenting with CKD, we have considered the UCI CKD dataset[4], a commonly used dataset found in the literature. This dataset was prepared and made public by Rubini et al. (2015). It contains diagnosis reports for 400 subjects, of which 250 are infected cases and 150 are normal cases. Each observation has 24 diagnostic attributes and 1 class attribute. Diagnostic attributes considered for a subject are of two types: nominal features (13 out of 24) that are: Specific gravity, Albumin, Sugar, Red blood cells counts, Pus cell, Pus cell clumps counts, Bacteria, Hypertension, Diabetes mellitus, Coronary artery disease, Appetite, Pedal Edema, and Anemia, and numerical features (11 out of 24) that are: Age, Blood Pressure, Blood Glucose Random, Amount of Urea in blood, Serum Creatinine, Sodium, Potassium, Hemoglobin, Packed Cell Volume, White Blood Cell Count and Red Blood Cell Count. The dataset has 400 instances out of which 244 instances are removed during experiments due to the presence of some missing values. All experiments are done on the remaining data samples.

7.1.4. Diabetes dataset

PIMA Indians Diabetes dataset[5] is a very popular dataset that was prepared and made public by Smith et al. (1988). This dataset is widely used for evaluating the performance of a newly designed decision support system for diabetes diagnosis. Each observation in the data set has eight diagnostic attributes: number of pregnancies,

[2] https://archive.ics.uci.edu/ml/datasets/heart+disease.
[3] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic).
[4] https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease.
[5] https://www.kaggle.com/uciml/pima-indians-diabetes-database.

body mass index (BMI), insulin level, glucose level, blood pressure (BP) level, skin thickness, Diabetes pedigree function, age, and 1 class attribute that indicates whether an observation is marked as infected or not. The dataset contains 768 observations, among which 500 are non-diabetic and 268 are diabetic cases. In this case, we have used all the observations since the dataset is not affected by missing values.

7.2. Parameter settings of BPSO

The main objective of the present work is to study the performances of different TFs when selecting diagnostic attributes using the BPSO algorithm. In other words, we have investigated the performance of the BPSO algorithm with varying TFs, which are here S-shaped, V-shaped, U-shaped, TVMS, and linear TF. For all cases, we have used Eq. (1) to calculate the velocity of the particles. An obtained velocity is then considered in the TFs to update the position of the particle (except linear TF as it does not consider the velocity parameter). To update the positions of the particles, the respective position update formula is used. To be specific, we have used Eq. (5), Eq. (10), Eq. (12), Eq. (18), and Eq. (8) for S-shaped, V-shaped, U-shaped, TVMS, and linear TFs, respectively.

It has been discussed earlier that Eq. (1) has three parameters: 'w', '$c1$', and '$c2$' that control the exploration and exploitation of the PSO algorithm. Therefore, to obtain a fair comparison among all TFs, these parameters need to be set correctly. In the literature, two different approaches are found to set the values of these parameters. A group of researchers (e.g., Rajamohana and Umamaheswari (2018) and Gunasundari et al. (2018)) kept the value of these coefficients fixed with the change of iteration, while others (for example, Chaturvedi et al. (2009), Xiong et al. (2019), and Elbedwehy et al. (2012)) suggested changing these values with increasing iteration number to achieve better exploration and exploitation capabilities. However, in our experiments, we have followed the setups as described by Chaturvedi et al. (2009). The initial value of 'w' is set to '1' and reduces as the number of iterations increases, as shown in Eq. (20).

$$w_{i+1} = 1.0 - \frac{i}{maxIter} \tag{20}$$

In Eq. (20), i is the current iteration number and $maxIter$ is the maximum iteration number set during an experiment. We also change the parameters' values: $c1$ and $c2$ following the work proposed by Chaturvedi et al. (2009). In this method, the value of $c1$ decreases with increasing iteration number (from 2.05 to 0.5) while the value of $c2$ increases (from 0.5 to 2.05) with increasing iteration (here, iteration is analogous to time). The initial values are $c1_0 = 2.05, c1_{maxIter} = 0.5, c2_0 = 0.5$, and $c2_{maxIter} = 2.05$.

$$c1_{i+1} = c1_i + \frac{i * (c1_{maxIter} - c1_i)}{maxIter} \tag{21}$$

$$c2_{i+1} = c2_i + \frac{i * (c2_{maxIter} - c2_i)}{maxIter} \tag{22}$$

In Eq. (21) and Eq. (22), $i = 1, 2, \ldots, maxIter$. Such parameter settings guide the BPSO to change from exploration to exploitation with the increase of iteration. Also, in our study, we have empirically set the number of particles in a swarm or population size at 50 (that is, the number of possible solutions as 50 from which the best is chosen) and $maxIter = 100$ to produce one result empirically.

7.3. Experimental setup

In this subsection, we have described the data preparation and experimental schemes that we have followed. At first, each dataset has been passed through a data cleaning process, in which missing values are removed. We have performed the experiments using hold-out test samples i.e., test samples are selected from the entire pre-processed dataset and kept aside for model evaluation. We

Table 1
Performances of different classifiers for disease prediction without feature selection. Additional information of datasets is also given.

Disease	Number of			Classification Accuracy (in %) using			
	Train Samples	Test Samples	Diagnostic Attributes	GNB	KNN	SVM	DT
Heart disease	242	61	13	86.89	68.85	70.49	81.97
Breast Cancer	546	137	9	95.62	94.89	94.89	93.43
CKD	124	32	24	100.00	81.25	81.25	96.87
Diabetes	614	154	8	76.62	66.23	76.62	75.97

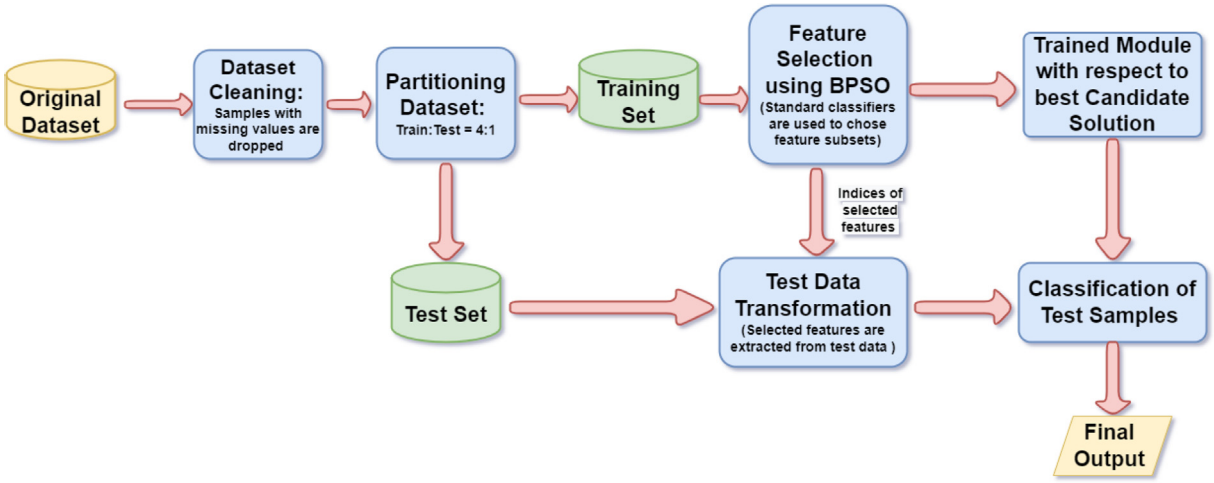


Fig. 3. The experimental setup used in the hold-out approach.

have followed the hold-out approach because we want the same test conditions for each BPSO setup (i.e., classifier and TF) while using cross validation would have made the experiments excessively time consuming. Unless otherwise specified, all the results mentioned are for these hold-out test samples only. We randomly selected 20% samples from the entire preprocessed dataset to prepare the test samples. For model training, we used the rest of the samples. The counts of train and test samples from each data set are shown in Table 1. Afterward, we have employed the BPSO-based feature selection with different TFs on the train samples to select a set of optimal features. In the BPSO, to decide on a better feature subset or candidate solution, we have evaluated the fitness score, which is controlled by the classification score and the number of selected features. The classification score generated during the feature selection phase is measured on the validation set which is selected from the training set (20% of the training samples). The BPSO returns feature indices of the near-optimal feature subset (i.e., features in the Leader particle) and the corresponding learning module. Finally, the near-optimal features are extracted from the test samples, which are then used to classify the samples with the help of the learned module. The entire experimental process following the hold-out test set approach is described in Fig. 3.

In the experiments conducted here, we have used four different classifiers, namely GNB, KNN, SVM, and DT, to perform classification tasks. For these classifiers, we have used the default parameters that are used in the machine learning library *Scikit-learn*. For example, $K = 5$ is considered the default parameter for the K-NN classifier, while for the SVM classifier, we have used the Radial Basis Function (RBF) as the kernel. In the case of the DT classifier, it uses max_depth by default, i.e., the nodes will be expanded until all leaves contain less than two samples.

The BPSO-based feature selection technique has certain randomness in its procedure like the creation of the initial population, velocity calculation, and deciding a bit as '0' or '1' using any TF. Therefore, it may produce slightly different results in different runs. However, if the algorithm converges after a certain number of iterations, then it can produce almost similar results in different runs, i.e., the standard deviation (SD) of performances over several runs remains very low. A lower SD value of the performances over several runs ensures the stability of the algorithm. Thus, we have performed BPSO-based feature selection 20 times on each dataset for each BPSO setup (consisting of the classifier and a TF). It is to be noted that we have used the same set of validation and training samples during the feature selection stage for all these 20 independent runs. This approach also helps us to identify a better BPSO setup by excluding unstable setups. We have recorded all accuracy values and finally reported the results in terms of maximum, minimum, average, and SD of the obtained accuracy. We have also recorded the set of diagnostic attributes for each disease dataset that provides the best and worst results in terms of classification accuracy.

7.4. Performance of disease prediction

This section describes the disease prediction capabilities of the BPSO-based decision support systems with varying TFs and classifiers in use. To start with we have first evaluated the performances of each classifier using all attributes present in the dataset under consideration. The classification performances of the classifiers used here are shown in Table 1. The results show that GNB has provided the best classification accuracy among these four classifiers for all datasets. Although GNB provides the best results, we have continued with all classifiers to investigate how the BPSO-based feature selection model performs when the TF is being changed. In other words, we have tried to investigate whether all the BPSO-based feature selection setups can improve the performance of disease prediction or if some of the setups do this. In other words, these experiments help us check whether there is any dependency between classifiers and TFs used when selecting optimal features using BPSO. By "dependency between classifiers and TFs", we mean that there is a specific BPSO setup (i.e., the classifier and TF) that improves classification performance on a specific dataset. If no such situation occurs, then we can say that there is no such dependency, at least for the datasets, classifiers, and TFs currently used. We have described the results of different BPSO setups in the following subsections for all the diseases considered here.

7.4.1. Performance on the Heart disease prediction

The results of different BPSO setups are recorded in Table 2. In this table, the terms "Max.", "Min.", "Avg.", "SD", "#SF (best)", "ICA", and "RFD" represent maximum classification accuracy, minimum classification accuracy, average classification accuracy, SD of classification accuracies, number of selected features in the best

Table 2
Performances of BPSO with different TFs and classifiers for Heart disease prediction. The meaning of the shortened column terms used in this table are defined in Section 7.4.1. The boldfaced numbers indicate the best scores in terms of maximum and average accuracy.

TF	Classifier	Classification Accuracy (in %)				#SF	Gain (in %)	
		Max.	Min.	Avg.	SD	(best)	ICA	RFD
S-shaped	GNB	95.08	90.16	92.46	01.76	6	05.57	69.23
	KNN	**96.72**	88.52	90.49	02.42	5	21.64	61.53
	SVM	91.80	86.89	89.18	01.38	5	18.69	61.53
	DT	91.80	86.89	88.36	01.63	6	06.39	53.85
V-shaped	GNB	93.44	90.16	91.97	01.21	5	05.08	61.53
	KNN	95.08	88.52	90.49	02.02	5	21.64	61.53
	SVM	95.08	90.16	**92.95**	01.90	6	**22.46**	53.85
	DT	91.80	88.52	90.16	01.34	6	08.19	53.85
U-shaped	GNB	95.08	90.16	92.30	01.35	5	05.41	61.53
	KNN	93.44	88.52	90.00	01.63	5	21.15	61.53
	SVM	93.44	90.16	92.46	01.15	6	21.97	53.85
	DT	91.80	88.52	89.67	01.35	6	07.70	53.85
TVMS	GNB	93.44	90.16	91.59	01.36	5	07.70	61.53
	KNN	93.44	88.52	90.33	01.80	5	21.48	61.53
	SVM	91.80	88.52	90.16	01.55	6	19.67	53.85
	DT	**96.72**	86.89	89.18	03.01	**4**	07.21	**69.23**
Linear	GNB	93.44	90.16	91.80	01.23	8	04.91	38.46
	KNN	93.44	90.16	90.98	01.39	7	22.13	46.15
	SVM	93.44	90.16	91.48	01.29	7	20.99	46.15
	DT	90.16	88.52	88.85	00.69	6	06.88	53.85

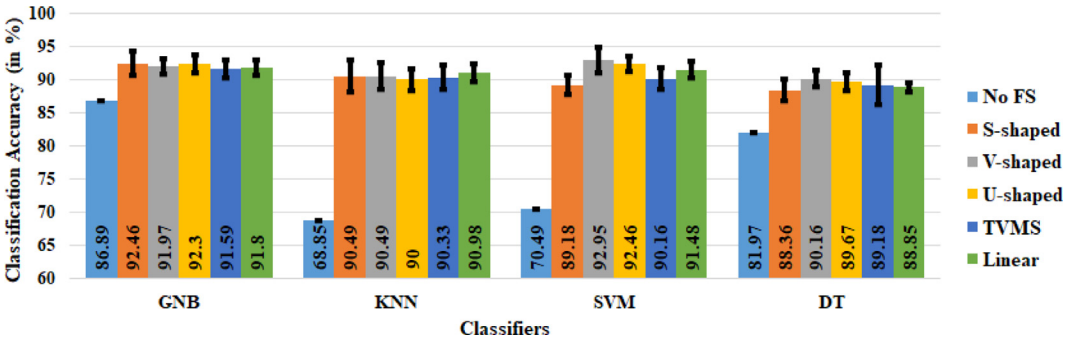


Fig. 4. Performance (average ± SD classification accuracy) of different BPSO setups for Heart disease prediction. Here the term "No FS" indicates the performance of an individual classifier without using the BPSO-based feature selection.

Table 3
Performance of BPSO with different TFs and classifiers for Breast Cancer prediction. The meaning of the shortened column terms used in this table are defined in Section 7.4.1. The boldfaced numbers indicate the best scores in terms of maximum and average accuracy.

TF	Classifier	Classification Accuracy (in %)				#SF	Gain (in %)	
		Max.	Min.	Avg.	SD	(best)	ICA	RFD
S-shaped	GNB	99.27	97.81	98.61	00.64	5	02.99	44.44
	KNN	99.27	97.81	98.54	00.60	5	03.65	44.44
	SVM	98.54	97.81	98.25	00.38	5	03.39	44.44
	DT	98.54	97.08	98.18	00.52	4	04.75	55.56
V-shaped	GNB	**99.82**	97.81	98.69	00.58	**3**	03.07	**66.67**
	KNN	99.27	97.81	98.61	00.41	5	03.72	44.44
	SVM	99.27	97.81	98.25	00.51	4	03.36	55.56
	DT	97.81	97.81	97.81	00.00	4	04.38	55.56
U-shaped	GNB	**99.82**	97.81	98.69	00.67	**3**	03.07	**66.67**
	KNN	99.27	97.81	98.47	00.54	**3**	03.58	**66.67**
	SVM	**99.82**	98.54	98.76	00.49	4	03.87	55.56
	DT	99.27	97.81	98.54	00.60	5	**05.11**	44.44
TVMS	GNB	99.27	97.81	98.47	00.41	**3**	02.85	**66.67**
	KNN	**99.82**	98.54	98.91	00.52	**3**	04.02	**66.67**
	SVM	99.27	97.81	98.18	00.52	4	03.29	55.56
	DT	99.27	97.81	98.39	00.58	**3**	04.96	**66.67**
Linear	GNB	99.27	97.81	98.61	00.41	5	02.99	44.44
	KNN	99.27	98.54	**98.98**	00.38	5	04.09	44.44
	SVM	99.27	98.54	**98.98**	00.38	4	04.09	55.56
	DT	99.27	97.81	98.32	00.58	5	04.89	44.44

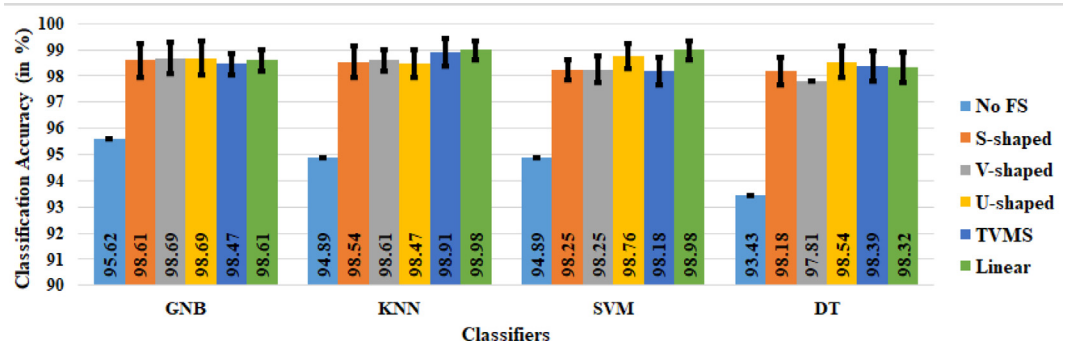


Fig. 5. Performance (average ± SD classification accuracy) of different BPSO setups for Breast Cancer prediction. Here, the term "No FS" indicates the performance of an individual classifier without using the BPSO-based feature selection.

case, increment in classification accuracy calculated by subtracting actual classification accuracy from obtained average classification accuracy using the corresponding BPSO setup, and reduced feature dimension calculated as $\frac{(d-d\prime)}{d} * 100$ (d, and $d\prime$ represent actual feature dimension and reduced feature dimension in the best case) respectively. From the results, we observe that the S-shaped TF with KNN and the TVMS with DT classifier provide the best accuracy among all BPSO sets with an accuracy of 96.72%. Whereas V-shaped TF with the SVM classifier performs best and S-shaped with the DT classifier performs worst in terms of average accuracy. Apart from these, comparing the results obtained without using BPSO (see Fig. 4) with the results obtained using BPSO with varying TFs, we have observed that the use of BPSO-based (with any TF) feature selection is always beneficial for the Heart disease

Table 4
Performance of BPSO with different TFs and classifiers for the prediction of CKD. The meaning of the shortened column terms used in this table are defined in Section 7.4.1. The boldfaced numbers indicate the best scores in terms of maximum and average accuracy.

TF	Classifier	Classification Accuracy (in %)				#SF	Gain (in %)	
		Max.	Min.	Avg.	SD	(best)	ICA	RFD
S-shaped	GNB	**100.00**	100.00	**100.00**	00.00	8	00.00	66.67
	KNN	**100.00**	96.77	99.54	00.17	9	18.29	62.50
	SVM	**100.00**	96.77	99.57	01.14	9	18.32	62.50
	DT	**100.00**	96.77	99.57	00.14	10	02.70	58.33
V-shaped	GNB	**100.00**	100.00	**100.00**	00.00	6	00.00	75.00
	KNN	**100.00**	100.00	**100.00**	00.00	7	15.75	70.83
	SVM	**100.00**	100.00	**100.00**	00.00	6	15.75	75.00
	DT	**100.00**	100.00	**100.00**	00.00	8	03.13	66.67
U-shaped	GNB	**100.00**	100.00	**100.00**	00.00	7	00.00	70.83
	KNN	**100.00**	100.00	**100.00**	00.00	7	**18.75**	70.83
	SVM	**100.00**	100.00	**100.00**	00.00	8	15.75	66.67
	DT	**100.00**	100.00	**100.00**	00.00	6	03.13	75.00
TVMS	GNB	**100.00**	100.00	**100.00**	00.00	**3**	00.00	**87.50**
	KNN	**100.00**	100.00	**100.00**	00.00	5	**18.75**	79.17
	SVM	**100.00**	100.00	**100.00**	00.00	4	15.75	83.33
	DT	**100.00**	100.00	**100.00**	00.00	6	03.13	75.00
Linear	GNB	**100.00**	100.00	**100.00**	00.00	8	00.00	66.67
	KNN	**100.00**	96.88	99.82	00.16	7	15.63	70.83
	SVM	**100.00**	96.88	99.82	00.12	7	18.57	70.83
	DT	**100.00**	93.75	99.69	00.30	6	02.82	75.00

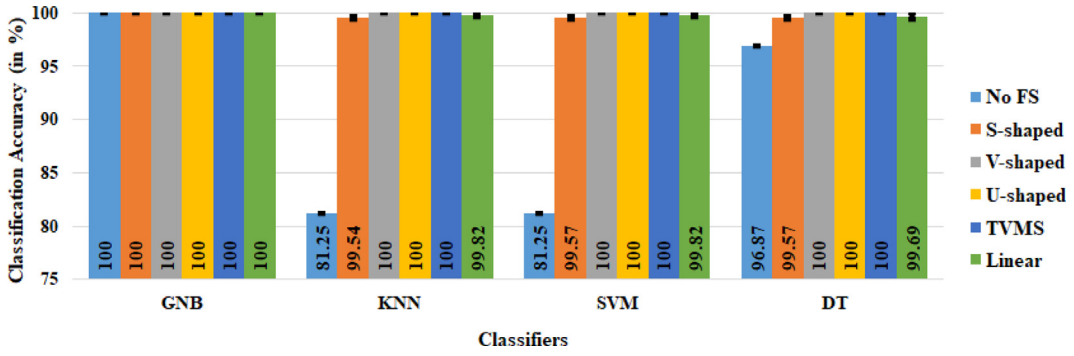


Fig. 6. Performance (average classification accuracy ± SD) of different BPSO setups for CKD prediction. Here, the term "No FS" indicates the performance of an individual classifier without using the BPSO-based feature selection.

prediction from diagnosis reports. From Fig. 4, it is also observed that the BPSO setup that includes the V-shaped TF and the SVM classifier provides the best average classification accuracy. Therefore, this setup could be used in the decision support system for Heart disease diagnosis.

7.4.2. Performance on the Breast Cancer prediction

The experimental results with different BPSO setups for Breast Cancer prediction are shown in Table 3. From these results, we can observe that V-shaped TF with GNB, U-shaped with GNB and SVM and TVMS with KNN classifiers provide maximum classification accuracy (99.82%) among all BPSO setups. However, in terms of average performance, Linear TF with KNN and SVM classifier performs best with 98.98% classification accuracy. In addition to these, while comparing the results obtained without using BPSO (see Fig. 5) with the results obtained with BPSO with varying TFs, we have observed that the use of BPSO-based feature selection (with any TF) is always beneficial for the prediction of this disease. Also, from Fig. 5, it is observed that the BPSO setup comprises V-shaped TF and GNB, U-shaped with KNN and SVM, and TVMS with KNN classifier provides the best average classification accuracy with minimal SD of the classification accuracy. Therefore, this setup can predict breast cancer using prediction based on diagnosis reports.

7.4.3. Results of CKD prediction

The experimental results of CKD prediction from the diagnosis report obtained under the present experimental setups are provided in Table 4. The results show that all setups (that is, regardless of the classifiers and TFs) produce 100% performance in the best case. However, in terms of average classification accuracy and SD of the classification accuracies over 20 independent runs, the V-shaped, U-shaped, and TVMS TFs mentioned in the present study are consistent with all classifiers and their SD is always 'zero' with 100% average classification accuracy (see Fig. 6). From Fig. 6 it is observed that all classifiers provide 100% average accuracy with 'zero' SD for V-shaped, U-shaped, and TVMS TFs. Comparing the results of Table 1 and Table 4, we can see that the accuracy of classifiers like KNN, SVM, and DT is increased. Therefore, these setups can be used to predict the disease while using the diagnostic attributes mentioned in the UCI CKD dataset.

7.4.4. Results of Diabetes prediction

The PIMA Indian dataset has been used for Diabetes disease prediction. The experimental results with varying BPSO setups in this

Table 5
Performance of BPSO with different TFs and classifiers for Diabetes prediction. Meaning of the shorten column terms used in this table are defined in Section 7.4.1. The boldfaced numbers indicate the best scores in terms of maximum and average accuracy.

TF	Classifier	Classification Accuracy (in %)				#SF	Gain (in %)	
		Max.	Min.	Avg.	SD	(best)	ICA	RFD
S-shaped	GNB	83.12	79.22	80.78	01.07	4	01.11	50.00
	KNN	80.52	76.62	77.86	01.28	4	11.63	50.00
	SVM	81.17	78.57	79.35	00.95	**3**	02.73	**62.50**
	DT	79.87	74.68	76.69	01.60	5	00.72	37.50
V-shaped	GNB	83.12	79.22	80.58	01.16	4	03.96	50.00
	KNN	81.17	77.92	79.68	00.97	4	**13.45**	50.00
	SVM	81.17	78.57	79.22	00.87	5	02.60	37.50
	DT	78.57	75.32	77.08	01.26	**3**	01.11	**62.50**
U-shaped	GNB	81.82	79.22	80.26	01.11	5	03.64	37.50
	KNN	81.82	77.27	79.48	01.76	**3**	13.25	**62.50**
	SVM	80.52	77.27	78.51	01.04	4	01.89	50.00
	DT	76.62	74.03	75.32	00.92	5	−0.65	37.50
TVMS	GNB	**84.41**	79.22	80.91	01.51	**3**	04.29	**62.50**
	KNN	81.17	77.27	78.77	01.23	**3**	12.54	**62.50**
	SVM	81.17	77.27	78.70	01.22	4	02.08	50.00
	DT	78.57	72.73	75.00	01.84	4	−0.97	50.00
Linear	GNB	82.47	80.52	**81.49**	00.70	4	04.87	50.00
	KNN	81.17	77.92	79.35	01.14	**3**	13.12	**62.50**
	SVM	80.52	76.62	78.12	01.40	4	01.50	50.00
	DT	79.22	75.32	76.62	01.33	5	00.65	37.50

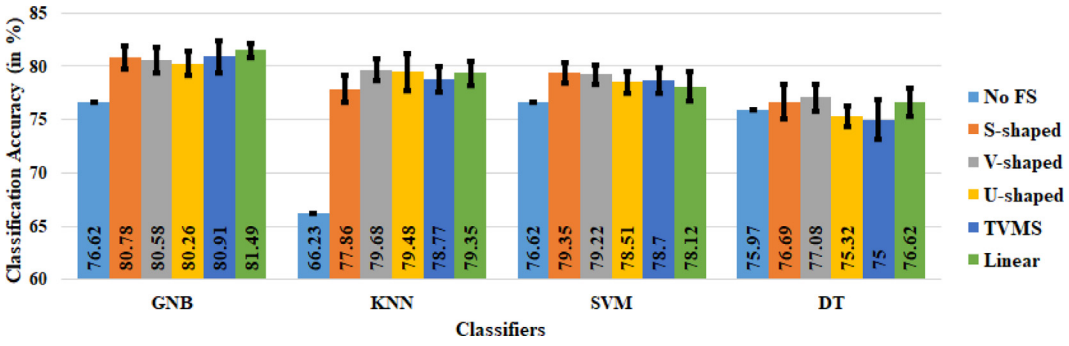


Fig. 7. Performance (average SD classification accuracy ±) of different BPSO setups for Diabetes prediction. Here, the term "No FS" indicates the performance of the individual classifier without using the BPSO-based feature selection.

dataset are shown in Table 5. From the results, we can see that TVMS TF with the GNB classifier performs the best in terms of maximum classification accuracy (84.41%) while Linear TF with the GNB classifier provides the best average classification accuracy (81.49%) with the lowest SD value. Therefore, this setup (i.e., Linear TF with GNB classifier) can be used to predict diabetes disease using diagnostic reports in the PIMA dataset. In addition to these, by comparing the results obtained without and with BPSO by varying TFs (see Fig. 7), we have observed that the use of BPSO-based feature selection (with any TF) is always beneficial for the prediction of this disease.

7.4.5. Overall performance analysis

The summary of the performance of BPSO with varying TFs and classifiers is recorded in Table 6. The results show that classification accuracies obtained in the best case for Heart disease, Breast Cancer, CKD, and Diabetes are 96.72%, 99.82%, 100.00%, and 84.41%, respectively. The TVMS TF provides the best results (in terms of maximum classification accuracy over 20 independent runs) for all four disease predictions (viz., Heart disease, Breast Cancer, CKD, and Diabetes) while V-shaped TF provides the best result for CKD and Breast Cancer, S-shaped TF provides the best result for Heart disease and CKD and U-shaped performs well in Breast cancer and CKD dataset. It is also observed that the SVM classifier's performance is poor compared to the GNB classifier when it accepts all the features present in the dataset (see Table 1). But this classifier helps to generate the best results for three diseases: Heart disease (refer to Table 2), Breast Cancer (refer to Table 3), and CKD (refer to Table 4) with V-shaped TF, Linear TF, and any of V-shaped, U-shaped, and TMVS respectively when considering the average classification accuracy over the 20 independent runs. These results indicate that feature selection using BPSO helps improve some classifiers' classification strength of some classifiers. This is because the BPSO-based feature selection helps to remove the irrelevant and redundant features from the feature set, and accordingly the classifier (here SVM) can generate better class-separating boundaries. As a result of this, we observe better performance for this classifier. If we compare BPSO setups (i.e., classifier + TF) in terms of the average accuracy obtained during the 20 independent runs, then V-shaped TF with SVM classifier performs the best for Heart disease prediction, Linear TF with KNN or SVM classifier performs the best for the Breast Cancer dataset, Linear TF with GNB classifier performs better in the Diabetes dataset and all classifiers (viz. GNB, KNN, SVM, and DT) with any of the TFs used provide 100% performs best for the prediction of CKD. Apart from these, results shown in Fig. 4, Fig. 5, Fig. 6, and Fig. 7 indicate that irrespective of BPSO setups feature selection using the BPSO algorithm is beneficial, as in each of the cases the prediction results improve or remain the same compared to the accuracy obtained using all features.

Table 6
Setups (i.e., combinations of classifier and TF) that perform well in comparison with the other setups. The results are shown for each disease in terms of the best case and average case performances

Disease	Best case performances			Average case performance		
	Classifier	TF	CA (in %)	Classifier	TF	CA (in %)
Heart disease	KNN DT	S-shaped TVMS	96.72	SVM	V-shaped	92.95
Breast Cancer	GNB KNN	V-shaped or U-shaped TVMS	99.82	SVM or KNN	Linear	98.98
CKD	All[+]	All[+]	100.00	GNB[*]	All[+]	100
Diabetes	GNB	TVMS	84.41	GNB	Linear	81.49

[+]Implies any classifier or TF.
[*]Indicates GNB classifier as it provides 100% accuracy irrespective of the TFs.

Table 7
Diagnostic attributes selected in the best case and appeared very frequently in independent experiments.

Disease	Selected Diagnosis Attributes
Heart disease	sex, exang, ca, and thal (in the best case) slope and oldpeak (very frequently)
Breast Cancer	Bare Nuclei, Normal Nucleoli, and Uniformity of cell size (in the best case) Uniformity of cell shape (very frequently)
CKD	al, sc, and rc (in the best case) su, pcc, bu, pcv, pc, ane, bp, dm, hemo, htn, and sg (very frequently)
Diabetes	Glucose, BMI, Pregnancies (in the best case) Age, and DiabetesPedigreeFunction (very frequently)

Apart from these, we can also analyze the performance of the said BPSO setups in terms of the number of selected features, performance gain in terms of improved classification performance, and the reduced feature dimension, which are shown in the under the column header "#SF", "ICA", and "RFD" respectively in the Table 2, Table 3, Table 4, and Table 5. It should be noted that the way these values are calculated is already discussed in Section 7.4.1. It could be observed that the best feature dimension reduction is obtained for many BPSO setups for Breast Cancer and Diabetes (6 times in each case) disease prediction while the same is achieved only once on the other two diseases once. From these tables, we observed that the minimum number of selected features or diagnostic attributes are 4, 3, 3, and 3 for Heart disease, Breast Cancer, CKD, and Diabetes respectively. It can also be observed from these tables that the selection of the least number of features does not imply the best performance; rather the classifier has the most impact on better performance while the TFs play important roles in selecting the least number of features. The TVMS TF selects the least number of features on all these datasets while the other four fail to achieve this for Heart disease and CKD prediction. We can also observe ICA value < 0 from Table 5, which proves that a bad choice of features might end up with reduced performance. Therefore, a feature selection technique should be chosen with caution. However, except for these two observations in Table 5, in all other cases, we have observed the ICA value is ≥ 0. Thus, from these observations we can safely comment that, in general, the use of a feature selection is beneficial not only to reduce the feature dimension, but also to improve the performance.

7.5. Selected diagnostic attributes

We have noted the diagnostic attributes that have been selected during different experiments and listed these attributes selected during the best case results, decided in terms of classification accuracy (see Table 6) and the attributes selected very frequently in Table 7. In this context, we would like to mention that we have considered a diagnostic attribute in a very frequently occurring set if it appears in at least 75% of the total number of experiments carried out for a specific disease. The total number of experiments carried out for a specific disease is 400 (that is, 20 (independent runs for a single setup) * 20 (5 TF variants and four classifiers)). Thus, if a diagnostic attribute is selected at least 300 times, then we have considered that this diagnosis attribute is the most frequently appearing attribute set. These results might help identify important diagnostic attributes for a specific disease, and an expert can draw some relation between an actual set of attributes and the selected attributes to improve the diagnosis process.

Table 8
Average execution time taken by different BPSO setups.

Classifier	Disease	Avg. time (in sec) for the different BPSO setups				
		S-shaped	V-shaped	U-shaped	TVMS	Linear
GNB	Heart disease	05.11	04.24	03.75	04.22	05.00
	Breast Cancer	07.23	07.03	05.90	05.15	06.63
	CKD	07.14	07.65	07.75	06.53	07.79
	Diabetes	11.22	11.13	09.43	08.55	10.15
KNN	Heart disease	17.58	13.74	14.30	14.41	06.70
	Breast Cancer	41.69	42.18	36.10	31.10	41.92
	CKD	13.96	14.39	13.58	13.63	84.76
	Diabetes	31.32	31.94	25.67	24.41	34.97
SVM	Heart disease	23.43	16.76	15.47	17.30	24.40
	Breast Cancer	26.52	28.09	26.55	25.11	27.20
	CKD	18.94	18.44	17.22	15.75	13.90
	Diabetes	86.09	91.96	80.86	65.46	85.55
DT	Heart disease	06.23	05.51	04.79	03.00	04.71
	Breast Cancer	07.09	06.30	05.16	03.88	05.79
	CKD	10.51	10.99	10.75	08.85	08.73
	Diabetes	11.47	09.00	06.89	06.95	09.12

Table 9
Test statistics (i.e., Q) values obtained using the Friedman statistical test over different independent runs of all BPSO setups.

Dataset	Heart Disease	Breast Cancer	CKD	Diabetes
Test statistics	15.9650	17.1864	13.9164	5.6457

7.6. Execution time

We also compared the performance of BPSO with different TFs in terms of execution time. Execution times on a machine with configurations - Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz, 512 GB SSD, and 16 GB RAM of these experiments are summarized in Table 8. This table lists the average time (in seconds) taken by 20 independent runs. These results show that most of the time the BPSO using the TVMS TF has converged to the best solution faster than its counterparts. It is also observed from the previous results that the BPSO with the TVMS TF has produced the best results in most cases (specifically see Table 6) when the best accuracy obtained is considered. Even the variation in the classification accuracies (recorded using SD of classification accuracies obtained over 20 independent runs (see Fig. 4, Fig. 5, Fig. 6 and Fig. 7)) is minimal in most scenarios. Apart from these, the theoretical explanation of all TFs used here (see Section 5) indicates that the TVMS TF is comparatively stronger than others. Therefore, considering all these facts, we recommend for using the TVMS TF in the BPSO-based diagnostic attribute selection process for disease prediction-based systems on diagnosis reports.

7.7. Statistical significance test

It is well known that the statistical significance test plays an important role in experiments having the impact of randomness (Mondal et al., 2022; Pramanik et al., 2022). In this study, we have executed all the BPSO setups (i.e., TF+ classifier) 20 times on each of the datasets to test the impact of the inbuilt randomness present in BPSO on the end outcome. Now the question arises whether, for a specific dataset, there exists any statistically significant difference among the outcomes from the BPSO setups (i.e., independent samples) over different runs (dependent samples). It would help us to decide whether the results are statistically significant or not. Such statistical significance tests can be performed using the analysis variance (ANOVA) with repeated measurements (or trials) test, which is a parametric test (Shaw et al., 2021). However, the observed data (classification accuracies) might not be normally distributed. Thus, we opt for the Friedman statistical test (Friedman, 1937), which is non-parametric in nature, and similar to the ANOVA with repeated measurement tests. When performing this test, we have considered each independent run as a treatment (or dependent samples) and each BPSO setup as a test case (i.e., independent samples or random experiments). It should be noted that 20 (5 TFs * 4 Classifiers) different BPSO setups are considered here. Therefore, the number of test cases (or independent variables) is 20. The degree of freedom (df)= number of dependent variables - 1, which is 19. Let the null hypothesis be "There is no significant difference among the performances of the BPSO setups over different runs for a specific dataset" and the corresponding alternative hypothesis is "There exist a significant difference among the performances of the BPSO setups over different runs for a specific dataset". We can reject the null hypothesis and accept the alternative hypothesis at $\alpha\%$ significance level if all the test statistic values (say, Q) are greater than χ^2_{df}-value at significance level $\alpha\%$ (we consider 5% significance level) i.e., in our case, χ^2_{19} with 5% significance level ($= 30.144$) $< Q$. The estimated Q-values for different datasets are recorded in Table 9. From the estimated Q-values recorded in Table 9, we can not reject the null hypothesis at a 5% significance level for any of the datasets used here. Therefore, we can safely comment that the performance of the different BPSO setups in different runs for all datasets is statistically significant at the significance level 5%.

7.8. Comparison with state-of-the-art methods

In this section, we have compared our findings with some state-of-the-art methods. Comparisons are made in terms of classification accuracy and the number of diagnostic attributes selected. Table 10, Table 11, Table 12, and Table 13 show the comparisons for prediction of Heart disease, Breast Cancer, CKD, and Diabetes, respectively. It is to be noted that here we have directly mentioned the results as reported in the state-of-the-art methods. Hence, all the results might not be directly comparable, since these techniques followed different experimental setups and dataset splits in some cases. However, they provide an overall idea about the performance of the current findings (i.e., the best BPSO setups) compared to the existing methods. These comparative results show that the models that perform best in the current study for each disease are comparable to the state-of-the-art methods considered here for comparison.

8. Additional experiments

Apart from these, we have also applied the model to three microarray datasets to observe how effective the feature selection technique is on a high-dimensional dataset. For our experiments, we have considered three standard datasets from the Biolab Repository namely, Leukemia[6], Diffuse large B cell lymphoma (DLBCL)[7], and Prostate cancer[8]. In the case of the Leukemia dataset, there are 72 instances with 47 acute lymphoblastic leukemia (ALL) and 25 acute myeloblastic leukemia (AML) with 5147 features, while the DLBCL dataset has 7070 features with 77 instances. Of these 77 instances, 58 instances belong to DLBCL, and 19 are Non-DLBCL. The prostate dataset has 12532 features, making this data set the largest in terms of the many features considered here for experiments. This dataset has 102 instances where 50 are considered normal cases and 52 have a Prostate tumor. A brief description of the datasets is provided in Table 14.

For microarray datasets, the feature count is very high. Hence, there may be a large number of correlated features. Therefore, to remove the correlated features at the initial stage, we have used a filter method called Xvariance (XV) (Alirezanejad et al., 2020). XV is used to rank the features present in a dataset. In doing so, we have taken the top 300-ranked features obtained after applying this filter method to test the performance of BPSO with varying TFs. Experiments are carried out with a hold-out test data approach (detailed in the Section 7.3). We have chosen the GNB as a classifier for all experiments. The experimental results are shown in Table 16, Table 16, and Table 17 for the Leukemia dataset, DLBCL dataset, and Prostate cancer dataset, respectively. It is worth mentioning that we have executed each experimental setup for 20 independent runs and recorded the results in terms of maximum

[6] https://file.biolab.si/biolab/supp/bi-cancer/projections/info/leukemia.html.
[7] https://file.biolab.si/biolab/supp/bi-cancer/projections/info/DLBCL.html.
[8] https://file.biolab.si/biolab/supp/bi-cancer/projections/info/prostata.html.

Table 10
Performance comparison of the best BPSO setup with some state-of-the-art methods for Heart disease prediction. Here "#SF", and "CA" indicate the number of selected features or diagnostic attributes and classification accuracy respectively.

Method	Feature Selector	Classifier	#SF	CA (in %)
Polat and Güneş (2009)	RBF with F-score	LS-SVM	6	83.70
Khemphila and Boonjing (2011)	Information Gain	NN	8	80.99
Peter and Somasundaram (2012)	CBFS with Bayes Theorem	KNN	3	85.55
Vivekanandan and Iyengar (2017)	Differential Evolution	NN	9	83.00
Gokulnath and Shantharajah (2019)	GA	SVM	7	88.34
Current Study	PSO with S-shaped TF PSO with TVMS TF	KNN DT	4	**96.72**

Table 11
Performance comparison of the best BPSO setup with some state-of-the-art methods for Breast Cancer prediction. Here "#SF", and "CA" indicate the number of selected features or diagnostic attributes and classification accuracy respectively.

Method	Feature Selector	Classifier	#SF	CA (in %)
Lavanya and Rani (2011)	Linear Forward Selection	CART	9	94.84
Alyami et al. (2017)	CBFS	SVM	5	96.57
Dhanya et al. (2019)	CBFS	RF	8	96.42
Hardani and Nugroho (2020)	RST with CBFS	SMO	9	94.66
Kumar and Singh (2021)	Enhanced GWO	SVM	6	98.24
Huang et al. (2017)	GA	Boosting ensemble with SVM	9	99.41
Present Study	PSO with V-shaped TF or PSO with U-Shaped TF or PSO with TVMS TF	GNB GNB KNN	3	**99.82**

Table 12
Performance comparison of the best BPSO setup with some state-of-the-art methods for CKD prediction. Here "#SF", and "CA" indicate the number of selected features or diagnostic attributes and classification accuracy, respectively.

Method	Feature Selector	Classifier	#SF	CA (in %)
Wibawa et al. (2017)	CBFS	AdaBoost	17	98.10
Shrivas et al. (2018)	UBFST	Ensemble of RF, SVM and CART	14	99.25
Lestari et al. (2020)	C4.5 using IG	Adaboost	12	98.33
Kadhum et al. (2021)	ELM-GWFP	ELM	10	98.10
Arulanthu and Perumal (2021)	Simulated Annealing	LR with RMSprop optimizer	13	98.25
Chetty et al. (2015)	best-fit	NB	12	99.00
Chetty et al. (2015)	best-fit	KNN	12	100.00
Senan et al. (2021)	RFE	SVM	-	96.67
Senan et al. (2021)	RFE	KNN	-	98.33
Senan et al. (2021)	RFE	DT	-	99.17
Senan et al. (2021)	RFE	RF	-	100.00
Present Study	PSO with U-shaped or V-shaped or TVMS or Linear TF	GNB or SVM or KNN or DT	3	**100.00**

Table 13
Performance comparison of the best BPSO setup with some state-of-the-art methods for Diabetes prediction. Here "#SF", and "CA" indicate the number of selected features or diagnostic attributes and classification accuracy, respectively.

Method	Feature Selector	Classifier	#SF	CA (in %)
Balakrishnan and Narayanaswamy (2009)	Symmetric Uncertainty with FCBF	SVM-RBF	4	77.99
Choubey et al. (2017)	GA	NB	4	78.69
Kewat et al. (2020)	GA with KNN	SVM	3	76.69
de Lima et al. (2020)	Modified F-score	Twin-bounded SVM	4	77.21
Rathi and Acharjya (2021)	GA	NB	7	76.30
Present Study	PSO with TVMS TF	GNB	3	**84.41**

Table 14
Details of three microarray datasets used in this work

Dataset	# Features	# Instances	Class Distribution
Leukemia	5147	72	ALL: 47 AML: 25
DLBCL	7070	77	Non-DLBCL: 19 DLBCL: 58
Prostate Cancer	12,532	102	Normal tissue: 50 Prostate tumor: 52

Table 15
Results of BPSO in combination with various TFs and GNB classifier on the Leukemia dataset. The mentioned number of selected features is for maximum classification accuracy. The meaning of the shortened column terms used in this table is defined in Section 7.4.1.

TF	Classification Accuracy (in %)				Avg. Time (in sec.)	#SF
	Max.	Avg.	Min.	SD		
S-shaped	100	94.00	93.33	2.11	20.15	107
V-shaped	100	98.33	93.33	1.89	12.36	049
U-shaped	100	99.58	93.33	1.67	15.12	009
TVMS	100	97.08	93.33	3.42	13.28	013
Linear	100	98.00	93.33	3.22	13.81	126

Table 16
Results of BPSO in combination with various TFs and GNB classifier on DLBCL dataset. The mentioned number of selected features is for maximum classification accuracy. The meaning of the shortened column terms used in this table is defined in Section 7.4.1.

TF	Classification Accuracy (in %)				Avg. Time (in sec.)	#SF
	Max.	Avg.	Min.	SD		
S-shaped	100	93.62	89.50	3.42	13.34	114
V-shaped	100	95.00	93.75	2.64	15.47	076
U-shaped	100	98.96	93.75	2.43	21.06	012
TVMS	100	98.44	93.75	2.83	12.94	013
Linear	100	93.12	87.50	3.55	13.54	126

Table 17
Results of BPSO in combination with various TFs and GNB classifier on Prostate cancer dataset. The mentioned number of selected features is for maximum classification accuracy. The meaning of the shortened column terms used in this table is defined in Section 7.4.1.

TF	Classification Accuracy (in %)				Avg. Time (in sec.)	#SF
	Max.	Avg.	Min.	SD		
S-shaped	85.71	74.40	66.67	6.55	16.94	144
V-shaped	76.19	72.62	66.67	3.37	13.49	056
U-shaped	95.24	88.14	81.19	5.53	14.43	006
TVMS	76.19	70.83	66.67	3.05	12.59	014
Linear	80.95	74.40	66.67	4.36	13.36	119

(Max. in Tables), minimum (Min. in Tables), average (Avg. in Tables), and SD of the obtained classification accuracy.

If we look at Table 15 and Table 16, one point is observable that although the maximum classification accuracy is 100% for all five TFs, S-shaped, V-shaped, and Linear TFs produce 100% classification accuracy with a much higher number of selected features than the other two. The same can be seen in the case of the Prostate dataset (refer to Table 17), the U-shaped and TVMS TFs select a very lower number of features than the other three. Although the use of U-shaped TF helps achieve maximum accuracy of 95.24%, the mean accuracy produced by the TF is also much higher than other TFs. From this set of experiments, we can safely comment that the U-shaped TF with GNB as a classifier produces better results on the microarray datasets used here than the other four TFs in terms of average classification accuracy and the number of selected features.

9. Advantages and limitations of the present study

For an unbiased view, in this section, we discuss the advantages and limitations of the present study.

The advantages are as follows:

- The present study shows that TFs play an important role while transforming an optimization algorithm from the continuous domain to the discrete domain.
- This study ensures that the underlying classifier of a wrapper-based feature selection has a significant impact on the end outcome.
- This study is able to find possibly the best BPSO setups (a combination of a TF and a classifier) for each disease under consideration.
- The obtained results are also proven to be statistically significant.
- The present study finds that TVMS TF with GNB as a classifier performs better than other BPSO setups.
- Comparative study shows the superiority of the best findings over many state-of-the-art methods in terms of classification accuracy and feature reduction capability.
- The best BPSO setups also perform well on large-dimensional feature vectors.

The limitations are as follows:

- The present study only investigates the effect of different TFs by considering only PSO, and hence these findings may differ if we use different meta-heuristic algorithms.
- In this study only the popularly used TFs are investigated, but not their variants.
- Only four diagnostic attribute-based datasets are considered here. However, we have not considered any image-based disease datasets for feature selection.

- The data with missing values are discarded during the data cleaning phase, and thus the behavior of these studies many vary slightly if we apply some missing value prediction algorithms, and consider all attributes' values.

10. Conclusion

In this study, we have investigated the performance of 20 different BPSO feature selection setups to predict four commonly found chronic diseases: Heart disease, Breast Cancer, CKD, and Diabetes. The different BPSO setups are constructed by varying the TFs and classifiers. Five major TF families - Linear, S-shaped, V-shaped, U-shaped, and TVMS - have been considered here. These setups are employed to select optimal diagnosis attributes that ultimately help identify the subject having a certain disease from the patient's data. Four popular classifiers: GNB, DT, SVM, and KNN are used for classification tasks. All BPSO setups are evaluated on publicly available datasets related to the chronic diseases considered here. After performing an exhaustive set of experiments, we have recommended a suitable BPSO setup (i.e., a combination of TF and classifier) for each of the disease datasets that performs better than many state-of-the-art techniques. We have obtained the maximum classification accuracy of 96.72%, 99.82%, 100%, and 84.41% for Heart disease, Breast Cancer, CKD, and Diabetes, respectively. In summary, we have tried to suggest a suitable TF for selecting features using BPSO based on the type of data and classification method used here.

Although the performance of the best BPSO setup, found in this study, is on par with the state-of-the-art methods, there is room for improvement. This study has applied various TFs to the classical PSO of the continuous domain and mapped it to a binary domain. However, there are many improved versions of PSO (improvement in terms of better initialization strategies, use of local search algorithms for improving the agents' performance, etc.) in the literature that can be considered in the future. Moreover, in the future, more such meta-heuristic algorithms can be analyzed to draw a better conclusive statement. The best BPSO setup recognizes Diabetes disease with an accuracy of 84.41%, which is far from its use in the practical field considering its deadly consequences. A similar comment can be made for Heart disease and Breast Cancer datasets, where the best BPSO setup of the present study correctly recognizes 96.72% and 99.82% of the test samples, respectively. Therefore, we will consider more sophisticated models that can improve these results. Apart from these, we have removed the data with missing value entries while preprocessing the datasets. This may result in improved classification accuracy (for example, 100% accuracy for CKD datasets) in a few cases as we have used partial samples from the entire dataset. Hence, some missing value imputation methods can be used in the future to use all samples. Last but not least, in the future, more datasets can be used for experimentation in order to understand the generalization ability of the BPSO setups considered here.

Availability of data and materials

All the data we use here are publicly available.

Consent for Publication

Not applicable.

Funding

This work was supported by the Russian Science Foundation (Project Number 22-41-04409).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aličković, E., Subasi, A., 2017. Breast cancer diagnosis using GA feature selection and Rotation Forest. Neural Comput. Appl. 28, 753–763.
Alirezanejad, M., Enayatifar, R., Motameni, H., Nematzadeh, H., 2020. Heuristic filter feature selection methods for medical datasets. Genomics 112, 1173–1181.
Almansour, N.A., Syed, H.F., Khayat, N.R., Altheeb, R.K., Juri, R.E., Alhiyafi, J., Alrashed, S., Olatunji, S.O., 2019. Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study. Comput. Biol. Med. 109, 101–111.
Alyami, R., Alhajjaj, J., Alnajrani, B., Elaalami, I., Alqahtani, A., Aldhafferi, N., Owolabi, T.O., Olatunji, S.O., 2017. Investigating the effect of correlation based feature selection on breast cancer diagnosis using artificial neural network and support vector machines. In: 2017 International Conference on Informatics, Health & Technology (ICIHT). IEEE, pp. 1–7.
Amirgaliyev, Y., Shamiluulu, S., Serek, A., 2018. Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods. In: 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT). IEEE, pp. 1–4.
Arulanthu, P., Perumal, E., 2021. Intelligent Chronic Kidney Disease Diagnosis System using Cloud Centric Optimal Feature Subset Selection with Novel Data Classification Model.
Balakrishnan, S., Narayanaswamy, R., 2009. Feature selection using FCBF in type II diabetes databases. Int. J. Comput. Internet Manage. 17, 50–58.
Banerjee, D., Chatterjee, B., Bhowal, P., Bhattacharyya, T., Malakar, S., Sarkar, R., 2021. A new wrapper feature selection method for language-invariant offline signature verification. Expert Syst. Appl. 186, 115756.
Begum, S., Chakraborty, D., Sarkar, R., 2015. Data classification using feature selection and kNN machine learning approach. In: 2015 International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, pp. 811–814.
Begum, S., Sarkar, R., Chakraborty, D., Sen, S., Maulik, U., 2021. Application of active learning in DNA microarray data for cancerous gene identification. Expert Syst. Appl. 177, 114914.
Beheshti, Z., 2020. A time-varying mirrored S-shaped transfer function for binary particle swarm optimization. Inf. Sci. 512, 1503–1542.
Chakraborty, A., Ghosh, K.K., De, R., Cuevas, E., Sarkar, R., 2021. Learning automata based particle swarm optimization for solving class imbalance problem. Appl. Soft Comput. 113, 107959.
Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. Comput. Electr. Eng. 40, 16–28.
Chatterjee, A., Malakar, S., Sarkar, R., Nasipuri, M., 2018. Handwritten Digit Recognition using DAISY Descriptor: A Study. In: 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT). IEEE, pp. 1–4.
Chaturvedi, K.T., Pandit, M., Srivastava, L., 2009. Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. Int. J. Electrical Power Energy Syst. 31, 249–257.
Chetty, N., Vaisla, K.S., Sudarsan, S.D., 2015. Role of attributes selection in classification of Chronic Kidney Disease patients. In: 2015 International Conference on Computing, Communication and Security (ICCCS). IEEE, pp. 1–6.
Choubey, D.K., Paul, S., Kumar, S., Kumar, S., 2017. Classification of PIMA Indian diabetes dataset using naive Bayes with genetic algorithm as an attribute selection. In: Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016), pp. 451–455.
Colagiuri, S., Guariguata, L., Motala, A., Ogurtsova, K., et al., 2019. Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, 9 (th) edition. Diabetes Res. Clin. Pract 157, 107843.
Daliri, M.R., 2012. Feature selection using binary particle swarm optimization and support vector machines for medical diagnosis. Biomed. Eng./Biomedizinische Technik 57, 395–402.
Das, S., Chatterjee, A., Dey, S., Saha, S., Malakar, S., 2023. Breast cancer detection from histology images using deep feature selection. In: Proceedings of International Conference on Frontiers in Computing and Systems. Springer, pp. 323–330.
Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., Guppy, K.H., Lee, S., Froelicher, V., 1989. International application of a new probability algorithm for the diagnosis of coronary artery disease. Am. J. Cardiol. 64, 304–310.
Dey, C., Bose, R., Ghosh, K.K., Malakar, S., Sarkar, R., 2021. LAGOA: Learning automata based grasshopper optimization algorithm for feature selection in disease datasets. J. Ambient Intell. Humanized Comput., 1–20
Dey, S., Roychoudhury, R., Malakar, S., Sarkar, R., 2022. An optimized fuzzy ensemble of convolutional neural networks for detecting tuberculosis from Chest X-ray images. Appl. Soft Comput., 108094

Dhanya, R., Paul, I.R., Akula, S.S., Sivakumar, M., Nair, J.J., 2019. A comparative study for breast cancer prediction using machine learning and feature selection. In: 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, pp. 1049–1055.
Elbedwehy, M.N., Zawbaa, H.M., Ghali, N., Hassanien, A.E., 2012. Detection of heart disease using binary particle swarm optimization. In: 2012 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, pp. 177–182.
Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. 32, 675–701.
Ghosh, K.K., Begum, S., Sardar, A., Adhikary, S., Ghosh, M., Kumar, M., Sarkar, R., 2021. Theoretical and empirical analysis of filter ranking methods: Experimental study on benchmark DNA microarray data. Expert Syst. Appl. 169, 114485.
Ghosh, M., Guha, R., Alam, I., Lohariwal, P., Jalan, D., Sarkar, R., 2020. Binary genetic swarm optimization: a combination of GA and PSO for feature selection. J. Intell. Syst. 29, 1598–1610.
Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R., Nasipuri, M., 2017. Memetic algorithm based feature selection for handwritten city name recognition. In: International Conference on Computational Intelligence, Communications, and Business Analytics. Springer, pp. 599–613.
Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R., Nasipuri, M., 2019. Feature selection for handwritten word recognition using memetic algorithm. In: Advances in Intelligent Computing. Springer, pp. 103–124.
Gokulnath, C.B., Shantharajah, S., 2019. An optimized feature selection based on genetic approach and support vector machine for heart disease. Cluster Comput. 22, 14777–14787.
Guha, R., Ghosh, M., Mutsuddi, S., Sarkar, R., Mirjalili, S., 2020. Embedded chaotic whale survival algorithm for filter–wrapper feature selection. Soft. Comput. 24, 12821–12843.
Gunasundari, S., Janakiraman, S., Meenambal, S., 2018. Multiswarm heterogeneous binary PSO using win-win approach for improved feature selection in liver and kidney disease diagnosis. Comput. Med. Imaging Graph. 70, 135–154.
Guo, S.-S., Wang, J.-S., Guo, M.-W., 2020. Z-shaped transfer functions for binary particle Swarm optimization algorithm. Comput. Intell. Neurosci. 2020.
Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Machine Learn. Res. 3, 1157–1182.
Hardani, D., Nugroho, H., 2020. Feature selection using rough set theory algorithm for breast cancer diagnosis. IOP Conference Series: Materials Science and Engineering, vol. 771, no. 1. IOP Publishing, p. 012017.
Heppner, F., 1990. A Stochastic Nonlinear Model for Coordinate Bird Flocks. The ubiquity of chaos.
Herliana, A., Arifin, T., Susanti, S., Hikmah, A.B., 2018. Feature selection of diabetic retinopathy disease using particle swarm optimization and neural network. In: In 2018 6th International Conference on Cyber and IT Service Management (CITSM). IEEE, pp. 1–4.
Huang, M.-W., Chen, C.-W., Lin, W.-C., Ke, S.-W., Tsai, C.-F., 2017. SVM and SVM ensembles in breast cancer prediction. PloS One 12, e0161501.
Islam, M.J., Li, X., Mei, Y., 2017. A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. Appl. Soft Comput. 59, 182–196.
Jain, I., Jain, V.K., Jain, R., 2018. Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification. Appl. Soft Comput. 62, 203–215.
Jain, M., Saihjpal, V., Singh, N., Singh, S.B., 2022. An overview of variants and advancements of pso algorithm. Appl. Sci. 12, 8392.
Kadhum, M., Manaseer, S., Dalhoum, A., et al., 2021. Evaluation feature selection technique on classification by using evolutionary ELM wrapper method with features priorities. J. Adv. Infor. Technol. 12.
Kaur, S., Kumar, Y., Koul, A., Kumar Kamboj, S., 2023. A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: open issues and challenges. Arch. Comput. Methods Eng. 30, 1863–1895.
Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, pp. 1942–1948.
Kennedy, J., Eberhart, R.C., 1997. A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, IEEE, pp. 4104–4108.
Kewat, A., Srivastava, P., Kumhar, D., 2020. Performance evaluation of wrapper-based feature selection techniques for medical datasets. In: Advances in Computing and Intelligent Systems. Springer, pp. 619–633.
Khemphila, A., Boonjing, V., 2011. Heart disease classification using neural network and feature selection. In: 2011 21st International Conference on Systems Engineering. IEEE, pp. 406–409.
Kumar, S., Singh, M., 2021. Breast cancer detection based on feature selection using enhanced grey wolf optimizer and support vector machine algorithms. Vietnam J. Comput. Sci. 8, 177–197.
Kumar, V., Minz, S., 2014. Feature Selection: A literature Review. SmartCR 4, 211–229.
Lavanya, D., Rani, D.K.U., 2011. Analysis of feature selection with classification: Breast cancer datasets. Indian J. Comput. Sci. Eng. (IJCSE) 2, 756–763.
Lee, S., Soak, S., Oh, S., Pedrycz, W., Jeon, M., 2008. Modified binary particle swarm optimization. Prog. Nat. Sci. 18, 1161–1166.
Lestari, A. et al., 2020. Increasing Accuracy of C4. 5 Algorithm using information gain ratio and adaboost for classification of chronic kidney disease. J. Soft Comput. Expl. 1, 32–38.
Li, A.-D., Xue, B., Zhang, M., 2021. Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. Appl. Soft Comput., 107302
de Lima, M.D., e Lima, J. d. O.R., Barbosa, R.M., 2020. Medical data set classification using a new feature selection algorithm combined with twin-bounded support vector machine. Med. Biolog. Eng. Comput. 58, 519–528.
Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowledge Data Eng. 17, 491–502.
Liu, J., Yang, R., Sun, S., 2011. The analysis of binary particle swarm optimization. J. Nanjing Univ. (Nat. Sci.) 47, 504–514.
Malakar, S., Ghosh, M., Bhowmik, S., Sarkar, R., Nasipuri, M., 2020a. A GA based hierarchical feature selection approach for handwritten word recognition. Neural Comput. Appl. 32, 2533–2552.
Malakar, S., Ghosh, M., Chaterjee, A., Bhowmik, S., Sarkar, R., 2020b. Offline music symbol recognition using Daisy feature and quantum Grey wolf optimization based feature selection. Multimedia Tools Appl. 79, 32011–32036.
Malakar, S., Roy, S.D., Das, S., Sen, S., Velásquez, J.D., Sarkar, R., 2022. Computer based diagnosis of some chronic diseases: a medical journey of the last two decades. Arch. Comput. Methods Eng., 1–43
Mandal, M., Ghosh, D., Acharya, S., Saha, N., Sarkar, R., 2022. MIRFCS: an ensemble of filter methods for classification of disease data. In: Computational Intelligence in Pattern Recognition. Springer, pp. 205–217.
Mandal, M., Singh, P.K., Ijaz, M.F., Shafi, J., Sarkar, R., 2021. A tri-stage wrapper-filter feature selection framework for disease classification. Sensors 21, 5571.
Mirjalili, S., Lewis, A., 2013. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evol. Comput. 9, 1–14.
Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.
Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., Noman, N., 2020. A Novel U-shaped transfer function for binary particle swarm optimisation. Soft Computing for Problem Solving, vol. 1138. Springer, pp. 241–259.
Mohan, S., Thirumalai, C., Srivastava, G., 2019. Effective heart disease prediction using hybrid machine learning techniques. IEEE Access 7, 81542–81554.
Mohiuddin, S., Sheikh, K.H., Malakar, S., Velásquez, J.D., Sarkar, R., 2023. A hierarchical feature selection strategy for deepfake video detection. Neural Comput. Appl. 35, 9363–9380.
Mondal, R., Malakar, S., Barney Smith, E.H., Sarkar, R., 2022. Handwritten english word recognition using a deep learning based object detection architecture. Multimedia Tools Appl., 1–26
Mukhopadhyay, S., Hossain, S., Malakar, S., Cuevas, E., Sarkar, R., 2022. Image contrast improvement through a metaheuristic scheme. Soft Comput., 1–20
Nadimi-Shahraki, M.H., Banaie-Dezfouli, M., Zamani, H., Taghian, S., Mirjalili, S., 2021a. B-MFO: a binary moth-flame optimization for feature selection from medical datasets. Computers 10, 136.
Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S., 2021b. An improved grey wolf optimizer for solving engineering problems. Expert Syst. Appl. 166, 113917.
Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S., Abualigah, L., 2022. Binary aquila optimizer for selecting effective features from medical data: A COVID-19 case study. Mathematics 10, 1929.
Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S., Faris, H., 2020. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. Appl. Soft Comput. 97, 106761.
Nguyen, B.H., Xue, B., Andreae, P., Zhang, M., 2021. A new binary particle swarm optimization approach: momentum and dynamic balance between exploration and exploitation. IEEE Trans. Cybernet. 51 (2), 589–603.
Nssibi, M., Manita, G., Korbaa, O., 2023. Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. Comput. Sci. Rev. 49, 100559.
Peter, T.J., Somasundaram, K., 2012. Study and development of novel feature selection framework for heart disease prediction. Int. J. Sci. Res. Publ. 2, 1–7.
Polat, H., Mehr, H.D., Cetin, A., 2017. Diagnosis of chronic kidney disease based on support vector machine by feature selection methods. J. Medical Syst. 41, 55.
Polat, K., Güneş, S., 2009. A new feature selection method on classification of medical datasets: Kernel F-score feature selection. Expert Syst. Appl. 36, 10367–10373.
Pramanik, R., Dey, S., Malakar, S., Mirjalili, S., Sarkar, R., 2022. Topsis aided ensemble of cnn models for screening covid-19 in chest X-ray images. Sci. Rep. 12, 15409.
Rajamohana, S., Umamaheswari, K., 2018. Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection. Comput. Electr. Eng. 67, 497–508.
Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2010. BGSA: binary gravitational search algorithm. Nat. Comput. 9, 727–745.
Rathi, R., Acharjya, D., 2021. A comparative study of genetic algorithm and neural network computing techniques over feature selection. Advances in Distributed Computing and Machine Learning, vol. 127. Springer, pp. 491–500.

Reynolds, C.W., 1987. Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 25–34.

Rubini, L., Soundarapandian, P., Eswaran, P., 2015. Chronic Kidney Disease.

Sarkar, S., Ghosh, M., Chatterjee, A., Malakar, S., Sarkar, R., 2018. An advanced particle swarm optimization based feature selection method for tri-script handwritten digit recognition. In: International Conference on Computational Intelligence, Communications, and Business Analytics. Springer, pp. 82–94.

Senan, E.M., Al-Adhaileh, M.H., Alsaade, F.W., Aldhyani, T.H., Alqarni, A.A., Alsharif, N., Uddin, M.I., Alahmadi, A.H., Jadhav, M.E., Alzahrani, M.Y., 2021. Diagnosis of chronic kidney disease using effective classification algorithms and recursive feature elimination techniques. J. Healthcare Eng. 2021.

Shaw, S.S., Ahmed, S., Malakar, S., Garcia-Hernandez, L., Abraham, A., Sarkar, R., 2021. Hybridization of ring theory-based evolutionary algorithm and particle swarm optimization to solve class imbalance problem. Complex Intell. Syst. 7, 2069–2091.

Shen, Q., Jiang, J.-H., Jiao, C.-X., Shen, G.-L., Yu, R.-Q., 2004. Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists. Eur. J. Pharm. Sci. 22, 145–152.

Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360). IEEE, pp. 69–73.

Shrivas, A., Sahu, S.K., Hota, H., 2018. Classification of chronic kidney disease with proposed union based feature selection technique. In: Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), pp. 26–27.

Sisodia, D., Sisodia, D.S., 2018. Prediction of diabetes using classification algorithms. Proc. Comput. Sci. 132, 1578–1585.

Smith, J.W., Everhart, J.E., Dickson, W., Knowler, W.C., Johannes, R.S., 1988. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Annual Symposium on Computer Application in Medical Care, American Medical Informatics Association, p. 261.

Street, W.N., Wolberg, W.H., Mangasarian, O.L., 1993. Nuclear feature extraction for breast tumor diagnosis. In: Biomedical Image Processing and Biomedical Visualization, SPIE volume 1905, pp. 861–870.

Sung, H., Ferlay, J., Siegel, R.L., Laversanne, M., Soerjomataram, I., Jemal, A., Bray, F., 2021. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer J. Clinicians.

Taghian, S., Nadimi-Shahraki, M.H., Zamani, H., 2018. Comparative analysis of transfer function-based binary Metaheuristic algorithms for feature selection. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP). IEEE, pp. 1–6.

Virani, S.S., Alonso, A., Aparicio, H.J., Benjamin, E.J., Bittencourt, M.S., Callaway, C.W., Carson, A.P., Chamberlain, A.M., Cheng, S., Delling, F.N. et al., 2021. Heart disease and stroke statistics—2021 update: a report from the American Heart Association. Circulation, pp. CIR-0000000000000950.

Vivekanandan, T., Iyengar, N.C.S.N., 2017. Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. Comput. Biol. Med. 90, 125–136.

Wang, L., Wang, X., Fu, J., Zhen, L., 2008. A novel probability binary particle swarm optimization algorithm and its application. J. Softw.

Wei, J., Zhang, R., Yu, Z., Hu, R., Tang, J., Gui, C., Yuan, Y., 2017. A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection. Appl. Soft Comput. 58, 176–192.

Wibawa, M.S., Maysanjaya, I.M.D., Putra, I.M.A.W., 2017. Boosted classifier and features selection for enhancing chronic kidney disease diagnose. In: 2017 5th International Conference on Cyber and IT Service Management (CITSM). IEEE, pp. 1–6.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1, 67–82.

Xiong, B., Li, Y., Huang, M., Shi, W., Du, M., Yang, Y., 2019. Feature selection of input variables for intelligence joint moment prediction based on binary particle Swarm optimization. IEEE Access 7, 182289–182295.

Yaghoubzadeh, R., Kamel, S.R., Barzgar, H., Moshajeri San'ati, B., 2021. The use of the binary bat algorithm in improving the accuracy of breast cancer diagnosis. Multidiscip. Cancer Investig. 5, 1–7.