Expert Systems With Applications 195 (2022) 116550

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Boolean Particle Swarm Optimization with various Evolutionary Population Dynamics approaches for feature selection problems

Thaer Thaher [a,b], Hamouda Chantar [c], Jingwei Too [d], Majdi Mafarja [e], Hamza Turabieh [f], Essam H. Houssein [g,*]

[a] Department of Engineering and Technology Sciences, Arab American University, Ramallah, Palestine
[b] Information Technology Engineering, Al-Quds University, Jerusalem, Palestine
[c] Faculty of Information Technology, Sebha University, Sebha, Libya
[d] Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia
[e] Department of Computer Science, Birzeit University, Birzeit, Palestine
[f] Department of Information Technology, Collage of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
[g] Faculty of Computers and Information, Minia University, Minia, Egypt

ARTICLE INFO

Keywords:
Particle Swarm Optimization (PSO)
Evolutionary Population Dynamics (EPD)
Natural selection
Feature selection
Metaheuristics

ABSTRACT

In the feature selection process, reaching the best subset of features is considered a difficult task. To deal with the complexity associated with this problem, a sophisticated and robust optimization approach is needed. This paper proposes an efficient feature selection approach based on a Boolean variant of Particle Swarm Optimization (BPSO) boosted with Evolutionary Population Dynamics (EPD). The proposed improvement assists the BPSO to avoid local optima obstacles via boosting its exploration ability. In the BPSO-EPD, the worst half of the solutions are discarded by repositioning them around the optimal solutions selected from the best half. Six natural selection mechanisms comprising Best-based, Tournament, Roulette wheel, Stochastic universal sampling, Linear rank, and Random-based are employed to select guiding solutions. To assess the performance of the proposed improvement, 22 well-regarded datasets collected from the UCI repository are employed. The experimental results demonstrate the superiority of the proposed EPD-based feature selection approaches, especially the BPSO-TEPD variant when compared with conventional BPSO and other five EPD-based variants. Taking SpecEW dataset as an example, an increment of 6.7% accuracy can be achieved for BSPO-TEPD. Consequently, BPSO-TEPD approach also outperformed other well-known optimizers, including two binary variants of PSO using S-shaped transfer function (SBPSO) and V-shaped transfer function (VBPSO), Binary Grasshopper Optimization Algorithm (BGOA), Binary Gravitational Search Algorithm (BGSA), Binary Ant Lion Optimizer (BALO), Binary Bat algorithm (BBA), Binary Salp Swarm Algorithm (BSSA), Binary Whale Optimization Algorithm (BWOA), and Binary Teaching-Learning Based Optimization (BTLBO). The result emphasizes the excellent behavior of EPD strategies in evolving the ability of BPSO when dealing with feature selection problems.

1. Introduction

In information systems, the data collected in real-world applications related to various fields such as industry, modern technology, and medicine have high dimensions, which consider a real challenge for data mining. Usually, for tasks such as classification, the real-world datasets often contain irrelevant and redundant patterns or features. Such patterns are uninformative, worthless, and make the task of automatic learning hard. Non-informative features have a direct negative impact on the performance of the machine learning classifiers in trams of both accuracy and cost of computation. Therefore, the process of exploring and eliminating uninformative patterns is essential for building effective machine learning classifiers (Hussain, Neggaz, Zhu, & Houssein, 2021).

Feature selection (FS) is an essential preprocessing procedure that is designed to find and eliminate uninformative features/patterns from the dataset under processing (Mafarja, Aljarah, Heidari, Hammouri, et al., 2018). In general, FS is classified into three board groups including supervised (Neggaz, Houssein, & Hussain, 2020), semi-supervised

* Corresponding author.
E-mail addresses: thaer.thaher@gmail.com (T. Thaher), ham.Chantar@sebhau.edu.ly (H. Chantar), jamesjames868@gmail.com (J. Too), mmafarja@birzeit.edu (M. Mafarja), h.turabieh@tu.edu.sa (H. Turabieh), essam.halim@mu.edu.eg (E.H. Houssein).

https://doi.org/10.1016/j.eswa.2022.116550
Received 14 August 2021; Received in revised form 11 December 2021; Accepted 14 January 2022
Available online 4 February 2022

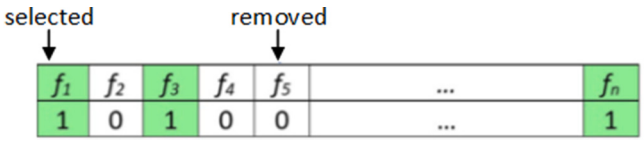

Fig. 1. Binary solution representation.

(Bellal, Elghazel, & Aussem, 2012) and unsupervised approach (Shang, Wang, Stolkin, & Jiao, 2017). The supervised FS approach relies on class labels in selecting the most appropriate features for classification tasks, while unsupervised FS approaches do not need labeled data. Alternatively, semi-supervised FS is appropriate when both labeled and unlabeled data are available in the dataset. In literature, many FS approaches belong to the three mentioned groups have been presented. For instance, the correlation-based feature selection (CFS) proposed by Neggaz, Houssein, et al. (2020) and spectral graph theory-based feature selection introduced by Zhao and Liu (2007) are samples of supervised FS methods. While feature selection technique using spectral analysis (Bellal et al., 2012) and forward feature selection (Ren, Qiu, Fan, Cheng, & Philip, 2008) represent types of semi-supervised FS approaches. In the case of unsupervised FS, an efficient algorithm named non-negative spectral learning and sparse regression-based dual-graph regularized feature selection (NSSRD) was introduced by Shang et al. in 2017 (Shang et al., 2017). In addition, another two approaches called subspace learning-based graph regularized FS and self-representation based dual-graph regularized FS are also unsupervised FS approaches proposed by Shang et al. in 2016 (Shang, Wang, Stolkin, & Jiao, 2016; Shang, Zhang, Jiao, Liu, & Li, 2016).

Based on the selection mechanism, FS approaches are split into two categories: filter and wrapper (Mafarja, Aljarah, Heidari, Hammouri, et al., 2018). In filter mode, feature subset selection is made separately from the learning classifier (Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2015). The quality of a feature is measured by giving a score to the feature using a statistical feature scoring method such as chi-square (Liu & Setiono, 1995), Gain Ratio (Quinlan, 1993), and Information gain (Quinlan, 1986). Features with scores less than a specific threshold are considered uninformative and discarded from the feature space. In wrapper mode, the quality of a subset of features is estimated based on a machine learning classifier (e.g., K-Nearest Neighbor) (Wang, An, Chen, Li, & Alterovitz, 2015). Las Vegas Wrapper (LVW) algorithm (Liu & Setiono, 1996) and a three-layer feedforward neural network-based approach (Setiono & Liu, 1997) represent examples of wrapper FS.

Generation of the subset of features is deemed as a search process for selecting a subset from a set of elements where complete, random, or a heuristic search is used (Dash & Liu, 1997; Siedlecki & Sklansky, 1988). In a complete search, all potential subsets of features are produced and examined. More precisely, if the given dataset has N features, then 2^N subsets will be produced and tested to find the best one. For big-size datasets, a complete search is infeasible due to its high cost of computation. Another potential way to generate a subset of features is to use a random search. In this way, searching for the subsequent subset of features is performed at random (Lai, Reinders, & Wessels, 2006). The worst scenario when using a random search is to produce all possible subsets of feature as in complete search strategy (Liu & Motoda, 1998; Talbi, 2009). Heuristic search is an alternative way to complete and random search for generating feature subsets. As defined in Talbi (2009), metaheuristic search is a top-level, general model that can be applied as guiding strategies when designing underlying heuristics to tackle particular optimization problems. In comparison with exact approaches (Guyon & Elisseeff, 2003; Zorarpacı & Özel, 2016), several metaheuristics including Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Ant Colony Optimization (ACO) (Dorigo, Birattari, & Stutzle, 2006), Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016), Ant Lion Optimization (ALO) (Mirjalili, 2015), Gray Wolf Optimizer (GWO) (Mirjalili, Mirjalili, & Lewis, 2014; Saremi, Mirjalili, & Mirjalili, 2015) and Firefly Algorithm (FA) (Yang, 2009) may approved notable capabilities in tackling FS problems. Moreover, FS has been utilized to resolve many classification problems belong to diverse fields like data mining (Piramuthu, 2004), path planning (Wu et al., 2017), pattern recognition (Gunal & Edizkan, 2008), power dispatch (Li et al., 2017) and others where FS can be utilized (Chandrashekar & Sahin, 2014).

Metaheuristic-based algorithms can be split into three branches: evolutionary-based algorithms (EAs), swarm-based algorithms (SAs), and trajectory-based algorithms (TAs) (Boussaïd, Lepagnot, & Siarry, 2013). EAs basically begin with a randomly generated population of individuals. In each generation, recombination and mutation are performed on parent individuals to obtain new offsprings that are higher quality than their parents. Genetic algorithm (GA) is an example of EAs (Holland, 1975). On the other side, TAs begin with only one potential solution, which is adapted repeatedly using neighboring operators until an optimal solution is reached (Al-Betar, Awadallah, Faris, Aljarah, & Hammouri, 2018). Examples of TAs are Simulated Annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983) and Tabu Search (TS) (Glover, 1986). SAs begin with a population of randomly generated solutions. These solutions are reformed at each generation depending on the previous generation. SAs are widely utilized to deal with FS problems where traditional approaches are hard to apply (BrezoAinik, Fister, & Podgorelec, 2018).

PSO is deemed as one of the most frequently applied swarm-based algorithms. It was proposed by Kennedy and Eberhart in 1995. It imitates the social behavior of a bevy of birds (Kennedy & Eberhart, 1995). PSO has gained popularity in solving a wide range of optimization problems due to its simplicity, efficiency, and low computation cost (Houssein, Ewees, & Abd ElAziz, 2018). Because of its characteristics, PSO has been applied to deal with many fields such as grid scheduling (Mathiyalagan, Dhepthie, & Sivanandam, 2010), medical diagnosis (Chandra, Bhat, & Singh, 2009), wavelength detection (Liang, Suganthan, Chan, & Huang, 2006), video abstraction (Fayk, El Nemr, & Moussa, 2010), and robot path planning (Masehian & Sedighizadeh, 2010). Furthermore, PSO can be utilized to find solutions for problems in both continuous and discrete search spaces. In 1997, Kennedy and Eberhart modified the original continuous version of the PSO algorithm to work in binary space (Kennedy & Eberhart, 1997). Tow main components distinguish between continuous and binary forms of PSO: A transfer function and new position updating formula. The transfer function is responsible for mapping continuous search space into discrete, whereas the new updating formula is applied to change the elements of the particle's position vector between two values, 0 and 1. Afterward, Shi and Eberhart (1998) introduced a new parameter named inertia weight that aims to improve the performance of PSO when solving complex optimization problems by making a balance between the local and global search of the PSO algorithm and avoids the PSO from falling into local optimum.

Since the appearance of the first version of Binary PSO in 1997 (Kennedy & Eberhart, 1997), many researchers have tried to enhance its ability in solving various discrete optimization problems by introducing several ideas. For instance, in 2004, Shen, Jiang, Jiao, Shen, and Yu (2004) proposed a modified BPSO for selecting variables in MLR and PLS modeling. In addition, in 2008, Wang et al. proposed a novel probability-based BPSO algorithm and evaluated its performance in solving the multidimensional knapsack problem (Wang, Wang, Fu, & Zhen, 2008). Chuang et al. introduced an enhanced BPSO via employing the catfish effect and applied it for feature selection (Chuang, Chang, Tu, & Yang, 2008). In Liu and Gu (2007), Binary PSO was applied for network reconfiguration. In 2013, to improve the performance of BPSO, Mirjalili and Lewis proposed a set of six transfer functions, divided into two groups named S-shaped and V-shaped for mapping the continuous search space into discrete search space. Experimental

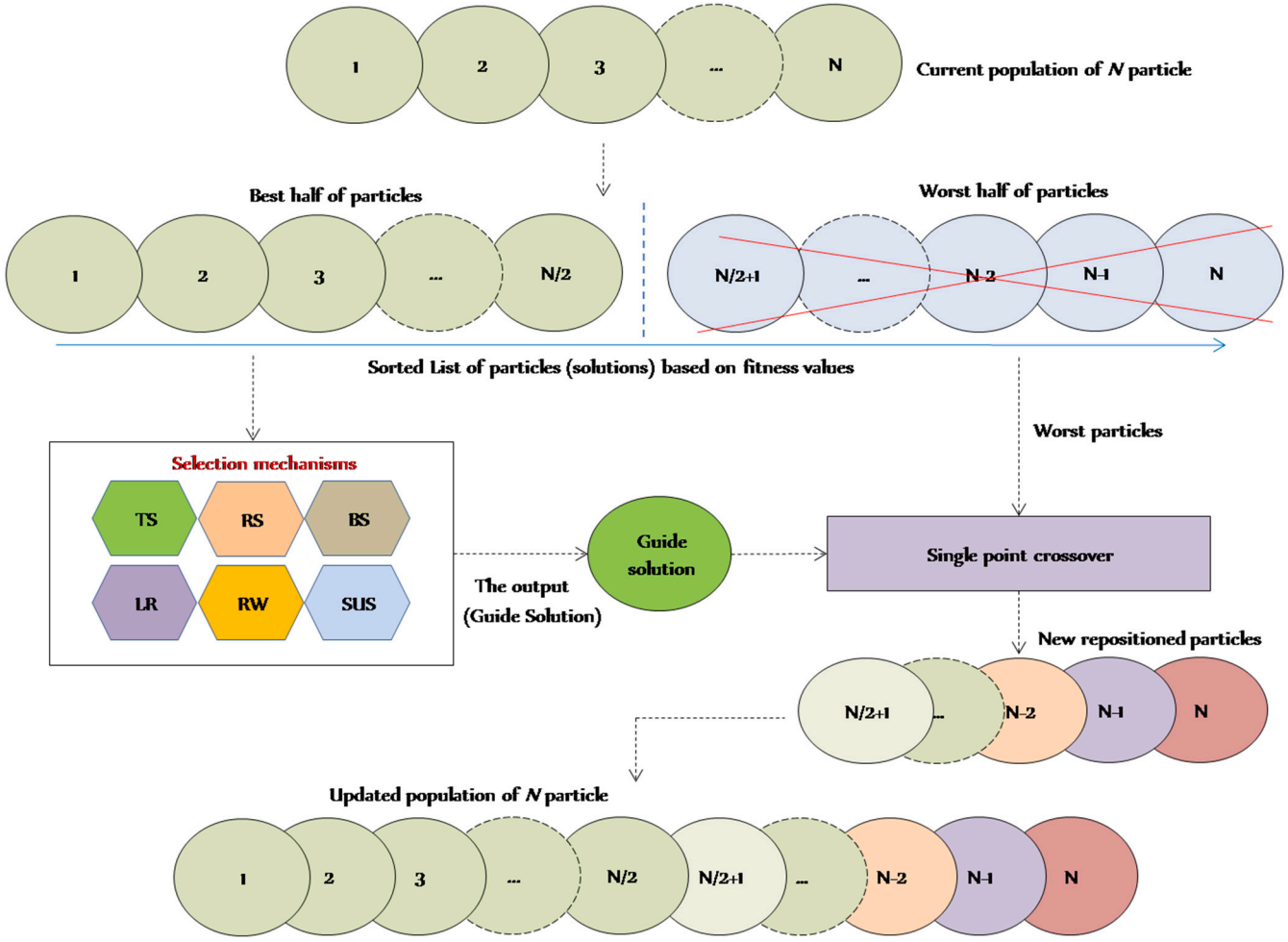

Fig. 2. Diagram of the proposed improvement.

results confirmed that V-shaped transfer functions could remarkably enhance the performance of conventional BPSO (Mirjalili & Lewis, 2013). Furthermore, in 2017, Islam, Li, and Mei (2017) claimed that existing BPSO versions are unable to provide the desired balance between exploration and exploitation due to the use of inappropriate transfer functions. They proposed a time-varying-based BPSO and experimentally demonstrated that TVT-BPSO is better than existing BPSO variants for low and high-dimensional classical 0–1 knapsack problems. In Marandi, Afshinmanesh, Shahabadi, and Bahrami (2006), a Boolean PSO was proposed using the Boolean algebra and applied to an antenna design problem. In addition, as stated in Deligkaris et al. (2009), a Boolean PSO was also applied for thinned planar array design.

According to Crawford et al. (2017), binarization schemes for adapting meta-heuristic algorithms to work in binary search space are divided into two board categories: two-step binarization and continuous-binary operator transformation. In two-step binarization, continuous operators are used to adapting continuous search space without modifying them. Two steps are performed to do the binarization. The first step is to introduce operators for transforming real-valued search space into integer one, whereas in the second step, integer space is converted into binary space. The commonly applied technique of this type is the transfer function and was originally proposed by Kennedy and Eberhart (1997) for BPSO. The transfer functions in the research (Crawford et al., 2017) is applied to provide probabilities for mapping solutions from R^n into $[0,1]^n$. Then, a binarization rule such as standard, complement, or Elitist is used to transform the particle P into a binary solution. Examples of using transfer functions are presented in Islam et al. (2017), Kennedy and Eberhart (1997) and Mirjalili and Lewis (2013). In Continuous-Binary Operator Transformation, the operators of the meta-heuristic are redefined. This type of binarization is further divided into two groups named modified algebraic operations and promising regions (Crawford et al., 2017). Modified algebraic operations (e.g., Boolean approach) use Boolean operations to transform real search space into binary. Examples of Boolean PSO are (Afshinmanesh, Marandi, & Rahimi-Kian, 2005; Deligkaris et al., 2009; Marandi et al., 2006). The proposed modified algebraic operations-based PSO by Afshinmanesh et al. (2005) yielded faster convergence speed and better local optimum reduction in comparison with transfer function-based Binary PSO. According to Afshinmanesh et al. (2005), using the transfer function for adapting PSO to work in binary search space has some drawbacks. For instance, the distance updating formula in Kennedy and Eberhart (1997) does not have a standard form. Another drawback of transfer function-based binary PSO is that the changing probability function has no monotonic form. This problem decreases the changing probability for some larger values of V_{id}. In addition, as stated in Islam et al. (2017), the transfer function in BPSO is considered as the main operator for managing exploration and exploitation, and hence applying an unsuitable transfer function can remarkably reduce the performance of the BPSO.

Evolutionary algorithms (EAs) are formulated to stimulate the evolution of individuals, starting from their basic states to some objectives enjoined on them. These models use some evolutionary operators such as mutation and recombination in the Genetic Algorithm (GA) or pheromone updating rules of ACO to some picked individuals depending on some selection techniques such as roulette wheel, tournament,

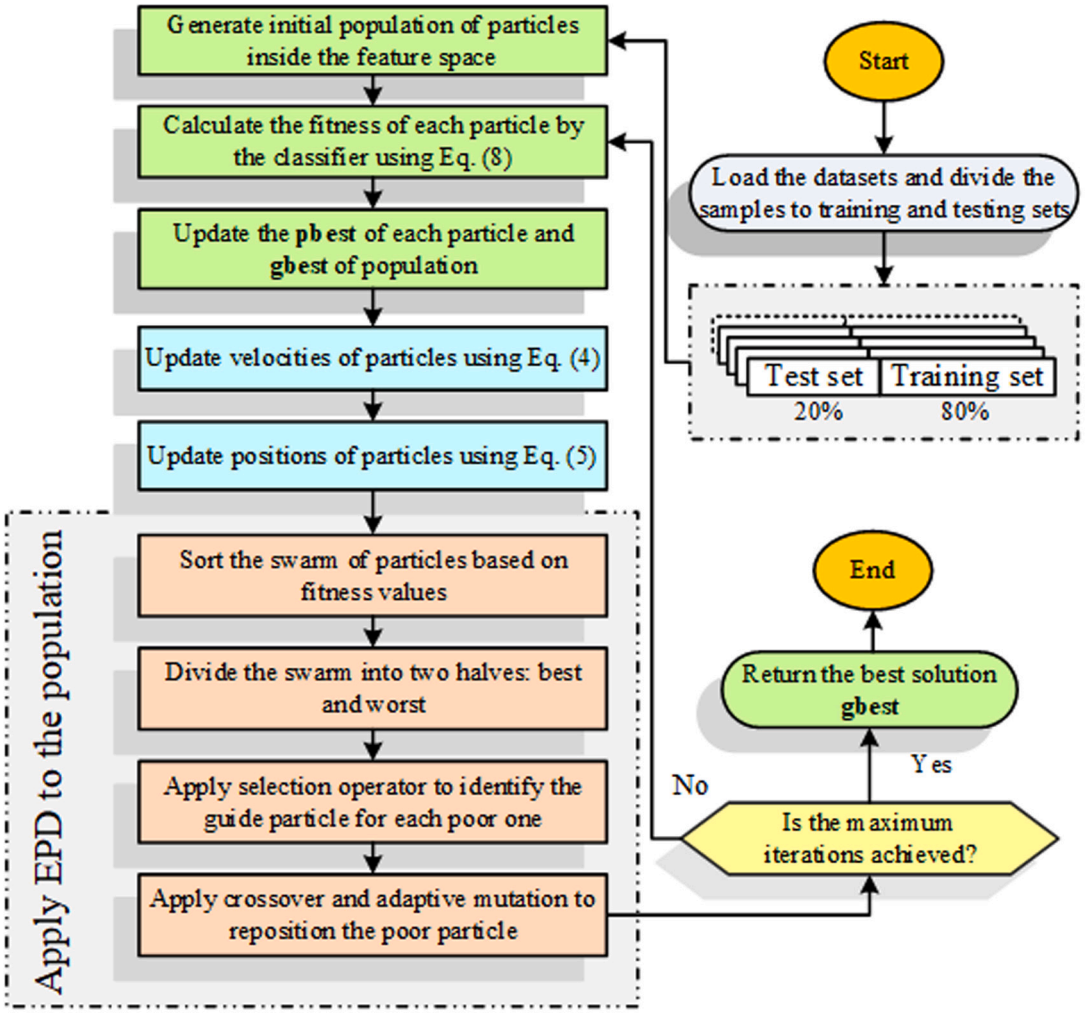

Fig. 3. The overall procedure of the proposed BPSO-EPD.

and random selection in the population to produce an offspring (Mafarja et al., 2017). Evolutionary Population Dynamics (EPD) is also an evolutionary operator. It is mainly based on the Self-organized criticality (SOC) theory (Bak, Tang, & Wiesenfeld, 1987). EPD manipulates the entire population instead of manipulating individuals (Lewis, Mostaghim, & Randall, 2008). Employing this operator with evolutionary algorithms will discard highly unfavorable individuals from the set of candidate solutions instead of refining the optimal individuals in the population (Boettcher & Percus, 1999). Extremal optimization (EO) is a metaheuristic algorithm that is based on the concept of EPD. It has been successfully applied in various research domains (Randall, Hendtlass, & Lewis, 2009; Tamura, Kitakami, & Nakada, 2013). In addition, EPD was used to enhance the performance of the GWO algorithm (Saremi et al., 2015).

FS is recognized as a complex and multi-objective optimization problem. Using search strategies such as exact and random for finding the ideal subset of features, especially from a huge set of features, is infeasible. However, for many classification tasks, metaheuristic-based FS approaches have been found very effective search strategies for finding optimal subsets of features. Furthermore, the PSO-based algorithms are successfully explored in different FS domains. According to the previous studies, the drawbacks of using transfer functions to convert the PSO algorithm into a binary show that efforts are still needed to propose new ideas for improving the performance of the binary version of the PSO algorithm. In addition, better performance of the Boolean variant of PSO in contrast with its transfer function-based peers reveals that the performance of the current binary PSO can be further improved by integrating Boolean PSO with other robust mechanisms such as evolutionary population dynamics and natural selection mechanisms. These mechanisms have shown excellent results

Table 1
Description of employed datasets.

Dataset	No. of features	No. of instances
Breastcancer	9	699
BreastEW	30	569
Exactly	13	1000
Exactly2	13	1000
HeartEW	13	270
Lymphography	18	148
M-of-n	13	1000
PenglungEW	325	73
SonarEW	60	208
SpectEW	22	267
CongressEW	16	435
IonosphereEW	34	351
KrvskpEW	36	3196
Tic-tac-toe	9	958
Vote	16	300
WaveformEW	40	5000
WineEW	166	476
Zoo	16	101
Clean1	13	178
Semeion	265	1593
Colon	2000	62
Leukemia	7129	72

Table 2
The detailed settings of the system.

Name	Setting
Hardware	
CPU	Intel Core(TM) i7-8550U
Frequency	2.2 GHz
RAM	8 GB
Hard drive	1 TB
Software	
Operating system	Windows 10 64 bit
Language	MATLAB R2018a

in boosting the efficiency of several binary versions of meta-heuristic algorithms such as whale optimization algorithm (WOA) (Hassouneh et al., 2021) and gray wolf optimizer (GWO) (Al-Betar et al., 2018) in the FS field. This is the foundation and motivation of this work as well, in which we propose a novel algorithm based on the modified algebraic operations instead of the transfer functions with various evolutionary population dynamics for FS tasks. The efficiency of the proposed algorithm developed in this research is evaluated on 22 well-regarded datasets obtained from the UCI repository. These datasets are chosen carefully with various properties (e.g., number of features, instances, and classes) to approve the efficiency and robustness of the proposed PSO-based FS approaches. The highest classification accuracy and least number of selected features are the main measures for evaluating the efficiency and robustness of the proposed approach. To sum up, the main contributions of this research is fourfold:

1. We propose a new Boolean PSO (BPSO) algorithm based on modified algebraic operations instead of the transfer functions for FS problems.
2. Various EPD mechanisms have been integrated with the BPSO to enhance its search power for the best subsets of features.
3. The convergence speed of BPSO are evaluated against the state-of-the-art meta-heuristic algorithms.
4. We perform a series of experiments to investigate the impact of the performance of the proposed algorithms over 22 well-regarded datasets obtained from the UCI repository.
5. BPSO outperforms other competitive algorithms in most cases.

The rest of the paper is organized as follows: Section 2 provides a brief overview of the related work. The proposed PSO methodology is described in Section 4. Section 5 presents a comprehensive comparative study on several benchmark functions to confirm and verify the performances of the proposed algorithm. Finally, Section 6 concludes the paper and suggests some directions for future studies.

2. Review of related works

Nowadays, wrapper FS has become a vital technique to seek out the most informative group of features, which was exceedingly useful in industry and medical applications (Sayed, Nabil, & Badr, 2016). Generally speaking, the wrapper FS treats the FS problem as a black box in which the meta-heuristic algorithm and classifier are used to evaluate and assess the optimal feature subset (Aghdam, Ghasem-Aghaee, & Basiri, 2009). In Kohavi and John (1998), the wrapper FS was introduced to improve the performance of the learning model. The study implied that the wrapper FS was excellent in eliminating unwanted information and enhancing the prediction power.

In 2013, Nakamura et al. (2013) designed a binary bat algorithm to identify the most informative features. They utilized the sigmoid function to restrict the position of the new bat on binary feature space. In the same year, another binary cuckoo search (BCS) algorithm was developed to tackle the FS problems (Rodrigues et al., 2013). Two years later, the authors in Rodrigues, Yang, De Souza, and Papa (2015) integrated the S-shape transfer function into the Flower Pollination Algorithm (FPA) for FS. Although the methods mentioned above have successfully applied to FS tasks, they suffered from the limitation of the S-shape function, which forces the solutions to take values of 1 or 0 (Mirjalili & Lewis, 2013).

Grasshopper Optimization Algorithm (GOA) was a recently established meta-heuristic algorithm (Saremi, Mirjalili, & Lewis, 2017), and it was first applied for FS in Aljarah, Ala'm et al. (2018). Mafarja, Aljarah, Heidari, Hammouri, et al. (2018) proposed a novel GOA approach for FS problems. The authors associated the binary GOA with EPD to re-position and improve the worst half of the population (Mafarja, Aljarah, Heidari, Hammouri, et al., 2018). Moreover, a binary GOA method was proposed to select the significant features from the datasets. In the proposed approach, a novel dynamic mutation operator was developed to boost the convergence rate (Mafarja et al., 2019). Another two recent GOA-based FS approaches were proposed in Zakeri and Hokmabadi (2019) and Hichem, Elkamel, Rafik, Mesaaoud, and Ouahiba (2019).

Recently, Mafarja and Mirjalili (2018) proposed the WOA approaches for wrapper FS. In their study, the crossover and mutation are utilized to evolve the exploitation and exploration abilities of the algorithm in searching the feature space (Mafarja & Mirjalili, 2018). The authors in Dong, Li, Ding, and Sun (2018) hybridized the GA with granular information to solve the feature optimization. They reported that the granular-based mechanism was beneficial in measuring the importance of the candidate features, which can significantly enhance the classification accuracy (Dong et al., 2018). Furthermore, a Gaussian mutational chaotic fruit fly optimization algorithm (MCFOA) was proposed to tackle the FS problems (Zhang et al., 2020). In addition, binary variants of the Butterfly Optimization Algorithm (BOA) were developed for wrapper FS in Arora and Anand (2019).

PSO can be considered the most popular meta-heuristic algorithm in the literature. It has the advantages of being simple, robust, and efficient when faced with complicated optimization problems (Li, Chen, Zhong, & Huang, 2019). Although PSO has proved its effectiveness in FS, it also has some shortcomings like premature convergence and eases to fall into local optimum (Xie et al., 2019). Therefore, Fong, Wong, and Vasilakos (2015) proposed an accelerated PSO swarm search to evaluate the promising features in the big data. They reported that the proposed method could often achieve higher accuracy compared to the other well-known FS algorithms. On the other hand, a potential PSO was designed in Tran, Xue, and Zhang (2017) for wrapper FS. In the proposed approach, a new representation was proposed to reduce the feature space (Tran et al., 2017). On the one hand, the authors in Gunasundari, Janakiraman, and Meenambal (2016) offered a velocity bounded Boolean PSO to tackle the FS problems in liver and kidney disease diagnosis. Moreover, another variant of PSO called multi-swarm heterogeneous binary PSO-based FS was developed in Gunasundari, Janakiraman, and Meenambal (2018). Overall, the survey reveals that the performance of Boolean PSO is still far from perfect. That is, the particles cannot effectively avoid the local minimal completely. Hence, it is necessary to enhance the performance of Boolean PSO by overcoming the challenges of Boolean PSO.

From the literature, it is seen that Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995) has become an increasingly important tool of SI that has been used in nearly all fields of optimization, engineering practice and real world problems, so forth (Houssein, Gad, Hussain, & Suganthan, 2021). The simplicity, efficiency, and low cost of computation make this algorithm very famous and strong in tackling a wide range of optimization problems. Lately, PSO has gained much consideration in past years, with several efforts to obtain the variant that achieves best on a wide range of optimization problems such as FMPSO (Xia et al., 2019), CAPSO (Beheshti & Shamsuddin, 2014), PSOG (Salajegheh & Salajegheh, 2019), PSOSCALF (Chegini, Bagheri, & Najafi, 2018), MOVPSO (Meza, Espitia, Montenegro, Giménez, & González-Crespo, 2017), MPSO (Tian & Shi, 2018), PSOTD (Chen et al., 2017), FST-PSO (Nobile et al., 2018), SPSO (Pedersen & Chipperfield,

Table 3
The average accuracy obtained by BPSO with different combinations of common parameters.

#iterations	70			100			150		
population	10	20	30	10	20	30	10	20	30
Breastcancer	0.967	0.980	0.976	0.980	0.976	0.981	0.971	**0.999**	0.976
BreastEW	0.970	0.986	0.982	0.983	0.970	0.985	**0.989**	0.968	0.968
CongressEW	0.984	0.962	0.987	0.973	0.970	0.965	0.969	0.964	**0.990**
Exactly	0.925	0.990	**1.000**	0.938	**1.000**	0.989	0.955	0.969	0.996
Exactly2	0.742	0.746	0.769	0.745	**0.792**	0.729	0.770	0.740	0.741
HeartEW	0.807	0.860	0.841	0.878	0.870	0.853	0.877	**0.909**	0.885
IonosphereEW	**0.996**	0.977	0.990	0.959	0.980	0.958	0.958	0.976	0.981
KrvskpEW	0.972	0.985	0.985	0.972	0.986	**0.988**	0.968	0.986	0.985
Lymphography	0.875	0.949	0.790	0.888	0.883	**0.981**	0.916	0.944	0.973
M-of-n	0.988	1.000	**1.000**	0.988	1.000	**1.000**	0.976	**1.000**	1.000
penglungEW	0.996	**1.000**	0.998	0.863	**1.000**	0.942	0.947	0.913	0.984
SonarEW	0.953	0.987	0.994	0.997	0.994	0.988	0.989	**1.000**	0.978
SpectEW	0.864	0.846	0.857	0.799	0.902	0.840	0.857	**0.938**	0.861
Tic-tac-toe	0.797	0.832	0.810	0.833	**0.841**	0.810	0.826	0.801	0.823
Vote	0.966	**1.000**	0.974	0.986	0.971	0.987	0.982	0.923	0.977
WaveformEW	0.765	0.780	**0.784**	0.759	0.774	0.774	0.758	0.776	0.777
WineEW	0.999	0.997	0.997	0.999	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Zoo	0.956	0.995	**1.000**	0.986	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Clean1	0.977	**0.991**	0.990	0.947	0.961	0.986	0.967	0.980	0.974
Semeion	0.983	0.994	0.993	0.994	0.997	**0.997**	0.996	0.990	0.996
Colon	0.846	0.562	0.849	0.874	0.759	0.821	0.846	0.672	**1.000**
Leukemia	**1.000**	0.933	0.929	0.929	0.887	**1.000**	0.942	0.933	0.813
Rank (F-test)	3.57	5.23	5.45	4.32	**5.68**	5.5	4.68	5.07	5.5

Table 4
The average accuracy obtained by BPSO-TEPD with various tournament sizes (k), and BPSO-LREPD with various values of (n^+) parameter.

Dataset	BPSO-TEPD					BPSO-LREPD			
	0.1 × N	0.3 × N	0.5 × N	0.7 × N	0.85 × N	$n = 1.1$	$n = 1.4$	$n = 1.7$	$n = 1.9$
Breastcancer	**1.000**	0.979	0.986	0.979	0.979	0.993	0.971	0.986	**0.993**
BreastEW	0.973	**1.000**	0.987	0.973	0.975	0.991	0.966	0.951	**0.999**
CongressEW	0.989	0.984	**0.994**	0.989	0.987	0.977	0.996	**1.000**	1.000
Exactly	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Exactly2	0.739	0.765	**0.804**	0.785	0.758	0.769	0.768	**0.797**	0.768
HeartEW	0.860	0.878	0.884	0.846	**0.889**	0.873	**0.924**	0.880	0.902
IonosphereEW	**0.999**	0.997	0.998	0.981	0.997	**0.990**	0.990	0.984	0.969
KrvskpEW	0.987	0.989	**0.990**	0.987	0.986	0.988	0.985	0.983	**0.990**
Lymphography	0.942	0.992	0.939	**0.999**	0.990	**0.962**	0.938	0.926	0.958
M-of-n	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
penglungEW	**1.000**	**1.000**	**1.000**	0.951	0.933	0.949	0.871	**0.991**	0.989
SonarEW	0.995	0.982	**0.998**	0.997	0.996	0.994	0.997	0.998	**1.000**
SpectEW	**0.914**	0.892	0.873	0.888	0.888	0.765	0.877	**0.936**	0.927
Tic-tac-toe	0.842	**0.858**	0.797	0.833	0.818	**0.849**	0.805	0.835	0.832
Vote	0.978	0.990	**0.998**	0.967	0.985	0.967	0.967	**0.983**	0.968
WaveformEW	**0.790**	0.784	0.776	0.789	0.789	0.787	0.777	0.780	**0.790**
WineEW	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Zoo	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Clean1	0.985	**0.999**	0.990	0.981	0.978	0.969	0.984	**0.993**	0.977
Semeion	0.996	0.997	0.997	**1.000**	0.998	0.995	**1.000**	**1.000**	0.994
Colon	**0.923**	0.792	**0.923**	0.672	0.892	0.828	**0.921**	0.774	0.731
Leukemia	**1.000**	0.900	**1.000**	**1.000**	0.971	0.933	0.927	**1.000**	**1.000**
Rank (F-test)	3.14	3.05	**3.45**	2.75	2.61	2.39	2.18	2.7	**2.73**

2010), THSPSO (Liu, Cui, Lu, Liu, & Deng, 2019), TSLPSO (Xu et al., 2019) and H-PSO-SCAC (Chen et al., 2018).

Definitely, several variants of PSO algorithm have been demonstrated to be efficient in feature selection domain, like BPSO (Mirjalili & Lewis, 2013), which is a binary variant of PSO that has been founded for handling binary optimization problems using the s-shaped and v-shaped transfer functions. A FS algorithm termed BPSO-SVM adapting the memory of both local and global optimum (LGO), and also rising the probability of particles' mutation for feature selection to avoid early convergence problem, and obtain high valuable features (Wei et al., 2017). Also, a modified version of discrete particle swarm optimization (PSO) algorithm for solving feature subset selection problem is introduced (Unler & Murat, 2010). This technique incorporates an adaptive feature selection strategy that dynamically accounts for the relationship and dependence of the features contained in the feature subset. In Moradi and Gholampour (2016), an innovative FS named HPSO-LS is introduced which utilizes a local search (LS) approach to select silent and less correlated features. In Chuang, Yang, and Li (2011), Logistic and tent maps are integrated in Binary particle swarm optimization (BPSO) to set the inertia weight of the BPSO. K-nearest neighbor algorithm with leave-one-out cross-validation acts as a classifier for assessing classification accuracies. The proposed BPSO algorithm has been applied efficiently to feature selection problems. In Jain, Jain, and Jain (2018), A hybrid approach for cancer classification is introduced in Jain et al. (2018). It integrates correlation-based feature selection (CFS) with enhanced Binary PSO using Naive Bayes algorithm. In addition, a new feature selection approach based on BPSO was proposed by Zhang, Gong, Hu, and Zhang (2015) in which a reinforced memory mechanism for updating the local leaders of particles to avoid the degradation of prominent genes in the particles, and a uniform combination for balancing the local exploitation and the global exploration were introduced. Also, one-nearest neighbor classifier was used to evaluate the classification accuracy of feature subsets. Further, a novel hybrid approach combining Binary Bat algorithm with

Table 5
Average accuracy results of the proposed variants with KNN classifier.

Dataset	Measure	BPSO	BPSO_TEPD	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	AVG	0.9767	0.9857	0.9643	**0.9929**	0.9857	**0.9929**	0.9855
	STD	0.0032	0.0000	0.0000	0.0000	0.0000	0.0000	0.0013
BreastEW	AVG	0.9649	0.9874	0.9778	**0.9985**	0.9784	0.9854	0.9915
	STD	0.0065	0.0050	0.0075	0.0040	0.0045	0.0062	0.0059
CongressEW	AVG	0.9762	0.9939	0.9881	0.9996	**1.0000**	0.9939	0.9866
	STD	0.0060	0.0058	0.0111	0.0021	0.0000	0.0058	0.0044
Exactly	AVG	0.9895	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	STD	0.0539	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Exactly2	AVG	0.7760	**0.8043**	0.7742	0.7677	0.7623	0.7390	0.7978
	STD	0.0200	0.0037	0.0072	0.0052	0.0073	0.0087	0.0406
HeartEW	AVG	0.8506	0.8840	0.8883	0.9025	0.9056	0.8691	**0.9142**
	STD	0.0217	0.0083	0.0034	0.0145	0.0057	0.0047	0.0114
IonosphereEW	AVG	0.9934	0.9977	0.9765	0.9690	0.9840	0.9873	**0.9981**
	STD	0.0089	0.0053	0.0113	0.0101	0.0115	0.0093	0.0049
KrvskpEW	AVG	0.9784	0.9899	0.9907	0.9902	0.9836	**0.9951**	0.9868
	STD	0.0057	0.0022	0.0018	0.0031	0.0014	0.0044	0.0012
Lymphography	AVG	0.9478	0.9385	**0.9978**	0.9578	0.8944	0.9400	0.9611
	STD	0.0226	0.0180	0.0085	0.0150	0.0126	0.0203	0.0126
M-of-n	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	STD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
penglungEW	AVG	0.9840	**1.0000**	0.9311	0.9889	**1.0000**	0.9400	0.9822
	STD	0.0296	0.0000	0.0276	0.0253	0.0000	0.0320	0.0300
SonarEW	AVG	0.9913	0.9984	0.9937	**1.0000**	0.9810	0.9889	0.9968
	STD	0.0117	0.0087	0.0107	0.0000	0.0131	0.0121	0.0082
SpectEW	AVG	0.8056	0.8728	0.8488	0.9272	0.8833	**0.9309**	0.8852
	STD	0.0144	0.0064	0.0085	0.0068	0.0086	0.0128	0.0075
Tic-tac-toe	AVG	0.8349	0.7965	**0.8403**	0.8325	0.8059	0.8229	0.8281
	STD	0.0140	0.0019	0.0055	0.0039	0.0036	0.0000	0.0000
Vote	AVG	0.9922	0.9983	**1.0000**	0.9683	0.9700	0.9944	0.9833
	STD	0.0137	0.0051	0.0000	0.0051	0.0068	0.0080	0.0000
WaveformEW	AVG	0.7686	0.7760	**0.7934**	0.7897	0.7840	0.7865	0.7730
	STD	0.0092	0.0070	0.0085	0.0057	0.0061	0.0066	0.0121
WineEW	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	STD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Zoo	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	STD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Clean1	AVG	0.9701	**0.9903**	0.9785	0.9767	0.9583	0.9861	0.9767
	STD	0.0122	0.0090	0.0086	0.0071	0.0095	0.0057	0.0127
Semeion	AVG	0.9955	0.9974	**0.9995**	0.9944	0.9984	0.9915	0.9936
	STD	0.0024	0.0017	0.0012	0.0021	0.0016	0.0017	0.0015
Colon	AVG	0.7128	**0.9231**	0.8487	0.7308	0.7256	0.7359	**0.9231**
	STD	0.0449	0.0000	0.0428	0.0525	0.0437	0.0388	0.0000
Leukemia	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9333	0.9333	0.9333
	STD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Mean rank	F-test	5.068	**3.364**	3.636	3.455	4.568	4.114	3.795
Overall rank		7	**1**	3	2	6	5	4

Particle Swarm Optimization Algorithm (HBBEPSO) for solving feature selection problem is introduced by Tawhid and Dsouza (2018). The proposed algorithm proved its ability to search the feature space for optimal features subset.

In the same context, a new optimization approach named catfish binary particle swarm optimization (CatfishBPSO) is utilized to enhance the performance of binary particle swarm optimization (BPSO) for feature selection (Chuang, Tsai, and Yang, 2011). K-nearest neighbor classifier with leave-one-out cross-validation (LOOCV) procedure was applied to assess the goodness of subsets of features generated by the proposed BPSO-based FS approach. In Lu, Liang, Ye, and Cao (2015), a feature selection method based on two improved PSO models using both inertia weight and functional constriction factor is proposed. In addition, Binary particle swarm optimization (BPSO) was hybridized with opposition-based learning to solve feature selection problem in text clustering (Bharti & Singh, 2016). In Xue, Zhang, and Browne (2014), three feature selection approaches were introduced based on new initializing and updating mechanisms in PSO with the goals of increasing the classification performance and minimizing each of number of selected features and the computational time. Two Boolean PSO approaches called Velocity Bounded BoPSO (VbBoPSO) and Improved Velocity Bounded BoPSO (IVbBoPSO) are proposed by Gunasundari et al. (2016) to solve feature selection problem in liver and kidney disease diagnosis. Also in Subasi (2013), a novel PSO-SVM approach has been proposed. It hybridizes the particle swarm optimization (PSO) and SVM to improve EMG signal classification accuracy. An automatic image annotation schema based on improved quantum PSO (IQPSO) is introduced in Jin and Jin (2015) for visual feature selection.

As discussed above, the literature survey on PSO shows that the local minima obstacle is not addressed sufficiently. It requires an efficient algorithm to evolve the performance of the PSO algorithm and its variants by handling the difficulties of the basic PSO and overcoming

Table 6
p-values of the Wilcoxon test for the accuracy results of BPSO-TEPD and other variants in Table 5 ($p \leq 0.05$ are significant and bolded).

Dataset	BPSO	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	**2.43E−13**	**1.69E−14**	**1.69E−14**	NaN	**1.69E−14**	3.34E−01
BreastEW	**1.85E−11**	**2.87E−06**	**2.42E−10**	**7.60E−08**	2.10E−01	**3.41E−03**
CongressEW	**3.86E−11**	**4.43E−02**	**2.12E−05**	**3.80E−06**	1.00E+00	**6.37E−06**
Exactly	1.61E−01	NaN	NaN	NaN	NaN	NaN
Exactly2	**1.42E−04**	**1.66E−12**	**3.76E−13**	**1.14E−12**	**8.36E−13**	6.39E−01
HeartEW	**6.45E−10**	**1.25E−02**	**3.99E−09**	**1.57E−11**	**5.86E−09**	**3.30E−11**
IonosphereEW	**3.92E−02**	**8.42E−10**	**7.31E−12**	**1.48E−06**	**8.26E−06**	7.29E−01
KrvskpEW	**6.27E−11**	2.08E−01	3.79E−01	**1.28E−10**	**7.45E−07**	**3.94E−08**
Lymphography	**5.84E−05**	**8.55E−13**	**5.23E−09**	**1.88E−11**	**2.16E−03**	**2.43E−10**
M-of-n	NaN	NaN	NaN	NaN	NaN	NaN
penglungEW	**5.55E−03**	**1.92E−12**	**2.14E−02**	NaN	**1.42E−10**	**2.70E−03**
SonarEW	**2.17E−03**	**1.64E−02**	3.34E−01	**1.33E−07**	**2.28E−04**	1.90E−01
SpectEW	**2.44E−12**	**2.68E−12**	**3.77E−13**	**1.05E−05**	**2.31E−12**	**3.00E−07**
Tic-tac-toe	**4.61E−13**	**4.53E−13**	**6.50E−14**	**2.51E−11**	**2.71E−14**	**2.71E−14**
Vote	**4.28E−02**	8.14E−02	**3.37E−13**	**1.26E−12**	**3.04E−02**	**3.94E−12**
WaveformEW	**2.06E−03**	**8.61E−10**	**3.52E−09**	**8.39E−05**	**7.09E−07**	2.70E−01
WineEW	NaN	NaN	NaN	NaN	NaN	NaN
Zoo	NaN	NaN	NaN	NaN	NaN	NaN
Clean1	**4.34E−08**	**2.02E−05**	**4.82E−07**	**4.27E−11**	**3.52E−02**	**5.36E−05**
Semeion	**1.24E−03**	**3.59E−06**	**5.85E−07**	**2.04E−02**	**1.13E−11**	**5.85E−10**
Colon	**3.07E−13**	**2.05E−10**	**5.95E−13**	**4.42E−13**	**4.46E−13**	NaN
Leukemia	NaN	NaN	NaN	**1.69E−14**	**1.69E−14**	**1.69E−14**

its drawbacks. In addition, it is clear that most of the binary variants of PSO are based on transfer functions. Modified algebraic operations-based PSO that use simple Boolean operations such as 'AND' ($\wedge$), 'XOR' ($\oplus$), and 'OR' ($\vee$) to transform real search space into binary has shown better performance in terms of convergence speed and local optimum avoidance in comparison with transfer function-based Binary PSO. In this regard, this paper proposes a novel Boolean PSO optimization algorithms with evolutionary population dynamics for solving the FS problem.

3. Preliminaries

This section introduces the classification algorithms used in this work. It also explains the particle swarm optimization algorithm (PSO) and its pseudo code.

3.1. Classification algorithms

3.1.1. K-nearest neighbor (K-NN) classifier

In literature (Oh, Lee, & Moon, 2004), K-NN is a non-parametric technique and is also regarded as one of the most widely applied in pattern recognition for classification and regression purposes. In addition, K-NN classifier is considered one of the simplest machine learning algorithms. For the test sample, K-NN uses the Euclidean distance for continuous variables to find the training nearest neighbors; thus, the majority class and its nearest neighbors are classified. The major property of K-NN is the need to define only one positive user-defined constant is called (K), and is used to classify new samples of data. Therefore, the overall classification accuracy will effect based on the k parameter value. If $k = 1$, then the object is simply assigned to the class of that identical nearest neighbor. Hence, to reduces the noise on the classification, a larger value of k is defined depends upon various metaheuristic algorithms.

Often, K-NN uses Euclidean distance for continuous variables as follows using Eq. (1):

$$ED(X_1, X_2) = \left(\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2\right)^{\frac{1}{2}} \quad (1)$$

where X_1 and X_2 denote two points with n dimensions.

3.1.2. Decision tree classifier (DT)

Decision Tree (DT) classifier is one of the most methods that is employed successfully in data mining, statistics, and machine learning (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). The main core of DT is to use the methods of knowledge acquisition and entropy. In general, DT employed the graph-based rules to classify data, where the final DT model is derived from the input dataset features (i.e., input features) with respect to the target value (Sahoo, Subudhi, Dash, & Sabut, 2020). In simple, the learning process of DT begins from the top point of the data and moves down leaf based on a set of discrete values, where each leaf node value refers to the predicted output. The DT classification algorithm begins from the root node. The decision is obtained at each branch of the tree to find the best model that fits the input features with output class based on the highest information gain from the set of the training dataset. Eq. (2) demonstrates the gained information from each node based on the reduction in entropy.

$$Gain(I, A) = Entropy(E) - \sum_{v \in Values(A)} \left(\frac{|I_q|}{|I|} Entropy(S_q)\right) \quad (2)$$

where I refers to data instances, A presents attribute with a value equals q, and S_q is the subset of I. Eq. (3) presents the calculations for Entropy, where P_i presents the proportion of orders in I that have the ith class value as output attribute.

$$Entropy(E) = -\sum_{i=1}^{C}(P \log_2 P_i) \quad (3)$$

DT can solve two types of problems: classification when data is discrete in nature, and regression when data is continuous in nature. Many algorithms are used in DT, such as Classification And Regression Tree (CART), ID3, and C4.5 algorithms.

3.2. Particle swarm optimization (PSO)

Kennedy and Eberhart proposed one of the most successful swarm optimization algorithm called PSO (Eberhart & Kennedy, 1995). The main idea is to simulate the performance of bird flocking based on a set of randomly initialized particles that explore and exploit the search space in order to find the optimal solution. In general, PSO is an iterative algorithm that each particle moves towards the best point (i.e., location). Each particle performs two processes: exploration and exploitation with respect to its best location and the best location for all particles. Each particle updates its position in the search space to achieve these two processes based on two factors: velocity and position

Table 7
The Average number of features results of the proposed variants with KNN classifier.

Dataset	Measure	BPSO	BPSO_TEPD	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	AVG	4.333	5.300	**3.067**	4.000	5.000	3.500	5.933
	STD	0.547	0.466	0.254	0.000	0.000	1.137	0.365
BreastEW	AVG	10.433	8.767	**8.133**	9.000	10.233	8.267	9.767
	STD	2.329	1.612	2.488	1.930	1.654	1.856	1.794
CongressEW	AVG	5.333	4.900	5.867	6.600	3.267	**2.867**	6.167
	STD	1.213	1.348	1.756	0.770	0.450	2.030	1.147
Exactly	AVG	6.367	**6.000**	**6.000**	**6.000**	**6.000**	**6.000**	**6.000**
	STD	0.556	0.000	0.000	0.000	0.000	0.000	0.000
Exactly2	AVG	4.767	9.900	**3.733**	9.367	5.833	3.833	6.200
	STD	4.207	0.548	1.874	0.718	3.174	1.206	3.347
HeartEW	AVG	5.167	**4.867**	7.067	5.900	6.567	6.000	5.133
	STD	1.177	1.279	0.365	0.403	0.858	0.455	3.224
IonosphereEW	AVG	9.967	9.033	8.767	9.633	10.600	**8.000**	8.767
	STD	1.691	2.076	2.012	1.608	2.253	1.554	1.591
KrvskpEW	AVG	18.867	17.633	18.467	19.700	16.067	20.733	**14.800**
	STD	3.371	2.619	1.408	1.579	2.690	1.741	4.003
Lymphography	AVG	7.333	5.333	4.767	7.200	7.000	7.833	**4.567**
	STD	1.900	1.348	0.568	0.484	1.509	1.984	1.331
M-of-n	AVG	6.167	**6.000**	**6.000**	**6.000**	**6.000**	**6.000**	**6.000**
	STD	0.379	0.000	0.000	0.000	0.000	0.000	0.000
penglungEW	AVG	96.700	**70.033**	76.200	83.133	71.767	77.933	80.567
	STD	6.535	3.891	7.568	7.262	5.036	7.772	10.156
SonarEW	AVG	19.200	19.067	19.867	19.500	19.633	17.433	**16.367**
	STD	2.809	2.288	2.886	3.371	2.399	2.373	2.748
SpectEW	AVG	7.400	**5.200**	7.333	6.467	5.667	9.100	8.333
	STD	1.499	1.584	1.729	1.042	0.758	1.470	1.124
Tic-tac-toe	AVG	6.933	6.033	**5.767**	6.967	6.733	6.000	6.000
	STD	0.254	0.183	0.430	0.183	0.691	0.000	0.000
Vote	AVG	5.833	3.967	**3.067**	3.900	4.400	4.333	4.400
	STD	1.147	1.938	0.254	1.605	1.653	2.023	0.675
WaveformEW	AVG	20.933	22.867	**19.233**	21.633	20.800	21.367	20.333
	STD	2.924	1.961	3.501	2.810	2.384	2.371	2.591
WineEW	AVG	5.400	**3.333**	4.033	4.033	4.200	3.433	6.167
	STD	1.192	0.922	0.183	0.183	0.610	0.898	0.379
Zoo	AVG	5.133	3.200	4.367	4.000	**2.200**	4.033	4.000
	STD	0.819	0.407	0.490	0.000	0.407	0.183	0.000
Clean1	AVG	61.900	55.167	**54.767**	59.400	58.633	62.633	57.333
	STD	5.281	5.742	6.484	4.724	4.803	6.840	5.585
Semeion	AVG	100.300	99.733	**93.867**	106.233	101.833	103.700	102.000
	STD	4.808	7.772	6.474	9.035	7.120	7.901	8.056
Colon	AVG	777.833	749.033	766.267	782.000	755.700	755.567	**745.800**
	STD	17.730	16.016	23.601	21.007	18.374	18.834	15.537
Leukemia	AVG	**3045.800**	3063.300	3082.800	3081.133	3074.533	3072.633	3073.333
	STD	36.936	29.744	36.309	31.638	28.472	28.420	26.935
Mean rank	F-test	5.182	**3.045**	3.227	4.909	4.068	3.841	3.727
Overall rank		7	**1**	2	6	5	4	3

updating rules. Eqs. (4) and (5) present the updating rules inside the PSO algorithm.

$$v_i^j(t+1) = \omega_1 v_i^j(t) + c_1 r_1 (pbest_i^j - x_i^j(t)) + c_2 r_2 (gbest_i^j - x_i^j(t)) \quad (4)$$

$$x_i^j(t+1) = x_i^j(t) + v_i^j(t+1) \quad (5)$$

where t refers to iteration number, the initial weight ω_1 is used to control the searching tendencies either for global and local solutions. $v_i^j(t)$ presents the current velocity at iteration t for jth dimension in ith agent, $x_i^j(t)$ is jth dimension in ith particle. Two random numbers between (0, 1) are expressed in terms r_1 and r_2, and c_1 and c_2 present the individual and social factors, respectively. The *pbest* and *gbest* refer to the personal and global best solutions (i.e., agent), respectively. Since each solution will follow *pbest* and *gbest* solutions, PSO shows a fast convergence behavior while performing the exploration and exploitation processes. However, PSO, in some cases, may be trapped in local optima problems when handling complex and multi-modal optimization problems (Sahu, Panigrahi, & Pattnaik, 2012). Algorithm 1 explores the standard pseudo-code of the PSO algorithm.

4. The proposed BPSO algorithm

Most currently available meta-heuristic algorithms have been initially proposed to optimize continuous optimization problems. However, to solve binary optimization problems such as FS, these algorithms need modifications and even adding new operators. The literature shows that there are three major categories of binarization techniques that are used to transform continuous algorithms into binary form. The first category is known as the two-steps binarization techniques, in which the operators of the algorithm stay unaltered. Thus, two

Table 8
p-values of the Wilcoxon test for the number of features obtained by BPSO-TEPD versus other variants in Table 7 ($p \leq 0.05$ are significant and bolded).

Dataset	BPSO	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	**4.13E−08**	**6.09E−13**	**2.90E−13**	**1.31E−03**	**3.45E−07**	**3.49E−07**
BreastEW	**4.49E−03**	1.28E−01	6.52E−01	**1.95E−04**	3.76E−01	**2.57E−02**
CongressEW	8.52E−01	**1.40E−02**	**5.06E−07**	**7.79E−06**	**2.44E−05**	**8.73E−04**
Exactly	**6.30E−04**	NaN	NaN	NaN	NaN	NaN
Exactly2	**1.09E−07**	**4.77E−13**	**4.57E−05**	**6.81E−11**	**6.36E−13**	**5.57E−12**
HeartEW	2.41E−01	**8.79E−10**	**1.27E−03**	**1.16E−05**	**6.89E−04**	1.00E+00
IonosphereEW	6.31E−02	6.43E−01	1.19E−01	**1.24E−02**	**2.97E−02**	6.05E−01
KrvskpEW	6.18E−02	4.85E−01	**3.37E−04**	**1.48E−02**	**1.41E−06**	**9.37E−03**
Lymphography	**4.17E−05**	2.96E−01	**4.42E−07**	**1.08E−04**	**1.90E−06**	**4.58E−03**
M-of-n	**2.14E−02**	NaN	NaN	NaN	NaN	NaN
penglungEW	**2.86E−11**	**5.60E−04**	**2.42E−09**	1.39E−01	**1.05E−05**	**1.71E−06**
SonarEW	8.93E−01	4.29E−01	5.72E−01	4.47E−01	**1.07E−02**	**1.15E−04**
SpectEW	**6.07E−06**	**7.90E−06**	**9.65E−05**	**1.41E−02**	**2.39E−09**	**2.14E−08**
Tic-tac-toe	**4.90E−12**	**3.62E−03**	**8.04E−13**	**1.56E−07**	3.34E−01	3.34E−01
Vote	**7.34E−04**	8.23E−02	3.52E−01	1.73E−01	6.55E−01	**2.11E−03**
WaveformEW	**1.21E−02**	**3.19E−05**	1.07E−01	**1.04E−03**	**5.70E−03**	**2.74E−05**
WineEW	**4.39E−08**	**1.03E−08**	**1.03E−08**	**1.34E−08**	5.69E−01	**1.58E−11**
Zoo	**4.36E−11**	**3.71E−10**	**3.78E−10**	**5.63E−10**	**4.38E−10**	**3.78E−10**
Clean1	**5.98E−05**	9.47E−01	**3.79E−03**	**1.68E−02**	**7.71E−05**	9.27E−02
Semeion	6.78E−01	**1.64E−03**	**1.39E−02**	3.58E−01	7.55E−02	2.22E−01
Colon	**4.25E−07**	**4.21E−03**	**4.42E−07**	2.43E−01	1.10E−01	3.95E−01
Leukemia	**2.28E−02**	**2.37E−02**	**4.92E−02**	2.97E−01	3.04E−01	2.28E−01

Algorithm 1 Pseudo-code of PSO algorithm

```
Initialize random population of particles Pr_i(i = 1, 2, ..., N) inside UB and LB
while Termination condition is not satisfied do
    for Each particle i do
        Calculate the fitness of all particles
        Update the velocity of all particles in the population using Eq. (4)
        Update the position of all particles in the population using Eq. (5)
        Evaluate the fitness f(Pr_i^j)
        if f(Pr_i^j) < f(pBest_i^j) then
            pBest_i^j ← Pr_i^j
        if f(P_i^j) < f(gBest_i^j) then
            gBest_i^j ← Pr_i^j
Return gBest
```

steps are applied in order to transform continuous solutions into binary ones following the original continuous iteration. The second category is named continuous-binary operator transformation, where the operators of the algorithm are reformed. In this category, modifications are made to algebraic operations of the search space (Crawford et al., 2017). Additionally, there is a third binarization category that depends on a clustering approach named K-means Transition Algorithm (KMTA), which was lately proposed by García, Crawford, Soto, and Astorga (2019) as a general binarization method.

Modified Algebraic Operations (MAO) can be categorized into two major approaches; Boolean Approach (BA) that is used to transform the real operators into binary, and Set-Based Approach (SBA) that is utilized for discrete problems. BA has been introduced as an efficient binarization method of PSO for handling various binary optimization problems (Afshinmanesh et al., 2005; Deligkaris et al., 2009; Marandi et al., 2006). Besides, it was applied with bitwise operations to adapt the Artificial Bee Colony (ABC) in a binary search space.

In this paper, we utilized the BA technique that belongs to the MAO method to convert the real operators of PSO into Boolean for tackling the FS problem. As we mentioned earlier, the BA technique applies Boolean operations to convert real search space into binary one (Afshinmanesh et al., 2005; Deligkaris et al., 2009; Marandi et al., 2006). The proposed MAO-based PSO by Afshinmanesh et al. (2005) performed better in terms of convergence speed and local optimum avoidance comparing with Binary PSO using transfer functions (i.e., two-steps binarization technique). However, according to Afshinmanesh et al. (2005), utilizing a transfer function for adapting PSO to deal with binary search space has some drawbacks. For example, the distance updating formula in PSO has no standard form (Kennedy & Eberhart, 1997). Another limitation of transfer function-based Binary PSO is that the changing probability function has no monotonic form. This problem reduces the changing probability for some larger values of V_{id}. In addition, as stated in Islam et al. (2017), the transfer function in the binary PSO algorithm is considered as the primary operator for controlling exploration and exploitation, and therefore using an inappropriate transfer function may significantly reduce the performance of the Binary PSO.

4.1. Boolean PSO (BPSO) with modified algebraic operations for feature selection

The Boolean PSO (BPSO) is a binary variant of PSO. It was initially proposed by Afshinmanesh et al. (2005). In the BPSO, the position (x_i) and the velocity (v_i) of the ith particle are decoded as N-bits binary vector. In this variant, the velocity and position are updated by using three Boolean operators: 'AND' ($\wedge$), 'XOR' ($\oplus$), and 'OR' ($\vee$). The Boolean rules for updating velocity and position are presented in Eqs. (6) and (7), respectively.

$$v_i^j(t+1) = \omega \wedge v_i^j(t) \vee [c_1 \wedge (pbest_i^j \oplus x_i^j(t))] \vee [c_2 \wedge (gbest_i^j \oplus x_i^j(t))] \quad (6)$$

$$x_i^j(t+1) = x_i^j(t) \oplus v_i^j(t+1) \quad (7)$$

where ω, c_1, and c_2 are uniformly distributed random numbers generated to be either 1 or 0. They are arbitrary binary numbers, selected with equal probability to be 0 or 1.

4.2. Formulation of feature selection problem

Adapting optimization algorithms to resolve any problem requires identifying two fundamental parts, including solution representation and fitness (evaluation) function. The central purpose of the FS process is to find the smallest features subset that leads to achieving the maximal classification accuracy. Consequently, FS can be defined as a complex multi-objective optimization problem. Decision-making in a

Table 9
Average fitness results of the proposed variants with KNN classifier.

Dataset	Measure	BPSO	BPSO_TEPD	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	AVG	0.0285	0.0208	0.0392	0.0121	0.0204	**0.0114**	0.0218
	STD	0.0035	0.0006	0.0003	0.0000	0.0000	0.0014	0.0008
BreastEW	AVG	0.0383	0.0155	0.0248	**0.0046**	0.0250	0.0173	0.0118
	STD	0.0061	0.0046	0.0077	0.0039	0.0043	0.0061	0.0058
CongressEW	AVG	0.0271	0.0093	0.0157	0.0048	**0.0022**	0.0080	0.0174
	STD	0.0062	0.0051	0.0100	0.0019	0.0003	0.0044	0.0040
Exactly	AVG	0.0157	**0.0050**	**0.0050**	**0.0050**	**0.0050**	**0.0050**	**0.0050**
	STD	0.0533	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Exactly2	AVG	0.2257	**0.2020**	0.2267	0.2378	0.2402	0.2616	0.2053
	STD	0.0175	0.0032	0.0059	0.0048	0.0072	0.0079	0.0376
HeartEW	AVG	0.1522	0.1189	0.1165	0.1015	0.0990	0.1346	**0.0892**
	STD	0.0217	0.0084	0.0037	0.0143	0.0058	0.0044	0.0103
IonosphereEW	AVG	0.0095	0.0051	0.0259	0.0336	0.0190	0.0150	**0.0045**
	STD	0.0087	0.0055	0.0109	0.0099	0.0116	0.0093	0.0048
KrvskpEW	AVG	0.0267	0.0150	0.0145	0.0153	0.0208	**0.0108**	0.0173
	STD	0.0053	0.0017	0.0015	0.0028	0.0013	0.0041	0.0017
Lymphography	AVG	0.0560	0.0640	**0.0050**	0.0460	0.1086	0.0640	0.0412
	STD	0.0217	0.0181	0.0086	0.0150	0.0123	0.0195	0.0132
M-of-n	AVG	0.0051	**0.0050**	**0.0050**	**0.0050**	**0.0050**	**0.0050**	**0.0050**
	STD	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
penglungEW	AVG	0.0189	**0.0022**	0.0706	0.0136	0.0022	0.0618	0.0201
	STD	0.0292	0.0001	0.0272	0.0249	0.0002	0.0317	0.0295
SonarEW	AVG	0.0119	0.0048	0.0097	**0.0033**	0.0222	0.0140	0.0059
	STD	0.0114	0.0087	0.0107	0.0006	0.0129	0.0119	0.0082
SpectEW	AVG	0.1960	0.1284	0.1532	0.0752	0.1182	**0.0728**	0.1176
	STD	0.0139	0.0058	0.0082	0.0065	0.0083	0.0128	0.0071
Tic-tac-toe	AVG	0.1721	0.2090	**0.1653**	0.1746	0.2006	0.1828	0.1777
	STD	0.0138	0.0021	0.0051	0.0038	0.0027	0.0000	0.0000
Vote	AVG	0.0116	0.0043	**0.0020**	0.0340	0.0326	0.0084	0.0194
	STD	0.0132	0.0047	0.0002	0.0041	0.0069	0.0068	0.0004
WaveformEW	AVG	0.2345	0.2277	**0.2095**	0.2137	0.2191	0.2168	0.2300
	STD	0.0093	0.0070	0.0090	0.0057	0.0063	0.0065	0.0122
WineEW	AVG	0.0045	**0.0028**	0.0034	0.0034	0.0035	0.0029	0.0051
	STD	0.0010	0.0008	0.0002	0.0002	0.0005	0.0007	0.0003
Zoo	AVG	0.0034	0.0021	0.0029	0.0027	**0.0015**	0.0027	0.0027
	STD	0.0005	0.0003	0.0003	0.0000	0.0003	0.0001	0.0000
Clean1	AVG	0.0333	**0.0130**	0.0246	0.0266	0.0448	0.0175	0.0265
	STD	0.0120	0.0089	0.0084	0.0070	0.0094	0.0056	0.0126
Semeion	AVG	0.0082	0.0064	**0.0041**	0.0096	0.0054	0.0123	0.0102
	STD	0.0024	0.0016	0.0012	0.0022	0.0016	0.0016	0.0015
Colon	AVG	0.2882	0.0799	0.1536	0.2705	0.2754	0.2652	**0.0799**
	STD	0.0444	0.0001	0.0423	0.0519	0.0432	0.0383	0.0001
Leukemia	AVG	**0.0043**	0.0043	0.0043	0.0043	0.0703	0.0703	0.0703
	STD	0.0001	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
Mean rank	F-test	5.432	**3.091**	3.636	3.500	4.477	3.977	3.886
Overall rank		7	**1**	3	2	6	5	4

Multi-objective optimization procedure can be accomplished after (posterior), during (interactive), or before (prior) the optimization process. Aggregation is known as one of the most common prior methods where multiple objectives are united into a single objective. Each objective is assigned a weight to decide its significance (Mirjalili & Dong, 2020). Based on that, the two substantial objectives of the FS problem can be combined as presented in Eq. (8) to assess the appropriateness of the selected subset of features.

$$\downarrow Fitness(S) = \alpha \times (1 - \gamma(S)) + \beta \times \frac{R}{T} \tag{8}$$

where $Fitness(S)$ denotes the fitness value of the subset S, $\gamma(S)$ represents the classification accuracy obtained from the S subset of features, R and T denote the number of selected features and the total number of features in the dataset respectively, α represents the weights of the classification accuracy, and β indicates the proportion of feature reduction. $\alpha \in [0,1]$ whereas $\beta = (1-\alpha)$ are adopted from Emary and Zawbaa (2016), Faris et al. (2018) and Mafarja et al. (2017). We are interested in developing an accurate classification model. For this reason, higher importance is given to classification in contrast with the number of selected features. Hence, the value of the parameter α was set 0.99. It was decided based on previous studies conducted by Neggaz, Ewees, Abd Elaziz, and Mafarja (2020) where several experiments were done to determine the optimal value of α.

Representation of the solution for the problem being resolved is another essential aspect that must be taken into consideration. FS is deemed as a binary optimization problem in which the potential solution (i.e., features subset) is formed as a binary vector. Each cell in the vector can hold one of two values (1 or 0). Zero value means that the corresponding feature is discarded, while one means that the corresponding feature is preserved (selected). Fig. 1 presents a pattern of binary solution for a dataset contains a total set of n features. The original real value PSO was adapted into a binary form to enable it to deal with FS problems.

Table 10
p-values of the Wilcoxon test for the fitness results of BPSO-TEPD and other variants in Table 9 ($p \leq 0.05$ are significant and bolded).

Dataset	BPSO	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	**5.57E−12**	**6.09E−13**	**2.90E−13**	**1.31E−03**	**1.44E−12**	**1.87E−08**
BreastEW	**3.14E−11**	**5.86E−05**	**6.62E−10**	**1.66E−10**	4.90E−01	**1.65E−02**
CongressEW	**4.31E−11**	**3.88E−04**	**4.63E−02**	**6.05E−12**	**6.39E−04**	**9.57E−11**
Exactly	**2.98E−04**	NaN	NaN	NaN	NaN	NaN
Exactly2	**1.52E−04**	**1.66E−12**	**1.19E−12**	**1.49E−12**	**8.36E−13**	6.39E−01
HeartEW	**4.44E−09**	**2.91E−02**	**1.11E−08**	**2.11E−09**	**2.07E−10**	**8.34E−10**
IonosphereEW	**4.40E−03**	**2.95E−09**	**6.15E−11**	**6.40E−07**	**1.27E−03**	6.43E−01
KrvskpEW	**3.51E−11**	2.16E−01	7.72E−01	**7.71E−11**	**1.69E−06**	**6.46E−05**
Lymphography	**2.41E−04**	**9.98E−12**	**2.15E−06**	**3.62E−10**	2.18E−01	**1.27E−08**
M-of-n	**2.14E−02**	NaN	NaN	NaN	NaN	NaN
penglungEW	**2.88E−11**	**2.87E−11**	**4.95E−11**	1.39E−01	**6.16E−10**	**6.14E−10**
SonarEW	**6.90E−03**	5.41E−02	6.45E−01	**2.13E−07**	**4.67E−02**	**7.40E−03**
SpectEW	**1.88E−11**	**1.74E−11**	**7.16E−12**	**3.50E−05**	**1.96E−11**	**1.10E−03**
Tic-tac-toe	**4.61E−13**	**4.53E−13**	**6.50E−14**	**1.32E−13**	**2.71E−14**	**2.71E−14**
Vote	**2.51E−04**	**2.77E−03**	**7.33E−12**	**1.28E−11**	**2.29E−04**	**1.18E−11**
WaveformEW	**6.09E−03**	**6.69E−10**	**2.32E−09**	**3.25E−05**	**6.76E−07**	3.83E−01
WineEW	**4.39E−08**	**1.03E−08**	**1.03E−08**	**1.34E−08**	5.69E−01	**1.58E−11**
Zoo	**4.36E−11**	**3.71E−10**	**3.78E−10**	**5.63E−10**	**4.38E−10**	**3.78E−10**
Clean1	**5.19E−09**	**1.29E−05**	**5.27E−08**	**1.26E−10**	**9.49E−04**	**1.24E−05**
Semeion	**8.93E−04**	**8.85E−08**	**1.45E−07**	7.58E−02	**4.00E−11**	**1.01E−09**
Colon	**2.99E−11**	**3.62E−10**	**2.99E−11**	**3.00E−11**	**3.00E−11**	3.95E−01
Leukemia	**2.28E−02**	**2.37E−02**	**4.92E−02**	**3.01E−11**	**3.01E−11**	**3.00E−11**

4.3. Evolutionary population dynamics (EPD)

Evolutionary Algorithms (EAs) are considered as stochastic search approaches in which a group of potential solutions called population is generated with random initial values and then progressively enhanced to become better adapted to the objectives demanded from them. Several EAs employ mutation techniques to modify the elected solutions, whereas others utilize crossover operators. These operators are applied to evolve the leading selected solutions that are most likely the optimal solutions. EPD can be defined as the process of discarding the worst solutions in the population via displacing them around the optimal solutions in the population. The EPD is essentially based on self-organized criticality (SOC) theory (Bak et al., 1987) where a local modification in the population could influence any individual in the population and gives delicate balances without the involvement of any external organizing force (Lewis et al., 2008). In GA, the evolutionary operators crossover and mutation are used to combine the best solutions found by the GA algorithm. On the other hand, in the EPD, the worst solutions are removed from the current generation. EPD is known as a simple and robust mechanism that can be integrated with population-based meta-heuristic algorithms. It begins by eliminating poor solutions from the population and then relocating them close to the optimal search agents.

In the this research, the removing worst solutions from the swarm are re-positioning around the best search agents (guide solutions). The guide solution is selected based on different natural selection criteria.

4.4. Natural selection mechanisms for EPD

The selection mechanism is considered a crucial part for any searching algorithm that brings a sufficient balance between intensification and diversification. The literature reveals that different techniques have been proposed for implementing selection mechanisms. The most popular approaches that were hired in this work comprising: Tournament (T), Linear rank (LR), Roulette wheel (RW), Random-based (R), Stochastic universal sampling (SUS), and best-based (B).

Selection approaches are basically based on the "survival of the best" principle (Talbi, 2009), where the most suited or fittest solution has a more prominent possibility of being picked and hence provides preferable exploitation of the optimal solution (intensification). Nevertheless, worst or unfavorable solutions are not omitted and have less opportunity to be picked, which leads to the exploration of the space (diversification). The critical factor that influences the capability of the selection process is named 'selective pressure'. It can be defined as the trend to pick the most suited (fittest) members of the present population (Back, 1994). Therefore, the balance between intensification and diversification can be affected by the quantity of selective pressure. This means that excessive pressure will lead to a bias in the direction of fittest solutions, and therefore, causes a loss of diversity and premature convergence. In contrast, a small amount of selective pressure can preserve diversity and lowers the speed of convergence.

4.4.1. Best-based selection method

The best-based method, also called Elitist-based, considers the fittest solution only by setting its selection probability equal to 1, while other solutions are neglected, and their selection probabilities are equal to 0. The best-based mechanism generates a population with properties derived from the optimal solution, and that will accelerate the convergence behavior of the algorithm (Mafarja et al., 2020; Razali, Geraghty, et al., 2011). Yet, large exploitation in this mechanism could cause a premature convergence issue (Mafarja et al., 2020).

4.4.2. Roulette wheel selection method

RW mechanism was firstly proposed by Holland (1975) GA. The prime concept of this mechanism is that every solution in the population has a non-zero chance to be picked. RW estimates the selection probability of the ith individual (p_i) in the population pool in proportion to its absolute fitness value ($f(x_i)$) divided by the total of fitness value for all individuals as in Eq. (9).

$$p_i = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)} \tag{9}$$

where N represents the population size, and $f(x_i)$ is estimated using Eq. (10) for any minimization optimization problem (Al-Betar et al., 2017).

$$f(x_i) = \frac{1}{1 + f(x_i)} \tag{10}$$

RW process can be viewed as a spinning roulette wheel that is partitioned into segments or portions with non-identical sizes. Each individual holds a portion that is proportional to its fitness value. Thus, the fittest individuals that occupy large portions have a higher chance of being picked than poor ones (i.e., individuals that occupy small segments).

The major characteristic of the RW selection mechanism compared to the best-based selection mechanism is that each individual in the

Table 11
Average running time results of the proposed variants with KNN classifier.

Dataset	Measure	BPSO	BPSO_TEPD	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Breastcancer	AVG	13.850	16.610	**13.544**	13.975	14.186	13.874	14.635
	STD	0.637	1.727	0.310	0.371	0.460	0.636	0.446
BreastEW	AVG	**14.298**	14.697	14.496	14.535	14.490	14.560	14.424
	STD	0.629	0.331	0.349	0.441	0.367	0.545	0.685
CongressEW	AVG	12.245	12.566	12.657	12.934	12.028	**12.021**	12.578
	STD	0.540	0.502	0.400	0.405	0.400	0.463	0.432
Exactly	AVG	**16.783**	17.789	17.359	17.532	18.604	17.046	18.818
	STD	0.738	0.422	0.551	0.351	0.395	0.406	0.525
Exactly2	AVG	18.104	23.018	**16.472**	22.227	18.722	16.868	19.017
	STD	4.791	0.873	1.466	1.210	2.400	0.692	3.189
HeartEW	AVG	**10.879**	11.072	11.244	11.139	11.337	11.575	11.326
	STD	0.271	0.218	0.326	0.225	0.336	0.309	0.373
IonosphereEW	AVG	11.775	11.868	11.964	11.779	**11.617**	12.126	11.871
	STD	0.527	0.478	0.319	0.303	0.442	0.299	0.334
KrvskpEW	AVG	172.472	173.554	178.107	179.187	168.602	182.418	**167.354**
	STD	21.461	11.519	11.589	8.390	11.940	11.412	14.933
Lymphography	AVG	10.712	**10.465**	10.570	11.015	10.563	10.793	10.860
	STD	0.401	0.289	0.271	0.271	0.261	0.382	0.343
M-of-n	AVG	16.928	17.483	17.219	**16.219**	16.328	17.271	17.464
	STD	0.539	0.540	0.343	0.559	0.333	0.457	0.551
penglungEW	AVG	**11.376**	11.939	11.813	11.922	11.843	11.823	12.099
	STD	0.289	0.215	0.282	0.309	0.278	0.375	0.512
SonarEW	AVG	**10.390**	10.922	10.915	10.895	10.823	11.000	10.605
	STD	0.278	0.452	0.288	0.290	0.289	0.372	0.253
SpectEW	AVG	11.287	**11.263**	11.474	11.390	11.314	11.508	11.478
	STD	0.420	0.344	0.312	0.381	0.363	0.387	0.377
Tic-tac-toe	AVG	19.153	17.310	**16.263**	18.469	17.686	16.756	16.987
	STD	0.728	0.580	0.589	0.593	0.965	0.506	0.374
Vote	AVG	**11.169**	11.368	11.498	11.549	11.591	11.514	11.744
	STD	0.386	0.312	0.241	0.378	0.353	0.345	0.337
WaveformEW	AVG	431.968	454.388	**428.325**	447.341	436.910	440.714	432.290
	STD	41.200	24.772	32.251	29.730	24.418	28.975	28.555
WineEW	AVG	**10.650**	10.683	10.760	11.007	10.716	10.850	10.977
	STD	0.340	0.340	0.425	0.332	0.433	0.329	0.475
Zoo	AVG	10.896	10.961	11.110	11.212	**10.566**	11.022	11.151
	STD	0.370	0.234	0.365	0.361	0.282	0.354	0.313
Clean1	AVG	**20.525**	21.165	21.773	21.843	21.613	22.477	21.578
	STD	0.729	0.869	0.865	0.847	0.709	0.724	0.582
Semeion	AVG	**181.144**	192.061	189.798	195.035	192.577	194.607	192.824
	STD	9.020	7.352	7.011	7.814	7.811	7.404	7.414
Colon	AVG	**14.900**	15.751	16.014	16.337	15.937	15.985	15.430
	STD	0.445	0.529	0.499	0.541	0.466	0.303	0.605
Leukemia	AVG	**42.765**	46.287	46.675	47.680	47.168	46.807	44.669
	STD	3.682	4.332	4.124	4.422	4.491	4.247	3.689
Mean rank	F-test	**2.000**	4.091	3.636	5.318	3.636	4.773	4.545
Overall rank		**1**	3	2	6	2	5	4

Table 12
Overall rank by the F-test for the proposed variants based on accuracy, number of features, fitness, and running time.

Measure	BPSO	BPSO_TEPD	BPSO_RWEPD	BPSO_LREPD	BPSO_REPD	BPSO_SEPD	BPSO_BEPD
Accuracy	5.068	**3.364**	3.636	3.455	4.568	4.114	3.795
Features	5.182	**3.045**	3.227	4.909	4.068	3.841	3.727
Fitness	5.432	**3.091**	3.636	3.500	4.477	3.977	3.886
Time	**2.000**	4.091	3.636	5.318	3.636	4.773	4.545
Average rank	4.420	**3.398**	3.534	4.296	4.187	4.176	3.989
Final rank	7	**1**	2	6	5	4	3

population has a chance to be picked. Hence, this will help in preserving the diversity of the population. Although, prominent solutions have high selection pressure, which may cause a bias in the direction of the optimal solutions, particularly in the initial phases of the search process, and consequently results in the problem of sliding into local optima. In addition, during the convergence of the population, when there are individuals with the same fitness values, it is hard to differentiate a higher quality individual (solution) (Eiben & Smith, 2003).

4.4.3. Linear rank-based selection method

Linear rank-based selection mechanism (LR) was introduced to overtake the drawbacks of RW selection mechanism (Baker, 1985). The primary idea of LR selection is that the selection probability of each

Table 13
The performance of BPSO-TEPD with KNN and DT classifiers in terms of accuracy, number of features, fitness, and running time.

Dataset	Measure	Accuracy		Features		Fitness		Time	
		KNN	DT	KNN	DT	KNN	DT	KNN	DT
Breastcancer	AVG	**0.986**	0.971	5.300	**2.000**	**0.021**	0.031	**16.610**	42.878
	STD	0.000	0.000	0.466	0.000	0.001	0.000	1.727	0.617
BreastEW	AVG	**0.987**	0.969	8.767	**5.967**	**0.015**	0.032	**14.697**	45.880
	STD	0.005	0.005	1.612	1.629	0.005	0.005	0.331	0.716
CongressEW	AVG	**0.994**	0.989	4.900	**3.000**	**0.009**	0.013	**12.566**	40.899
	STD	0.006	0.000	1.348	0.000	0.005	0.000	0.502	0.401
Exactly	AVG	**1.000**	0.992	**6.000**	**6.000**	**0.005**	0.013	**17.789**	46.304
	STD	0.000	0.044	0.000	0.000	0.000	0.043	0.422	0.566
Exactly2	AVG	**0.804**	0.765	9.900	**7.067**	**0.202**	0.239	**23.018**	49.480
	STD	0.004	0.007	0.548	1.143	0.003	0.006	0.873	1.595
HeartEW	AVG	**0.884**	0.860	**4.867**	5.033	**0.119**	0.142	**11.072**	41.696
	STD	0.008	0.009	1.279	0.964	0.008	0.009	0.218	0.671
IonosphereEW	AVG	**0.998**	0.962	9.033	**7.733**	**0.005**	0.040	**11.868**	46.471
	STD	0.005	0.009	2.076	1.143	0.005	0.008	0.478	0.976
KrvskpEW	AVG	0.990	**0.994**	**17.633**	18.400	0.015	**0.012**	173.554	**85.706**
	STD	0.002	0.001	2.619	0.814	0.002	0.001	11.519	2.167
Lymphography	AVG	0.939	**0.966**	**5.333**	6.000	0.064	**0.038**	**10.465**	41.594
	STD	0.018	0.006	1.348	0.000	0.018	0.006	0.289	0.613
M-of-n	AVG	**1.000**	**1.000**	**6.000**	**6.000**	**0.005**	**0.005**	**17.483**	44.832
	STD	0.000	0.000	0.000	0.000	0.000	0.000	0.540	0.608
penglungEW	AVG	**1.000**	0.961	**70.033**	85.033	**0.002**	0.042	**11.939**	53.817
	STD	0.000	0.041	3.891	7.577	0.000	0.040	0.215	1.105
SonarEW	AVG	**0.998**	0.963	19.067	**12.200**	**0.005**	0.038	**10.922**	47.566
	STD	0.009	0.018	2.288	1.827	0.009	0.018	0.452	1.205
SpectEW	AVG	0.873	**0.909**	**5.200**	7.633	0.128	**0.093**	**11.263**	43.169
	STD	0.006	0.006	1.584	1.217	0.006	0.005	0.344	1.524
Tic-tac-toe	AVG	0.797	**0.864**	6.033	**6.000**	0.209	**0.142**	**17.310**	49.462
	STD	0.002	0.002	0.183	0.000	0.002	0.002	0.580	1.255
Vote	AVG	**0.998**	0.967	**3.967**	4.000	**0.004**	0.036	**11.368**	40.761
	STD	0.005	0.000	1.938	0.000	0.005	0.000	0.312	0.470
WaveformEW	AVG	0.776	**0.792**	22.867	**19.667**	0.228	**0.211**	454.388	**304.205**
	STD	0.007	0.009	1.961	2.537	0.007	0.009	24.772	20.898
WineEW	AVG	**1.000**	**1.000**	3.333	**3.033**	0.003	**0.003**	**10.683**	41.070
	STD	0.000	0.000	0.922	0.183	0.001	0.000	0.340	1.132
Zoo	AVG	**1.000**	**1.000**	3.200	**3.033**	0.002	**0.002**	**10.961**	40.561
	STD	0.000	0.000	0.407	0.183	0.000	0.000	0.234	0.635
Clean1	AVG	**0.990**	0.969	55.167	**50.433**	**0.013**	0.034	**21.165**	184.742
	STD	0.009	0.014	5.742	4.599	0.009	0.014	0.869	8.953
Semeion	AVG	**0.997**	0.976	99.733	**89.933**	**0.006**	0.027	192.061	**169.072**
	STD	0.002	0.004	7.772	6.108	0.002	0.004	7.352	7.958
Colon	AVG	**0.923**	0.859	**749.033**	765.433	**0.080**	0.143	**15.751**	88.605
	STD	0.000	0.029	16.016	18.926	0.000	0.029	0.529	3.165
Leukemia	AVG	**1.000**	**1.000**	**3063.300**	3083.233	**0.004**	0.004	**46.287**	178.919
	STD	0.000	0.000	29.744	31.698	0.000	0.000	4.332	14.485
Rank	W|T|L	13|4|5	5|4|13	8|2|12	12|2|8	14|1|7	7|1|14	19|0|3	3|0|19

particle is decided depending on the rank of the individual instead of its absolute fitness value. Each particle obtains a selection probability according to a linear mapping function, as given in Eq. (11).

$$p_j = \frac{1}{N} \times (\eta^+ - (\eta^+ - \eta^- \times \frac{j-1}{N-1})) \tag{11}$$

where j indicates the rank of the solution, η^+ denotes the expected value of the optimal particle (solution) in the swarm, and it is equal to $(N \times P_1)$, whereas η^- denotes the expected value the worst particle (solution), and it is equal to $(N \times P_N)$. P_1 and P_N denote the optimal and worst solutions respectively. η^+ and η^- are used to determine the slope of the linear function. The selective pressure of LR relies on the value of η^+ where a larger value implies a larger selective pressure (Baker, 1985).

After calculating the selection probabilities, the selection procedure is done by the roulette wheel mechanism. It is remarkable that relying on the rank rather than fitness values avoids the controlling of prominent solutions because all particles in the population have equal selection probabilities, hence preventing the appearance of a premature convergence problem. Although, this mechanism might decrease the convergence speed. In addition, in terms of computation, the LR selection method is considered costly since the swarm requires sorting in each generation (Razali et al., 2011).

4.4.4. Stochastic universal sampling method

The stochastic Universal Sampling (SUS) selection mechanism is an extended form of the proportional selection mechanism proposed by Baker (1987) in 1987. the basic idea of SUS is to determine a selection probability for each particle based on its fitness value belonged to the total fitness values of all particles of the current generation. Simply, in the RW method, the wheel is rotated N times to pick N members (particles), whereas SUS spins the wheel only one time to pick N members (Baker, 1987). The main drawback of this method is that

once the swarm is converged, it becomes difficult for SUS to behave properly (Al-Betar et al., 2018).

4.4.5. Tournament selection method

The tournament selection mechanism was introduced by Goldberg, Korb, and Deb (1990) in 1989. It is known as simple, powerful, and the most common selection approach in EAs (Goldberg & Deb, 1991). It is deemed as a two-step selection approach. The first step begins with a random selection of a group of k individuals (particles) from the current population, where k represents the size of the tournament. Then, the fittest solution (particle) among the dominant ones in the tournament is picked. For each individual in the selected group, a selection probability is calculated based on Eq. (12).

$$p_j = \frac{1}{N^t} \times [(N - j + 1)^t - (N - j)^t] \tag{12}$$

The main advantage of the tournament selection method is that it maintains the diversity of the population via assigning for each individual (particle) an equal probability of being chosen for guiding the poor solutions, even though this might lower the speed of convergence. Also, the tournament size (k) is a crucial operator that is responsible for controlling the selection pressure and hence, compromising between the exploitative and exploitative capabilities (Back, 1994; Mafarja et al., 2017). Larger values of k lead to higher selection pressure and, therefore, will produce a bias in the direction of the best solutions (intensification). On the other hand, lower values of k provide less selection pressure, and that will switch the search process towards a random manner (diversification). However, deciding the favorable value of k parameter is a complicated task, and heavily relies on the type of the problem being dealt with (Al-Betar et al., 2017; Hassouneh et al., 2021).

4.4.6. Random-based selection method

The random-based selection approach picks a solution (particle) from the current population in a random way. All solutions have an equal chance to be selected despite their fitness values. Accordingly, the selection pressure is equalized. Generally, this method causes slow convergence (Hassouneh et al., 2021).

4.5. BPSO with EPD strategy

As mentioned above, the EPD eliminates the worst solutions from the swarm and replaces them by generating neighbor solutions around the good ones. It is an efficient operator for population-based algorithms. Therefore, it is easily applied to the BPSO, which is a population-based algorithm. In simple, the process starts by sorting the swarm based on the fitness value and then divided into two halves. The worst half of the swarm is eliminated and re-positioned using six different selection operators based on the good half of the swarm. Algorithm 2 presents the proposed evolutionary population dynamics based Boolean PSO. Moreover, Algorithm 3 illustrates the tournament selection based Boolean PSO approach.

Each solution in the worst half is re-positioned around the guide solution, which is selected using six different selection strategies. So, six variants of BPSO-EPD were presented:

- BPSO_BEPD: The best solution obtained so far is considered as a guide solution.
- BPSO_RWEPD: Roulette wheel selection method is used to identify the guide solution.
- BPSO_LREPD: The guide solution is selected using the Linear rank selection mechanism.
- BPSO_SEPD: The guide solution is selected using SUS method.
- BPSO_TEPD: Tournament selection method is involved to select the guide solution.
- BPSO_REPD: Random solution from the best half is selected as a guide.

Algorithm 2 Pseudo-code of BPSO-EPD algorithm

Initialize random population of particles $Pr_i (i = 1, 2, \ldots, N)$ inside UB and LB
while Termination condition is not satisfied **do**
  for Each particle i **do**
    Calculate the fitness of all particles
    Update the velocity of all particles in the population using Eq. (6)
    Update the position of all particles in the population using Eq. (7)
    Evaluate the fitness $f(Pr_i^j)$
    if $f(Pr_i^j) < f(pBest_i^j)$ **then**
      $pBest_i^j \leftarrow Pr_i^j$
    if $f(P_i^j) < f(gBest_i^j)$ **then**
      $gBest_i^j \leftarrow Pr_i^j$
  Sort the population based on the fitness value
  Select guide solution $gSol$ from the best half of the population using EPD strategy
  for $i = (\frac{N}{2}) + 1$ to N **do**
    Apply single point crossover to re-position ith particle around $gSol$.
Return $gBest$

Algorithm 3 Pseudo-code of BPSO_TEPD algorithm

Initialize random population of particles $Pr_i (i = 1, 2, \ldots, N)$ inside UB and LB
while Termination condition is not satisfied **do**
  for Each particle i **do**
    Calculate the fitness of all particles
    Update the velocity of all particles in the population using Eq. (6)
    Update the position of all particles in the population using Eq. (7)
    Evaluate the fitness $f(Pr_i^j)$
    if $f(Pr_i^j) < f(pBest_i^j)$ **then**
      $pBest_i^j \leftarrow Pr_i^j$
    if $f(P_i^j) < f(gBest_i^j)$ **then**
      $gBest_i^j \leftarrow Pr_i^j$
  Sort the population based on the fitness value
  // Selection of gSol (guide solution) using tournament selection
  Identify the tournament size k
  m = generate random index within [0, N]
  set optimal = m
  set j = 2
  while $(j \leq k)$ **do**
    m = generate random index within [0, N]
    if (fitness(X_m) < fitness($X_{optimal}$)) **then**
      optimal = m
    j = j + 1
  gSol = optimal
  for $i = (\frac{N}{2}) + 1$ to N **do**
    Apply single point crossover to re-position ith particle around $gSol$.
Return $gBest$

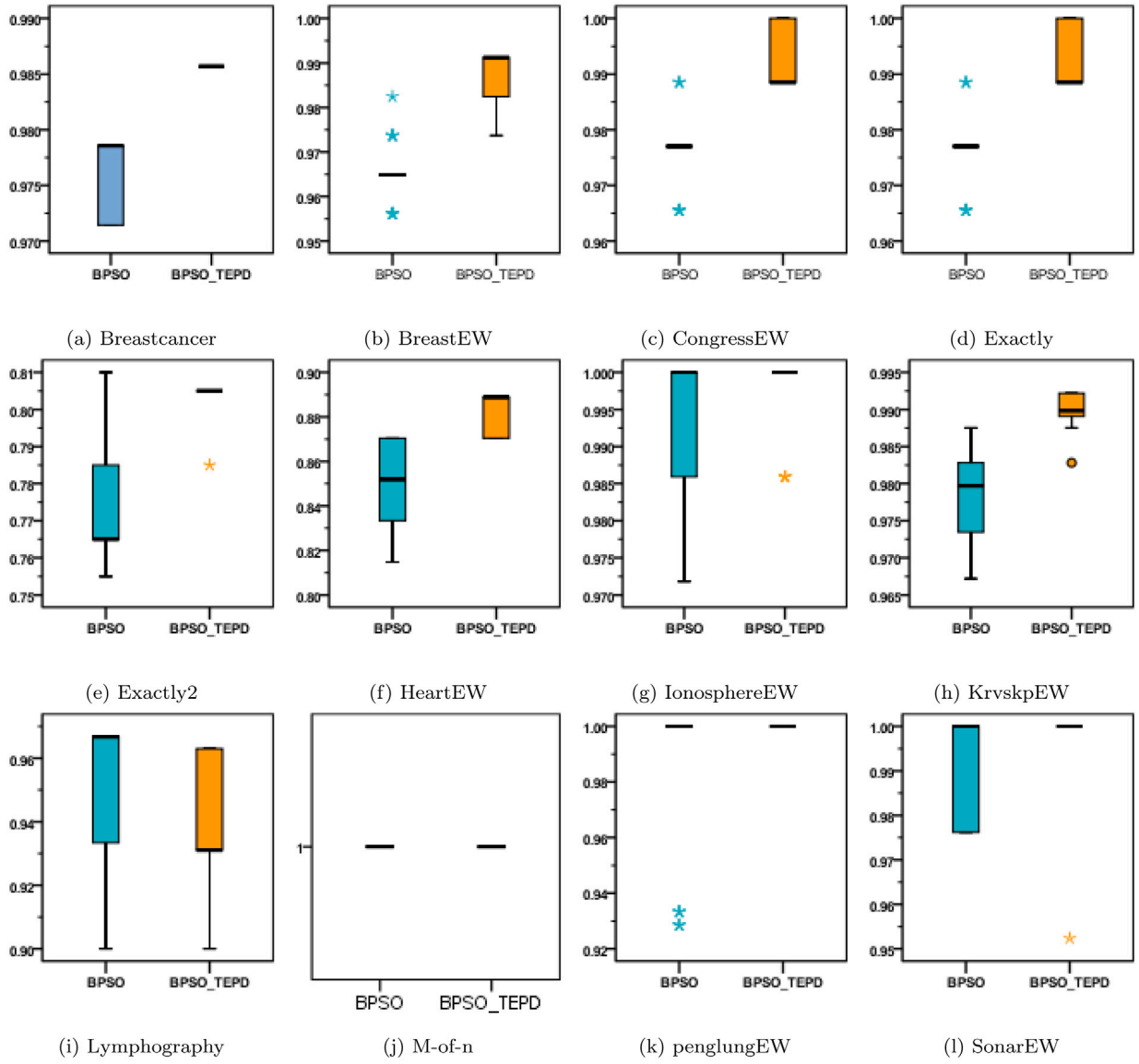

Fig. 4. Box-Plots of classification accuracy for BPSO-TEPD versus BPSO in dealing with the first 12 datasets.

After identifying the guide solution for each poor one, two evolutionary operators (mutation and crossover) are utilized to generate a new solution. Firstly, a simple mutation operator with adaptive probability (that is linearly decreases over the curse of iterations) is applied to the guide solution to find a better solution near the selected solution and escape the premature convergence. Afterward, the poor solution is re-positioned around the resulting solution via employing a single-point crossover operator. Fig. 2 illustrates the proposed improvement of BPSO algorithm. It shows how the EPD strategy removes the worst solutions (particles) from the population based on their fitness values, and replaces them with neighbor solutions around guide ones generated by a selection mechanism (e.g., roulette wheel selection). In addition, Fig. 3 shows the flowchart of the proposed BPSO-EPD approach.

5. Experimental results and simulations

5.1. Experimental setup

To study the effectiveness of the proposed binary BPSO variants, 22 well-regarded datasets obtained from UCI repository (Lichman, 2013) are utilized here. These problems are chosen carefully with various details and properties (e.g., number of features, instances, and classes) to cover varied types of real-life tasks. Table 1 provides a concise description for each employed dataset.

In this work, we followed the hold-out method to estimate the performance of used FS approaches. Each dataset is split into two portions, where 80% of the data was preserved for training the machine learning model while the rest of 20% of the data was employed to evaluate how well that model performs on unseen data. This method was chosen according to many previous studies related to FS problems (Faris et al., 2018; Mafarja et al., 2020; Mafarja & Mirjalili, 2017; Thaher et al., 2021) to make the results comparable. Due to the stochastic nature of meta-heuristic algorithms, each experiment was repeated for 30 trials with a randomly set seed. Accordingly, all the statistics are presented in terms of the average value (AVG) and the standard deviation (STD). Besides, Two non-parametric statistical tests, namely Wilcoxon rank-sum and Friedman, were performed with 5% degree of significance to show the significance of the results. The interest in non-parametric statistical analysis has grown recently in the field of computational intelligence (Derrac, García, Molina, & Herrera, 2011).

To study the impact of the proposed EPD approaches, we provide a comprehensive comparison between the results of the basic BPSO and six BPSO variants (BPSO-TEPD, BPSO-RWEPD, BPSO-LREPD,

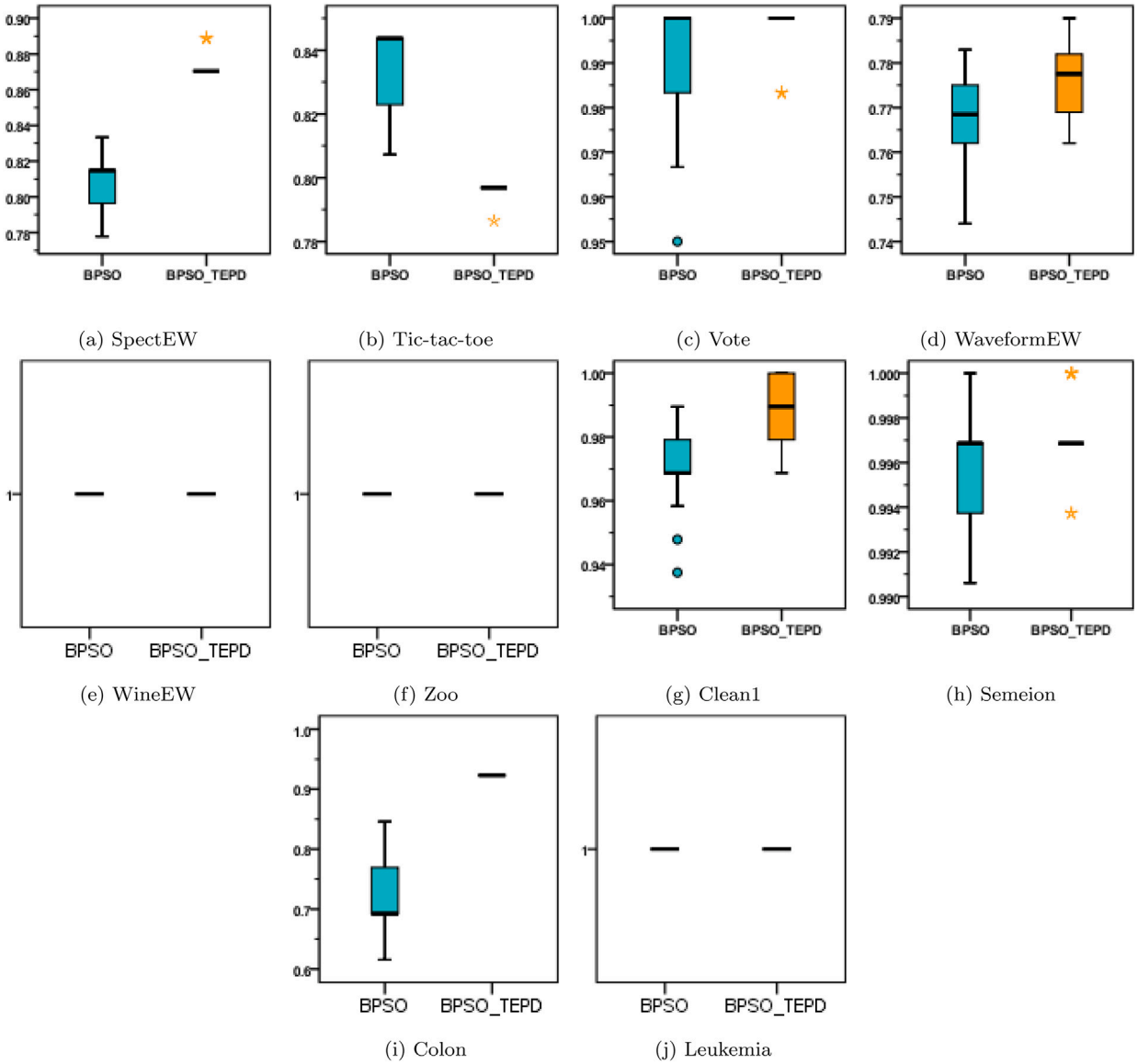

Fig. 5. Box-Plots of classification accuracy for BPSO-TEPD versus BPSO in dealing with the last 10 datasets.

BPSO-REPD, BPSO-SEPD, and BPSO-BEPD). Furthermore, the best FS approach among tested BPSO variants was then compared to other well-known FS approaches comprising two Binary variants of PSO using S-shaped transfer function (SBPSO) and V-shaped transfer function (VBPSO), Binary Grasshopper Optimization Algorithm (BGOA), Binary Gravitational Search Algorithm (BGSA), Binary Ant Lion Optimizer (BALO), Binary Bat algorithm (BBA), Binary Salp Swarm Algorithm (BSSA), Whale Optimization Algorithm (BWOA), and Binary Teaching-Learning Based Optimization (BTLBO). The maximum number of iterations is set to 100 and the population size is 20. These common parameters were set based on the results of several trials. Four criteria were used for comparison: classification accuracy, number of selected features, fitness values, and the running time. For the fitness formula in Eq. (8), α is set to 0.99 and β is set to 0.01 according to recommended settings in related works (Chantar, Thaher, Turabieh, Mafarja, & Sheta, 2021; Faris et al., 2018; Thaher & Arman, 2020). Please note that in all reported tables, the best-obtained results are highlighted using a boldface format.

5.2. System details

All experiments in this study are coded in MATLAB 2018 licensed software under a same computing system. Table 2 shows the details of the system and used environments.

5.3. Assessment of parameters settings

In the first part of the experiment, we investigate and analyze the impact of the parameter settings on proposed approaches.

5.3.1. Impact of common parameters on BPSO

Firstly, we study the effect of common parameters (population size and maximum iterations) on classification accuracy. Table 3 depicts the average accuracy obtained by BPSO with different combinations of the common parameters. It can be seen that the optimal ranking of 5.68 was achieved when population = 20 and maximum iterations = 100. The results imply that neither a lower nor higher population size and maximum iterations can benefit the algorithm. Hence, a sensitivity analysis on these common parameters is critically important.

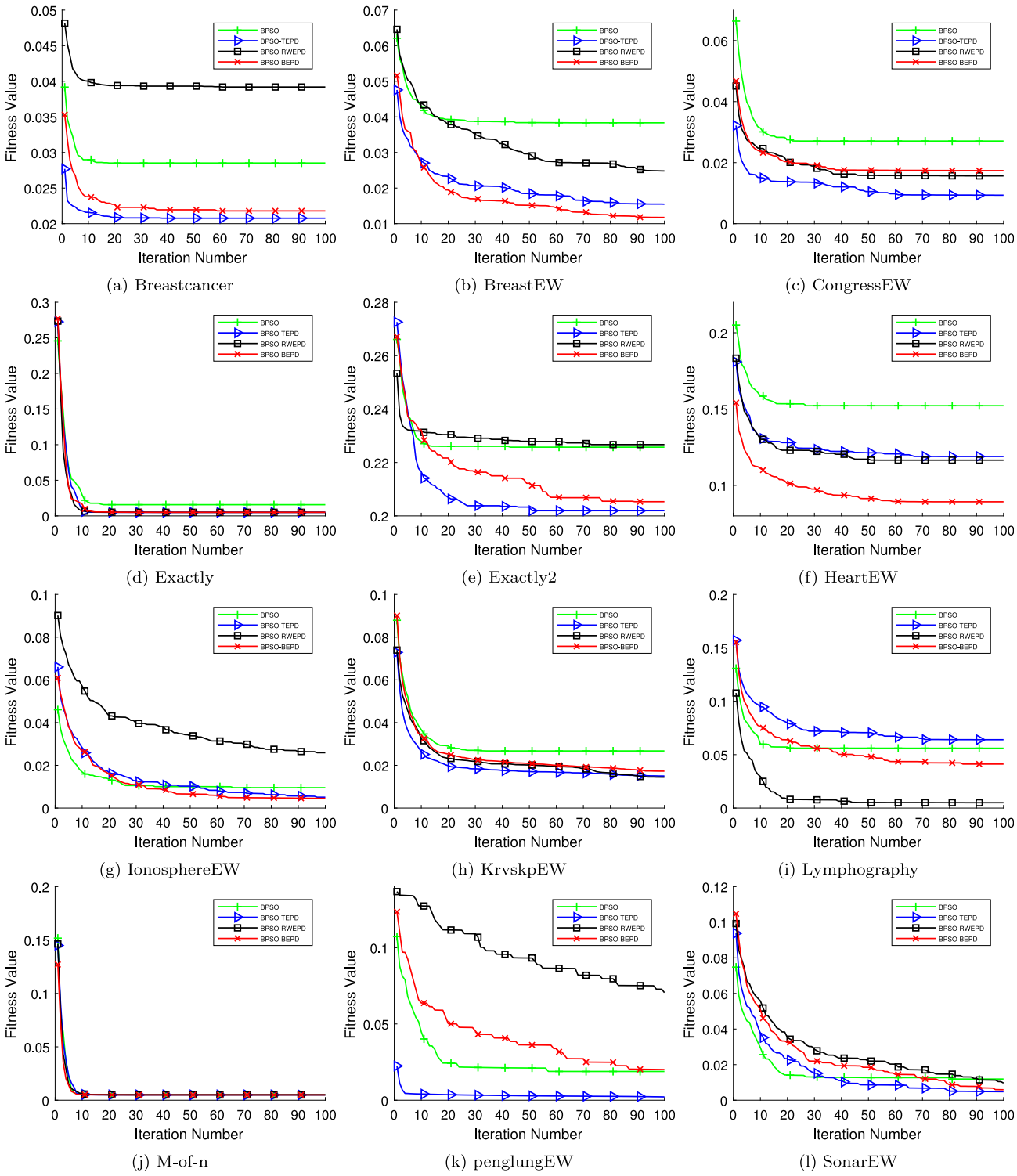

Fig. 6. Convergence curves of the top three variants versus the original BPSO for the first 12 datasets.

5.3.2. Impact of tournament size on BPSO-TEPD

Secondly, we study the effect of the tournament size. Table 4 outlines the average accuracy obtained by BPSO-TEPD with various tournament sizes (10%, 30%, 50%, 70%, and 85%). From Table 4, BPSO-TEPD with 50% tournament size perceived the optimal accuracy in 12 datasets, which showed better classification results. Hence, BPSO-TEPD with 50% tournament size was used in the rest of this work.

5.3.3. Impact of parameter (n^+) on BPSO-LREPD

Thirdly, we analyze the impact of the parameter (n^+) on BPSO-LREPD. Based on the results obtained in Table 4, the best accuracy was perceived by $n = 1.7$ and $n = 1.9$ in most datasets. However, BPSO-LREPD ($n = 1.9$) scored a higher ranking of 2.73 in which a promising performance can be ensured.

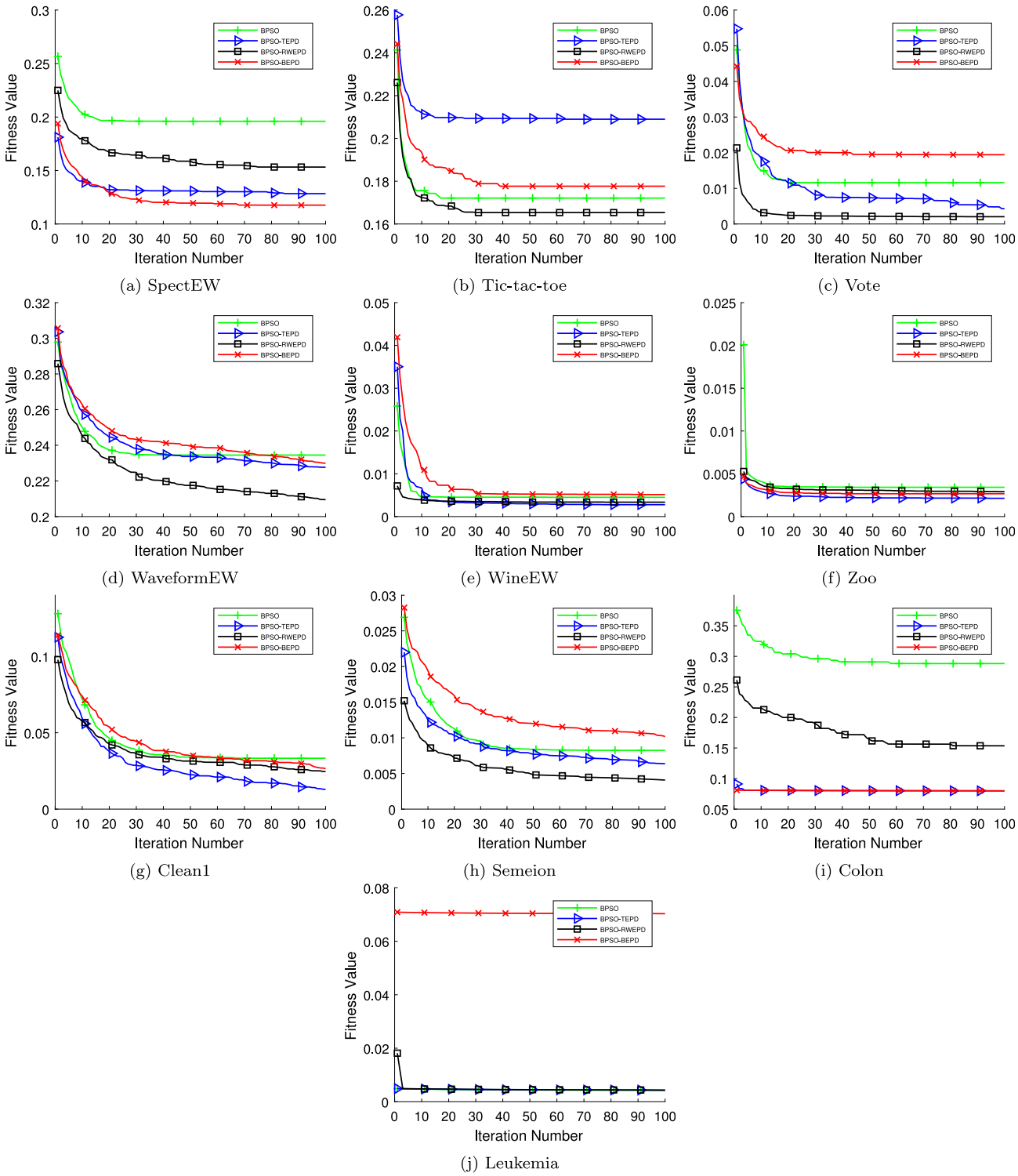

Fig. 7. Convergence curves of the top three variants versus the original BPSO for the last 10 datasets.

5.4. Results of different variants with KNN classifier

In the previous section, the sensitivity analysis of the proposed approaches have been studied. This section aims to investigate the efficiency of the six proposed BPSO variants (BPSO-TEPD, BPSO-RWEPD, BPSO-LREPD, BPSO-REPD, BPSO-SEPD, and BPSO-BEPD) in solving the FS tasks. The accuracy results of the six proposed variants are shown in Table 5. In Table 5, the AVG denotes the average result while the STD presents the consistency and robustness of the algorithm in feature selection. Correspondingly, small standard deviation suggests that the algorithm is highly consistent. Among the proposed approaches, the best approach was found to be BPSO-TEPD. As compared to the conventional BPSO, the BPSO-TEPD can often yield a higher accuracy, which enables an accurate classification process.

Inspecting the boxplot results in Figs. 4 and 5, the proposed BPSO-TEPD achieved the highest median value in most cases. The results

Table 14
Comparison between BPSO-TEPD and other optimizers based on average accuracy results.

Dataset	Measure	BPSO_TEPD	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Breastcancer	AVG	0.9857	0.9929	0.9807	0.9929	0.9721	0.9836	0.9395	0.9857	0.9641	**1.0000**
	STD	0.0000	0.0000	0.0065	0.0000	0.0051	0.0033	0.0209	0.0000	0.0040	0.0000
BreastEW	AVG	0.9874	**0.9959**	0.9924	0.9751	0.9617	0.9754	0.9357	0.9658	0.9629	0.9836
	STD	0.0050	0.0045	0.0045	0.0052	0.0081	0.0043	0.0227	0.0048	0.0107	0.0045
CongressEW	AVG	0.9939	**0.9981**	0.9931	0.9885	0.9605	0.9824	0.8977	0.9839	0.9594	0.9678
	STD	0.0058	0.0044	0.0057	0.0000	0.0089	0.0066	0.0635	0.0057	0.0094	0.0047
Exactly	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7582	0.9980	0.6217	**1.0000**	0.8082	0.9995
	STD	0.0000	0.0000	0.0000	0.0000	0.0952	0.0048	0.0861	0.0000	0.1307	0.0015
Exactly2	AVG	**0.8043**	0.7648	0.7497	0.7822	0.7450	0.7690	0.6807	0.7397	0.7488	0.7700
	STD	0.0037	0.0119	0.0074	0.0130	0.0120	0.0067	0.0384	0.0089	0.0096	0.0000
HeartEW	AVG	0.8840	0.9235	0.8920	0.9043	0.8321	**0.9377**	0.6833	0.8858	0.8333	0.9080
	STD	0.0083	0.0064	0.0195	0.0070	0.0275	0.0157	0.0732	0.0098	0.0344	0.0133
IonosphereEW	AVG	**0.9977**	0.9808	0.9751	0.9803	0.9601	0.9826	0.8831	0.9418	0.9812	0.9808
	STD	0.0053	0.0069	0.0109	0.0088	0.0129	0.0061	0.0282	0.0071	0.0093	0.0094
KrvskpEW	AVG	0.9899	0.9852	0.9846	**0.9911**	0.9141	0.9798	0.7916	0.9741	0.9505	0.9686
	STD	0.0022	0.0022	0.0044	0.0021	0.0270	0.0020	0.1094	0.0035	0.0314	0.0048
Lymphography	AVG	0.9385	**0.9844**	0.9766	0.9199	0.7833	0.8889	0.7692	0.8911	0.9142	0.9065
	STD	0.0180	0.0169	0.0305	0.0157	0.0227	0.0182	0.0809	0.0174	0.0208	0.0133
M-of-n	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.8797	0.9993	0.7458	**1.0000**	0.9447	0.9997
	STD	0.0000	0.0000	0.0000	0.0000	0.0710	0.0037	0.0850	0.0000	0.0654	0.0013
penglungEW	AVG	**1.0000**	0.9356	0.9444	0.9133	**1.0000**	0.8667	0.8511	0.9356	0.9844	0.9533
	STD	0.0000	0.0122	0.0253	0.0286	0.0000	0.0000	0.0693	0.0122	0.0287	0.0311
SonarEW	AVG	**0.9984**	0.9817	0.9794	0.9786	0.9690	0.9873	0.8817	0.9825	0.9230	0.9881
	STD	0.0087	0.0120	0.0174	0.0170	0.0155	0.0121	0.0431	0.0107	0.0231	0.0121
SpectEW	AVG	0.8728	0.8901	0.8593	0.8556	0.8648	**0.9451**	0.8500	0.8648	0.8543	0.9136
	STD	0.0064	0.0083	0.0143	0.0090	0.0147	0.0124	0.0673	0.0099	0.0135	0.0101
Tic-tac-toe	AVG	0.7965	0.8021	0.8453	0.8125	0.7674	0.8385	0.6908	0.8125	0.7872	**0.8490**
	STD	0.0019	0.0000	0.0161	0.0000	0.0245	0.0000	0.0567	0.0000	0.0271	0.0000
Vote	AVG	**0.9983**	0.9656	0.9806	0.9756	0.9444	0.9739	0.8800	0.9806	0.9644	0.9561
	STD	0.0051	0.0042	0.0063	0.0085	0.0091	0.0084	0.0901	0.0063	0.0095	0.0082
WaveformEW	AVG	**0.7760**	0.7717	0.7694	0.7604	0.7082	0.7510	0.6731	0.7426	0.7345	0.7406
	STD	0.0070	0.0066	0.0084	0.0058	0.0120	0.0066	0.0379	0.0077	0.0142	0.0068
WineEW	AVG	**1.0000**	**1.0000**	0.9954	**1.0000**	0.9852	**1.0000**	0.8944	**1.0000**	0.9593	**1.0000**
	STD	0.0000	0.0000	0.0105	0.0000	0.0159	0.0000	0.0932	0.0000	0.0298	0.0000
Zoo	AVG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9968	**1.0000**	0.7968	**1.0000**	0.9651	**1.0000**
	STD	0.0000	0.0000	0.0000	0.0000	0.0121	0.0000	0.0684	0.0000	0.0214	0.0000
Clean1	AVG	**0.9903**	0.9674	0.9781	0.9726	0.9337	0.9104	0.8479	0.9375	0.9431	0.9417
	STD	0.0090	0.0060	0.0107	0.0070	0.0130	0.0111	0.0243	0.0072	0.0164	0.0059
Semeion	AVG	0.9974	0.9935	0.9939	0.9871	0.9797	0.9913	0.9720	**0.9996**	0.9813	0.9876
	STD	0.0017	0.0011	0.0027	0.0019	0.0026	0.0013	0.0092	0.0011	0.0045	0.0025
Colon	AVG	0.9231	0.7667	0.8128	0.8359	0.9231	0.8385	0.7103	0.8462	**0.9487**	0.6333
	STD	0.0000	0.0140	0.0388	0.0524	0.0000	0.0235	0.0331	0.0000	0.0369	0.0331
Leukemia	AVG	**1.0000**	0.9956	0.9333	0.9556	0.9333	0.9333	**1.0000**	0.9333	**1.0000**	0.9422
	STD	0.0000	0.0169	0.0000	0.0320	0.0175	0.0000	0.0000	0.0000	0.0000	0.0231
Mean rank	F-test	**2.773**	3.750	4.568	4.523	7.818	5.114	9.591	5.341	6.909	4.614
Overall rank		1	2	4	3	9	6	10	7	8	5

evidently prove the superiority of BPSO-TEPD against BPSO. The excellence performance of BPSO-TEPD to solve the FS problem mainly benefits from the tournament based EPD strategy that empowers the search by re-positioning the worst solutions in the population.

Among the BPSO variants, the result shows that the performance of the BPSO-TEPD was dominant. The findings of F-test in Table 5 supported this argument. In addition, the results of Wilcoxon test in Table 6 confirm that the classification performance of BPSO-TEPD was significantly better than other variants for most of the datasets. The observed improvement in BPSO-TEPD are attributed to the efficacy of the tournament selection mechanism in selecting the guide solution in the EPD strategy. This selection method allows potential solution to be chosen as the guide solution through a competition. In other words, BPSO-TEPD is able to select advisable guide solution that can help the particles to visit the promising regions. Moreover, the proper selection of tournament size offers a better trade-off between exploration and exploitation. Therefore, BPSO-TEPD can usually select the most relevant features from the search space when compared to other variants. On the one hand, the worst variant was found to be BPSO-REPD. This is expected because the BPSO-REPD selects the guide solution using a random manner. Some of the guide solutions might be built up by a weak solution, which may lead to unsatisfactory results.

Figs. 6 and 7 exhibit the convergence curves of the top three BPSO variants and BPSO approaches. Intuitively, the convergence curve highlights how the fitness value changes over the course of iterations. From these Figures, it is seen that the convergence rate of the EPD based BPSO variants were much faster than the conventional BPSO. These results affirm the excellent behavior of the EPD strategies in improving the convergence speed. Among the BPSO variants, BPSO-TEPD can often accelerate to find the global solution. Taking a high dimensional dataset (Leukemia) as an example, the BPSO-TEPD has converged faster to reach the global minimum.

Table 15
Comparison between BPSO-TEPD and other optimizers based on average number of features.

Dataset	Measure	BPSO_TEPD	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Breastcancer	AVG	5.300	4.000	4.300	5.167	5.633	4.200	4.233	8.000	**3.500**	4.000
	STD	0.466	0.000	0.702	0.379	1.402	0.407	1.382	0.000	0.731	0.000
BreastEW	AVG	8.767	10.800	10.867	11.133	14.100	19.767	12.033	15.533	**6.300**	12.300
	STD	1.612	2.355	2.177	2.623	2.940	2.029	3.102	1.852	1.368	2.548
CongressEW	AVG	4.900	5.000	4.600	4.633	7.833	10.000	6.667	6.733	**4.333**	5.400
	STD	1.348	0.000	1.404	0.850	1.599	0.871	2.496	1.701	1.516	2.268
Exactly	AVG	6.000	6.000	6.000	6.000	7.733	6.833	6.267	6.667	**5.467**	6.833
	STD	0.000	0.000	0.000	0.000	2.164	0.699	1.741	0.547	1.479	0.592
Exactly2	AVG	9.900	10.733	6.833	5.000	3.867	10.467	5.800	9.300	1.933	**1.000**
	STD	0.548	0.640	1.577	2.877	2.315	0.900	2.203	0.952	1.856	0.000
HeartEW	AVG	4.867	8.267	**3.933**	6.167	6.933	7.800	5.567	8.100	5.667	7.067
	STD	1.279	1.048	1.780	0.379	1.596	0.805	1.942	1.125	1.422	0.944
IonosphereEW	AVG	9.033	12.333	11.967	12.500	16.133	21.867	12.100	17.167	**5.600**	14.333
	STD	2.076	2.233	2.697	2.488	2.515	3.391	2.808	2.534	1.453	2.857
KrvskpEW	AVG	17.633	20.567	18.400	21.800	20.067	27.767	14.967	23.067	**7.167**	19.800
	STD	2.619	2.388	3.286	1.730	2.703	1.073	2.895	2.449	2.561	2.369
Lymphography	AVG	5.333	6.600	5.400	7.100	9.400	11.100	7.633	9.533	**4.167**	9.100
	STD	1.348	1.773	2.111	1.322	2.078	1.269	1.712	1.852	1.206	1.373
M-of-n	AVG	6.000	6.000	6.000	6.000	8.200	6.833	6.200	6.633	**5.967**	6.867
	STD	0.000	0.000	0.000	0.000	1.769	0.791	1.400	0.556	1.033	0.681
penglungEW	AVG	70.033	116.867	110.867	61.400	146.500	149.167	130.600	152.067	**8.533**	133.400
	STD	3.891	4.066	7.333	22.412	4.833	5.434	12.604	5.813	2.240	12.571
SonarEW	AVG	19.067	25.633	25.567	30.300	29.933	42.767	23.800	34.433	**10.633**	26.767
	STD	2.288	3.023	3.626	6.148	3.973	3.910	4.514	3.739	3.306	4.477
SpectEW	AVG	**5.200**	8.700	8.267	8.600	10.233	14.900	9.600	11.900	5.933	8.133
	STD	1.584	1.466	2.016	1.589	2.046	1.029	2.594	2.171	1.507	2.030
Tic-tac-toe	AVG	6.033	6.000	6.800	6.000	5.933	7.000	4.733	7.000	**4.467**	6.000
	STD	0.183	0.000	0.484	0.000	0.785	0.000	1.507	0.000	1.332	0.000
Vote	AVG	3.967	5.200	4.567	**2.667**	7.200	9.267	5.900	7.767	4.533	6.800
	STD	1.938	0.551	0.971	1.539	2.325	1.015	2.454	2.402	1.548	1.955
WaveformEW	AVG	22.867	20.600	21.833	24.767	21.400	33.433	17.200	24.967	**9.900**	21.067
	STD	1.961	3.024	2.937	2.402	3.092	1.406	3.699	2.883	2.928	3.403
WineEW	AVG	3.333	**3.300**	5.900	4.167	6.400	6.867	5.133	4.933	4.000	3.733
	STD	0.922	0.466	0.712	0.461	1.192	0.730	1.871	0.583	1.114	0.785
Zoo	AVG	3.200	4.367	**2.433**	3.033	6.600	6.067	7.533	5.100	4.867	2.733
	STD	0.407	0.615	0.817	0.183	1.714	0.907	1.852	0.845	0.776	0.640
Clean1	AVG	55.167	76.633	72.467	82.333	81.267	129.467	67.267	100.167	**12.533**	72.633
	STD	5.742	5.786	5.900	16.898	5.971	11.860	8.292	7.634	4.321	7.194
Semeion	AVG	99.733	125.633	120.933	137.533	129.800	207.667	102.400	165.933	**71.633**	124.200
	STD	7.772	8.853	5.394	27.413	8.596	9.196	14.229	10.840	20.995	9.174
Colon	AVG	749.033	915.067	857.233	217.433	949.200	1400.700	801.600	1075.767	**3.600**	868.067
	STD	16.016	12.825	26.882	128.487	13.540	245.317	47.099	104.900	1.632	48.188
Leukemia	AVG	3063.300	3404.200	3298.000	567.533	3527.500	3488.333	2891.500	3506.500	**4.800**	3096.833
	STD	29.744	37.756	36.021	374.679	42.163	20.866	271.777	25.084	2.107	133.862
Mean rank	F-test	3.500	5.205	4.273	4.864	7.545	9.182	4.818	8.568	**1.682**	5.364
Overall rank		2	6	3	5	8	10	4	9	**1**	7

Table 7 outlines the average number of selected features. As can be seen, BPSO-TEPD yielded competitive results as compared to the other BPSO variants. In Table 7, the F-test revealed that the BPSO-TEPD was the best approach in feature reduction, followed by BPSO-RWEPD. From Table 8, the capability of BPSO-TEPD in selecting the minimal features was significantly better than other methods in most cases. That is, BPSO-TEPD can always remove the irrelevant and redundant features that are not contributed to the classification process.

From the empirical analysis in Tables 5 and 7, the BPSO-TEPD devoted the optimal results in both average accuracy and feature size. In comparison with other BPSO variants, the proposed EPD and tournament selection operator in BPSO-TEPD no only enhances its searching ability but also accomplishes excellent solution across all 22 datasets. Thus, BSPO-TEPD can always achieve higher accuracy while maintaining minimal number of selected features.

Furthermore, the average fitness results are presented in Table 9. Based on the results obtained, BPSO-TEPD scored the first rank while the BPSO-LREPD and BPSO-RWEPD achieved the second and third ranks. The experimental results (refer Tables 9 and 10) expose the supremacy of BPSO-TEPD in finding the exact minimum value.

Table 11 shows the average running time results. In comparison with conventional BPSO, the BPSO variants were more time consuming in most datasets. This is because an additional computational cost is required to re-position half of the worst solutions in the population.

The overall performance of the proposed approaches is summarized in Table 12. From Table 12, one can see that the performance of BPSO variants were better than conventional BPSO algorithm. The result affirms the excellent behavior of EPD strategies in evolving the manifestation of BPSO when dealing with FS tasks. The reason is that EPD strategies help the particle to retain high diversity in exploring and exploiting the feature space. Besides, EPD strategies take advantages of top solutions to lead the poor solutions to move towards a better position. This, in turn, will improve the potential of particles in finding

Table 16
Comparison between BPSO-TEPD and other optimizers based on average fitness.

Dataset	Measure	BPSO_TEPD	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Breastcancer	AVG	0.021	0.012	0.024	0.014	0.035	0.022	0.039	0.024	0.040	**0.005**
	STD	0.001	0.000	0.007	0.000	0.004	0.003	0.007	0.000	0.004	0.000
BreastEW	AVG	0.015	**0.008**	0.011	0.028	0.043	0.031	0.043	0.039	0.039	0.020
	STD	0.005	0.004	0.004	0.005	0.008	0.004	0.009	0.004	0.011	0.004
CongressEW	AVG	0.009	**0.005**	0.010	0.014	0.044	0.024	0.054	0.020	0.043	0.035
	STD	0.005	0.004	0.006	0.001	0.009	0.006	0.011	0.005	0.009	0.004
Exactly	AVG	**0.005**	**0.005**	**0.005**	**0.005**	0.246	0.008	0.263	0.006	0.194	0.006
	STD	0.000	0.000	0.000	0.000	0.094	0.005	0.110	0.000	0.129	0.002
Exactly2	AVG	**0.202**	0.242	0.254	0.220	0.256	0.237	0.285	0.265	0.250	0.229
	STD	0.003	0.011	0.007	0.010	0.013	0.007	0.014	0.009	0.008	0.000
HeartEW	AVG	0.119	0.083	0.110	0.100	0.172	**0.068**	0.237	0.120	0.170	0.097
	STD	0.008	0.007	0.020	0.007	0.027	0.015	0.027	0.010	0.034	0.013
IonosphereEW	AVG	**0.005**	0.023	0.028	0.023	0.044	0.024	0.075	0.063	0.020	0.023
	STD	0.005	0.007	0.011	0.009	0.013	0.006	0.018	0.007	0.009	0.009
KrvskpEW	AVG	**0.015**	0.021	0.020	0.015	0.091	0.028	0.077	0.032	0.051	0.037
	STD	0.002	0.002	0.004	0.002	0.027	0.002	0.021	0.003	0.031	0.005
Lymphography	AVG	0.064	**0.019**	0.026	0.083	0.220	0.117	0.131	0.113	0.087	0.098
	STD	0.018	0.016	0.030	0.016	0.022	0.018	0.040	0.017	0.020	0.013
M-of-n	AVG	**0.005**	**0.005**	**0.005**	**0.005**	0.126	0.006	0.127	0.006	0.060	0.006
	STD	0.000	0.000	0.000	0.000	0.070	0.004	0.069	0.000	0.065	0.002
penglungEW	AVG	**0.002**	0.067	0.058	0.088	0.005	0.137	0.041	0.068	0.016	0.050
	STD	0.000	0.012	0.025	0.028	0.000	0.000	0.033	0.012	0.028	0.030
SonarEW	AVG	**0.005**	0.022	0.025	0.026	0.036	0.020	0.057	0.023	0.078	0.016
	STD	0.009	0.012	0.017	0.016	0.015	0.012	0.018	0.011	0.023	0.012
SpectEW	AVG	0.128	0.113	0.143	0.147	0.139	**0.061**	0.086	0.140	0.147	0.089
	STD	0.006	0.008	0.014	0.008	0.014	0.012	0.021	0.009	0.013	0.010
Tic-tac-toe	AVG	0.209	0.203	0.162	0.193	0.238	0.169	0.238	0.194	0.216	**0.157**
	STD	0.002	0.000	0.016	0.000	0.024	0.000	0.021	0.000	0.026	0.000
Vote	AVG	**0.004**	0.038	0.022	0.026	0.060	0.032	0.048	0.024	0.038	0.048
	STD	0.005	0.004	0.006	0.007	0.009	0.009	0.014	0.006	0.009	0.007
WaveformEW	AVG	**0.228**	0.231	0.234	0.244	0.294	0.255	0.288	0.261	0.265	0.262
	STD	0.007	0.007	0.008	0.006	0.012	0.007	0.012	0.008	0.014	0.007
WineEW	AVG	0.003	**0.003**	0.010	0.003	0.020	0.006	0.022	0.004	0.044	0.003
	STD	0.001	0.000	0.010	0.000	0.015	0.001	0.013	0.000	0.029	0.001
Zoo	AVG	0.002	0.003	**0.002**	0.002	0.008	0.004	0.104	0.003	0.038	0.002
	STD	0.000	0.000	0.001	0.000	0.012	0.001	0.014	0.001	0.021	0.000
Clean1	AVG	**0.013**	0.037	0.026	0.032	0.071	0.097	0.116	0.068	0.057	0.062
	STD	0.009	0.006	0.011	0.007	0.013	0.011	0.013	0.007	0.016	0.006
Semeion	AVG	**0.006**	0.011	0.011	0.018	0.025	0.016	0.020	0.007	0.021	0.017
	STD	0.002	0.001	0.003	0.002	0.002	0.001	0.004	0.001	0.004	0.002
Colon	AVG	0.080	0.236	0.190	0.164	0.081	0.167	0.245	0.158	**0.051**	0.367
	STD	0.000	0.014	0.038	0.052	0.000	0.023	0.029	0.001	0.037	0.033
Leukemia	AVG	0.004	0.009	0.071	0.045	0.071	0.071	0.003	0.071	**0.000**	0.062
	STD	0.000	0.017	0.000	0.032	0.017	0.000	0.000	0.000	0.000	0.023
Mean rank	F-test	**2.523**	3.591	4.318	4.432	8.250	5.591	8.523	6.091	6.818	4.864
Overall rank		**1**	2	3	4	9	6	10	7	8	5

the global optimum. Inspecting the result, BPSO-TEPD perceived the optimal F-test score in terms of accuracy, feature size, and fitness. Hence, it can be inferred that the BPSO-TEPD was the best BPSO variant in current work.

5.5. Comparing of top variant with KNN and DT classifiers

After testing all BPSO variants with KNN classifier, we found that BPSO-TEPD (top BSPO variant) revealed the best results among other approaches, thus, we are interested to test this approach with the DT classifier. In this experiment, the DT is utilized to assess the efficiency of proposed approach. Table 13 presents the performance of BPSO-TEPD with KNN and DT. In terms of accuracy and fitness, the KNN model can often retain the optimal performance in most cases. Moreover, the KNN model yielded the lowest computational cost due to its simplicity and ease of implementation. Even though DT model can guarantee a lower number of selected features, it cannot ensure higher classification accuracy. From the analysis, the KNN model was found to more appropriate for FS problems.

5.6. Comparison of BPSO-TEPD with other well-known optimizers

In this sub-section, the performance of BPSO-TEPD is compared to other well known optimizers. For comparison purposes, the optimizers SBPSO, VBPSO, BGOA, BGSA, bALO, BBA, BSSA, BWOA, and BTLBO were used. Table 14 presents the average accuracy results. In Table 14, the BPSO-TEPD scored the highest accuracy in 12 datasets, followed by SBPSO approach (7 datasets). Our results reveal that the BPSO-TEPD outperformed the other well known optimizer in FS tasks. The reason for the superior performance of BPSO-TEPD is that the tournament based EPD strategy is integrated to strengthen the exploration patterns

Table 17
Comparison between BPSO-TEPD and other optimizers based on average running time.

Dataset	Measure	BPSO_TEPD	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Breastcancer	AVG	16.610	15.285	13.801	15.625	13.915	14.980	13.847	15.376	**10.813**	13.926
	STD	1.727	1.456	0.495	1.666	0.392	0.311	0.517	1.904	0.730	0.344
BreastEW	AVG	14.697	14.382	14.067	15.032	14.662	15.948	14.378	16.103	**12.551**	14.669
	STD	0.331	0.269	0.391	0.426	0.546	0.411	0.524	0.435	0.621	0.381
CongressEW	AVG	12.566	11.983	12.065	12.507	13.044	12.134	12.664	12.831	**10.077**	13.005
	STD	0.502	0.265	0.469	0.613	0.401	0.383	0.521	0.255	0.916	0.314
Exactly	AVG	17.789	17.082	17.564	16.648	17.945	21.050	17.043	21.004	**14.086**	17.652
	STD	0.422	0.328	0.607	1.027	0.717	0.761	0.887	0.503	1.667	0.650
Exactly2	AVG	23.018	20.762	18.977	16.946	18.577	21.820	18.442	21.335	**8.678**	18.519
	STD	0.873	0.911	1.485	1.575	0.844	0.728	0.939	0.724	3.099	0.554
HeartEW	AVG	11.072	11.008	11.061	11.419	11.209	11.391	10.788	10.979	**10.216**	11.614
	STD	0.218	0.345	0.272	0.298	0.320	0.346	0.237	0.319	0.571	0.403
IonosphereEW	AVG	11.868	11.357	11.506	11.819	12.049	12.384	11.813	12.510	**10.491**	12.212
	STD	0.478	0.277	0.264	0.536	0.388	0.378	0.524	0.444	0.821	0.282
KrvskpEW	AVG	173.554	172.352	174.449	192.403	170.902	245.451	167.992	255.238	**63.077**	169.458
	STD	11.519	6.087	10.997	8.846	7.055	8.923	9.158	8.271	9.005	5.849
Lymphography	AVG	10.465	9.746	10.271	10.753	10.696	10.414	10.303	10.636	**9.201**	10.896
	STD	0.289	0.332	0.228	0.307	0.329	0.434	0.377	0.334	0.645	0.290
M-of-n	AVG	17.483	18.001	17.912	17.569	17.421	20.765	16.675	20.541	**14.313**	17.327
	STD	0.540	0.490	0.638	0.477	0.584	0.639	0.605	0.853	0.769	0.559
penglungEW	AVG	11.939	11.299	11.073	11.414	13.360	11.968	12.081	11.984	**11.027**	12.873
	STD	0.215	0.281	0.186	0.486	0.495	0.401	0.444	0.401	0.445	0.568
SonarEW	AVG	10.922	10.424	10.856	10.997	10.979	11.380	10.822	11.253	**10.250**	11.346
	STD	0.452	0.225	0.477	0.362	0.359	0.331	0.501	0.330	0.272	0.441
SpectEW	AVG	11.263	10.946	11.049	11.114	11.236	10.818	10.881	11.125	**10.661**	11.138
	STD	0.344	0.362	0.261	0.430	0.331	0.482	0.442	0.447	0.591	0.358
Tic-tac-toe	AVG	17.310	16.296	16.720	16.557	15.814	21.229	15.751	22.370	**12.901**	15.846
	STD	0.580	0.524	0.656	0.583	0.630	0.678	0.768	0.624	1.128	0.409
Vote	AVG	11.368	11.155	11.502	11.520	11.370	10.823	11.420	11.048	**9.804**	11.543
	STD	0.312	0.416	0.328	0.273	0.349	0.333	0.421	0.468	0.701	0.442
WaveformEW	AVG	454.388	420.869	438.185	493.620	432.885	649.208	416.085	631.707	**119.791**	423.303
	STD	24.772	12.895	24.353	24.265	14.901	22.656	22.848	19.179	30.201	13.831
WineEW	AVG	10.683	10.314	10.597	10.476	10.494	10.693	10.543	10.473	**9.307**	10.799
	STD	0.340	0.279	0.489	0.362	0.336	0.291	0.435	0.288	0.580	0.335
Zoo	AVG	10.961	10.635	10.922	11.102	11.030	10.964	10.968	10.972	**10.333**	10.898
	STD	0.234	0.490	0.363	0.258	0.449	0.365	0.342	0.501	0.369	0.369
Clean1	AVG	21.165	22.569	22.471	24.427	24.249	31.794	22.565	30.705	**12.640**	23.938
	STD	0.869	0.712	0.847	1.213	0.672	1.060	1.018	0.901	0.629	0.594
Semeion	AVG	192.061	205.191	205.215	237.349	212.803	362.222	201.375	348.390	**86.944**	206.132
	STD	7.352	5.489	4.628	17.998	4.506	11.687	5.993	5.671	15.850	3.513
Colon	AVG	15.751	15.459	15.566	15.434	27.439	17.788	15.595	18.017	**11.081**	19.720
	STD	0.529	0.340	0.481	0.491	0.944	0.650	0.547	0.697	1.254	0.704
Leukemia	AVG	46.287	46.407	46.998	43.597	74.605	58.932	**37.287**	49.444	52.535	45.792
	STD	4.332	4.641	4.819	3.209	2.759	5.725	1.952	2.520	2.802	1.792
Mean rank	F-test	6.182	3.955	4.727	6.227	6.773	7.682	3.818	7.727	**1.318**	6.591
Overall rank		5	3	4	6	8	9	2	10	**1**	7

Table 18
Overall rank by the F-test for all optimizers based on accuracy, number of features, fitness, and running time.

Measure	BPSO_TEPD	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Accuracy	**2.773**	3.750	4.568	4.523	7.818	5.114	9.591	5.341	6.909	4.614
Features	3.500	5.205	4.273	4.864	7.545	9.182	4.818	8.568	**1.682**	5.364
Fitness	**2.523**	3.591	4.318	4.432	8.250	5.591	8.523	6.091	6.818	4.864
Time	6.182	3.955	4.727	6.227	6.773	7.682	3.818	7.727	**1.318**	6.591
Average rank	**3.744**	4.125	4.472	5.011	7.597	6.892	6.688	6.932	4.182	5.358
Final rank	1	2	4	5	10	8	7	9	3	6

and re-position the worst solutions during the FS process. The classification performance of the proposed BPSO algorithm is summarized in Fig. 8.

Table 15 outlines the results of the average number of features. Based on the results obtained, the best algorithm that contributed to the minimal number of features was BWOA. As compared to other algorithm, BWOA can usually select fewer features. The second best algorithm was found to be BPSO-TEPD. Although BSPO-TEPD cannot guarantee the lowest number of features, it is good at choosing the most relevant features that yield to high classification accuracy.

Figs. 9 and 10 illustrate the convergence curves of proposed BPSO algorithm and other algorithms. Meanwhile, the results of average fitness is shown in Table 16. As can be observed, BPSO-TEPD gave better convergence behavior in most cases. The results of F-test in Table 16

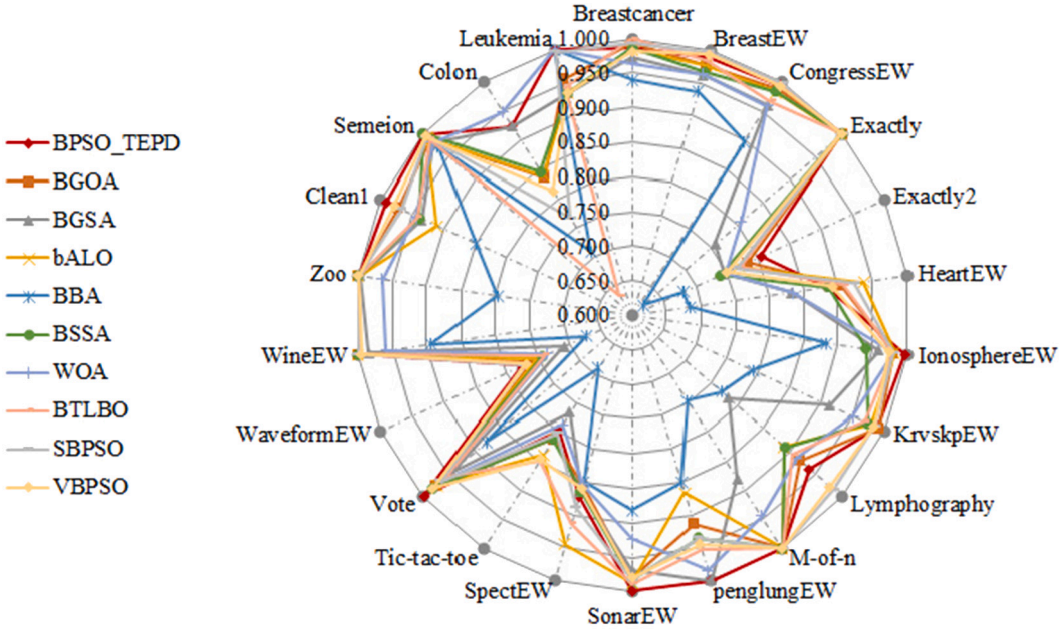

Fig. 8. Classification accuracy results of BPSO-TEPD compared to other optimizers.

Table 19
p-values of the Wilcoxon test for the classification accuracy of BPSO-TEPD versus the other optimizers in Table 14 ($p \leq 0.05$ are significant and bolded).

Dataset	SBPSO	VBPSO	BGOA	BGSA	bALO	BBA	BSSA	BWOA	BTLBO
Breastcancer	**1.69E−14**	**1.38E−04**	**1.69E−14**	**4.96E−13**	**1.31E−03**	**1.09E−12**	NaN	**3.15E−13**	**1.69E−14**
BreastEW	**1.36E−07**	**2.81E−04**	**1.62E−09**	**1.41E−11**	**7.31E−10**	**1.46E−11**	**6.29E−12**	**5.32E−11**	**2.49E−03**
CongressEW	**3.25E−03**	6.11E−01	**2.36E−05**	**9.28E−12**	**1.01E−07**	**1.40E−11**	**4.25E−07**	**9.43E−12**	**2.73E−12**
Exactly	NaN	NaN	NaN	**4.52E−12**	**2.15E−02**	**4.50E−12**	NaN	**4.72E−08**	8.14E−02
Exactly2	**1.81E−13**	**9.62E−13**	**3.34E−12**	**7.74E−13**	**3.60E−13**	**1.69E−12**	**4.70E−13**	**3.48E−13**	**2.71E−14**
HeartEW	**9.50E−13**	**3.67E−02**	**1.71E−10**	**5.08E−11**	**1.23E−11**	**8.47E−12**	1.44E−01	**4.69E−08**	**3.33E−09**
IonosphereEW	**1.39E−10**	**5.18E−11**	**1.60E−09**	**6.32E−12**	**1.75E−10**	**4.43E−12**	**1.28E−12**	**7.37E−09**	**2.37E−09**
KrvskpEW	**1.98E−09**	**1.21E−07**	5.25E−02	**2.38E−11**	**2.22E−11**	**2.39E−11**	**2.25E−11**	**2.32E−11**	**2.33E−11**
Lymphography	**6.05E−12**	**1.75E−08**	**4.66E−03**	**7.43E−12**	**2.60E−10**	**1.05E−10**	**2.57E−10**	**8.78E−03**	**5.42E−07**
M-of-n	NaN	NaN	NaN	**5.74E−11**	3.34E−01	**1.20E−12**	NaN	**6.58E−05**	1.61E−01
penglungEW	**1.17E−13**	**8.99E−11**	**2.59E−13**	NaN	**1.69E−14**	**8.53E−13**	**1.17E−13**	**5.47E−03**	**1.83E−08**
SonarEW	**1.25E−07**	**1.15E−06**	**4.27E−07**	**4.77E−10**	**4.35E−05**	**1.73E−12**	**1.15E−07**	**1.50E−12**	**1.01E−04**
SpectEW	**1.15E−09**	**2.04E−05**	**2.16E−09**	**8.11E−03**	**2.04E−12**	2.78E−01	**6.55E−04**	**3.41E−08**	**2.03E−12**
Tic-tac-toe	**2.71E−14**	**6.36E−12**	**2.71E−14**	**3.96E−06**	**2.71E−14**	**7.29E−12**	**2.71E−14**	**5.05E−03**	**2.71E−14**
Vote	**1.40E−13**	**1.19E−11**	**1.33E−11**	**8.64E−13**	**8.74E−12**	**3.52E−12**	**1.19E−11**	**1.72E−12**	**1.08E−12**
WaveformEW	**2.25E−02**	**3.06E−03**	**1.43E−09**	**2.97E−11**	**5.32E−11**	**2.97E−11**	**2.93E−11**	**3.28E−11**	**2.92E−11**
WineEW	NaN	**2.14E−02**	NaN	**1.02E−05**	NaN	**1.53E−11**	NaN	**4.69E−09**	NaN
Zoo	NaN	NaN	NaN	1.61E−01	NaN	**9.77E−13**	NaN	**5.36E−09**	NaN
Clean1	**8.03E−11**	**5.23E−05**	**7.66E−09**	**1.84E−11**	**1.55E−11**	**2.06E−11**	**1.31E−11**	**2.29E−11**	**9.21E−12**
Semeion	**4.11E−11**	**9.02E−07**	**4.50E−12**	**7.01E−12**	**4.81E−12**	**9.96E−12**	**1.02E−06**	**9.19E−12**	**7.77E−12**
Colon	**2.71E−14**	**4.46E−13**	**1.76E−10**	NaN	**6.12E−14**	**1.97E−13**	**1.69E−14**	**6.18E−04**	**1.97E−13**
Leukemia	1.61E−01	**1.69E−14**	**5.88E−08**	**1.77E−13**	**1.69E−14**	NaN	**1.69E−14**	NaN	**1.97E−11**

support this clarification. In the datasets like Exactly2, IonosphereEW, Vote, and Clean1, the BPSO-TEPD was able to converge faster to find the global minimum.

The results of computational time is presented in Table 17. Judging from Table 17, BWOA was the fastest algorithm in this work. On the one hand, the BPSO-TEPD also provided a competitive result, and it ranked fifth in the computational analysis. The overall performance of BPSO-TEPD is concluded in Table 18. Also, the *p*-values of Wilcoxon test is displayed in Table 19. The experiment results imply that the proposed BPSO-TEPD was the most effective FS approach when dealing with FS problems. Evidently, BPSO-TEPD is known to be the best FS algorithm in this work. Owing to EPD strategy, the proposed algorithm randomly re-position the worst solutions, which can effectively improve the diversity during the FS process. Moreover, the tournament selection operator allows the particles to learn from the potential guider in which a better exploration and exploitation can be ensured. All in all, these mechanisms have made the BPSO-TEPD superior.

5.7. Comparison of BPSO-TEPD with other approaches from the literature

In the final part of the experiments, we compare the performance of proposed BPSO algorithm with other FS methods in the literature. Table 20 shows the classification accuracy results. It is seen that the BPSO-TEPD perceived the highest accuracy in most datasets (16 datasets). Evidently, BPSO-TEPD demonstrated supremacy accuracy as compared to other FS methods. On the whole, it can be inferred that the BPSO-TEPD is a valuable FS tool when applied to solve the FS problems in classification tasks.

6. Conclusion and future works

In this paper, an efficient feature selection technique based on a Boolean variant of Particle Swarm Optimization (BPSO) integrated with Evolutionary Population Dynamics (EPD) was proposed as wrapper approaches to handle the feature selection problems. Various natural

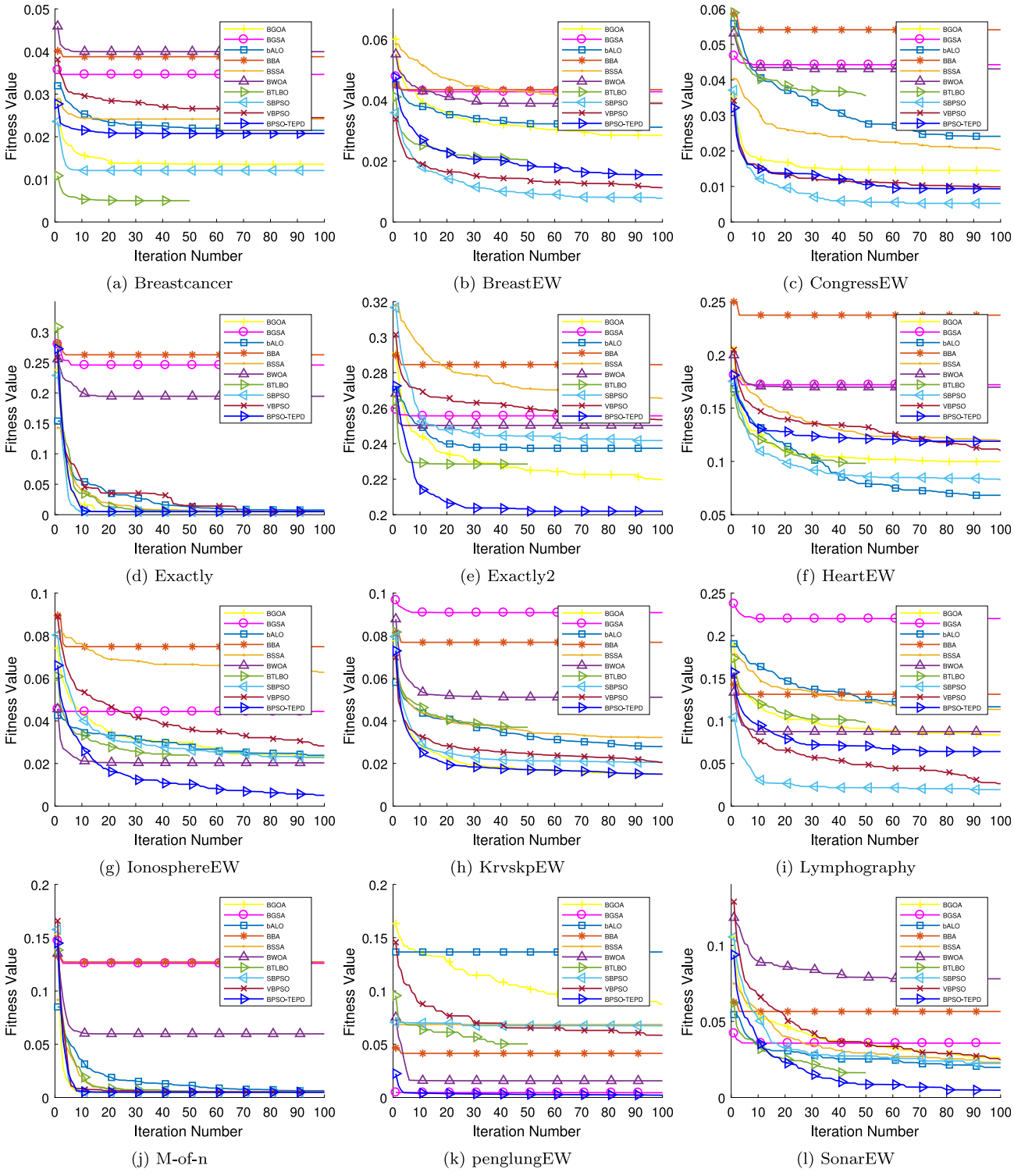

Fig. 9. Convergence curves of the proposed BPSO algorithm and other algorithms for the first 12 datasets.

selection schemes comprising (random selection, linear rank-based selection, tournament selection, roulette wheel selection, and stochastic universal sampling selection) were utilized and integrated into the BPSO algorithm. The essential idea is to improve the quality of the guide solution along with the worst solutions in the swarm, which boosted the particles (solutions) to avoid the local optima obstacle while exploring the search space. The proposed approaches were evaluated on 22 public datasets from UCI repository. Among the introduced BPSO versions, tournament selection based Boolean PSO (BPSO-TEPD) approach has gained the best performance when compared to the

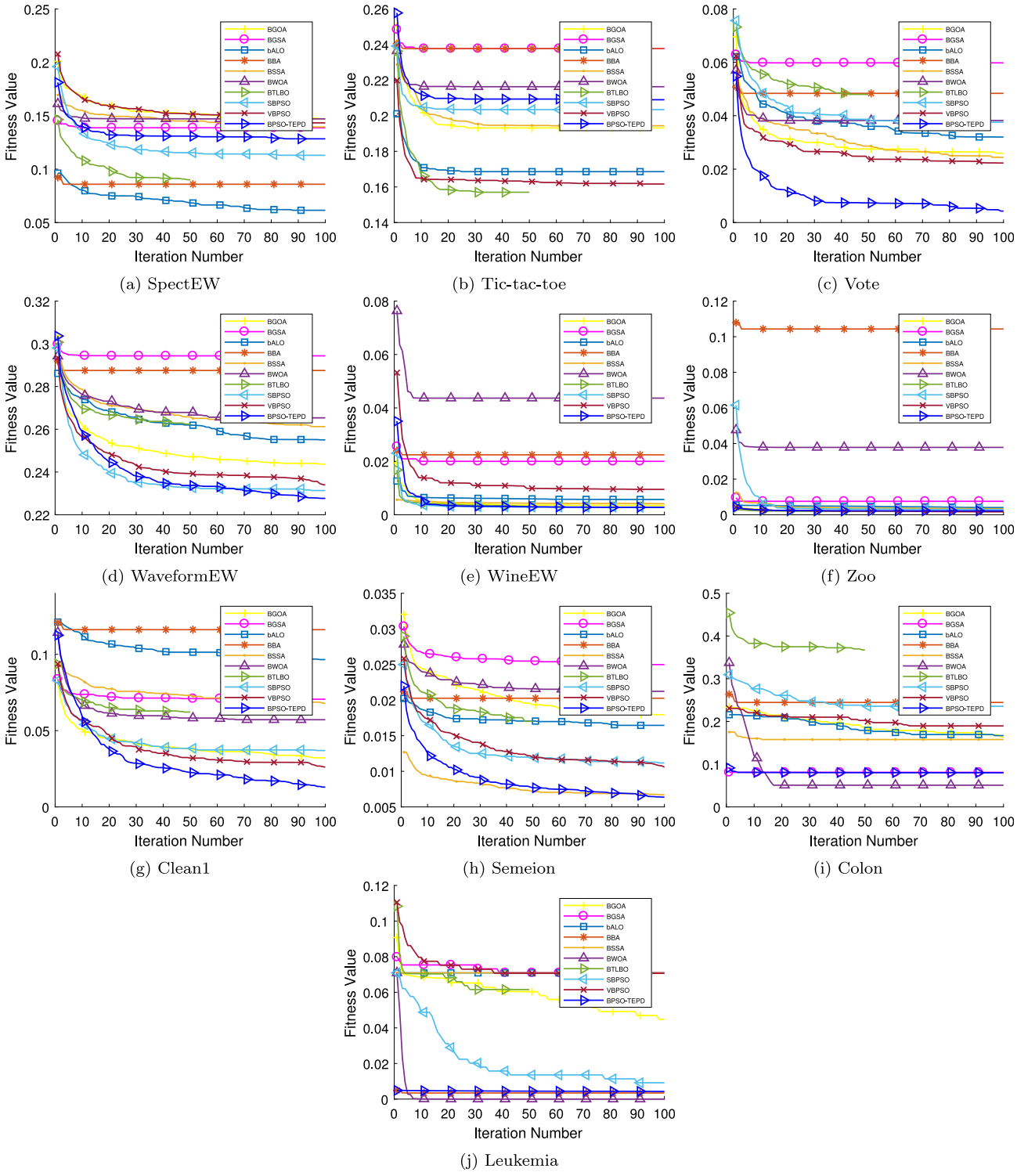

Fig. 10. Convergence curves of the proposed BPSO algorithm and other algorithms for the last 10 datasets.

other EPD based approaches in terms of classification accuracy and least number of selected features. It also outperformed several state-of-the-art approaches including SBPSO, VBPSO, BGOA, BGSA, bALO, BBA, BSSA, BWOA, and BTLBO. The result emphasized the excellent behavior of EPD strategies in enhancing the ability of Boolean PSO in resolving FS problems. Our future studies will concentrate on exploiting the efficiency of the proposed EPD based Boolean PSO in solving other challenging data mining problems.

Table 20
Classification accuracy results of the proposed BPSO algorithm versus other approaches from the literature.

Dataset	BPSO-TEPD	BSSA-S3-CP	WOA-CM	BGOA-EPD-T	GA	PSO	bGWO2	BDA-TVv4	BGOA-M	TCSSA3
Breastcancer	**0.986**	0.977	0.968	0.980	0.968	0.967	0.975	0.977	0.9743	-
BreastEW	**0.987**	0.948	0.971	0.947	0.939	0.933	0.935	0.974	0.9697	-
CongressEW	0.994	0.963	0.792	0.964	0.932	0.928	0.938	**0.995**	0.9764	0.9704
Exactly	**1.000**	0.980	0.956	0.999	0.674	0.688	0.776	0.929	**1**	0.99693
Exactly2	0.804	0.758	**1.000**	0.780	0.746	0.730	0.750	0.726	0.7352	0.7672
HeartEW	0.884	0.861	0.742	0.833	0.780	0.787	0.776	**0.886**	0.8358	0.83309
IonosphereEW	**0.998**	0.918	0.919	0.899	0.814	0.819	0.834	0.925	0.9458	0.9376
KrvskpEW	**0.990**	0.964	0.866	0.968	0.920	0.941	0.956	0.971	0.9736	0.9692
Lymphography	**0.939**	0.890	0.807	0.868	0.696	0.744	0.700	0.895	0.9118	0.84437
M-of-n	**1.000**	0.992	0.926	**1.000**	0.861	0.921	0.963	0.973	**1**	0.9992
penglungEW	**1.000**	0.878	0.972	0.927	0.584	0.584	0.584	0.807	0.9342	0.90721
SonarEW	**0.998**	0.937	0.852	0.912	0.754	0.737	0.729	0.995	0.9147	0.94808
SpectEW	0.873	0.836	**0.991**	0.826	0.793	0.822	0.822	0.876	0.8261	0.83333
Tic-tac-toe	0.797	0.821	0.785	0.808	0.719	0.735	0.727	**0.822**	0.7912	0.7974
Vote	**0.998**	0.951	0.939	0.966	0.904	0.904	0.920	0.962	0.9633	0.95489
WaveformEW	0.776	0.734	0.753	0.737	0.733	0.762	**0.789**	0.749	0.7511	0.73643
WineEW	**1.000**	0.993	0.959	0.989	0.937	0.933	0.920	0.999	0.9888	0.99775
Zoo	**1.000**	**1.000**	0.980	0.993	0.855	0.861	0.879	0.983	0.9575	0.99281
Clean1	**0.990**	0.880	–	0.863	–	–	–	–	–	0.91359
Semeion	**0.997**	0.980	–	0.976	–	–	–	–	–	0.97996
Colon	**0.923**	0.686	0.909	0.870	–	–	–	–	–	0.65699
Leukemia	**1.000**	0.989	0.982	0.931	–	–	–	–	–	0.95092

BSSA-S3-CP (Faris et al., 2018), WOA-CM (Mafarja & Mirjalili, 2018), BGOA-EPD-T (Mafarja et al., 2017), GA (Emary, Zawbaa, & Hassanien, 2016), PSO (Emary et al., 2016), bGWO2 (Emary et al., 2016), BDA-TVv4 (Mafarja, Aljarah, Heidari, Faris, et al., 2018), BGOA-M (Mafarja, Aljarah, Faris, et al., 2018), TCSSA3 (Aljarah, Mafarja et al., 2018).

CRediT authorship contribution statement

Thaer Thaher: Conceptualization, Methodology, Formal analysis, Software, Validation, Writing – original draft, Writing – review & editing. **Hamouda Chantar:** Resources, Writing – original draft, Writing – review & editing. **Jingwei Too:** Resources, Writing – original draft, Writing – review & editing. **Majdi Mafarja:** Supervision, Methodology, Formal analysis, Resources, Writing – review & editing. **Hamza Turabieh:** Resources, Writing – review & editing. **Essam H. Houssein:** Supervision, Conceptualization, Methodology, Formal analysis, Software, Validation, Writing – review & editing. All authors read and approved the final paper.

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.

Acknowledgments

The authors would like to acknowledge Taif University Researchers Supporting Project Number (TURSP-2020/125), Taif University, Taif, Saudi Arabia.

Compliance with ethical standards

This article does not contain any studies with human participants or animals performed by any of the authors.

References

Afshinmanesh, F., Marandi, A., & Rahimi-Kian, A. (2005). A novel binary particle swarm optimization method using artificial immune system. In *EUROCON 2005 - The international conference on "computer as a tool"*: *Vol. 1*, (pp. 217–220). http://dx.doi.org/10.1109/EURCON.2005.1629899.

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications*, *36*, 6843–6853.

Al-Betar, M. A., Awadallah, M. A., Faris, H., Aljarah, I., & Hammouri, A. I. (2018). Natural selection methods for grey wolf optimizer. *Expert Systems with Applications*, *113*, 481–498. http://dx.doi.org/10.1016/j.eswa.2018.07.022, URL: https://www.sciencedirect.com/science/article/pii/S0957417418304378.

Al-Betar, M., Awadallah, M., Faris, H., Yang, X. -S., Khader, A. T., & Alomari, O. (2017). Bat-inspired algorithms with natural selection mechanisms for global optimization. *Neurocomputing*, *273*, http://dx.doi.org/10.1016/j.neucom.2017.07.039.

Aljarah, I., Ala'M, A. -Z., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cognitive Computation*, 1–18.

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., Zhang, Y., & Mirjalili, S. (2018). Asynchronous accelerating multi-leader salp chains for feature selection. *Applied Soft Computing*, *71*, 964–979.

Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, *116*, 147–160.

Back, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence* (pp. 57–62). IEEE.

Bak, P., Tang, C., & Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Physical Review Letters*, *59*, 381.

Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of an international conference on genetic algorithms and their applications* (pp. 101–111). Hillsdale, New Jersey.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms on genetic algorithms and their application* (pp. 14–21). Hillsdale, NJ, USA: L. Erlbaum Associates Inc., URL: http://dl.acm.org/citation.cfm?id=42512.42515.

Beheshti, Z., & Shamsuddin, S. M. H. (2014). CAPSO: Centripetal accelerated particle swarm optimization. *Information Sciences*, *258*, 54–79.

Bellal, F., Elghazel, H., & Aussem, A. (2012). A semi-supervised feature ranking method with ensemble learning. *Pattern Recognition Letters*, *33*, 1426–1433. http://dx.doi.org/10.1016/j.patrec.2012.03.001, URL: https://www.sciencedirect.com/science/article/pii/S0167865512000736.

Bharti, K. K., & Singh, P. K. (2016). Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. *Applied Soft Computing*, *43*, 20–34.

Boettcher, S., & Percus, A. G. (1999). Extremal optimization: Methods derived from co-evolution. In *Proceedings of the 1st annual conference on genetic and evolutionary computation*: *Vol. 1*, (pp. 825–832). Morgan Kaufmann Publishers Inc..

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems*, *86*, 33–45.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117. http://dx.doi.org/10.1016/j.ins.2013.02.041, Prediction, Control and Diagnosis using Advanced Neural Computations. URL: https://www.sciencedirect.com/science/article/pii/S0020025513001588.

BrezoAinik, L., Fister, I., & Podgorelec, V. (2018). Swarm intelligence algorithms for feature selection: A review. *Applied Sciences*, *8*, http://dx.doi.org/10.3390/app8091521, URL: https://www.mdpi.com/2076-3417/8/9/1521.

Chandra, S., Bhat, R., & Singh, H. (2009). A PSO based method for detection of brain tumors from MRI. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 666–671).

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers and Electrical Engineering*, *40*, 16–28.

Chantar, H., Thaher, T., Turabieh, H., Mafarja, M., & Sheta, A. (2021). BHHO-TVS: A binary harris hawks optimizer with time-varying scheme for solving data classification problems. *Applied Sciences*, *11*, 6516.

Chegini, S. N., Bagheri, A., & Najafi, F. (2018). PSOSCALF: A new hybrid PSO based on Sine cosine algorithm and Levy flight for solving optimization problems. *Applied Soft Computing*, *73*, 697–726.

Chen, Y., Li, L., Peng, H., Xiao, J., Yang, Y., & Shi, Y. (2017). Particle swarm optimizer with two differential mutation. *Applied Soft Computing*, *61*, 314–330.

Chen, K., Zhou, F., Yin, L., Wang, S., Wang, Y., & Wan, F. (2018). A hybrid particle swarm optimizer with sine cosine acceleration coefficients. *Information Sciences*, *422*, 218–241.

Chuang, L. -Y., Chang, H. -W., Tu, C. -J., & Yang, C. -H. (2008). Improved binary PSO for feature selection using gene expression data. *Computational Biology and Chemistry*, *32*, 29–38.

Chuang, L. -Y., Tsai, S. -W., & Yang, C. -H. (2011). Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, *38*, 12699–12707.

Chuang, L. -Y., Yang, C. -H., & Li, J. -C. (2011). Chaotic maps based on binary particle swarm optimization for feature selection. *Applied Soft Computing*, *11*, 239–248.

Crawford, B., Soto, R., Astorga, G., García, J., Castro, C., & Paredes, F. (2017). Putting continuous metaheuristics to work in binary search spaces. *Complexity*, *2017*.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, *1*, 131–156.

Deligkaris, K. V., Zaharis, Z. D., Kampitaki, D. G., Goudos, S. K., Rekanos, I. T., & Spasos, M. N. (2009). Thinned planar array design using Boolean PSO with velocity mutation. *IEEE Transactions on Magnetics*, *45*, 1490–1493. http://dx.doi.org/10.1109/TMAG.2009.2012687.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*, 3–18.

Dong, H., Li, T., Ding, R., & Sun, J. (2018). A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Applied Soft Computing*, *65*, 33–46.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*, 28–39.

Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks: Vol. 4*, (pp. 1942–1948). Citeseer.

Eiben, A., & Smith, J. (2003). *Introduction to evolutionary computing*. *Computing Reviews*, *45*, http://dx.doi.org/10.1007/978-3-662-05094-1.

Emary, E., & Zawbaa, H. M. (2016). Impact of chaos functions on modern swarm optimizers. *PLoS One*, *11*, Article e0158738.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, *172*, 371–381.

Faris, H., Mafarja, M., Heidari, A. A., Aljarah, I., Al-Zoubi, A., Mirjalili, S., et al. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, *154*, 43–67. http://dx.doi.org/10.1016/j.knosys.2018.05.009.

Fayk, M. B., El Nemr, H. A., & Moussa, M. M. (2010). Particle swarm optimisation based video abstraction. *Journal of Advanced Research*, *1*, 163–167. http://dx.doi.org/10.1016/j.jare.2010.03.009, URL: https://www.sciencedirect.com/science/article/pii/S2090123210000408.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*, 37.

Fong, S., Wong, R., & Vasilakos, A. V. (2015). Accelerated PSO swarm search feature selection for data stream mining big data. *IEEE Transactions on Services Computing*, *9*, 33–45.

García, J., Crawford, B., Soto, R., & Astorga, G. (2019). A clustering algorithm applied to the binarization of swarm intelligence continuous metaheuristics. *Swarm and Evolutionary Computation*, *44*, 646–664.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*, 533–549. http://dx.doi.org/10.1016/0305-0548(86)90048-1, Applications of Integer Programming. URL: https://www.sciencedirect.com/science/article/pii/0305054886900481.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms: Vol. 1*, (pp. 69–93). Elsevier.

Goldberg, E. D., Korb, B., & Deb, K. (1990). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, *3*, 493–530.

Gunal, S., & Edizkan, R. (2008). Subspace based feature selection for pattern recognition. *Information Sciences*, *178*, 3716–3726.

Gunasundari, S., Janakiraman, S., & Meenambal, S. (2016). Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis. *Expert Systems with Applications*, *56*, 28–47.

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. *Computerized Medical Imaging and Graphics*, *70*, 135–154.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Hassouneh, Y., Turabieh, H., Thaher, T., Tumar, I., Chantar, H., & Too, J. (2021). Boosted whale optimization algorithm with natural selection operators for software fault prediction. *IEEE Access*, *9*, 14239–14258. http://dx.doi.org/10.1109/ACCESS.2021.3052149.

Hichem, H., Elkamel, M., Rafik, M., Mesaaoud, M. T., & Ouahiba, C. (2019). A new binary grasshopper optimization algorithm for feature selection problem. *Journal of King Saud University-Computer and Information Sciences*.

Holland, H. J. (1975). *Adaptation In Natural And Artificial Systems*. University of Michigan Press.

Houssein, E. H., Ewees, A. A., & Abd ElAziz, M. (2018). Improving twin support vector machine based on hybrid swarm optimizer for heartbeat classification. *Pattern Recognition and Image Analysis*, *28*, 243–253.

Houssein, E. H., Gad, A. G., Hussain, K., & Suganthan, P. N. (2021). Major advances in particle swarm optimization: Theory, analysis, and application. *Swarm and Evolutionary Computation*, *63*, Article 100868.

Hussain, K., Neggaz, N., Zhu, W., & Houssein, E. H. (2021). An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. *Expert Systems with Applications*, *176*, Article 114778.

Islam, M. J., Li, X., & Mei, Y. (2017). A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. *Applied Soft Computing*, *59*, 182–196. http://dx.doi.org/10.1016/j.asoc.2017.04.050, URL: https://www.sciencedirect.com/science/article/pii/S1568946417302326.

Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Applied Soft Computing*, *62*, 203–215.

Jin, C., & Jin, S. -W. (2015). Automatic image annotation using feature selection based on improving quantum particle swarm optimization. *Signal Processing*, *109*, 172–181.

Kennedy, J., & Eberhart, R. (1995). A new optimizer using particle swarm theory. In *Micro machine and human science, 1995. MHS '95., proceedings of the sixth international symposium on* (pp. 39–43). http://dx.doi.org/10.1109/MHS.1995.494215.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Systems, man, and cybernetics, 1997. computational cybernetics and simulation., 1997 IEEE international conference on: Vol. 5*, (pp. 4104–4108). IEEE.

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by simulated annealing. *Science*, *220*, 671–680. http://dx.doi.org/10.1126/science.220.4598.671.

Kohavi, R., & John, G. H. (1998). The wrapper approach. In *Feature extraction, construction and selection* (pp. 33–50). Springer.

Lai, C., Reinders, M. J., & Wessels, L. (2006). Random subspace method for multivariate feature selection. *Pattern Recognition Letters*, *27*, 1067–1076.

Lewis, A., Mostaghim, S., & Randall, M. (2008). Evolutionary population dynamics and multi-objective optimisation problems. *Multi-Objective Optimization in Computational Intelligence: Theory And Practice*, 185–206.

Li, Y., Chen, Y., Zhong, J., & Huang, Z. (2019). Niching particle swarm optimization with equilibrium factor for multi-modal optimization. *Information Sciences*, *494*, 233–246.

Li, Y. Z., Jiang, L., Wu, Q., Wang, P., Gooi, H. B., Li, K., et al. (2017). Wind-thermal power system dispatch using MLSAD model and GSOICLW algorithm. *Knowledge-Based Systems*, *116*, 94–101.

Liang, J. J., Suganthan, P. N., Chan, C. C., & Huang, V. L. (2006). Wavelength detection in FBG sensor network using tree search DMS-PSO. *IEEE Photonics Technology Letters*, *18*, 1305–1307. http://dx.doi.org/10.1109/LPT.2006.875525.

Lichman, M. (2013). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Liu, H. -R., Cui, J. -C., Lu, Z. -D., Liu, D. -Y., & Deng, Y. -J. (2019). A hierarchical simple particle swarm optimization with mean dimensional information. *Applied Soft Computing*, *76*, 712–725.

Liu, Y., & Gu, X. (2007). Skeleton-network reconfiguration based on topological characteristics of scale-free networks and discrete particle swarm optimization. *IEEE Transactions on Power Systems*, *22*, 1267–1274. http://dx.doi.org/10.1109/TPWRS.2007.901486.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers.

Liu, H., & Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. (pp. 388–391). ISBN: 0818673125.

Liu, H., & Setiono, R. (1996). Feature selection and classification-a probabilistic wrapper approach. (pp. 419–424).

Lu, Y., Liang, M., Ye, Z., & Cao, L. (2015). Improved particle swarm optimization algorithm and its application in text feature selection. *Applied Soft Computing*, *35*, 629–636.

Mafarja, M., Aljarah, I., Faris, H., Hammouri, A., Al-Zoubi, A., & Mirjalili, S. (2018). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, *117*, http://dx.doi.org/10.1016/j.eswa.2018.09.015.

Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Ala'M, A. -Z., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, *117*, 267–286.

Mafarja, M., Aljarah, I., Heidari, A. A., Faris, H., Fournier-Viger, P., Li, X., et al. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Systems*, *161*, 185–204. http://dx.doi.org/10.1016/j.knosys.2018.08.003, URL: http://www.sciencedirect.com/science/article/pii/S095070511830399X.

Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A. -Z., et al. (2017). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems*.

Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Ala'M, A. -Z., et al. (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems*, *145*, 25–45.

Mafarja, M., Heidari, A. A., Habib, M., Faris, H., Thaher, T., & Aljarah, I. (2020). Augmented whale feature selection for IoT attacks: Structure, analysis and applications. *Future Generation Computer Systems*, *112*, 18–40. http://dx.doi.org/10.1016/j.future.2020.05.020, URL: https://www.sciencedirect.com/science/article/pii/S0167739X19313561.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, *260*, 302–312.

Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, *62*, 441–453.

Marandi, A., Afshinmanesh, F., Shahabadi, M., & Bahrami, F. (2006). Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna. In *2006 IEEE international conference on evolutionary computation* (pp. 3212–3218). http://dx.doi.org/10.1109/CEC.2006.1688716.

Masehian, E., & Sedighizadeh, D. (2010). A multi-objective PSO-based algorithm for robot path planning. In *2010 IEEE international conference on industrial technology* (pp. 465–470).

Mathiyalagan, M., Dhepthie, U., & Sivanandam, S. (2010). Grid scheduling using enhanced PSO algorithm. *International Journal on Computer Science and Engineering*, *2*.

Meza, J., Espitia, H., Montenegro, C., Giménez, E., & González-Crespo, R. (2017). MOVPSO: Vortex multi-objective particle swarm optimization. *Applied Soft Computing*, *52*, 1042–1057.

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software*, *83*, 80–98.

Mirjalili, S., & Dong, J. (2020). *Multi-objective optimization using artificial intelligence techniques*. http://dx.doi.org/10.1007/978-3-030-24835-2.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, *43*, 117–130.

Nakamura, R. Y. M., Pereira, L. A. M., Rodrigues, D., Costa, K. A. P., Papa, J. P., & Yang, X. -S. (2013). Binary bat algorithm for feature selection. In *Swarm intelligence and bio-inspired computation* (pp. 225–237). Elsevier.

Neggaz, N., Ewees, A. A., Elaziz, M. A., & Mafarja, M. (2020). Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Systems with Applications*, *145*, Article 113103. http://dx.doi.org/10.1016/j.eswa.2019.113103, URL: https://www.sciencedirect.com/science/article/pii/S0957417419308206.

Neggaz, N., Houssein, E. H., & Hussain, K. (2020). An efficient henry gas solubility optimization for feature selection. *Expert Systems with Applications*, *152*, Article 113364.

Nobile, M. S., Cazzaniga, P., Besozzi, D., Colombo, R., Mauri, G., & Pasi, G. (2018). Fuzzy self-tuning PSO: A settings-free algorithm for global optimization. *Swarm and Evolutionary Computation*, *39*, 70–85.

Oh, I. -S., Lee, J. -S., & Moon, B. -R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 1424–1437.

Pedersen, M. E. H., & Chipperfield, A. J. (2010). Simplifying particle swarm optimization. *Applied Soft Computing*, *10*, 618–628.

Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, *156*, 483–494.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1993). *C4. 5: programs for machine learning*. Morgan kaufmann.

Randall, M., Hendtlass, T., & Lewis, A. (2009). Extremal optimisation for assignment type problems. *Biologically-Inspired Optimisation Methods: Parallel Algorithms, Systems And Applications*, *210*, 139–164.

Razali, N. M., Geraghty, J., et al. (2011). Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering*: *Vol. 2*, (pp. 1–6). International Association of Engineers Hong Kong.

Ren, J., Qiu, Z., Fan, W., Cheng, H., & Philip, S. Y. (2008). Forward semi-supervised feature selection. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 970–976). Springer.

Rodrigues, D., Pereira, L. A., Almeida, T., Papa, J. a. P., Souza, A., Ramos, C. C., et al. (2013). BCS: A binary cuckoo search algorithm for feature selection. In *2013 IEEE international symposium on circuits and systems* (pp. 465–468). IEEE.

Rodrigues, D., Yang, X. -S., De Souza, A. N., & Papa, J. a. P. (2015). Binary flower pollination algorithm and its application to feature selection. In *Recent advances in swarm intelligence and evolutionary computation* (pp. 85–100). Springer.

Sahoo, S., Subudhi, A., Dash, M., & Sabut, S. (2020). Automatic classification of cardiac arrhythmias based on hybrid features and decision tree algorithm. *International Journal of Automation and Computing*, 1–11.

Sahu, A., Panigrahi, S. K., & Pattnaik, S. (2012). Fast convergence particle swarm optimization for functions optimization. *Procedia Technology*, *4*, 319–324.

Salajegheh, F., & Salajegheh, E. (2019). PSOG: Enhanced particle swarm optimization by a unit vector of first and second order gradient directions. *Swarm and Evolutionary Computation*.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30–47.

Saremi, S., Mirjalili, S. Z., & Mirjalili, S. M. (2015). Evolutionary population dynamics and grey wolf optimizer. *Neural Computing and Applications*, *26*, 1257–1263.

Sayed, S. A.-F., Nabil, E., & Badr, A. (2016). A binary clonal flower pollination algorithm for feature selection. *Pattern Recognition Letters*, *77*, 21–27.

Setiono, R., & Liu, H. (1997). Neural-network feature selector. *IEEE Transactions on Neural Networks*, *8*, 654–662.

Shang, R., Wang, W., Stolkin, R., & Jiao, L. (2016). Subspace learning-based graph regularized feature selection. *Knowledge-Based Systems*, *112*, 152–165.

Shang, R., Wang, W., Stolkin, R., & Jiao, L. (2017). Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection. *IEEE Transactions on Cybernetics*, *48*, 793–806.

Shang, R., Zhang, Z., Jiao, L., Liu, C., & Li, Y. (2016). Self-representation based dual-graph regularized feature selection clustering. *Neurocomputing*, *171*, 1242–1253.

Shen, Q., Jiang, J. -H., Jiao, C. -X., Shen, G. -., & 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. *European Journal of Pharmaceutical Sciences*, *22*, 145–152. http://dx.doi.org/10.1016/j.ejps.2004.03.002.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No.98TH8360)* (pp. 69–73). http://dx.doi.org/10.1109/ICEC.1998.699146.

Siedlecki, W., & Sklansky, J. (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, *2*, 197–220.

Subasi, A. (2013). Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders. *Computers in Biology and Medicine*, *43*, 576–586.

Talbi, E. -G. (2009). *Metaheuristics: from design to implementation: Vol. 74*. John Wiley & Sons.

Tamura, K., Kitakami, H., & Nakada, A. (2013). Distributed modified extremal optimization using Island model for reducing crossovers in reconciliation graph. *Engineering Letters*, *21*.

Tawhid, M. A., & Dsouza, K. B. (2018). Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems. *Applied Computing and Informatics*.

Thaher, T., & Arman, N. (2020). Efficient multi-swarm binary Harris Hawks optimization as a feature selection approach for software fault prediction. In *2020 11th International conference on information and communication systems* (pp. 249–254). IEEE.

Thaher, T., Mafarja, M., Turabieh, H., Castillo, P. A., Faris, H., & Aljarah, I. (2021). Teaching learning-based optimization with evolutionary binarization schemes for tackling feature selection problems. *IEEE Access*, *9*, 41082–41103. http://dx.doi.org/10.1109/ACCESS.2021.3064799.

Tian, D., & Shi, Z. (2018). MPSO: Modified particle swarm optimization and its applications. *Swarm and Evolutionary Computation*, *41*, 49–68.

Tran, B., Xue, B., & Zhang, M. (2017). A new representation in PSO for discretization-based feature selection. *IEEE Transactions on Cybernetics*, *48*, 1733–1746.

Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, *206*, 528–539.

Wang, A., An, N., Chen, G., Li, L., & Alterovitz, G. (2015). Accelerating wrapper-based feature selection with K-nearest-neighbor. *Knowledge-Based Systems*, *83*, 81–91.

Wang, L., Wang, X., Fu, J., & Zhen, L. (2008). A novel probability binary particle swarm optimization algorithm and its application. *Journal of Software*, *3*, 28–35.

Wei, J., Zhang, R., Yu, Z., Hu, R., Tang, J., Gui, C., et al. (2017). A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection. *Applied Soft Computing*, *58*, 176–192.

Wu, J., Wang, H., Li, N., Yao, P., Huang, Y., & Yang, H. (2017). Path planning for solar-powered UAV in urban environment. *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2017.10.037, URL: http://www.sciencedirect.com/science/article/pii/S0925231217316880.

Xia, X., Xing, Y., Wei, B., Zhang, Y., Li, X., Deng, X., et al. (2019). A fitness-based multi-role particle swarm optimization. *Swarm and Evolutionary Computation*, *44*, 349–364.

Xie, Y., Zhu, Y., Wang, Y., Cheng, Y., Xu, R., Sani, A. S., et al. (2019). A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment. *Future Generation Computer Systems*, *97*, 361–378.

Xu, G., Cui, Q., Shi, X., Ge, H., Zhan, Z. -H., Lee, H. P., et al. (2019). Particle swarm optimization based on dimensional learning strategy. *Swarm and Evolutionary Computation*, *45*, 33–51.

Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, *18*, 261–276.

Yang, X. -S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169–178). Springer.

Zakeri, A., & Hokmabadi, A. (2019). Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Systems with Applications*, *119*, 61–72.

Zhang, Y., Gong, D., Hu, Y., & Zhang, W. (2015). Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing*, *148*, 150–157.

Zhang, X., Xu, Y., Yu, C., Heidari, A. A., Li, S., Chen, H., et al. (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Systems with Applications*, *141*, Article 112976.

Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on machine learning* (pp. 1151–1157). ACM.

Zorarpacı, E., & Özel, S. A. (2016). A hybrid approach of differential evolution and artificial bee colony for feature selection. *Expert Systems with Applications*, *62*, 91–103.