International Journal of Multidisciplinary Research and Growth Evaluation.

Feature selection using chaotic particle swarm optimization

Samuel-Soma M Ajibade [1], Kayode Akintoye [2], Sushovan Chaudhury [3*], Mbiatke Anthony Bassey [4]
[1, 2] Department of Computer Science, The Federal Polytechnic, Ado Ekiti, Nigeria
[3] CSE Department, University of Engineering and Management, Kolkata, India
[4] Department of Business Administration, UTHM, Batu Pahat, Malaysia

* Corresponding Author: **Sushovan Chaudhury**

Article Info

ISSN (online): 2582-7138
Volume: 04
Issue: 01
January-February 2023
Received: 10-12-2022;
Accepted: 01-01-2023
Page No: 160-170

Abstract
A very successful global optimization technique is particle swarm optimization (PSO). PSO has some drawbacks, including slow convergence, early convergence, and getting stranded at local optima. In order to enhance the search process, the chaotic map and dynamic-weight are introduced into PSO in this paper. By modifying the position update formula and the chaotic dynamic-weight Particle Swarm Optimization (CPSO) inertia weight, they effectively balance the local and global PSO feature selection processes. The effectiveness of CPSO was compared to three metaheuristic methods: Differential Evolution (DE), Genetic Algorithm (GA), and PSO, using eight numerical functions. Four datasets are used to evaluate this approach. The findings show that by balancing the exploration and exploitation search processes, the CPSO is an effective feature selection strategy that produces high-quality results. The suggested CPSO approach successfully classified features using the KNN Classifier for the four datasets.

Keywords: Feature selection, Particle swarm optimization, Chaos, Inertia Weight

1. Introduction

Technology has become so ubiquitous in recent years that it has resulted in the creation of numerous datasets with a large number of features and patterns for classification. Due to the large amount of redundant and irrelevant features in these datasets, there is a good likelihood that the precision of training the classifier will suffer, as well as the classifier's training speed. Dimensionality reduction is a fundamental step in making data mining algorithms more efficient and effective. Feature Extraction (FE) and Feature Selection (FS) are the two main types of dimensionality reduction (FS). The FE method creates a new subset of functions from unique capabilities, whereas the FS method selects a subset from the original capabilities. By removing irrelevant and redundant capabilities, and so improving the overall performance of the set of rules (Gopika, 2018) [1]. The FS is a more widely used technique for data and system knowledge mining than the FE. Wrapper, filter, and embedding approaches are the three categories of FS techniques. A function subset is evaluated in wrappers using the type performance of a chosen type algorithm. Meanwhile, any classification algorithm's filters study the function subset's best in an unbiased approach. Embedded strategies make the capability selection at some time during the modeling algorithm's implementation. Filter techniques, in particular, are based on the characteristics of facts, which include distance, consistency, dependency, data, and correlation. Wrappers are frequently more expensive in comparison to the filter approach in terms of computational value. The wrapper technique, which employs machine learning algorithms, is commonly used to estimate the feature subset. However, the cost of processing remains high for high-dimensional data, but they are primarily appropriate for improving an algorithm's prediction chances.

A number of feature selection strategies based on metaheuristic techniques have been presented in recent years (Mafarja and Mirjalili, 2018, Ajibade, S. 2020) [2, 4]. Because of their potential global search capabilities, these metaheuristic strategies have gained a lot of attention due to their good performance in solving the feature selection problem. Evolutionary based algorithms, physics-based algorithms, swarm-based algorithms, and human-based algorithms are the four types of metaheuristics optimization algorithms (Mirjalili and Lewis, 2016). [3]

Eberhart and Kennedy (1995) [5], created particle swarm optimization (PSO) as a nature-inspired algorithm that replicated the sociological behavior of individuals in fowl and fish swarms. PSO is considered to be less expensive to compute, and it easily integrates with other heuristic approaches such as GA, ACO, and DE (Moradi, 2016, Ajibade and Adediran 2016) [6, 7]. Furthermore, PSO has a few parameters that are simple to fine-tune and implement, making it an effective FS technique. Though the application of PSO has made the search for optimal feature subsets more successful, it still has several flaws, such as premature convergence in complex optimization problems. The value of reducing the classification error rate is overshadowed by the quantity of features in the search process. Another flaw is that it selects features that are comparable in the final feature subset (Sengupta *et al.*, 2019) [8]. Furthermore, the PSO does not typically use feature correlation information to guide the search process, resulting in the look-alike features being selected the majority of the time in the closing feature subset, lowering classifier performance (Mafarja and Mirjalili, 2018, Ajibade, Ahmad and Shamsuddin, 2018) [9, 10, 2]. Chaos theory is a common mathematical concept that has been shown to be effective in improving metaheuristic algorithm performance. Chaos theory has been widely employed in the literature as a supplement to the mathematical model in order to improve the algorithm's global convergence capacity and avoid converging at local optima. This research focuses on the impact of a chaotic map on PSO performance.

The rest of this paper is organized as follows: Section 2 discusses the overview of PSO algorithm. The research methodology and research framework are provided in section 3. The proposed approach is discussed in Section 4, the experimental results are presented and analyzed in section 5. Finally, in Section 6, conclusions and future work is given.

2. Overview of Particle Swarm Optimization (PSO)

PSO is a technique for optimizing that was created by Eberhart and Kennedy (1995) [5]. For guiding the molecules to explore global optimal outcomes, this optimization technique offers an easy approach to replicate swarm characteristics in bird groupings and fish schools. PSO was defined by (Wang *et al.*, 2018) [19] using three simple actions: separation, alignment, and cohesion. The swarming local flock mates are usually avoided in the separation approach, but the action in the alignment method tends in the direction of the local flock mates, and the cohesiveness action tends towards the average location of local flock mates. It makes use of the social interaction concept to find solutions to problems without ignoring the problem gradient being optimized, obviating the requirement for differentiation of the optimization problem, as required by traditional optimization procedures (Yan et al., 2018) [13].

The algorithm iterates until the resolution is found. Each particle regulates its forward motion in relation to its previous best position and the current best position of other particles in order to find the global optimal solution. "The former location of particle i is represented by a vector in the search process, which is $x_i = (x_{i1},x_{i2},..., x_{id})$, where d denotes the dimension of the problem space. The maximum flying velocity of particle i is $v_i = (v_{i1},v_{i2},..., v_{id})$, and the range of v_{id} is $[- V_{max},V_{max}]$, where V_{max} is the maximum flying velocity, which is used to keep the value within a tolerable range. Furthermore, $pbest_i = [pbest_{i1}, pbest_{i2},..., pbest_{id}]$ denotes the current best location of particle i and $gbest = [gbest_1, gbest_2,..., gbest_d]$ indicates the current best location of the total population" (Ajibade *et al.*, 2020) [4]. By updating the flying velocity and location of all individuals in the swarm according to Eq 1 and Eq 2, the PSO approach recognizes the global optimal solution.

$$v_{id}^{(t+1)}=v_{id}(t)+r_1(t)\times C_1\times(pbest_{id}^{(t)}-x_{id}^{(t)})+r_2(t)\times C_2\times(gbest^{(t)}-x_{id}^{(t)}) \quad (1)$$

$$x_{id}^{(t+1)}=x_{id}^{(t)}+v_{id}^{(t+1)} \quad (2)$$

where r1 and r2 are two uniformly distributed values in the range [0,1], and c1 and c2 are acceleration parameters, which are generally set to 2.0. As a result of using the inertia weight w, Eq 1 is improved to Eq 3.

$$V_{id}^{(t+1)}=\omega\times v_{id}(t)+r_1(t)\times C_1\times(pbest_{id}^{(t)}-x_{id}^{(t)})+r_2(t)\times C_2\times (gbest^{(t)} - x_{id}^{(t)}) \quad (3)$$

The first component on the right-hand side of Eq 3 represents the inertia weight. Excitations towards promising sites in the search universe, as determined by personal and global best locations, are represented by the other two words. Eq 4 gives a mechanism for updating (Shi and Eberhart, 1998) [14].

$$w(t) = w_{max} - (w_{max} - w_{min}) \times \frac{t}{T} \quad (4)$$

For which t and T are the present and max iterations, accordingly, while max and min are the user-defined inertia weight constraints. When exploring the PSO, Xue et al., (2015) [15] considered the variety of attributes when conducting upgrades of pbest and gbest approaches, since it subsequently minimises number of attributes while in comparison with outdated updates of pbest and gbest methodologies without deteriorating classification. When updating pbest and gbest during the PSO search process, the steps are considered the attributes numbers, which reduces the numbers of attribute over the conventional updating pbest and gbest mechanism.

1. Initialize population
2. Repeat
3. Evaluated particles founded on the calculated fitness estimates
4. Each particle is modified based on the best solution of current particle achieved (Pbest, particle best) and the best of the population (gbest, global best).
5. The particles tend to move to best solution (pbest and gbest) in the search space by the information spreading to the swarm.
6. Calculate the new velocities, v of particles with Eq 5.

$$V(i+1) = w*(v(i)) + \phi 1*rand(0,1)*x(i) - gbest(i) + \phi 2 rand(0,1)(pbest(i) - x(i)) \quad \ldots(5)$$

Where, i is the index variable,
x is the solution (particle),
$\phi 1$ and $\phi 2$ are variables used to determine the significant of pbest and gbest,
rand (0,1) is the function to generate random value with normal distribution within the range of 0 and 1.

7. Update the current best particles positions if the fitness value of the new particle is lower than the fitness value of the current best particles.

UNTIL predefined maximum loop value is reached, or the fitness value has convergence

Fig 1: PSO Algorithm (Rennard, 2006)

Fig. 1 shows the PSO algorithm. There are various measures that can be used for the improvement of PSO. The hugeness of the populace is one of them. Huge populace enhances the probability of speedy and exact connection. Secondly, attaining an equilibrium linking exploration and exploitation is another procedure. At the initial stages of iteration, greater exploration results in higher probability of achieving an outcome closer to global optima. PSO has been utilized as an efficient method in feature selection. However, it has a few downsides, for example, premature convergence, high computational complexity, slow convergence. Additionally, it has the inadequacies of impulsive convergence with frailty in fine-tuning close local optimum positions (Rahman *et al.*, 2017) [18]. One potential explanation behind this is the way that PSO doesn't efficiently deal with the association linking exploitation (local search) with exploration (global search), consequently it habitually intersects to a local minimum so quick. Subsequently, to defeat the previously mentioned limitations and difficulties, Wang *et al.*, (2018) [19] and Sengupta *et al.*, (2019) [8] proposed that so as to have progressively proficient performance, suitable core update position formula and efficient strategy ought to be employed for adjustment of the global exploration with local exploitation**.** According to Nagra *et al.*, (2020) [20] and a few techniques for adjusting boundaries which includes inertia weight, and acceleration coefficients for PSO have been suggested to improve the exploration performance of PSO.

3. Research Methodology

The existing techniques need to be improved by the hybrid of chaotic dynamic weight PSO and DE. The method of research used in designing the proposed algorithm used to improve classification performance is described. The operational framework of this study is also presented.

3.1 Research Framework

The framework of the research contains all the procedures involved in order to accomplish the research objectives. The framework of this research comprises of 3 stages and each stage gives the result that is essential for the following stage. The outline of the framework for designing and developing the proposed CPSO algorithm is seen in Fig 2. The Phase 1 of the framework described the process of data pre-processing techniques that were used. Phase 2 focused on the initialization of PSO where chaotic logistic map was introduced into PSO in order to avoid premature convergence. Phase 3 shows the development of CPSO in which the particle swarm optimization was combined with chaotic dynamic weight (CPSO), so as to improve the convergence speed of PSO technique. The performance of this technique was measured using eight benchmark functions. The 8-benchmark function is given in Fig 4 (Kohli *et al.*, 2018; Houssein *et al.*, 2021) [22, 23].

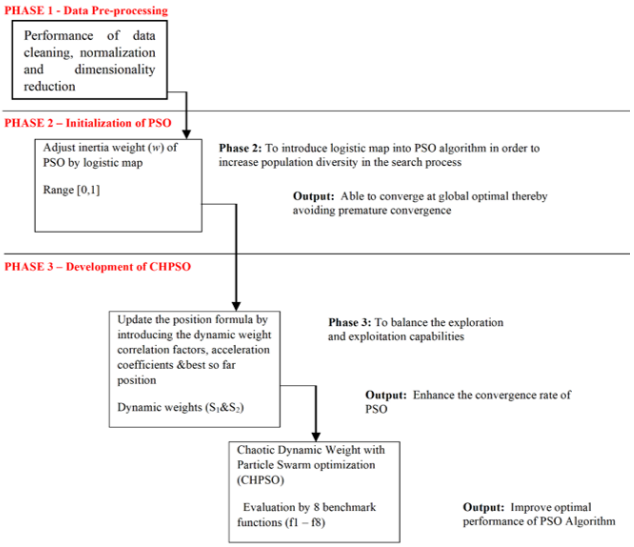

Fig 2: Operational Framework

3.2 Phase 1: Data Preprocessing

This section addresses the issue of the interference of different data attribute scaling. The attributes of the data are designed for the use of their current operational system. Usually, these raw attributes reduce the performance of machine learning classification power, because those with values that are higher dominate the attributes that have lower values. As a result, all of the original attributes were subjected to a normalization (linear) transform in order to produce new attributes which possesses improved properties and a more uniform distribution. "This will help the predictive power of the classifiers, it improves the accuracy of the classifier as well as reduces the computational time". Min-max normalization technique was adopted to scale all the numerical attributes to a new range denoted as [newMax$_A$ newMin$_A$]. This is because it preserves the relationship between the original values and the normalized ones. To obtain the transformed values, the expression in Equation 6. Was applied.

$$V^i = \frac{v - \text{MinA}}{\text{MaxA} - \text{MinA}} (newMax_A - newMin_A) + \text{newMin}_A \quad (6)$$

"where v' is the new normalized value, v is the current data point, $MinA$ and $MaxA$ are the lower and upper boundaries of the original attribute, $newMaxA$ and $newMinA$ are the respective lower and upper target boundaries. In this study, $newMaxA$ and $newMinA$ are set as 0 and 1 respectively". Each dataset is then partitioned into ratio 2:1 for training and testing sets. Dimensionality reduction was also performed in that it uses the mechanisms of encoding to reduce the size of dataset. Reduction can be lossless and lossy based on the retrieved information from decoding. Wavelet Transforms and Principal Component Analysis (PCA) are two effected methods for lossy reduction.

3.3 Phase 2: Initialization of PSO

Phase 2 contains the research activities which involves introducing chaos maps to adjust the inertia weight in order to increase the diversity of population in the search process of PSO to avoid premature convergence and improve the convergence performance of PSO by developing the ability to increase the convergence speed. Despite its convergence rate, the population maintenance of PSO is little, thereby causing it to converge prematurely. Hence, in order to lessen this impact and improve the convergence performance, a chaos dynamic weight particle swarm optimization (CPSO) was developed by introducing the chaos map into PSO algorithm and this is done by using the logistic map to adjust the inertia weight of PSO. The chaotic map positively has an impact on the convergence rate of PSO algorithm because this map induces chaos in the viable region which is predictable for a minimal initial time and stochastic for longer time period. "The range of the logistic map is always [0,1] and any number within that range can be chosen as the initial value, nevertheless it should be well-known that the initial value is able to have important effects on the pattern of fluctuation of some chaotic maps including logistic map" (Kohli & Arora, 2018) [22]. "This kind of chaotic maps are selected with various attitudes, while the initial value is set to 0.7 for all of them" (Mirjalili, 2019) [25]. Statistically, the inertia weight ω is represented in Eq 7.

$$\omega = {}_{\varphi} * cos\left(\frac{Mj}{Mmax}\right) * \pi) + \boldsymbol{\tau} \quad (7)$$

where $\boldsymbol{\varphi}$ and $\boldsymbol{\tau}$ are constant ($\boldsymbol{\varphi}$ = 1/3, $\boldsymbol{\tau}$ = 0.6), M_j is the current iteration number while M_{max} is the maximum iteration number.

3.4 Phase 3: Development of CPSO

This Phase comprises the research activities which involves modifying the position update formula so as to balance the exploration and exploitation capabilities to enhance escaping from local optimal. The exploitation (local search) and exploration (global search) capabilities need to be competently balanced (Marcelino *et al.*, 2018; Braik *et al.*, 2020) [26, 27]. Performing update on the position formula of PSO helps for adjustment of the global explore and local exploit. To accomplish this, two modifications are brought into the position update formula that are equipped for improving the PSO algorithm performance. In Equation 2, the location of the particle in the next generation is mostly determined by the current location, which is identified as x_{id} (t), and the flight speed, is called v_{id} (t+1). "This may reduce the ability to switch between exploitation and exploration in the process of searching" (Marcelino *et al.*, 2018) [26]. As a result of this link, two distinct weight rectification factors, s_1 and s_2, are introduced into Equation 2, as shown in Equation 8. These variables aid in changing the exploitation and exploration search measure, increasing the chances of achieving a faster rate of convergence. Eq 8. gives the location of each particle in the swarm.

$$x_{id}(t+1) = s_1(t) \text{ x } x_{id}(t) + s_2(t) \text{ x } v_{id}(t+1) \quad (8)$$

The fitness of the particle swarm optimization that are initialized in the search space are evaluated by using eight standard benchmark functions to test the CPSO performance.

4. The Proposed Approach

In this section, the proposed Chaotic Particle Swarm Optimization (CPSO) is presented.

4.1 Inertia Weight

The algorithm finds the resolution by operating in a series of cycles. The particles individually regulate the frontward direction in an attempt to find the global optimal solution, and this is as a result of its prior best position and the current best position of all the other elements. "The vector $x_i = (x_{i1}, x_{i2}, ..., x_{id})$ represents the former location of particle i in the search process and furthermore, $v_i = (v_{i1}, v_{i2}, ..., v_{id})$ denotes particle i's flying velocity, and vid's range is $[-V_{max}, V_{max}]$, where V_{max} is the maximum flying velocity, which is utilized to keep the value within a tolerable range" (Ajibade *et al.*, 2020) [4]. The PSO technique finds the global optimum resolution via upgrading the flight velocity and location of all the objects in the swarm according to Equation 1. The inertia weight (ω) controls the level of contribution of prior particle velocity to the current velocity and strikes a balance between exploration and exploitation characteristics of PSO. As a result of using the inertia weight (ω), Equation 1 is improved to Eq 3.

4.1.1. Chaotic Inertia Weight PSO (CIW-PSO)

The chaotic map is used to fine-tune PSO's inertia weight. The ability of PSO to escape from local optima when solving multimodal functions can be improved by modifying inertia weight using chaotic maps. The chaotic map is chosen to optimize the inertia weight of PSO, according to Niu *et al.*, (2017) [29] and Wei *et al.*, (2020) [30]. The logistic map's

ergodicity, irregularity, and non-repetition qualities allow it to boost population diversity in the search process while simultaneously increasing the ability to converge at the global optimal, avoiding premature convergence. The range of the logistic map is always [0,1] and any number within that range can be chosen as the initial value. Nevertheless, it should be well-known that the initial value is able to have important effects on the pattern of fluctuation of some chaotic maps including logistic map. This kind of chaotic maps are selected with various attitudes, while the initial value is set to 0.7 for all of them (Kohli *et al.*, 2018; Marcelino *et al.*, 2018) [22, 26]. The inertia weight ω is represented in Eq 7.

4.2 Chaotic Maps

"Chaos is a deterministic, arbitrary-like approach found in non-linear, dynamic, non-period, non-merging, and constrained systems" (Kohli & Arora, 2018) [22]. Chaos is a basic and unpredictable deterministic dynamical framework that occurs numerically, and the system that entails chaos can be considered as a source of unpredictability. "Several chaotic maps have been proposed in the literature, each with its own set of mathematical equations and because of their dynamic behavior, which assists optimization procedures in probing the search space globally, chaotic maps have been widely employed in the field of enhancement since a decade ago" (Kohli & Arora, 2018) [22]. Numbers in the range [0,1] can be used as the beginning value in chaotic maps. Some of the well-known chaotic maps are:

Cubic Map: The cubic map produces values within the range [-1.5,1.5]. It is almost similar to Sine map. It is defined as:

$$X_{k+1} = 3x_k (1-X^2_k) \quad (9)$$

where X_k is the kth chaotic number, with k denoting the iteration number.

Sine Map: The sine map is a unimodal map and it is represented by:

$$X_{k+1} = \frac{a}{4} \sin(\pi x_k) \quad (10)$$

where X_k is the kth chaotic number, with k denoting the iteration number.

Chebyshev Map

The Chebyshev map is represented by Eq 11.

$$X_k+1 = \cos(K \cos\text{-}1 (X_k)) \quad (11)$$

where X_k is the kth chaotic number, with k denoting the iteration number where $-1 < X_k < +1$; n=0,1,2…

Circle Map

The map contains two parameters: *a* (nonlinearity strength) and *b* (bilinearity strength) (externally applied frequency). It's a one-dimensional map that maps a circle into itself, and Eq 12 is the equation that represents it.

$$X_{k+1} = (x_k + b - \frac{a}{2\pi} \sin(2\pi x_k)) \bmod (1) \quad (12)$$

Logistic Map: The equation of logistic map is a non-linear dynamic of population which is biological evidencing chaotic behaviour. The map is represented by:

$$X_{k+1} = ax_k (1 - X_k) \quad (13)$$

where X_k is the kth chaotic number, with k denoting the iteration number.

In the method of using the logistic chaotic map

The logistic map formula is used to generate a chaotic random sequence Z between [0, 1], which is then passed through the carrier map in Eq 14 and included into gbest, a neighboring area, to complete the local chaotic search:

$$Z \rightarrow Y: X = gbest + R \times \cos(Z) \quad (14)$$

R is the search radius, which is used to limit the scope of local chaotic search. $R \in [0.1, 0.4]$.

4.3 Feature Selection using chaotic PSO (CPSO)

The chaotic time-series sequences, such as logistic map is applied in optimization problems to solve the challenges of global convergence rate and premature convergence (Demir *et al.*, 2020) [32]**.** To overcome the challenges of slow convergence, premature convergence, and to avoid being trapped in the local optimum, this research work proposes chaotic weight particle swarm optimization algorithm known as CPSO. In this algorithm, there are two main climaxes which are explained below.

First, "the chaos map in the CPSO has the properties of ergodicity, non-repetition, and sensitivity, and it can perform straightforward searches at faster rates than stochastic searches, which rely heavily on probabilities" (Gotmare *et al.*, 2018) [33]. As a result, the inertia weight is adjusted using the chaotic map. In making decisions that affect the speed of convergence, the inertia weight in Equation 7 is vital. In Equation 7, the inertia weight reduces linearly as the number of iterations increases, reducing the convergence speed and accuracy of PSO (Shi & Eberhart, 1998; Taherkhani & Safabakhsh, 2016; Abualigah *et al.*, 2021) [14, 37, 34]. In the PSO algorithm (kumar & Bharti 2019) [38] proposed that ω need not to repeatedly decrease linearly rather should linearly increase in order to avoid premature convergence and enhance escaping from local optimal. According to Niu *et al.*, (2017) [29] & Wang *et al.*, (2018) [19], the chaotic map was chosen to adjust the inertia weight ω. Second, the powers of exploration and exploitation are mutually exclusive. "The exploration and exploitation abilities must be well matched in order to achieve higher optimization performance, exploration refers to the ability to locate the global ideal in the solution space, whereas exploitation refers to the ability to use information from previous solutions to find a superior one". According to Equation 2, the current position, denoted by $x_{id}(t)$, and the flight velocity, denoted by v_{id} (t + 1), have a crucial role in the location of the future generation particle. In the search process, this can impair the ability to balance exploration and exploitation. As a result, three key adjustments are recommended to increase PSO performance: the introduction of dynamic weights, the acceleration coefficient, Combining the best-so-far position with the other two components to update the new position. Two additional parameters have been added to the position update equation, which have the potential to improve the PSO algorithm's optimal performance. Equation 2 incorporates two dynamic weight adjustment factors, s_1 and s_2, based on this relationship. Furthermore, the particle's search space is expanded by incorporating these two dynamic correction

variables, allowing the particles to explore a new location. As a result, incorporating the two dynamic weight adjustment elements increases the possibilities of achieving excellent performance. The proposed Equation 15 is used to update the position of each particle in the swarm.

$$x_{id}(t+1) = s_1(t) \times x_{id}(t) + s_2(t) \times v_{id}(t+1) + \rho x\, gbest^d \times \psi \quad (15)$$

As shown in Eq 15, these two parameters (s_1 and s_2) regulates the exploitation and exploration processes via monitoring the impact of the prior solution x_{id} (t) and the velocity v_{id} (t + 1), $gbest_d$ is the best so far position that accelerates the convergence rate, whereas ρ is a random number between 0 and 1 and ψ is the acceleration coefficient that controls the maximum step size. In the PSO searching process, the dynamic weight and acceleration coefficient are given as fitness functions. These three changes were suggested during the search stage to fine-tune the parameters used to determine new particle placements. Nonlinear functions of particle position define the two modification factors. The variables s_1, s_2, and ψ are defined in Eq 16, Eq 17 and Eq 18:

$$s_1(t) = \psi = \frac{1}{(1+\exp(a.(-\min(SP)/\max(SP))))t} \quad (16)$$

$$s_2(t) = 1 - s_1(t) \quad (17)$$

$$SP(t) = x_{id}(t-1) \times [x_{id}(t-1)]^T \quad (18)$$

where SP is the particle position value; t is the current iteration, a is a constant (a=2). The pseudo code of CPSO is shown in Fig. 3.

1: ***Input:*** NP, D, c_1, c_2, ω, M_{max}, V_{min}, V_{max}
2: Randomly initialize each particle position X*i* (*i*=1,2, …, *NP*) in the swarm
3: Generate the initial velocities V*i* (*i*=1,2, …, *NP*) for each particle randomly
4: Calculate the fitness value of each particle using the objective function *f*
5: Set *pbest* and *gbest* in the swarm
6: Update the particle velocity V_i using Eq 3.
7: Update the particle position X_i using Eq 2.
8: Chaotic search *K* times near gbest
9: Update the position of particle velocity by adjusting inertia weight with chaotic logistic map using Eq 14.
10: *K* chaotic search points near $Pbest_i$ are obtained
11: Calculate weight factors s_1 and s_2 using Eq 16. And Eq 17.
12: Update particle position of each particle by proposed update equation using Eq 15.
13: ***end***

Fig 3: The Proposed CPSO Algorithm

5. Experimental Result

In this study, four well-known datasets were used to evaluate the performance of the proposed technique. These datasets are XAPI dataset, UCI student dataset, Quality Wine dataset and Breast Cancer Wisconsin dataset respectively. The first dataset (XAPI dataset) was gotten from an LMS called Kalboard 360 (Amrieh *et al.*, 2015) [40], while the other datasets were retrieved from the UCI data repository. Eight numerical benchmark functions are used to test the performance of the proposed algorithm with 30 and 50 dimensions and the benchmark functions are listed in Fig 4. All of the functions chosen are well-known in the literature on global optimization. Unimodal and multimodal test functions are included in the test functions.

CPSO is run on a PC with a 2.60GHz Intel Core (i5-2540M) processor and 6GB RAM running Microsoft Windows 7 Professional with Matlab 9.3 (R2017b) (64-bit). Table 1 lists the parameters that were used. The average NFE, fitness value means, standard deviation, average classification accuracy, average number of selected features, and metaheuristic technique performance evaluation were all presented. The KNN classifier is utilized to evaluate the metaheuristics technique in 30 runs. Each dataset is split into two parts: 70 percent was used for training and 30% was used for testing.

No	Function	Formula	Value	Dim	Range	Properties
1	Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	0	30	[-100, 100]	Unimodal, Separable
2	Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	0	30	[-30, 30]	Unimodal, Inseparable
3	Quartic	$f(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	0	30	[-1.28, 1.28]	Unimodal, Inseparable
4	Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}}+1$	0	30	[-600, 600]	Multimodal, Inseparable
5	Sumsquare	$f(x)=\sum_{i=1}^{n}ix_i^2$	0	30	[-5.12, 5.12]	Unimodal, Separable
6	Schwefel	$f(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	0	30	[-500, 500]	Multimodal, Separable
7	Rastrigin	$f(x)=\sum_{i=1}^{n}x_i^2-10\cos(2\pi x_i)+10$	0	30	[-5.12, 5.12]	Multimodal, Separable
8	Michalewicz5	$f(x)=-\sum_{i=1}^{n}\sin(x_i)(\sin(ix_i^2/\pi))^{2m}$	-4.688	5	[0, π]	Multimodal, Separable

Fig 4: Commonly used Numerical Benchmark Functions for Comparison

5.1 Result and Discussion

To test the efficiency of the chaos map which was merged with dynamic weight, there was a comparison of the original PSO algorithm with the improved PSO called CPSO, as well as other metaheuristics like DE and GA, in terms of convergence speed, as measured by the number of function evaluations (NFE) and solution accuracy (mean fitness values, SD). Convergence time is expressed in seconds (s). A smaller NFE indicates a faster convergence rate. The results are provided in Section 5.1.1.

5.1.1 Comparison on NFE, Mean Fitness Value and Standard Deviation for Proposed CPSO and other metaheuristics algorithms

To tackle the problem of sluggish convergence, the performance of PSO has been improved in this section by merging the chaotic dynamic weight with PSO. Four unimodal and four multimodal functions make up the benchmark functions. For 8 benchmark functions in dimension 30, Table 2 compares the avg NFE of the GA, DE, PSO and CPSO algorithms.

Table 1: Parameter Setting

Parameter	Description of Parameter	Setting Value
Pop size (NP)	Particle and Population size	50
Dimension (D)	Dimension of the obj. function i.e. No of independent variables	30,50
Acceleration coefficient (c_1, c_2)	The cognitive component (c_1) and social component (c_2) are used to find the optimal solution of the algorithm. ($c_1 = c_2$)	2.0
Inertia Weight (ω)	Determines the next position of particle	(0.4, 0.9)
Value to reach (VTR)	Determines the termination criteria to which the best value found by the algorithm is reduced to.	10^{-12}
MAX_{NFE}	Max NFE is set as a stoppage criterion so as to make sure that the algorithms being compared have sampled the search space equal number of time	20000
V_{max}, V_{min}	They are set for values of velocity which are for each of the particles to regulate the velocity within a realistic range	(4, -4)
$Rand_1$, $Rand_2$	They are two uniform random numbers	(0,1)

Table 2: Average NFE of GA, DE, PSO and the proposed CPSO for D=30

	Average NFE			
Function	**GA**	**DE**	**PSO**	**CPSO**
*f*1	20014	**3784**	16984	7124
*f*2	20019	8419	17169	**3879**
*f*3	20013	16542	20001	**9898**
*f*4	14574	**7601**	17146	8145
*f*5	12491	7883	18633	**6635**
*f*6	20000	**5934**	20001	16252
*f*7	20055	**2778**	8226	10163
f8	13682	**5113**	20001	6689

From Table 2, the proposed CPSO requires least average NFE to realize the acceptable solution on the functions and because when a smaller value of NFE is obtained, then this denotes that the rate of convergence is higher. The table shows that the proposed CPSO algorithm has a higher convergent rate when compared with the original PSO because the population diversity of PSO was improved to give the proposed CPSO a better performance. However, the DE algorithm shows faster convergence in functions (*f*1, *f*4, *f*6, *f*7, *f*8) as compared to the proposed CPSO which outperforms other algorithms in functions (*f*2, *f*3, *f*5). Showing a faster convergence means that it is able to converge at an optimal solution within the shortest period of time and this is measured in seconds (s). For 8 benchmark functions in dimension 50, Table 3 compares the avg NFE of the GA, DE, PSO and CPSO algorithms.

Table 3: Average NFE of GA, DE, PSO and the proposed CPSO for D =50

	Average NFE			
Function	**GA**	**DE**	**PSO**	**CPSO**
*f*1	20025	**8539**	11702	10905
*f*2	20016	20001	12525	**8191**
*f*3	20013	**19521**	20000	19948
*f*4	18308	**7306**	18627	18462
*f*5	19536	14473	20000	**8239**
*f*6	18999	**14896**	20000	20001
*f*7	20018	3093	**1335**	13711
*f*8	16941	**10829**	20000	19976

Table 3 shows that CPSO takes the fewest avg NFE number of function evaluations to provide a satisfactory result on the functions. Because the population diversity of PSO was increased to give the proposed CPSO a better performance, the proposed CPSO obtained a better convergent rate over PSO and GA. However, the DE algorithm surpasses the proposed CPSO in most functions (*f*1, *f*3, *f*4, *f*6, *f*8) while the proposed CPSO outperforms DE in functions (*f*2 and *f*5) and the PSO algorithm outperforms the other algorithms in function *f*7.

In 30 independent runs, Tables 4 demonstrate avg mean fitness values and standard deviation (SD) by GA, DE, PSO, and the proposed CPSO for dimensions 30. Eight benchmark functions are utilized. The proposed CPSO algorithm yielded the best mean fitness for three functions (*f*1, *f*3, and *f*5), whereas the DE algorithm yielded the best mean fitness function for four functions (*f*4, *f*6, *f*7, and *f*8), and the PSO algorithm yielded the greatest mean fitness function value for (*f*2). DE provides the best performance in dimension 30 when compared to the proposed CPSO, PSO, and GA. The result in the table implies that the proposed CPSO even though it has a better performance than the original PSO in escaping from local optimal, it still has high chances of getting stuck in local optimal than the DE algorithm.

Table 4: Average mean fitness and standard deviation (SD) of functions for DE, GA, PSO and the proposed CPSO in 30 runs for D=30

D = 30								
Function	**DE**		**GA**		**PSO**		**CPSO**	
	Mean	S.D	Mean	S.D	Mean	S.D	Mean	S.D
*f*1	4.8E-06	3.02E-06	4.09E-16	1.94E-15	1.2E-201	1.98E-22	**5.49E-23**	**0.640088**
*f*2	5.92E-06	3.29E-06	1.866335	3.29E-14	**7.63E-15**	**1.487041**	6.9E-22	3.74E-21
*f*3	5.17E-06	3.08E-06	1.53E-23	7.9E-23	0.216669	122.8667	**238.7992**	**0.408601**
*f*4	**5.812331**	**1.214471**	3.49063	5.192793	0.398026	4.165899	2.255447	2.006048
*f*5	5.53E-06	3.42E-06	0.530645	0.568466	0.051125	1.801896	**5.603973**	**0.280023**
*f*6	**5.374099**	**0.021788**	0.068839	0.042799	0.04746	2.621595	0.054535	0.039852
*f*7	**7.13E-07**	**2.27E-06**	4.48E-11	2.45E-10	2E-16	3.87E-06	4.97E-06	9.25E-16
*f*8	**9.435544**	**0.669746**	6.62E-06	2.97E-06	0.329245	1.891857	5.39E-15	8.99E-15

In Table 5, the proposed CPSO method yielded the highest mean fitness for two functions (*f*2, *f*4 and *f*7), whereas the DE algorithm yielded the highest mean fitness function for four functions (*f*1, *f*5, *f*6, and *f*8) and GA yielded the highest mean fitness function value for one function (*f*3). When compared to the proposed CPSO method, PSO, and GA, it is found that DE performs best in dimension 50. The proposed CPSO algorithm outperforms the original PSO. Again, the result in the Table implies that the proposed CPSO although it gives a better performance than the original PSO in escaping from local optimal, it still possesses high chances of getting stuck in local optimal than the DE algorithm which has proven to have more capability of jumping out of local optimal and converge at global solutions and that is why the DE outperforms every other algorithm.

Table 5: Average mean fitness and standard deviation (SD) of functions for DE, GA, PSO and the proposed CPSO in 30 runs for D=50

D = 50								
Function	**DE**		**GA**		**PSO**		**CPSO**	
	Mean	S.D	Mean	S.D	Mean	S.D	Mean	S.D
*f*1	**13.9898**	**1.38E-05**	7.7795E-06	2.21E-06	3.53E-06	5.1975	3.75E-18	1.73E-17
*f*2	7.0194E-06	6.11E-18	4.07E-18	3.45E-06	70.28041	23.27691	**1.62E-06**	**2.44E-06**
*f*3	185.893255	55.56902	**2.894472**	**3.469198**	2931.641	1015.749	372.8988	812.7194
*f*4	21.07463	26.6819	22.39456	25.14135	226.1307	66.2979	**8.37941774**	**2.211705**
*f*5	**1.05637886**	**0.820624**	4.775809	3.084979	47.41923	11.92975	2.327313	5.310664
*f*6	**0.127584**	**0.058745**	0.44720163	0.078309	48.44681	15.95185	0.141085	0.079965
*f*7	1.07E-20	3.5E-06	4.8E-08	2.61E-07	4.8407E-06	0.001491	**0.001162**	**4.04E-20**
*f*8	**16.51541**	**0.001183**	0.60455	0.675166	0.00187485	1.346292	4.48E-08	4.58E-08

For 30 runs, Table 6 illustrates the average classification accuracy and number of minimal selected features. CPSO has the highest classification accuracy for all four datasets with the smallest number of specified features, as shown in the table. The results successfully confirm the findings of Amrieh et al., (2015) [40] & Mohammadi-Balani *et al.*, (2021) [43], which show that classification accuracy may be improved with a small number of characteristics. The result also means that the performance algorithm's construction cost and computing time are lowered throughout the training and classification stages.

Table 6: Average classification accuracy and average number of selected features for 30 runs

Datasets		GA	DE	PSO	CPSO
XAPI dataset	Average Classification Accuracy	70.78	71.72	76.39	78.71
	Number of Selected Features	10	8	7	5
UCI student dataset	Average Classification Accuracy	82.63	87.18	85.04	88.46
	Number of Selected Features	10	9	9	7
Wine Quality	Average Classification Accuracy	80.00	80.21	81.74	82.83
	Number of Selected Features	8	8	6	5
Breast Cancer Wisconsin	Average Classification Accuracy	90.02	92.21	94.65	98.53
	Number of Selected Features	8	7	4	3

Table 7 displays the results of four meta-heuristic feature selection strategies, as well as the Precision, Recall, and F-measure values calculated using the KNN classifier. The proposed CPSO algorithm had the greatest overall performance in accurately categorizing the features for the four datasets, as indicated in the table.

Table 7: Performance Evaluation of Meta heuristics Feature Selection Techniques

	DE			GA			PSO			CPSO		
Datasets	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**
XAPI dataset	73.74	75.00	74.36	73.74	75.00	74.36	73.94	75.10	74.52	75.15	75.92	75.71
UCI student dataset	91.71	88.23	90.00	91.15	88.17	89.64	92.22	89.36	90.80	99.11	95.69	97.36
Wine Quality	93.67	69.02	79.48	93.67	68.66	79.24	93.67	89.02	91.29	91.77	89.43	90.58
Breast Cancer Wisconsin	98.34	96.50	97.41	97.52	95.10	96.29	98.25	99.22	98.73	97.31	97.96	97.63

6. Conclusion and Future Work

CPSO, a chaotic variation of the PSO algorithm, was invented and applied to feature selection problems in this study. The logistic chaotic map was utilized to replace the random operator in the algorithm that regulates the balance of exploration and exploitation. The proposed algorithm's performance was assessed using four benchmark datasets. The results were compared to the GA, DE, and PSO meta heuristic feature selection algorithms. CPSO performed well, and the findings demonstrated the power of the logistic chaotic map in improving the quality of the PSO algorithm. On most traditional benchmark functions, the CPSO algorithm developed in this paper outperformed all other algorithms in terms of convergence speed and bigger optimization accuracy. It also performs better in categorization, making it a viable alternative for numerical optimization tasks. However, better strategies for parameter tuning can be further developed and the application of CPSO algorithm can be studied in solving practical problems. The proposed CPSO algorithm displayed good optimization performance when evaluated with the eight (four unimodal and four multimodal) benchmark functions, however the DE algorithm seem to perform better for most of the numerical functions for the average NFE when compared to our proposed CPSO technique. Therefore, a future work could be to examine the hybrid of DE with the proposed CPSO algorithm. Also, in the future more multimodal and unimodal benchmark functions can be used to further evaluate the algorithm for a wider view of comparison.

7. Refernces

1. Gopika N, ME AMK. Correlation based feature selection algorithm for machine learning. In: 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Institute of Electrical and Electronics Engineers; c2018;692-695.
2. Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection. Applied Soft Computing. 2018;62:441-453.
3. Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software. 2016;95:51-67.
4. Ajibade S. Particle Swarm Optimization with Chaotic Dynamic Weight for Feature Selection Enhancement. Journal of Science, Engineering Technology and Management. 2020;1(2):1-5.
5. Eberhart R, Kennedy J. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, IEEE; c1948-1995; 4:1942.
6. Moradi P, Gholampour M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Applied Soft Computing. 2016;43:117-130.
7. Ajibade SS, Adediran A. An overview of big data visualization techniques in data mining. International Journal of Computer Science and Information Technology Research. 2016;4(3):105-113.
8. Sengupta S, Basak S, Peters RA. Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives. Machine Learning and Knowledge Extraction. 2019;1(1):157-191.
9. Mafarja MM, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing. 2018;260:302-312.
10. Ajibade SSM, Ahmad NB, Shamsuddin SM. A data mining approach to predict academic performance of students using ensemble techniques. In: International Conference on Intelligent Systems Design and Applications, Springer; c2018;749-760.
11. Ajibade SSM, Ahmad NBB, Zainal A. A hybrid chaotic particle swarm optimization with differential evolution for feature selection. In: 2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA); c2020; 1-6.
12. Wang D, Tan D, Liu L. Particle swarm optimization algorithm: An overview. Soft Computing. 2018;22(2):387-408.
13. Yan Y, Zhang R, Wang J, Li J. Modified PSO algorithms

with Request and Reset for leak source localization using multiple robots. Neurocomputing. 2018;292:82-90.
14. Shi Y, Eberhart R. A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360); c1998;69-73.
15. Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation. 2015;20(4):606-626.
16. Rennard JP. Pierre Collet. Handbook of Research on Nature-Inspired Computing for Economics and Management; c2006;28.
17. Ajibade SSM, Ahmad NB, Shamsuddin SM. An heuristic feature selection algorithm to evaluate academic performance of students. In: 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC); c2019;110-114.
18. Rahman L, Setiawan NA, Permanasari AE. Feature selection methods in improving accuracy of classifying students' academic performance. In: 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE); c2017;267-271.
19. Wang D, Tan D, Liu L. Particle swarm optimization algorithm: An overview. Soft Computing. 2018;22(2):387-408.
20. Nagra AA, Han F, Ling QH, Abubaker M, Ahmad F, Mehta S, Apasiba AT. Hybrid self-inertia weight adaptive particle swarm optimization with local search using C4.5 decision tree classifiers for feature selection problems. Connection Science. 2020;32(1):16-36.
21. Ajibade SSM, Ahmad NBB, Shamsuddin SM. Educational data mining: enhancement of student performance model using ensemble methods. In: IOP Conference Series: Materials Science and Engineering; IOP Publishing; 2019;551:012061.
22. Kohli M, Arora S. Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of Computational Design and Engineering. 2018;5(4):458-472.
23. Houssein EH, Gad AG, Hussain K, Suganthan PN. Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application. Swarm and Evolutionary Computation. 2021;63:100868.
24. Ajibade SSM, Ahmad NBB, Shamsuddin SM. A novel hybrid approach of Adaboostm2 algorithm and differential evolution for prediction of student performance. International Journal of Scientific and Technology Research. 2019;8(07):65-70.
25. Mirjalili S. Biogeography-based optimisation. In: Evolutionary Algorithms and Neural Networks; Springer; c2019;57-72.
26. Marcelino C, Almeida P, Pedreira C, Caroalha L, Wanner E. Applying C-DEEPSO to solve large scale global optimization problems. In: 2018 IEEE Congress on Evolutionary Computation (CEC); IEEE; c2018;1-6.
27. Braik M, Sheta A, Al-Hiary H. A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. Neural Computing and Applications; c2020;1-33.
28. Ajibade SSM, Dayupay J, Ngo-Hoang DL, Oyebode OJ, Sasan JM. Utilization of Ensemble Techniques for Prediction of the Academic Performance of Students. Journal of Optoelectronics Laser. 2022;41(6):48-54.
29. Niu P, Chen K, Ma Y, Li X, Liu A, Li G. Model turbine heat rate by fast learning network with tuning based on ameliorated krill herd algorithm. Knowledge-Based Systems. 2017;118:80-92.
30. Wei CL, Wang GG. Hybrid annealing krill herd and quantum-behaved particle swarm optimization. Mathematics. 2020;8(9):1403.
31. Ajibade SSM, Ogunbolu MO, Chweya R, Fadipe S. Improvement of Population Diversity of Meta-heuristics Algorithm Using Chaotic Map. In: International Conference of Reliable Information and Communication Technology; Springer, Cham; c2022;95-104.
32. Demir FB, Tuncer T, Kocamaz AF. A chaotic optimization method based on logistic-sine map for numerical function optimization. Neural Computing and Applications. c2020;1-13.
33. Gotmare A, Keskar NS, Xiong C, Socher R. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. arXiv preprint, 2018. arXiv:1810.13243.
34. Abualigah L, Yousri D, Abd Elaziz M, Ewees AA, Al-qaness MA, Gandomi AH. Aquila Optimizer: A novel meta-heuristic optimization Algorithm. Computers & Industrial Engineering. 2021;157:107250.
35. Ajibade SSM, Ahmad NBB, Zainal A. Analysis of Metaheuristics Feature Selection Algorithm for Classification. In: International Conference on Hybrid Intelligent Systems; Springer, Cham; c2021;214-222.
36. Shi Y, Eberhart R. A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360); IEEE; c1998;69-73.
37. Taherkhani M, Safabakhsh R. A novel stability-based adaptive inertia weight for particle swarm optimization. Applied Soft Computing. 2016;38:281-295.
38. Kumar L, Bharti KK. An improved BPSO algorithm for feature selection. In: Recent Trends in Communication, Computing, and Electronics; Springer, Singapore; c2019;505-513.
39. Rabbi F, Ayaz M, Dayupay JP, Oyebode OJ, Gido NG, Adhikari N, *et al*. Gaussian Map to Improve Firefly Algorithm Performance. In: 2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC); IEEE; c2022;88-92.
40. Amrieh EA, Hamtini T, Aljarah I. Preprocessing and analyzing educational data set using X-API for improving student's performance. In: 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT); IEEE; c2015;1-5.
41. Wei Q, Guo Z, Lau HC, He Z. An artificial bee colony-based hybrid approach for waste collection problem with midway disposal pattern. Applied Soft Computing. 2019;76:629-637.
42. Ajibade SSM, Oyebode OJ, Dayupay JP, Gido NG, Tabuena AC, Kilag OKT. Data Classification Technique for Assessing Drug Use in Adolescents in Secondary Education. Journal of Pharmaceutical Negative Results. c2022;971-977.
43. Mohammadi-Balani A, Nayeri MD, Azar A, Taghizadeh-Yazdi M. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. Computers &

Industrial Engineering. 2021;152:107050.
44. Ajibade SSM, Mejarito C, Egere OM, Adediran AO, Gido NG, Bassey MA. An Analysis of the Impact of Social Media Addiction on Academic Engagement of Students. Journal of Pharmaceutical Negative Results. 2022;1390-1398.
45. Sokkhey P, Navy S, Tong L, Okazaki T. Multi-models of Educational Data Mining for Predicting Student Performance in Mathematics: A Case Study on High Schools in Cambodia. IEIE Transactions on Smart Processing and Computing. 2020;9(3):217-229.
46. Chaudhury S, Oyebode OJ, Hoang DLN, Rabbi F, Ajibade SSM. Feature Selection for Metaheuristics Optimization Technique with Chaos. In: 2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA); IEEE; 2022; 365-370.
47. Ajibade SSM, Adhikari N, Ngo-Hoang DL. An Analysis of Social Networking for E-learning in Institutions of Higher Learning using Perceived Ease of use and Perceived Usefulness. Journal of Scientometric Research. 2022;11(2):246-253.