Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 5 (2023) 1007

Journal of the Nigerian Society of Physical Sciences

Hyper-parameter tuning for support vector machine using an improved cat swarm optimization algorithm

Silifat Adaramaja Abdulraheem, Salisu Aliyu*, Fatimah Binta Abdullahi

Department of Computer Science, Faculty of Physical Science, Ahmadu Bello University Zaria, Nigeria

Abstract

Support vector machine (SVM) is a supervised machine learning algorithm for classification and regression problems. SVM performs better when combined with other classifiers or optimized with an optimization algorithm. The SVM parameters such as kernel and penalty have good performance on the classification accuracy. Recently, a lot of evolutionary optimization algorithms were used for optimizing the SVM. In this paper, an Improved Cat Swarm Optimization (ICSO) was proposed for optimizing the parameters of SVM with the aim of enhancing its performance. CSOs have the problem of a low convergence rate and are easily trapped in local optima. To address this problem, a new parameter was added to the velocity for the tracing mode and the Opposition-Based Learning (OBL) technique was used to modify the CSO algorithm (ICSO-SVM). A new parameter was introduced to guide the cats' positions to the local and global best positions in the velocity tracing mode of the CSO algorithm. The proposed algorithm was verified using 15 datasets from the University of California Irvine (UCI) data repository and also six different performance metrics were used. The experimental results clearly indicate that the proposed method performs better than the other state-of-the-art methods.

DOI:10.46481/jnsps.2023.1007

Keywords: Cat swarm optimization, Support vector machine, Opposition-based learning

Article History :
Received: 28 August 2022
Received in revised form: 24 June 2023
Accepted for publication: 21 July 2023
Published: 08 September 2023

Communicated by: T. Latunde

1. Introduction

Machine Learning (ML) is an artificial intelligence subfield that uses machines to understand big datasets with no training data [1]. In recent years, other ML algorithms acquire developed experience, including artificial neural networks, support vector machines, random forests, naive Bayes, and K-nearest neighbors. ML techniques have also been applied to solve other complex problems [2]. This subfield of artificial intelligence has received increasing attention as computers have gained the mathematical capacity to develop complex models capable of processing and learning from unstructured raw data [1]. For example, clinicians are required to generate real organized attributes to interpret input, the input is converted by an algorithm to its original form, and then recognizes the attributes to the related desired results. The processes that machine used to learn are known as "Algorithms". Machine learning algorithms can be categorized into four basic types: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Supervised Learning uses the labeled examples to adapt to what they have learnt previously to a new data for

*Corresponding author: Tel.: +2348067993631
Email address: aliyusalisu@abu.edu.ng (Salisu Aliyu)

future events prediction. Supervised Learning is like giving students answer and asking them to show their work. How well a machine-learning system works depends on the type of data and the learning functions of the algorithms [3].

Artificial Intelligence (AI) has become more prevalent in business and industry. It has the ability to communicate, discover things, learn and work. Various researchers have defined AI as a domain of science that authorizes machines to find and fix problems the way humans do [4]. AI has also revolutionized as many people as possible through smartphones, in healthcare, as well as assist in recognizing people's problems and developing solutions for them [5]. AI has assisted and improves human performance in various aspects of system management. For example, artificial intelligence (AI) have been used to boost organization's performance, reliability, client satisfaction, and return on capital while at the time empowering employees [6]. Artificial intelligence (AI) has transformed trade, the financial system, and society by converting decision making policies and individuals' experiences [7].

Support Vector Machine (SVM) are supervised learning algorithm used for many classification and regression problems [8]. SVM is derived from the statistical learning principle and theory that provides a framework for analyzing the process of collecting data and making predictions based on the data. In short, it allows hyper-plane space selection to strongly represent the fundamental feature in the target space. As noted, SVM is used in solving classification and regression problems in the fields such as pattern recognition and speech recognition [9]. SVM is a novel machine learning algorithm that has recently attracted the interest of researchers. This is due to its high performance and a strong record of achieving high accuracy levels in the shortest possible time. Support Vector Machine performance are determined based on how the parameters are chosen [10]. The process of selecting the set of optimal parameters for a machine learning algorithm is known as hyper-parameter optimization problem.

A variety of bio-inspired algorithms, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) are proposed for solving optimization problems in several areas, including intrusion detection scheduling [11]. Research has shown that the CSO algorithm performs better than the Particle Swarm Optimization and the Genetic Algorithms [12]. CSO, is an evolutionary optimization algorithm that replicates the cats' natural behavior [12]. Cat swarm optimization (CSO) is an optimization algorithm that focuses on how cats search and hunt for their prey. Cat's behaviors are categorized into two modes: the seeking mode and the tracing mode. Seeking mode refers to a cat's ability to be observant in its environment while sleeping. Tracing mode on the other hand, simulates how cats track and catch their prey [13]. The convergence speed is one of the issues to consider when using this optimization technique. It is worth noting that majority of the optimization algorithms have relatively slow convergence rates.

CSO algorithm has been improved by different approaches which including the following: In Ref. [14], the authors improved the CSO's exploration capability by employing opposition-based learning. Gomathy also used opposition-based learning and the Cauchy mutation operator to prevent it from falling into local optima. Similarly, in [15], the authors proposed an Average-Inertia weighted to overcome the premature convergence problem caused by low diversity. Therefore, CSO have the problem of low convergence rate and hence, can easily be trapped in local optima. The use of CSO for optimizing the parameters of SVM is also affected by the premature convergence problem of the CSO. To address this problem, the velocity model for the Cat Swarm tracing mode was modified using Opposition-Based Learning (OBL) technique and used to improve the CSO algorithm.

This article aims to tune the SVM parameters using an improved CSO algorithm. The CSO algorithm was improved by modifying the tracing mode velocity and then applying the Opposition-Based Learning (OBL). OBL, though a technique inspired by the inverse relationship between objects has proven to be effective [16]. Soft computing algorithms that have used the concept of OBL to enhance their efficiency include optimization methods, fuzzy systems, artificial neural networks, and reinforcement learning. The main contributions of this paper are; the proposed improved CSO algorithm, the application of the improved CSO algorithm for the hyper-parameter tuning of the SVM classifiers and the application of the hybrid ICSO-SVM algorithm for data classification.

The remaining part of this paper is arranged as follows: section two discusses preliminary concepts. Section three describes the methodology. Section four discusses the experiments conducted. Finally, section five concludes the work with some recommendations for future directions.

In Ref. [17], the authors present a Parallel Cat Swarm Optimization (PCSO) that improves the convergence speed as the population size reduces. Lin & Chien [18], proposed the CSO+SVM data classification model that integrates cat swam optimization with the SVM classifier. The training of the parameter optimization can help improve classification accuracy. Their experiment was tested on some datasets from the UCI data repository and the comparison was made on GA+SVM demonstrating that the CSO+SVM approach outperforms GA+SVM in terms of performance. Santosa & Ningrum [19] used CSO to cluster data of which their results shows that CSO outperformed the K-means and the PSO.

Orouskhani *et al*. [15] added an inertia value to the velocity model to make a balance between the exploitation and the exploration phase. They observed that the function (w) is best chosen in the range [0.4, 0.9], where it is set to 0.9 at the start of the operation and gradually decreases to the end. High numbers of (w) have helped with the global searches, while low numbers of (w) help with the local searches. Their results also indicate that CSO achieved a better result than PSO. Tsai *et al*. [20] enhanced the PCSO by incorporating the Taguchi orthogonal array approach and naming it an enhanced parallel cat swarm optimization (EPCSO).

Orouskhani *et al*. [21] modified the CSO algorithm by proposing three major changes. To begin, they added a variable inertia value to the velocity formula and as the dimension numbers increases, this value gradually decreases. Secondly, the constant (C) was changed to a variable value and finally, they reformed the position equation using information from other dimensions. Lin *et al*. [12] proposed a modified cat swarm optimization (MCSO) to achieve the capability to search the neighborhood of the optimum solution by combining parameter optimization and feature selection for the SVMs. In comparison to the original CSO algorithm, the experimental results indicated that MCSO performed better with small sets of features for the given UCI datasets.

Hadi & Saba [22] combined two principles to enhance the algorithm, named ICSO. The first principle is borrowed from PCSO, parallel tracing mode and information exchange. The second AICSO is derived from a principle of adding an inertia weight to the position formula using a parameter called MR in the original CSO algorithm. Manurung *et al*. [23] introduced the use of Genetic Algorithm (GA) to find the solution to the problem by determining the optimal parameter value for the SVM. Their findings suggest that using a support vector algorithm and a genetic algorithm together improves classification accuracy. Nie *et al*. [24] proposed a new CSO algorithm that adjusts MR parameter dynamically and its performance on the CSO. In addition, a Cauchy operator was used to improve the global search ability.

Lin *et al*. [25] presented the modified cat swarm optimization (MCSO) algorithm, that enhances the computation time within the search area, and then using the MCSO algorithm to select features in a big data classification task. Their findings indicate that the MCSO algorithm outperformed the traditional CSO. Mohapatra *et al*. [26] proposed the modified cat swarm optimization algorithm for improving the exploration performance of cats. A mutation operator was introduced in their research to mutate the best cat's position to achieve good results. The updated CSO algorithm outperformed other algorithms in gene data analysis and classification.

Cho & Hoang [27] proposed the Particle Swarm Optimization (PSO) based Support Vector Machine for feature selection and SVM parameter tuning. Kumar & Sahoo [28] proposed the use of opposition-based learning on an improved CSO algorithm (OBL-ICSO) that introduces the Cauchy operator to enhance the CSO algorithm's exploration phase. Nie *et al*. [24] developed the quantum-behaved cat swarm optimization (QCSO), that merged the CSO algorithm with quantum mechanics. Tharwat *et al*. [29] proposed Bat Algorithm (BA) for the SVM parameters optimization in order to reduce classification error. The result of the BA-SVM algorithm was compared with Grid Search. The proposed model was capable of determining the optimal SVM parameter values while avoiding the local optima problem. Siqueira *et al*. [30] invented the Boolean Binary Cat Swarm Optimization Algorithm (BBCSO) to modify the cats' modes of operation and demonstrate a new method for determining the position and velocity of the cat using Boolean functions.

Tharwat *et al*. [31] proposed the Dragonfly Algorithm (DA) to enhance the SVM parameter. Their proposed model, known as DA-SVM, was tested using six datasets taken from the UCI data repository. Pappula & Ghosh [32] proposed a normal mutation strategy-based on cat swarm optimization (NMCSO) to address the problem of the CSO algorithm's premature convergence and easy trapping in local optima. The performance was evaluated using benchmark unimodal, rotated, unrotated and shifted multimodal problems to demonstrate the performance of the proposed approach. Tharwat *et al*. [33] proposed a Social Ski-Driver (SSD) optimization algorithm inspired by various evolutionary optimization algorithms for improving SVM parameters with the goal of improving classification performance. The results demonstrate that the SSD-SVM algorithm can find near-optimal values for SVM parameters.

Siqueira *et al*. [30] presents a novel version of BBCSO namely "Simplified Binary CSO" (SBCSO) to avoid the problem of lack of velocity control and the tendency to turn once, when updating positions. It also lowers computational costs. Gomathy [14] described a modified cat swarm optimization (ECSO) algorithm for feature extraction. The ECSO was used to extract features from speech signal, and the support vector neutral network (SVNN) was used to classify the extracted features based on emotion. The performance of their work as compared to CSO-SVNN and PSO-SVNN was also excellent.

2. Background Theory

2.1. Support Vector Machine (SVM)

Support Vector Machine (SVM) was developed in 1992, by Boser, Guyon, and Vapnik. SVM is a machine learning algorithm that is significantin the classification of patterns. Support Vector Machine is a classifier that was created for binary classification. SVMs have been shown to outperform traditional machines learning and used for solving classification problems in fields such as pattern recognition and speech recognition [9]. SVM stands out from other classification algorithms due to its strong generalization performance and track record of high accuracy in training datasets [10]. One of the most difficult aspects of classification is the separation of data in different formats, which makes linear separation difficult [34]. Choosing the best kernel function and changing the SVM learning parameters are the most difficult aspects of using the SVM.

2.1.1. Concept of SVM

Support Vector Machine's main goal is to find the best hyper plane for separating two classes. Given a training dataset $(x_1,\ y_1),\ (x_2,\ y_2),\ldots\ldots,(x_N,\ y_N)$, where $x_1 \in \mathbb{R}^n$ is the n dimensional characteristic vector, $y_i \in \{-1,\ +1\}$ is the class label and N represent the total number of records from training datasets. The hyper plane is defined by $(\omega,\ b)$, where ω is the weight vector and b is a bias. The following function can be classified as a new object x; $g(x) = sign\left(w^T . x + b\right) = sign(f(x))$. To solve the problem of linearly inseparable data sets, the kernel function was use to map data from non-linearly separable to linearly separable, such that, x_i will be replaced with K, where K denotes the higher-dimensional mapping, K in this mapping is called kernel function [35]. By constructing a hyper plane, common kernel functions such as the polynomial, sigmoid and Radial-basis kernel function (RBF) are used to divide the feature space. During classification training, the kernel functions could be used to choose training examples along the function's edge [36].

2.1.2. Polynomial Kernel Function

The polynomial kernel function takes more time to practice in SVM and produces better results than the Radial Basis Function in the earlier research. The parameters of this kernel are the gamma slope, where r is the *constant term* and d is the *polynomial degree* [37, 38]. The Polynomial Kernel Function is given as in equation (1):

$$K\left(x_i, x_j\right) = \left(\sigma x_i^T\ X_j + r\right)^d,\ \sigma > 0. \tag{1}$$

2.1.3. Radial-Basis Function (RBF)

The RBF kernel, unlike the linear kernel, maps nonlinearly samples into a faster space, allowing it to deal with situations where the interaction among labeled class and features is dynamic. Furthermore, the linear kernel is a unique example of RBF because the linear kernel with a penalty parameter C achieves similar results as the RBF kernel with some parameters as shown in equation (2)(C, Gamma) [39].

$$K\left(x_i, x_j\right) = exp\left(-\sigma \left|x_i - x_j\right|^2\right),\ \sigma > 0. \tag{2}$$

The adjustable parameter σ is critical to kernel performance and should be properly tuned. When the equation is exceeded, the higher-dimensional projection loses its nonlinearity and behaves nearly linearly. If the feature is undervalued, it lacks regularization, making the judgment boundary critical input susceptible to noise in training performance. Therefore, the width parameter σ determines SVM output [39].

2.1.4. Optimization of the SVM parameters

In SVM, the penalty parameter (C) and the kernel parameter (σ) have a great impact on the classification performance. Parameter C refers to the misclassification or error term. The misclassification or error term determine how much error is acceptable. The changing value of C affects classification accuracy and the support vectors. The Kernel parameter (σ) transform a low-dimensional input space into a higher-dimensional space. RBF was chosen as the kernel function because of its ability to handle non-linear trials with fewer hyper-parameters than the polynomial kernel and it produces good performance in SVM classification.

2.2. Cat Swarm Optimization (CSO)

Cat swarm optimization (CSO) is a recent computer intelligence optimization algorithm that observes the cats' common behavior [13]. Each cat represents 'a solution set' with the following attributes: Position-calculated by the fitness function, Fitness value- defined by the fitness function then the Flag to classify the cat into Seeking or Tracing mode. The CSO algorithm solves the problem by first determining the number of cats to use for the iteration, and then using these cats to solve the problem. The best position from the cats would be the final solution. The CSO retains the best solution in memory until the terminal criteria are met. The common termination criteria are the number of iterations, the amount of improvement, and the running time. Furthermore, two major components are described for CSO, i.e., seeking (searching) and tracing mode. These two modes describe the characteristics of cats when they are at rest, although they are also aware of their surroundings for possible prey. In the algorithm, the two modes are combined using a mixture ratio (MR). This parameter, which is selected from the range [0,1], specifies how many cats are in the seeking and the tracing mode. Figure 1 shows the flowchart of the CSO algorithm.

2.2.1. Seeking Mode

This model represents a cat's resting skill, in which the cat does further thought and makes choices about where to go next [35]. In seeking mode, the cat moves around the search area while remaining alert. In seeking mode, the Seeking Memory Pool (SMP), Seeking the Range of the selected Dimension (SRD), Self-Position Consideration (SPC) and the counts of dimension to change (CDC) are the important operators in this mode. SMP determined the number of cats to be generated. An SMP value of 6 for example, means each cat can store 6 candidate solution sets. SPC has a Boolean value, that signifies if correct, it will retain the existing solution set in one memory location and will not be modified. SRD denoted the difference between the current and previous dimensions of the cat selected for mutationand the CDC defined the amount of dimensions a cat position has experienced for mutation [22]. The steps for seeking mode are given as:

1. Create SMP to duplicates the current cat's location, if SPC is real, one of the copies keeps the current cat's location and becomes a contender instantly. However, before becoming a contender, the other cats must be updated. Otherwise, all SMP copies can perform scanning, causing their locations to shift.
2. By varying CDC percent of measurements, each copy to be modified changes location at random. The CDC percent of measurement is chosen first. The current value of the SRD percent will be reduced over time at random for each chosen dimension. The copies become the contender, after being modified,
3. Using the fitness value, calculate each contender's fitness score.
4. Determine the likelihood that each candidate will be chosen. If the fitness values of all candidates are the same, then the chosen probability (Pi) is one. Otherwise, P_i is calculated via Eq. (3). P_i is the probability that this contender will be chosen; This contender's fitness value is FSi, and the maximum and minimum total fitness values are FSmax and FSmin, respectively. If finding the solution set with the best fitness score is the goal, FS_b = FSmin; otherwise, FS_b = FSmax. When this formula is used to calculate the probability, the stronger contender has a higher chance of being chosen, and vice versa as given below:

$$P_i = \frac{FS_i - FS_b}{FS_{max} - FS_{min}}\quad . \tag{3}$$

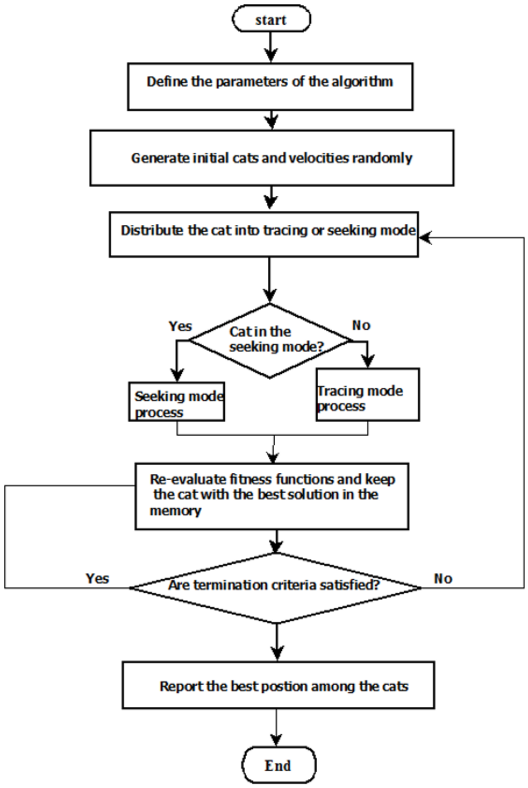

Figure 1. Flowchart of CSO algorithm [13].

1. Pick one contender at random based on the probability (P_i), move the current cat to this location until the contender has been chosen.

2.2.2. *Tracing Mode*

Tracing mode identifies cats tracing a destination [22]. When a cat hunts its prey, its position and velocity are changed. This mode is equivalent to a global search. The tracing mode cat has a velocity and position that are similar to those of the global best cat, (gbest). The following are the process of tracing mode:
The velocity of cat k in dimension d is given as

$$V_{k,d\ new} = V_{k,d} + r_1 \times c_1 \times \left(P_{gbest} - X_{k,d}\right). \tag{4}$$

The new position of each cat is calculated by

$$X_{k,d\ new} = X_{k,d} + V_{k,d}, \tag{5}$$

where, $X_{k,d\ new}$ and $V_{k,d\ new}$ are the position and velocity of the new cat respectively. The global best position of a cat is denoted by P_{gbest} , c_1 is a constant and r_1 is a random number between 0 and 1, $V_{k,d}$ represent the old velocity of the cat k, $X_{k,d}$ represent the old position of the cat k in the d dimension.

3. Methodology

This section describes the proposed ICSO-SVM model as depicted in Figure 2 for determining the optimal SVM parameter values. Any machine learning classification work must begin with preprocessing and feature selection, after obtaining fifteen (15) datasets from the University of California Irvine (UCI) machine learning data repository. Our contribution begins with improving Cat Swarm Optimization Algorithm.

3.1. *Improved CSO Algorithm*

A new parameter was introduced to guide the cats' positions to local and global best positions in the velocity tracing mode of the CSO algorithm. The new velocity update is given as:

$$V_{k,d\ new} = wV_{k,d} + r_1 \times c_1 \left(P_{gbest} - X_{k,d}\right), \tag{6}$$

where c_1 is acceleration coefficient and usually defined constant of 2.05, r is a random number between range of [0,1], P_{gbest} denotes the global best position of a cat, $V_{k,d}$ and $X_{k,d}$ represent the current position and velocity of a cat respectively, $V_{k,d\ new}$ represent the new velocity updated of a cat and w is an inertia weight that direct the cat's position toward local and global positions. The value of inertia weight (w) was chosen randomly and experimental results shows that it is better to choose w in the range of [0.4, 0.9]. The largest value for w in the first iteration (w=0.9) and then reduced to 0.4 in the next iterations. For inertia weight less than 1, the velocities decrease over time, resulting in convergence behavior while inertia weight greater than 1 velocity increase over time, causing cats to divert at the end beyond the boundaries of the search space.

3.2. *Opposition-Based Learning (OBL)*

Many researchers have used Opposition-Based Learning [40] as an optimization technique to improve the performance of their population ideas and local search results. The OBL main idea is to consider an estimate and its associated opposite estimate that is nearest to the global optimum in order to get the best solution. In the search area, the OBL technique searches in both directions. One is the initial solution and the other is the opposite solution. The opposite point in the D-dimensional space can be defined as:

Let $x = x_1, x_2, \ldots\ldots, x_D$ *and* $x_i \in [a_i, b_i]$, $i = 1, 2 \ldots, D$ *be a point D-dimensional space.*

The opposite point $x' = x'_1, x'_2, \ldots, x'_D$ is computed as:

$$x' = a_i + b_i - x_i. \tag{7}$$

3.3. *Data Description and Preprocessing*

The University of California Irvine (UCI) machine learning data repository "https://archive.ics.uci.edu/ml/index.php" dataset was used to evaluate the proposed system. The UCI dataset is a collection of databases, domain structures, and data generators that machine learning researchers use to predict higher evaluations of machine learning algorithms. The UCI has a number of datasets, but only fifteen were utilized to evaluate our system, which includes: iris, ionosphere, dermatology, hepatitis, sponge, soya beans, pen digits, annealing, KDD, breast cancer, Contraceptive method choice (CMC), sonar, balance scale and mushroom as described in Table 1. Although, these datasets consist of a number of features, however, not all the features are relevant. As a result, there was a requirement for feature selection to help reduce the training time and improve classification performance. Principal Component Analysis (PCA) is an approach used to reduce the number of features in large datasets to a smaller number while keeping the significant data in the large set. PCA was used in accordance with the ranker search. The dimensionality was reduced by selecting a sufficient number of eigenvectors to allow a certain percentage of the variance in the original data of which the default is 0.95 (95%).

3.4. Dataset Classification

After improving the CSO, PCA was utilized for the feature reduction in the prediction model, and the dataset classification step was initiated as depicted in the flowchart shown in Figure 2. The classification was performed with the aid of an ICSO optimized SVM. Algorithm 1 shows the pseudo code of the proposed ICSO-SVM algorithm.

Algorithm 1. Proposed ICSO-SVM algorithm

```
Begin
  Input: Parameters of the algorithm and dataset
  Output: optimized (C, σ)
  Initialize the cat population Xi (I = 1, 2, ..., n), V and MR, SMP as parameters
  Compute the opposite position(X) and velocity (V)of each cat using equation (7)
  while (the stop criterion is not satisfied)
    Calculate the fitness function values for each cat and sort them
      Xg= cat with the best solution (the position of the best cat)
        for = I: N do
          If SPC = 1
            Start seeking mode process
          Else
            Start tracing mode process
            Update the new velocity of cat using equation (6)
          End if
        End for
    bestCats = find_best_cat_solution (newPopulation)
    Find parameter (C, σ) value using the lower and upper boundary of the best cats
  End while
  Classify data with optimized SVM parameter (C, σ)
  End
```

3.5. Constructing SVM Classifiers with the Best Parameter Set

RBF was chosen as the kernel function because of its ability to handle non-linear trials with fewer hyper-parameters than the polynomial kernel and produce good performance in SVM classification. It is useful when the attributes and class label don't really have linear relationships. Furthermore, the RBF uses Gamma (G). The value of G and C are selected based on the best solution generated from the improved CSO (ICSO). There are a number of parameters when it comes to training a model but hyper parameters tuning allows us to select the best parameters value for the purpose of improving the learning process and consequently a more efficient model.

3.6. Evaluation metrics and parameter settings

The parameter settings for the proposed algorithm are listed in Table 2. The model's performance was validated using the following metrics:

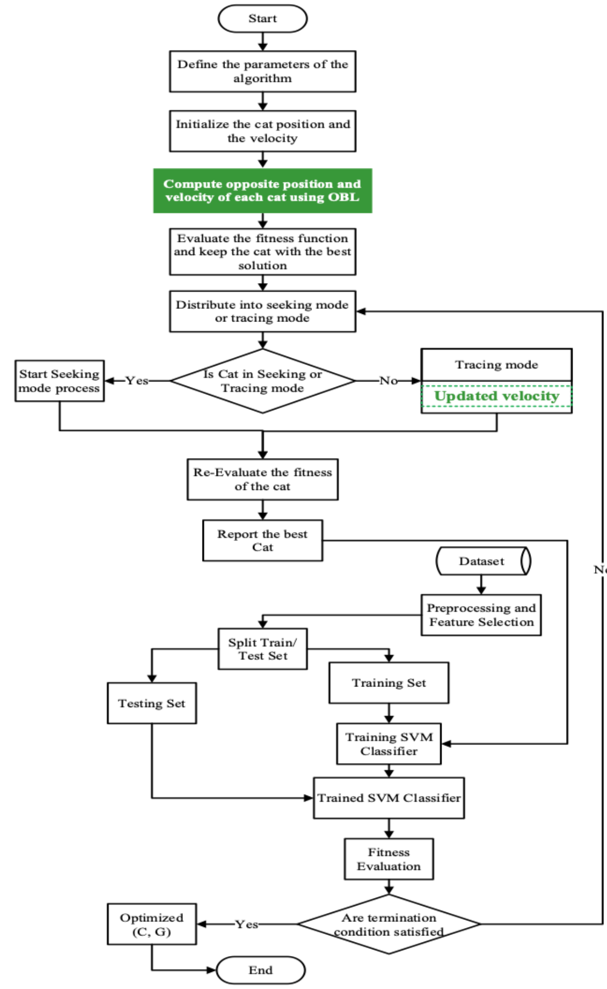

Figure 2. Flowchart of the proposed algorithm (ICSO-SVM).

1. Accuracy: is determined by the number of positive predictions over the total number of predictions and multiplying the result by 100 using equation (8).

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \text{x } 100, \tag{8}$$

where TP, TN, FP and FN are true positives, true negatives, false positives, and false negatives respectively.

2. Precision: the number of positive observations that are correct via:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{9}$$

3. Recall: calculate the number of samples in a class that the model correctly predicts using

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}. \tag{10}$$

4. F-measure: is gotten from the value of precision and recall calculated using

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall+precision}}. \tag{11}$$

Table 1. Description of datasets utilized in the study.

S/N	DATASET	NO. OF FEATURES	NO. OF INSTANCE
1.	Iris	4	150
2.	Ionosphere	34	351
3.	Dermatology	34	366
4.	Hepatitis	19	155
5.	Annealing	38	798
6.	Balance scale	4	625
7.	Contraceptive method choice (CMC)	9	1473
8.	KDD	42	625
9.	Breast cancer	9	286
10.	Soyabeans	35	307
11.	Glass	10	214
12.	Sonar	60	208
13.	Pendigits	17	3498
14.	Sponge	33	10
15.	Mushroom	22	8124

5. Standard deviation: measure the amount of dispersion in a set of values. The low value indicates that the algorithm is reliable with the same value which is better whereas a high standard deviation value implies unstable outcomes. This will be calculated using

$$\text{Standard deviation} = \sqrt{\frac{1}{P-1}\sum_{i=1}^{P}(B_i - Mean)^2}, \quad (12)$$

where B_i stands for each value gotten for standard deviation at run i and P denotes the number of values in the sample.

6. Average accuracy (CA): This function returns the mean of the classification performance obtained after running the algorithm P times the average value of the classifier's output when run P times using

$$CA = \frac{1}{P}\sum_{i=1}^{P} CA^i, \quad (13)$$

where CA^i stands for the accuracy value acquired at runs i.

7. T-Test: This is a statistical test used to calculate the significant difference between the mean of two groups of data and it is calculated using

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{(s^2(\frac{1}{n_1} + \frac{1}{n_2}))}}, \quad (14)$$

where, t represents the student's t-test, $\bar{x}$ denotes mean, s stands for standard deviation, n is the variable set size.

4. Results and Discussions

4.1. Experimental Testbed

The experiment was run on a java NetBeans platform with Weka.jar libraries to enable access to the Weka functionalities on the computer system with Intel Core^{TM} i3-7100U 2.40 GHz CPU and 4.0 GB RAM. The model runs for 20 times independently with 50 iterations. First, we perform the feature reduction on the 15 UCI datasets to determine which attributes have a high impact on our prediction and which does not. Follow by attribute selection using the principal analysis component, to determine attributes with significant impact from others with no impact. The next step is the dataset classification, which

Table 2. Parameter settings.

Parameters	Value or Range
Population size	100
Dimension size	Also known as number of attributes in the dataset
SMP	10
MR	0.5
c_1	2
r_1	[0,1]
W	[0.4, 0.9]

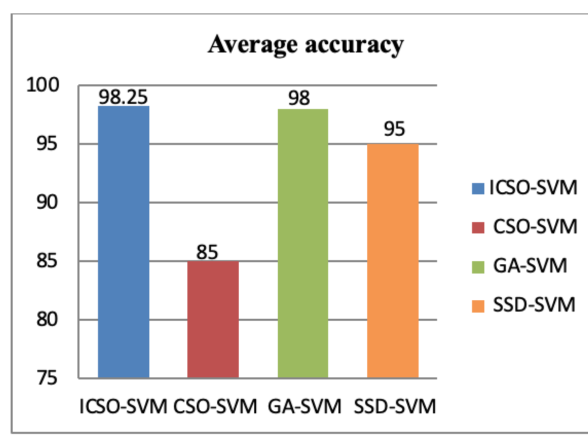

Figure 3. Classification accuracy comparison of the proposed model and some of the existing models.

Table 3. Results of the proposed algorithm with different datasets.

S/N	Datasets	Accuracy%	Precision%	Recall%	F-measure%
1.	Iris	99.0	97.0	96.0	96.6
2.	Ionosphere	98.0	94.7	95.5	95.0
3.	Dermatology	98.4	98.3	98.3	98.3
4.	Hepatitis	91.4	92.8	95.1	93.6
5.	Annealing	97.4	94.2	1.0	97.6
6.	Balance scale	76.7	52.9	91.8	67.1
7.	Contraceptive method choice (CMC)	43.6	38.1	37.5	37.8
8.	KDD	97.0	93.4	95.6	94.4
9.	Breast cancer	96.2	89.0	90.6	89.7
10.	Soyabeans	98.2	98.3	98.3	98.3
11.	Glass	89.2	70.2	77.6	73.7
12.	Sonar	99.2	97.4	98.4	97.8
13.	Pendigits	94.7	95.6	95.5	95.5
14.	Sponge	90.0	93.6	94.8	94.1
15.	Mushroom	99.4	99.4	99.2	99.3

Table 4. Comparison of ICSO-SVM, CSO-SVM, GA-SVM and SSD-SVM with some datasets.

S/N	Dataset	ICSO-SVM%	CSO-SVM%	GA-SVM%	SSD-SVM%
1.	Iris	99	96	100	100
2.	Ionosphere	98	87	98	97
3.	Sonar	99	82	98	88
	Average	98.25	85	98	95

Table 5. Comparison between ICSO-SVM and CSO-SVM in terms of average accuracy and standard deviation

S/N	Datasets	Accuracy		Standard deviation	
		ICSO-SVM	CSO-SVM	ICSO-SVM	CSO-SVM
1	Iris	99.0	96.0	0.022	0.070
2	Ionosphere	98.0	87.1	0.025	0.078
3	Dermatology	98.4	97.4	0.025	0.012
4	Hepatitis	91.4	88.6	0.075	0.063
5	Annealing	97.4	99.0	0.027	0.067
6	Balance scale	76.7	87.5	0.141	0.141
7	CMC	43.6	43.0	0.339	0.339
8	KDD	97.0	96.1	0.027	0.014
9	Breast cancer	96.2	75.4	0.030	0.073
10	Soyabeans	98.0	97.0	0.025	0.013
11	Glass	89.2	59.2	0.079	0.070
12	Sonar	99.2	82.8	0.022	1.020
13	Pendigits	94.7	93.0	0.040	0.141
14	Sponge	90.0	85.6	1.167	1.201
15	Mushroom	99.4	98.0	0.022	0.001
	Average	91.21	85.71	0.13	0.22

comes after attribute reduction. We then used an improved cat swarm optimization algorithm to optimized the support vector machine parameters and a significant value attribute to classify the dataset. The dimension reduction size obtained was based on the selected datasets.

4.2. Results

In this section, the experiments for the performance of the proposed ICSO-SVM algorithm were conducted and the results are presented.

4.3. Discussion of results

The evaluation metrics for all datasets used in the study is presented in Table 3. From the table, our ICSO-SVM obtain a higher accuracy value for 12 out of the 15 datasets used. The ICSO-SVM results were also compared with the existing algorithms the CSO-SVM proposed by Idris *et al.* [35], GA-SVM [41] and SSD-SVM [33] in Table 4. The comparison between

Table 6. Finding T-Test for the ICSO-SVM and CSO-SVM.

S/N	Datasets	ICSO-SVM vs CSO-SVM < 0.05
1	Iris	0.000174
2	Ionosphere	1.95967E-05
3	Dermatology	0.002328
4	Hepatitis	0.052191
5	Annealing	4.45475E-09
6	Balance scale	0.001201
7	CMC	0.234302
8	KDD	0.076091
9	Breast cancer	4.64205E-06
10	Soyabeans	0.000587
11	Glass	7.52885E-08
12	Sonar	4.25683E-07
13	Pendigits	0.227348
14	Sponge	0.004457
15	Mushroom	0.038723
	Average	0.04249

Table 7. Comparison between ICSO-SVM and CSO-SVM in terms of precision.

S/N	Datasets	Precision	
		ICSO-SVM	CSO-SVM
1	Iris	97.0	90.7
2	Ionosphere	94.7	82.4
3	Dermatology	98.3	93.2
4	Hepatitis	92.8	86.6
5	Annealing	94.2	78.7
6	Balance scale	52.9	77.3
7	CMC	38.1	44.0
8	KDD	93.4	94.0
9	Breast cancer	89.0	60.8
10	Soyabeans	98.3	87.6
11	Glass	70.2	53.4
12	Sonar	97.4	74.8
13	Pendigits	95.6	89.3
14	Sponge	93.6	64.5
15	Mushroom	99.4	100
	Average	86.99	78.48

ICSO-SVM and CSO-SVM in terms of average accuracy and standard deviation is presented in Table 5. Table 5 demonstrates the experimental results of the average accuracy and standard deviation of the proposed ICSO-SVM and CSO-SVM algorithms for the 15 datasets used in this study. With regards to the standard deviation of the compared algorithms, ICSO-SVM recorded the lowest values of 0.02 in 8 datasets having an average standard deviation value of 0.13, while the CSO-SVM produced 0.22 average standard deviation as shown graphically in Figure 5. This proves that ICSO-SVM produce a more stable result than CSO-SVM. For the average accuracy shown in Figure 4, the proposed ICSO-SVM has the highest value of 91.21% compared to the CSO-SVM with value of 85.71%.

After finding stability, the T-Test was used to check the significant difference between the proposed ICSO-SVM and the

Table 8. Comparison between ICSO-SVM and CSO-SVM in terms of Recall.

S/N	Datasets	Recall	
		ICSO-SVM	CSO-SVM
1	Iris	97.0	90.7
2	Ionosphere	94.7	82.4
3	Dermatology	98.3	93.2
4	Hepatitis	92.8	86.6
5	Annealing	94.2	78.7
6	Balance scale	52.9	77.3
7	CMC	38.1	44.0
8	KDD	93.4	94.0
9	Breast cancer	89.0	60.8
10	Soyabeans	98.3	87.6
11	Glass	70.2	53.4
12	Sonar	97.4	74.8
13	Pendigits	95.6	89.3
14	Sponge	93.6	64.5
15	Mushroom	99.4	100
	Average	84.34	82.56

Table 9. Comparison between ICSO-SVM and CSO-SVM in terms of F-measure.

S/N	Datasets	F-measure	
		ICSO-SVM	CSO-SVM
1	Iris	97.0	90.7
2	Ionosphere	94.7	82.4
3	Dermatology	98.3	93.2
4	Hepatitis	92.8	86.6
5	Annealing	94.2	78.7
6	Balance scale	52.9	77.3
7	CMC	38.1	44.0
8	KDD	93.4	94.0
9	Breast cancer	89.0	60.8
10	Soyabeans	98.3	87.6
11	Glass	70.2	53.4
12	Sonar	97.4	74.8
13	Pendigits	95.6	89.3
14	Sponge	93.6	64.5
15	Mushroom	99.4	100
	Average	88.58	79.96

CSO-SVM as shown in Table 6. From the table, 3 of the datasets obtained p-values 0.234302, 0.076091, and 0.227348 > 0.05 for the pairs of CMC (ICSO-SVM) against CMC (CSO-SVM), KDD (ICSO-SVM) against KDD (CSO-SVM) and pendigits (ICSO-SVM) against pendigits (CSO-SVM) respectively.Therefore, the average of the p-value is 0.04249 < 0.05 which means there is a clear difference between the proposed model and the existing model. Tables 7, 8 and 9 show the average precision, recall and F-measure for ICSO-SVM and CSO-SVM respectively. In Table 7, it can be seen that iris, ionosphere, dermatology, hepatitis, annealing, soyabeans, sonar, pendigits, and sponges has the highest value for precision metric in the ICSO-SVM algorithm which shows the number of correct positive observations. From Table 8, iris, ionosphere,

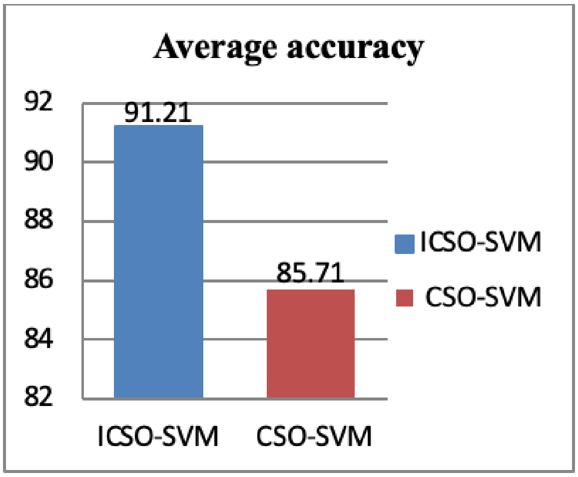

Figure 4. Comparison of average accuracy between ICSO-SVM and CSO-SVM.

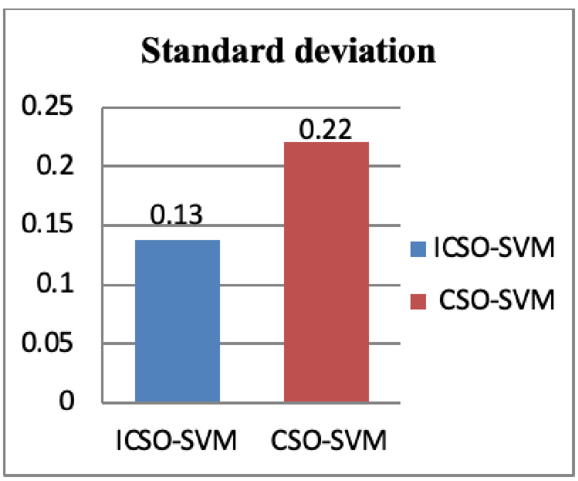

Figure 5. Standard deviation comparison between ICSO-SVM and CSO-SVM.

dermatology, hepatitis, annealing, soyabeans, sonar, pendigits, and sponge datasets have the highest value for recall metric in the ICSO-SVM which produced the number of datasets that the model correctly predicts. Table 9 recorded the F-measure result which was obtained from the precision and recall metrics, for iris, ionosphere, dermatology, hepatitis, annealing, breast cancer, soyabeans, glass, sonar, pendigits, and Sponge with the highest values gotten from the ICSO-SVM algorithm. Figure 6 shows that the proposed algorithm ICSO-SVM has an average precision value of 86.99% while the CSO-SVM has an average value of 78.48%. The average recall value of 84.34% and 82.56 % for ICSO-SVM and CSO-SVM are shown respectively in Figure 7. The F-measure obtained an average value of 88.58% for ICSO-SVM and 79.96% for the CSO-SVM algorithms as shown in Figure 8.

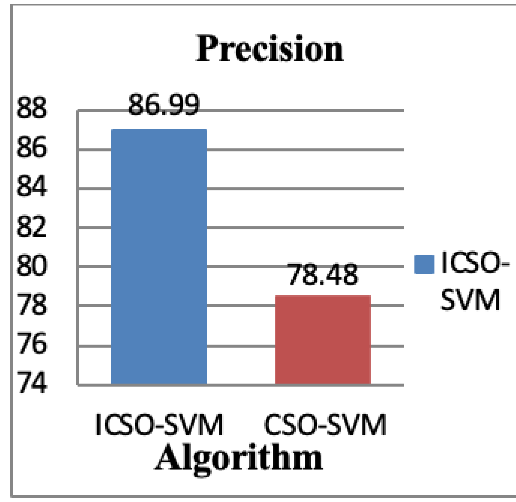

Figure 6. Average precision among the algorithms.

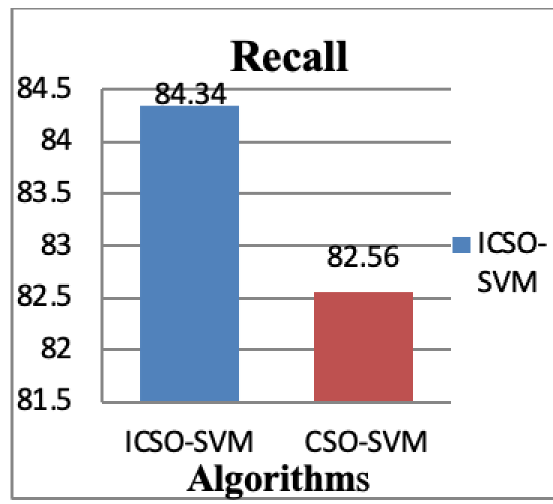

Figure 7. Average recall for the algorithms.

4.4. *Comparison of results*

The work presented in this paper, was compared with the existing work of CSO-SVM proposed by Idris *et al.* [35], GA-SVM presented by Huang & Wang [41], and SSD-SVM introduced by Pappula & Ghosh [33]. The comparison was based on the accuracy of each of the models as recorded in Table 4 and graphically presented in Figure 3.

From Table 4, it can be observed that our model demonstrated a promising performance in terms of accuracy compared to other models. The iris, ionosphere, and sonar datasets result present an average accuracy value of 98.25% for the ICSO-SVM algorithm, CSO-SVM has an average accuracy value of 85%, GA-SVM produces an average accuracy value of 98% and SSD-SVM generate an average accuracy value of 95% respectively.

Based on the results of the experiment, it was discovered

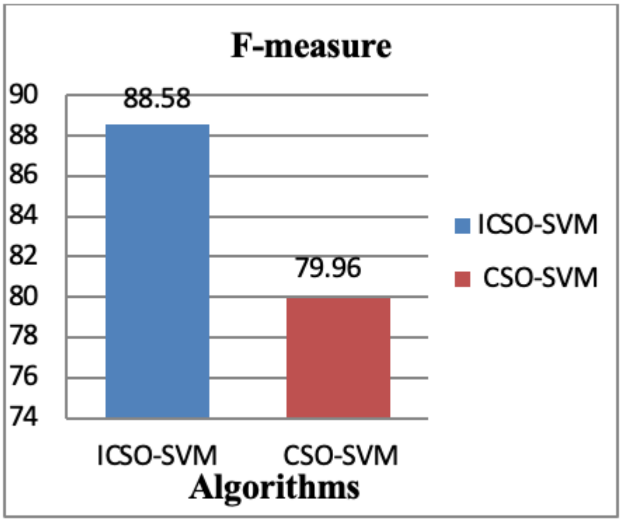

Figure 8. Average F-measure among the algorithms.

that the ICSO-SVM outperforms the existing CSO algorithms in terms of classification, precision, recall, and F-measure. To improve the convergence of the CSO algorithm, a new parameter (inertia weight) was added to the velocity tracing mode to prevent it being trapped in local optima, and the Opposition-based learning was used. We therefore conclude that, the ICSO-SVM algorithm outperform the CSO-SVM algorithm based on the performance measure used in this study.

5. Conclusion

In the study, we proposed an improved Cat Swarm Optimization (CSO) algorithm for optimizing the support vector machine parameters (SVM). The improvement starts with adding a new parameter to the velocity for the tracing mode then using Opposition-Based Learning (OBL) to improve the convergence of the CSO algorithm in order to prevent it from being stuck in local optima. PCA was used as a feature reduction technique. The ICSO-SVM was compared with the state-of-the-art CSO-SVM algorithm using 15 datasets taken from the UCI data repository. The classification was done on the optimized ICSO-SVM using the classification accuracy, precision, recall, and F-measure as the performance metrics. Similarly, the standard deviation was used to test the convergence stability and robustness of the algorithms, while the T-Test was used to determine the significant difference between the proposed ICSO-SVM and the CSO-SVM algorithm. The algorithm was implemented using Java programming language.

The experimental results demonstrate that ICSO-SVM and CSO-SVM obtains an average accuracy value of 91.2% and 85.71% over15 datasets respectively. Furthermore, the proposed ICSO-SVM method outperforms the state-of-the-art CSO-SVM with a lower standard deviation on iris, ionosphere, dermatology, hepatitis, annealing, KDD, breast cancer, soyabeans, glass, sonar, pen digits and mushroom datasets. For future studies, the authors intend to focus on incorporating the proposed algorithm in real-world problem, such as face detection, image classification, engineering applications. In addition, the ICSO-SVM can be applicable to variety of other datasets. Furthermore, other optimization algorithms can also be considered for hyper-parameter tuning of SVM.

Acknowledgment

The authors thank the reviewers and editors for their valuable comments and suggestions.

References

[1] J. T. Senders, M. M. Zaki, A. V. Karhade, B. Chang, W. B. Gormley, M. L. Broekman, T. R. Smith & O. Arnaout "An introduction and overview of machine learning in neurosurgical care", Acta Neurochirurgica **160** (2018) 29. https://link.springer.com/article/10.1007/s00701-017-3385-8

[2] C. L. Udeze, I. E. Eteng & A. E. Ibor, "Application of Machine Learning and Resampling Techniques to Credit Card Fraud Detection", Journal of the Nigerian Society of Physical Sciences **4** (2022) 769. https://doi.org/10.46481/jnsps.2022.769.

[3] D. Opeoluwa, E. Gbenga, N. Juliana, A. U. Terang & S. A. Akinwumi, "COVID-19 Risk Factors, Economic Factors, and Epidemiological Factors nexus on Economic Impact : Machine Learning and Structural Equation Modelling Approaches", Journal of the Nigerian Society of Physical Sciences **3** (2021) 395. https://doi.org/10.46481/jnsps.2021.173.

[4] U. B. Mir, S. Sharma, A. K. Kar & M. P. Gupta, "Critical success factors for integrating artificial intelligence and robotics", Digital Policy Regulation and Governance **22** (2020) 307. https://doi.org/10.1108/DPRG-03-2020-0032.

[5] M. A. Goralski & T. K. Tan, "Artificial intelligence and sustainable development", International Journal of Management Education **18** (2020) 1. https://doi.org/10.1016/j.ijme.2019.100330.

[6] N. Malik, S. N. Tripathi, A. K. Kar & S. Gupta, "Impact of artificial intelligence on employees working in industry 4.0 led organizations", International Journal of Manpower **43** (2021) 334. https://doi.org/10.1108/ijm-03-2021-0173.

[7] S. M. C. Loureiro, J. Guerreiro & I. Tussyadiah, "Artificial intelligence in business: State of the art and future research agenda", Journal Business Research **129** (2021) 911. https://doi.org/10.1016/j.jbusres.2020.11.001.

[8] B. E. Boser, V. N. Vapnik & I. M. Guyon, *A Training Algorithm Margin for Optimal Classifiers*, Proceedings of the fifth Annual Workshop on Computational Learning Theory, 1992, pp 144–152. https://doi.org/10.1145/130385.130401.

[9] D. S. Kim & J. S. Park, *Network-based intrusion detection with support vector machines* in *Information Networking. ICOIN 2003. Lecture Notes in Computer Science*, K. H. Kahng (Ed.), Vol 2662, Springer, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-540-45235-5_73.

[10] A. C. Enache & V. V. Patriciu, "Intrusions detection based on support vector machine optimized with swarm intelligenc", S2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2014, pp. 153-158. https://ieeexplore.ieee.org/document/6840052.

[11] Y. Li, L. Guo, Z. H. Tian & T. B. Lu, "A lightweight web server anomaly detection method based on transductive scheme and genetic algorithms", Computer Communications **31** (2008) 4018. https://doi.org/10.1016/j.comcom.2008.08.009.

[12] K. C. Lin, Y. H. Huang, J. C. Hung & Y. T. Lin, "Feature Selection and Parameter Optimization of Support Vector Machines Based on Modified Cat Swarm Optimization", International Journal of Distributed Sensor Networks **11** (2015) 365869. https://doi.org/10.1155/2015/365869.

[13] S. C. Chu & P. W. Tsai, "Computational intelligence based on the behavior of cats", International Journal of Innovative, Computing Information and Control **3** (2007) 163. https://api.semnticscholar.org/CorpusID:58516637.

[14] M. Gomathy, "Optimal feature selection for speech emotion recognition using enhanced cat swarm optimization algorithm", International Journal of Speech Technology **24** (2021) 155. https://doi.org/10.1007/s10772-020-09776-x.

[15] M. Orouskhani, M. Mansouri & M. Teshnehlab, "Average-inertia weighted cat swarm optimization", Proceedings of Second International Conference on Advances in Swarm Intelligence (2011) 321. https://doi.org/10.1007/978-3-642-21515-5_38.

[16] S. Mahdavi, S. Rahnamayan & K. Deb, "Opposition based learning: A literature review", Swarm Evolutionary Computation **39** (2018) 1. https://doi.org/10.1016/j.swevo.2017.09.010.

[17] P. W. Tsai, J. S. Pan, S. M. Chen, B. Y. Liao & S. P. Hao, "Parallel cat swarm optimization", International Conference on Machine Learning and Cybernetics. ICMLC **6** (2008) 3328. https://doi.org/10.1109/ICMLC.2008.4620980.

[18] K. C. Lin & H. Y. Chien, "CSO-based feature selection and parameter optimization for support vector machine", 2009 Joint Conferences on Pervasive Computing JCPC **2009** (2009) 783. https://doi.org/10.1109/JCPC.2009.5420080.

[19] B. Santosa & M. K. Ningrum, "Cat swarm optimization for clustering", 2009 International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 2009, pp. 54-59. https://doi.org/10.1109/SoCPaR.2009.23.

[20] P. W. Tsai, J. S. Pan, S. M. Chen & B. Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method", Expert Systems Applications **39** (2012) 6309. https://doi.org/10.1016/j.eswa.2011.11.117.

[21] M. Orouskhani, Y. Orouskhani, M. Mansouri & M. Teshnehlab, "A Novel Cat Swarm Optimization Algorithm for Unconstrained Optimization Problems", International Journal of Information Technology and Computer Science **5** (2013) 32. https://doi.org/10.5815/ijitcs.2013.11.04.

[22] I. Hadi & M. Sabah, "Improvement Cat Swarm Optimization for Efficient Motion Estimation", International Journal of Hybrid Information Technology **8** (2015) 279. https://dx.doi.org/10.14257/ijhit.2015.8.1.25.

[23] J. Manurung, H. Mawengkang & E. Zamzami, "Optimizing Support Vector Machine Parameters with Genetic Algorithm for Credit Risk Assessment", Journal of Physics Conference Series **930** (2017) 8. https://doi.org/10.1088/1742-6596/930/1/012026.

[24] X. Nie, W. Wang & H. Nie, "Chaos quantum-behaved cat swarm optimization algorithm and its application in the PV MPPT", Computational Intelligence and Neuroscience **2017** (2017) 1583847. https://doi.org/10.1155/2017/1583847.

[25] K. C. Lin, K. Y. Zhang, Y. H. Huang, J. C. Hung & N. Yen, "Feature selection based on an improved cat swarm optimization algorithm for big data classification", Journal of Supercomputing **72** (2016) 3210. https://doi.org/10.1007/s11227-016-1631-0.

[26] P. Mohapatra, S. Chakravarty & P. K. Dash, "Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system", Swarm Evolutionary Computation **28** (2016) 144. https://doi.org/10.1016/j.swevo.2016.02.002.

[27] M. Y. Cho & T. T. Hoang, "Feature Selection and Parameters Optimization of SVM Using Particle Swarm Optimization for Fault Classification in Power Distribution Systems", Computational Intelligence and Neuroscience **2017** (2017) 4135465. https://doi.org/10.1155/2017/4135465.

[28] Y. Kumar & G. Sahoo, "An improved cat swarm optimization algorithm based on opposition-based learning and Cauchy operator for clustering", Journal of Information Processing Systems **13** (2017) 1000. https://doi.org/10.3745/JIPS.02.0022.

[29] A. Tharwat, A. Ella & B. E. Elnaghi, "A BA-based algorithm for parameter optimization of Support Vector Machine", Pattern Recognition Letters **93** (2017) 13. https://doi.org/10.1016/j.patrec.2016.10.007.

[30] H. Siqueira, E. Figueiredo, M. Macedo, C. J. Santana, C. J. A. Bastos-Filho & A. A. Gokhale, "Boolean Binary Cat Swarm Optimization Algorithm", 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Gudalajara, Mexico, 2018, pp. 1-6. https://doi.org/10.1109/LA-CCI.2018.8625226.

[31] A. Tharwat, T. Gabel & A. E. Hassanien, "Parameter optimization of support vector machine using dragonfly algorithm", Advances in Intelligent System and Computing **639** (2018) 309. https://doi.org/10.1007/978-3-319-64861-3_29.

[32] L. Pappula & D. Ghosh, "Cat swarm optimization with normal mutation for fast convergence of multimodal functions", Journal of Applied Soft Computing **66** (2018) 473. https://doi.org/10.1016/j.asoc.2018.02.012.

[33] A. Tharwat, T. Gabel & T. Gabel, "Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm", Neural Computing and Applications **32** (2020) 6925. https://doi.org/10.1007/s00521-019-04159-z.

[34] M. A. Nanda, K. B. Seminar, D. Nandika & A. Maddu, "A comparison study of kernel functions in the support vector machine and its application for termite detection", Information **9** (2018). https://doi.org/10.3390/info9010005.

[35] S. Idris, O. Oyefolahan Ishaq & N. N. Juliana, "Intrusion Detection System Based on Support Vector Machine Optimised with Cat Swarm Optimization Algorithm", 2019 2^{nd} International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), Zaria, Nigeria, 2019, pp. 1-8. https://doi.org/10.1109/NigeriaComputConf45974.2019.8949676.

[36] S. A. Mulay, P. R. Devale & G. V. Garje, "Intrusion Detection System Using Support Vector Machine and Decision Tree", International Journal of Computer Applications **3** (2010) 40. https://doi.org/10.5120/758-993.

[37] B. Y. Sun, D. S. Huang, L. Guo & Z. Q. Zhao, "Support vector machine committee for classification", International Symposium on Neural Network **3173** (2004) 648. https://doi.org/10.1007/978-3-540-28647-9_106.

[38] J. H. Min & Y. C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters", Expert System Applications **28** (2005) 603. https://doi.org/10.1016/j.eswa.2004.12.008.

[39] R. Amami, D. Ben Ayed & N. Ellouze, "Practical Selection of SVM Supervised Parameters with Different Feature Representations for Vowel Recognition", Computer Science Bibliography **7** (2015) 418. https://doi.org/10.48550/arXiv.1507.06020.

[40] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence", International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 2005, pp. 695-701. https://doi.org/10.1109/CIMCA.2005.1631345.

[41] C. Huang & C. Wang, "A GA-based feature selection and parameters optimization for support vector machines", Expert Systems with Applications **31** (2006) 231. https://doi.org/10.1016/j.eswa.2005.09.024.