

## Research Article

# Optimization of SVM Parameters with Hybrid CS-PSO Algorithms for Parkinson's Disease in LabVIEW Environment

Duygu Kaya 

*Department of Electrical-Electronics Engineering, Faculty of Engineering, Firat University, Elazig, Turkey*

Correspondence should be addressed to Duygu Kaya; [dgur@firat.edu.tr](mailto:dgur@firat.edu.tr)

Received 10 December 2018; Accepted 7 April 2019; Published 2 May 2019

Academic Editor: Jan Aasly

Copyright © 2019 Duygu Kaya. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Optimization is the process of achieving the best solution for a problem. LabVIEW based on an SVM model is proposed in this paper to get the best SVM parameters using the hybrid CS and PSO method. PCA is used as a preprocessor of SVM for reducing the dimension of data and extracting features of training samples. Also, SVM parameters are optimized for Parkinson's disease data by combining CS and PSO. The designed system is used to determine the best SVM parameters, and it is compared to PSO and CS optimization methods and found that the used CS-PSO hybrid optimization method is better. The hybrid model shows that the accuracy of the performance achieved is 97.4359%. Also, the data classification results obtained by using SVM parameters determined by optimization are measured by precision, recall, *F1* score, false positive rate (FPR), false discovery rate (FDR), false negative rate (FNR), negative predictive value (NPV), and Matthews' correlation coefficient (MCC) parameters.

## 1. Introduction

Parkinson's disease (PD) is a neurological disorder that affects the standard of life of the patients and their relatives. PD is more widespread in countries where the pretty old population is high. According to statistics, in the US, about one million people will be affected with Parkinson's disease (PD) by 2020 and more than 10 million people worldwide will be living with PD. The possibility of men with Parkinson's disease is 1.5 times higher than that of women [1]. The main symptoms of this disease are tremor-trembling, stiffness in the body, slowness of movements, and impaired balance. As the disease progresses, patients may experience difficulty in vital tasks such as walking, speech, swallowing, and chewing, emotional changes, and sleep disorders. PD symptoms occur slowly, and in some people, the disease progresses faster than others. The intensity of the symptoms varies from person to person and does not create the same effect [2]. Although there is no treatment method to eliminate this disease completely, drug treatment is applied to reduce the symptoms seen in the early stages of the disease. For diagnosis of the disease, walking and vocal analysis methods are used. Machine learning methods have

been used to diagnose the disease [3]. Among supervised learning algorithms, support vector machine (SVM) is based on statistical learning theory and is the most effective algorithm for predicting performance for nonlinear problems. It has high generalization capabilities. SVM is a powerful technique for overcoming classification problems, image processing, and disease diagnosis with excellent performance.

Before the SVM is applied, a "feature transformation" operation must be performed, which is the process of transforming the data into a new set of data at a dimension that can express less features. With this, dimension of data reduces and excessive numbers of unimportant features are removed. In this study, PCA is used for dimension reduction.

In cases where the difference between the reduced dimension data is too high, the normalization process is performed to handle the data in a single order. In addition, normalization is also used which makes use of mathematical functions to move data in different scaling systems to a common system and make them comparable. In this study, the *Z*-score normalization method is used. *Z*-score is calculated by subtracting the average value from each variable

value and dividing the obtained difference by the standard deviation.

Particle swarm optimization (PSO) is an optimization technique designed to deliver the best solution to the system. Also, the cuckoo search algorithm (CS) is another optimization method. It has less parameters, easy to implement, and efficient. In this paper, both of them are used for optimization of SVM parameters.

*1.1. Motivation.* Accurate and reliable diagnosis is very important for human health. In this study, different optimization algorithms have been used to obtain the best SVM parameters for predicting Parkinson's disease. The proposed hybrid CS-PSO-SVM model provided an accuracy of 97.4359% and is superior to the PSO-SVM and CS-SVM models.

### 1.2. Contribution

- (i) A different work environment for researchers has been proposed using LabVIEW, a visual programming language instead of only text-based programming languages
- (ii) Hybrid optimization methods are used for obtaining the best SVM parameters

*1.3. Sections.* The paper is organized as follows: In Section 2, used data are described. In Section 3, PCA, SVM, PSO, and CS are explained. In Section 4, the LabVIEW programming language and information about its features are given. In Section 5, the proposed model CS-PSO-SVM is described. In Section 6, experimental results are given. The last section of the study includes discussion and conclusions.

## 2. Dataset and Features

Appropriate C value parameter is chosen [4]. The attributes of these data which are related to biomedical voice measurements are given in Table 1.

## 3. Description of Used Techniques

Dimension reduction and normalization procedures were performed to extract properties from the used data to ensure that they are in a single order. Then, optimization methods are applied to SVM. Figure 1 shows the diagram of the used techniques.

*3.1. Principal Component Analysis (PCA).* PCA is a technique that has a wide range of uses for reducing the insignificant features of the data. The idea underlying the PCA is to represent a data plane by separating it into orthogonal axes to reflect the data in small linear combinations. In other words, PCA reduces the data dimension to extract features. Figure 2 shows the used dimension reduction program on LabVIEW.

TABLE 1: Attributes of used data.

Number	Attributes	Information
1	MDVP:Fo (Hz)	Average vocal fundamental frequency
2	MDVP:Fhi (Hz)	Maximum vocal fundamental frequency
3	MDVP:Flo (Hz)	Minimum vocal fundamental frequency
4	MDVP:Jitter (%)	Several measures of variation in the fundamental frequency
5	MDVP:Jitter (Abs)	
6	MDVP:RAP	
7	MDVP:PPQ	
8	Jitter:DDP	
9	MDVP:Shimmer	Several measures of variation in amplitude
10	MDVP:Shimmer (dB)	
11	Shimmer:APQ3	
12	Shimmer:APQ5	
13	MDVP:APQ	Two measures of ratio of noise-to-tonal components in the voice status
14	Shimmer:DDA	
15	NHR	Two nonlinear dynamic complexity measures
16	HNR	
17	RPDE	Signal fractal scaling exponent
18	D2	
19	DFA	Three nonlinear measures of the fundamental frequency variation
20	Spread1	
21	Spread2	
22	PPE	

*3.2. Normalization with Z-Score.* Statistical normalization is performed to treat the data in a single order when there is a lot of difference between the data. Also, another objective is to use mathematical functions to translate data from different systems into a common system and make them comparable. In the Z-score normalization, the numbers are normalized to the distance of the mean value. In addition, dividing by standard deviation, the mobility between numbers (rate of change) normalizes the distance to the average. In other words, the mean and standard deviation values are taken into account. Standard deviation, mean, and Z-score are calculated by using the following equations, respectively:

$$s = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n}},$$

$$\bar{x} = \frac{\sum_i x_i}{n},$$

$$z = \frac{x_i - \bar{x}}{s}. \quad (1)$$

*3.3. Support Vector Machine (SVM).* SVM is an important tool for machine learning (ML) derived from statistical

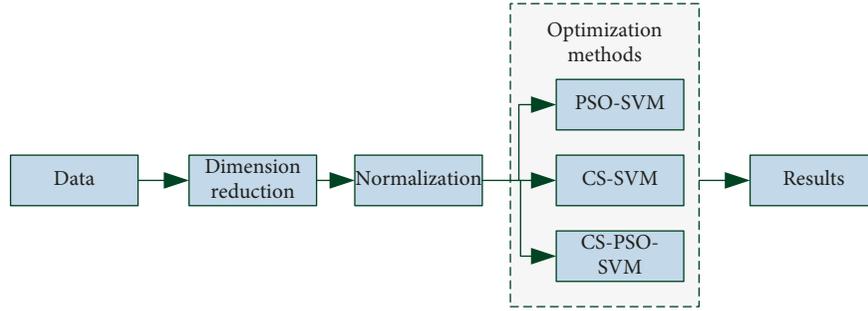


FIGURE 1: The diagram of the used techniques.

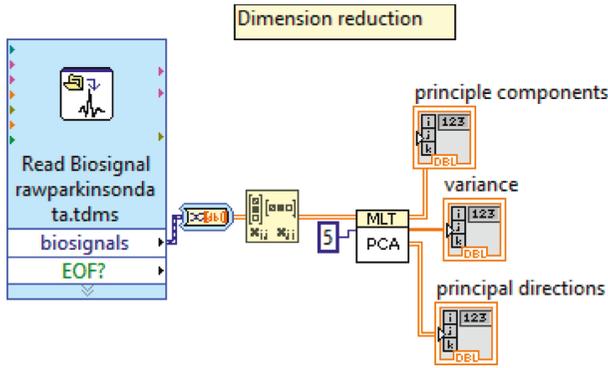


FIGURE 2: Dimension reduction program for PCA on LabVIEW.

learning theory [5–8]. SVM is one of the most used algorithms for classification and regression tasks because it has high performance and generalization capability [9–14]. The main idea behind SVM is to obtain a linear discriminatory function that separates classes from each other by a hyperplane. SVM finds the hyperplane which maximizes the margin between samples and the class boundary for linearly separable data, but for real-world applications, as shown in Figure 3, the nonlinear transformations with the kernel functions are essential to move the datasets to spaces that can linearly separate and classify them [15, 16]. Transfer process of input data to the property plane is shown in Figure 4.

Theoretically, any linearly separable SVM can be correctly classified. For a linearly separable dataset, there are  $n$  samples of training data with two classes expressed as

$$(x_1, y_1), \dots, (x_n, y_n), \quad x \in R^d, y \in \{1, -1\}. \quad (2)$$

These data can be separated from each other by the separator function given by

$$D(x) = (w \times x) + w_0. \quad (3)$$

The following equations are used for correct classification:

$$(w \times x_i) + w_0 \geq +1, \quad y_i = +1, \quad (4)$$

$$(w \times x_i) + w_0 \leq -1, \quad y_i = -1. \quad (5)$$

The appropriate values of  $w$  and  $w_0$  are calculated to find the optimal separator hyperplane. For real-world data, data samples cannot be distinguished linearly. For this reason,

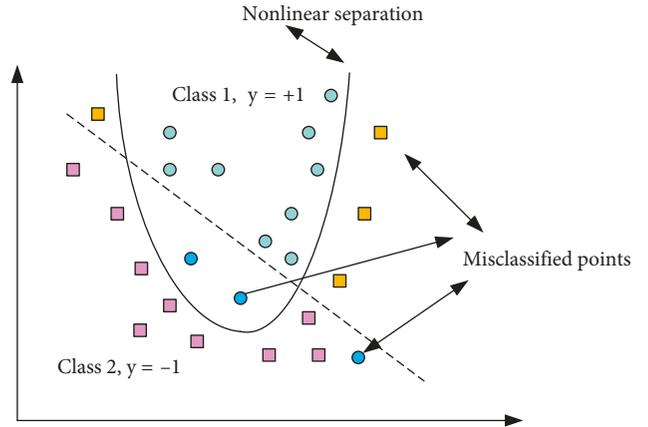


FIGURE 3: Nonlinear SVM.

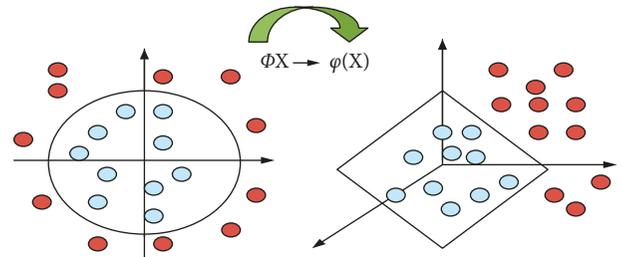


FIGURE 4: Transfer of input data to the property plane.

defining a feature mapping function is needed. This function is called the kernel function. The basic idea of kernel methods is to use nonlinear mapping on the input plane first and then apply a linear algorithm to the new input. The training phase is the  $K$  kernel function of the data in this plane:

$$K = \varphi(x_i) \cdot \varphi(x_j), \quad (6)$$

which will depend on the inner products. Decision function is

$$f(x) = \sum_{i=1}^{ls} \alpha_i y_i \varphi(x_i) \varphi(x) + b = \sum_{i=1}^{ls} \alpha_i y_i K(x_i, x) + b. \quad (7)$$

In this function,  $\alpha_i$  values are positive Lagrange multipliers.  $ls$  is the number of support vectors, and  $x_i$  is the support vector [17].

Polynomial kernel, sigmoid kernel, and Gaussian kernel functions are used commonly to find the optimal hyperplane to distinguish linearly nonseparable data.

(i) Gaussian kernel:

$$\begin{aligned} K(x_i, x_j) &= \exp\left(-\frac{\|x - y\|^2}{\varphi^2}\right), \\ K(x_i, x_j) &= e^{-\|x^2\|} e^{-\|y^2\|} e^{2xy^T}, \\ K(x_i, x_j) &= e^{-\|x^2\|} e^{-\|y^2\|} \sum_{n=0}^{\infty} \frac{2^n}{n!} (xy^T)^n. \end{aligned} \quad (8)$$

(ii) Polynomial kernel:

$$K(x_i, x_j) = (x_i \cdot x_j + a)^b. \quad (9)$$

$xy$  is the dot product of  $x$  and  $y$ .  $n$ th order of this product is a polynomial kernel. The infinite totaled expression containing all polynomial kernel from the 0th to the infinite order is Gaussian kernel. Therefore, Gaussian is a special kernel and shows good performance.

In order to classify with SVM, the first thing to do is to select a kernel function and related parameters that allow linear separation of the data. For classification of data, the following equation is obtained:

$$K_{ij} = y_i y_j \varphi(x_i) \cdot \varphi(x_j). \quad (10)$$

Appropriate  $C$  value parameter is chosen and  $\alpha$  is found:

$$\begin{aligned} \max \quad & \left[ \sum_{i=1}^L \alpha_i - \frac{1}{2} \alpha^T K_\alpha \right] \\ & 0 \leq \alpha_i \leq C \\ \text{s.t} \quad & \sum_{i=1}^L \alpha_i y_i = 0. \\ & \omega = \sum_{i=1}^L \alpha_i y_i \varphi(x_i). \end{aligned} \quad (11)$$

Provided the  $0 \leq \alpha_i \leq C$  condition, support vectors  $V$  are determined:

$$b = \frac{1}{N_v} \sum_{v \in V} \left( y_v - \sum_{i \in V} \alpha_i y_i \varphi(x_i) \varphi(x_v) \right), \quad (12)$$

$$y' = \text{sgn}(\omega \varphi(x') + b).$$

**3.4. Particle Swarm Optimization (PSO).** Optimization is the process of achieving the best solution for a problem. Since the methods used in optimization problems defined by mathematical functions are not flexible and the desired result cannot be achieved, new methods have been developed with reference to natural phenomena and PSO is the most common of these algorithms. Inspired by fish and insects moving in flocks, Kennedy and Dr. Eberhart developed PSO in 1995 [18, 19]. It has been shown that the random movements of animals that move in flocks to meet their vital needs are influenced by the other members of the flock and are easier to reach for the purpose of the flock. This process is done for determining the location of the particle with the best position in the stream and the other particles to move in that direction. The particles aim to improve their next position based on their past experience and the individual with the best position in the pack. The PSO algorithm is an evolutionary algorithm like genetic algorithm (GA). However, PSO is faster than GA because there are no operators such as crossover and mutation.

The basic PSO algorithm: every individual in the swarm can be a solution, and every individual is represented by the dimension vector:

$$x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}) \in S. \quad (13)$$

The speed of each individual in the herd is randomly generated. Each individual has the same speed as in equation (14):

$$v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}) \in S. \quad (14)$$

The best local and global positions are determined. Here, the position of each individual is defined as follows:

$$p_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD}) \in S. \quad (15)$$

Each individual in the PSO adjusts its position around the individual to pbest, global, and gbest. The speed and position information of the individuals are given in the following equations:

$$v_i^{(t+1)} = v_i^{(t)} + c_1 r_{i1} * (pbest_i^{(t)} - x_i^{(t)}) + c_2 r_{i2} * (gbest - x_i^{(t)}), \quad (16)$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{t+1}, \quad i = 1, \dots, P. \quad (17)$$

Here,  $c_1$  and  $c_2$  are two social and cognitive acceleration parameters.  $r_1$  and  $r_2$  are random numbers between [0, 1]. Figure 5 shows the pseudocode of the PSO.

**3.5. Cuckoo Search Algorithm (CS).** CS is a next-generation optimization method based on the hatching parasitic nature of cuckoo birds [21–23]. The most effective features that separate cuckoos from other birds and subject them to the optimization algorithm are aggressive breeding strategies. If the host bird finds that the eggs are not its own, it shows the behavior of throwing the egg from the nest or abandoning the nest. If the eggs are not recognized, the host bird sits on these eggs and the condition of brood parasitism arises. In

---

```

Step 1: Create starting swarm, speeds and positions.
Step 2: Calculate the fitness value of all the particles in the swarm.
Step 3: Compare all the particles to the best of the previous generation. Change
        the place if it's better.
Step 4: Compare best local values amongst themselves and assign the best one
        as the best of the global.
Step 5: Refresh speed and position values.
Step 6: Stopping criteria
        YES Results
        If NO go to step 2

```

---

FIGURE 5: The pseudocode of the PSO [20].

the CS algorithm, each egg represents a new solution, while each solution is a cuckoo egg. The purpose of this is to use better new solutions to replace the existing poor solution in the nest. As with any optimization problem, CS also has some restrictions that are each cuckoo can leave only one egg in a randomly selected nest and if the nest has a high-quality egg, then the egg is transferred to future generations.

Dropped eggs can be familiarized by the host with a probability of  $P_a$ . The probability of  $P_a$  changes in the range  $[0, 1]$ , and the number of host nests is fixed. Heuristic optimization algorithms perform global and local searches while approaching the best solution. The CS algorithm is an algorithm that is used in combination with global random walk and local random walk approaches.

The global random walk performed with a Levy flight is performed with equation (18). Here, the value produced by the Levy flight is weighted by the variable  $\alpha$  and is summed to its old position. Thus, new locations are found.  $s$  and  $\lambda$  are control parameters:

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda). \quad (18)$$

Local random walk is performed with equation (19). Here,  $x_j^t$  and  $x_k^t$  are the random permutations,  $H(u)$  the Heaviside function,  $s$  the step length, and  $\alpha$  a random real number from the Gaussian distribution:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(P_a - \epsilon) \otimes (x_j^t - x_k^t). \quad (19)$$

#### 4. LabVIEW

The emerging technology needed the development of object-oriented programming languages instead of text-based programming languages. Thus, visual programming was possible without writing code. With National Instruments' development of the LabVIEW program, it was possible to program the model graphically with ready-made functions, and there was no need to write code. With LabVIEW (Laboratory Virtual Instrument Engineering Workbench), it was possible to make programs more quickly and to avoid time loss. LabVIEW generally uses a data flow model instead of text codes. Also, LabVIEW has an ability of multiple parallel processes [24].

LabVIEW consists of two components: the first one is the front panel that is the user interface and the second one is the block diagram in which graphical codes are shown. Both of them are shown in Figures 6 and 7, respectively. Inputs connected to the virtual instrument on the front panel are called controls, while the outputs are called indicator. The control palette is used in the front panel, and the function palette is used in the block diagram. The control palette allows access to various controls and indicators and is displayed only on the front panel. In the same way, the function palette also allows access to blocks with various functions to design a system and is displayed only in the block diagram. With LabVIEW, subVI can be created just like a VI. Also, a subVI can be created from code already within another VI. The created subVI, with the customized icon and the configured terminals, is used within other VIs repeatedly. The subVI prevents the program from appearing too crowded. As a matter of fact, subVI was used in this study.

#### 5. The Proposed Methods

In this study, PSO-SVM, CS-SVM, and CS-PSO-SVM methods are compared with each other. The created hybrid program in the LabVIEW environment is shown in Figure 8. Optimization algorithms are used to find the best SVM parameters. To get these parameters, subVI is created for each optimization method. subVI is frequently used in LabVIEW such as in other programming languages. With subVI, the created program is simplified and is prevented from appearing crowded.

The classification performance results obtained by using SVM parameters determined by optimization are measured by accuracy, precision, recall,  $F1$  score, false positive rate (FPR), false discovery rate (FDR), false negative rate (FNR), negative predictive value (NPV), and Matthews' correlation coefficient (MCC) parameters. These parameters are obtained by the confusion matrix. A confusion matrix is shown in Table 2.

5.1. *Accuracy.* Accuracy is the correct classification ratio:

$$\text{accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \quad (20)$$

5.2. *Precision.* Precision is a situation that shows success in a positively predicted situation:

$$\text{precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}. \quad (21)$$

5.3. *Recall.* Recall shows how well the positive cases are estimated:

$$\text{recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (22)$$

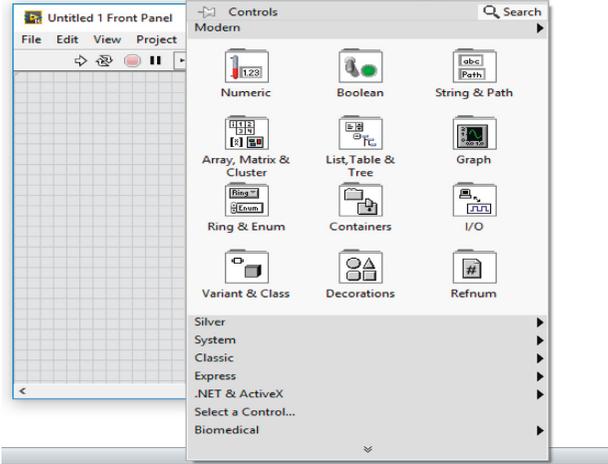


FIGURE 6: LabVIEW front panel.

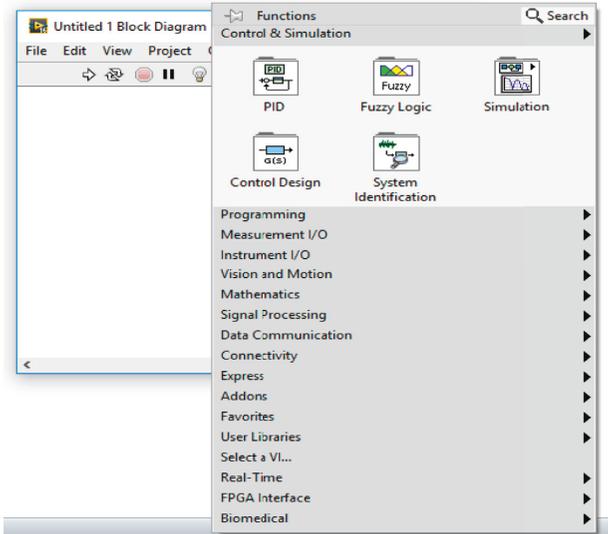


FIGURE 7: LabVIEW block diagram.

5.4. *F1 Score*. *F1* score is the harmonic average of precision and recall:

$$F1 \text{ score} = \frac{2(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}. \quad (23)$$

5.5. *False Positive Rate (FPR)*. *FPR*, sometimes called the fall-out, is the ratio of misclassified events (*FP*) to all actual negative events:

$$FPR = \frac{FP}{(FP + TN)}. \quad (24)$$

5.6. *False Discovery Rate (FDR)*. *FDR* is the expected percent of false predictions in a set of predictions:

$$FDR = \frac{FP}{(FP + TP)}. \quad (25)$$

5.7. *False Negative Rate (FNR)*. *FNR*, sometimes called the miss rate, is the proportion of individuals with a known positive condition for which the test result is negative:

$$FNR = \frac{FN}{(FN + TP)}. \quad (26)$$

5.8. *Negative Predictive Value (NPV)*. *NPV* is the proportion of individuals with a negative test result for which the true condition is negative:

$$NPV = \frac{TN}{(TN + FN)}. \quad (27)$$

5.9. *Matthews' Correlation Coefficient*. *MCC* is a reliable metric used to assess the quality of binary classifiers by taking into account *TP*, *TN*, *FN*, and *FP*. In fact, *MCC* is a correlation coefficient between the actual and predictor labels. This parameter takes a value between  $-1$  and  $+1$ . The  $+1$  coefficient means an excellent estimate,  $0$  indicates that the classifier is not better than random estimates, and  $-1$  means a discrepancy between the actual and predicted values [25]:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (28)$$

## 6. Experimental Results

The dataset used in the study is obtained from [4] and has 195 instances and 22 attributes. It is composed of a range of biomedical voice measurements from 31 people. The dataset information is listed in Table 3. Table 4 lists the appropriate values of the parameters for PSO-SVM, CS-SVM, and CS-PSO-SVM, and the comparison of the models is given in Table 5.

The  $P_a$  parameter in Table 4 is usually selected in the range  $[0, 1]$ . In this study, the program was run for different  $P_a$  parameters, and more successful results were obtained for  $P_a = 0.262$ .

As can be seen in Table 5, the performance of the proposed hybrid model is more superior to that of others.

The population average fitness value for the used dataset is shown in Figure 9 for each method. Also, error rates are shown in Figure 10.

## 7. Discussion and Conclusion

Accurate and reliable diagnosis is very important for human health. Different optimization algorithms have been used for optimizing the SVM parameters in this paper. The aim of this paper is to find the best SVM parameters with the hybrid CS-PSO optimization method and obtain best classification accuracy. For this, to analyze the performances of the used methods, the programs were run several times, and the results are presented as tables. Table 4 shows the appropriate

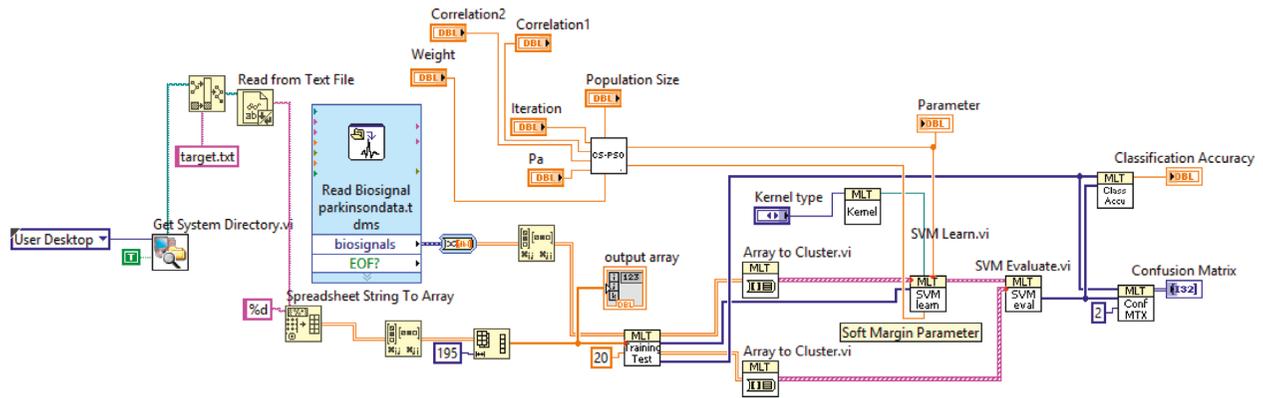


FIGURE 8: The created hybrid program.

TABLE 2: Confusion matrix.

Prediction	Actual	
	Positive	Negative
Positive	TP	FP
Negative	FN	TN

TABLE 3: Dataset information.

Number of instances	Number of attributes	Normal	PD
195	22	8	23

TABLE 4: Parameter settings.

Method	Population size	Iteration	$c_1$	$c_2$
PSO-SVM	18	120	1.3	1.87
CS-SVM ( $P_a = 0.262$ )	18	120	—	—
CS-PSO-SVM ( $P_a = 0.262$ )	18	120	1.3	1.87

TABLE 5: Obtained results.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 measure (%)	FPR	FDR	FNR	NPV	MCC
PSO-SVM	82.05	88.89	57.14	69.57	0.04	0.1111	0.4286	0.80	0.6051
CS-SVM	92.3077	83.33	90.91	86.96	0.0714	0.1667	0.0909	0.9630	0.8167
CS-PSO-SVM ( $P_a = 0.262$ )	97.4359	100	90.91	95.24	0	0	0.0909	0.9655	0.9369

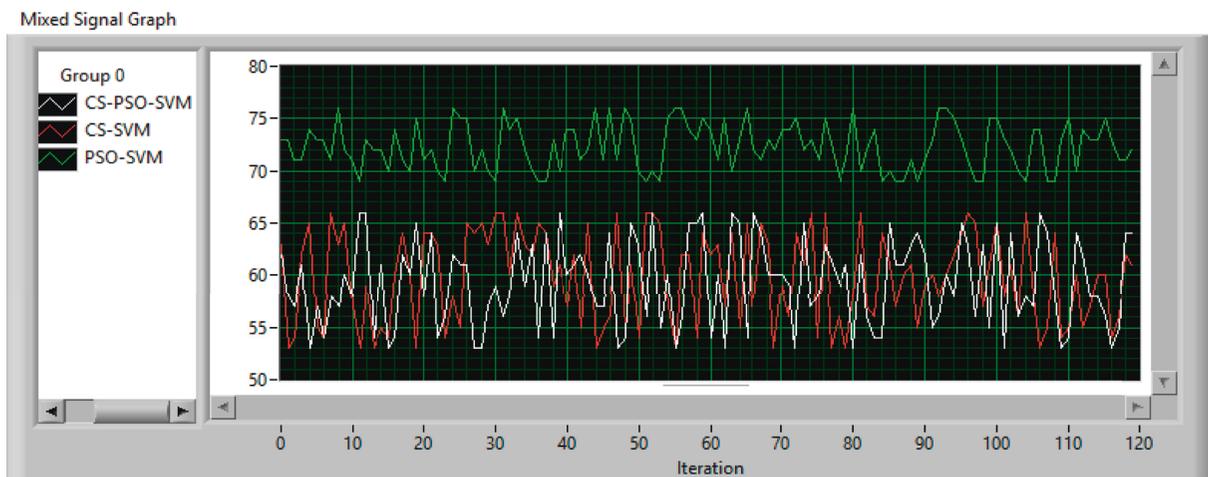


FIGURE 9: Population average fitness value.

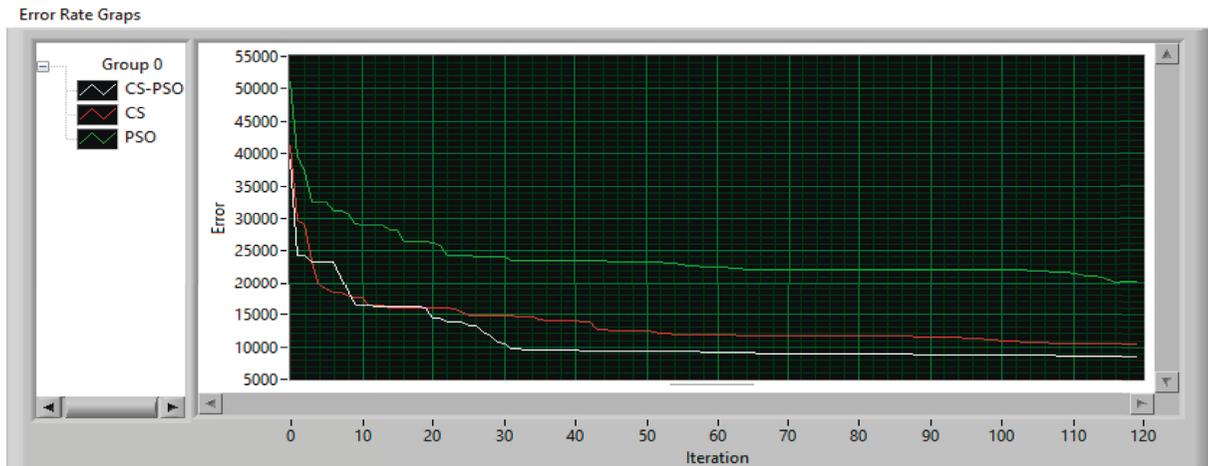


FIGURE 10: Error rate of each method.

algorithm parameters of different methods used for classification. The performances of the used models are shown in Table 5, and the results show that the best result is obtained by the hybrid model (CS-PSO).

The proposed model achieves a classification accuracy of 97.4359%, while this rate is 92.3077% in CS-SVM and 82.05% in PSO-SVM. The MCC contains all parameters in the confusion matrix. The higher value of MCC proves that the proposed classification method is successful. As shown in Table 5, the highest MCC value was obtained with the proposed hybrid algorithm.

As results of this study, hybrid models created by combining the good characteristics of different optimization algorithms can be used to find the parameters of the classification methods, and the success rate of the model can be increased.

In an increasingly widespread LabVIEW environment, it is possible to quickly create subprograms and to obtain results quickly.

## Data Availability

The data that support the findings of this study are available from the authors upon reasonable request.

## Conflicts of Interest

The author declares that there are no conflicts of interest.

## References

- [1] <https://parkinson.org/Understanding-Parkinsons/Causes-and-Statistics/Statistics>.
- [2] <https://www.worldpdcoalition.org/page/AboutParkinson>.
- [3] X. Liu and H. Fu, "PSO-based support vector machine with cuckoo search technique for clinical disease diagnoses," *Scientific World Journal*, vol. 2014, Article ID 548483, 7 pages, 2014.
- [4] <https://ftp.ics.uci.edu/pub/machine-learning-databases/>.
- [5] T. Frieß, N. Cristianini, and C. Campbell, "The kernel-adatron: a fast and simple learning procedure for support vector machines," in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*, pp. 188–196, Madison, WI, USA, July 1998.
- [6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [7] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.
- [8] B. Gu and V. S. Sheng, "A robust regularization path algorithm for support vector classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1241–1248, 2017.
- [9] U. R. Acharya, S. V. Sree, A. P. C. Alvin, and J. S. Suri, "Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework," *Expert Systems*, vol. 39, no. 10, pp. 9072–9078, 2012.
- [10] H. Drucker, C. Burges, L. Kaufman, A. Smola, and Vapnik, "Support vector regression machines," in *Neural Information Processing Systems*, M. Moser, M. Jordan, and T. Petshe, Eds., vol. 9, pp. 155–161, MIT Press, Cambridge, MA, USA, 1997.
- [11] J. Tian, Q. Hu, X. Ma, and M. Han, "An improved KPCA/GA-SVM classification model for plant leaf disease recognition," *Journal of Computational Information Systems*, vol. 8, no. 18, pp. 7737–7745, 2012.
- [12] T. Fletcher, "Support vector machines explained," <https://www.tristanfletcher.co.uk/SVM%20Explained.pdf>, 2014.
- [13] Reyaz-Ahmed, Y.-Q. Zhang, and R. W. Harrison, "Granular decision tree and evolutionary neural SVM for protein secondary structure prediction," *International Journal of Computational Intelligence Systems*, vol. 2, no. 4, pp. 343–352, 2009.
- [14] A. An, C. Angulo, and Y. Sun, "Support vector regression with interval-input interval-output," *International Journal of Computational Intelligence Systems*, vol. 1, no. 4, pp. 299–303, 2008.
- [15] M. Xi, J. Sun, L. Liu, F. Fan, and X. Wu, "Cancer feature selection and classification using a binary quantum-behaved particle swarm optimization and support vector machine," *Computational and Mathematical Methods in Medicine*, vol. 2016, Article ID 3572705, 9 pages, 2016.
- [16] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory—COLT'92*, pp. 144–152, ACM, Pittsburgh, PA, USA, July 1992.

- [17] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [18] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, pp. 39–43, Nagoya, Japan, October 1995.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November 1995.
- [20] F. Serbet, T. Kaya, and M. T. Ozdemir, "Design of digital IIR filter using particle swarm optimization," in *Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 202–204, Opatija, Croatia, May 2017.
- [21] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, Waltham, MA, USA, 1st edition, 2014.
- [22] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.
- [23] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo Search optimization—a review," *Materials Today: Proceedings*, vol. 4, no. 8, pp. 7262–7269, 2017.
- [24] A. S. Kehtarnavaz and O. Mahotra, *Digital Signal Processing Laboratory: LabVIEW-Based FPGA Implementation*, BrownWalker Press, Boca raton, FL, USA, 2010.
- [25] M. Behroozi and A. Sami, "A multiple-classifier framework for Parkinson's disease detection based on various vocal tests," *International Journal of Telemedicine and Applications*, vol. 2016, Article ID 6837498, 9 pages, 2016.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

