

How to cite this article:

Mohammed, A. J., Ghathwan, K. I., & Yusof, Y. (2020). A hybrid least squares support vector machine with bat and cuckoo search algorithms for time series forecasting. *Journal of Information and Communication Technology, 19(3)*, 351-379. <https://doi.org/10.32890/jict2020.19.3.3>

A HYBRID LEAST SQUARES SUPPORT VECTOR MACHINE WITH BAT AND CUCKOO SEARCH ALGORITHMS FOR TIME SERIES FORECASTING

¹Athraa Jasim Mohammed, ¹Khalil Ibrahim Ghathwan & ²Yuhanis Yusof

¹*Computer Science Department, University of Technology, Iraq*

²*School of Computing, Universiti Utara Malaysia, Malaysia*

10872, 110039@uotechnology.edu.iq; yuhanis@uum.edu.my

ABSTRACT

Least Squares Support Vector Machine (LSSVM) has been known to be one of the effective forecasting models. However, its operation relies on two important parameters (regularization and kernel). Pre-determining the values of parameters will affect the results of the forecasting model; hence, to find the optimal value of these parameters, this study investigates the adaptation of Bat and Cuckoo Search algorithms to optimize LSSVM parameters. Even though Cuckoo Search has been proven to be able to solve global optimization in various areas, the algorithm leads to a slow convergence rate when the step size is large. Hence, to enhance the search ability of Cuckoo Search, it is integrated with Bat algorithm that offers a balanced search between global and local. Evaluation was performed separately to further analyze the strength of Bat and Cuckoo Search to optimize LSSVM parameters. Five evaluation metrics were utilized; mean average percent error (MAPE), accuracy, symmetric mean absolute percent error (SMAPE), root mean square percent error (RMSPE) and fitness value. Experimental results on diabetes forecasting demonstrated that the proposed

BAT-LSSVM and CUCKOO-LSSVM generated lower MAPE and SMAPE, at the same time produced higher accuracy and fitness value compared to particle swarm optimization (PSO)-LSSVM and a non-optimized LSSVM. Following the success, this study has integrated the two algorithms to better optimize the LSSVM. The newly proposed forecasting algorithm, termed as CUCKOO-BAT-LSSVM, produces better forecasting in terms of MAPE, accuracy and RMSPE. Such an outcome provides an alternative model to be used in facilitating decision-making in forecasting.

Keywords: Machine learning, data mining, time series forecasting, least squares support vector machine, particle swarm optimization.

INTRODUCTION

Time Series Forecasting is a machine learning field, where it uses historical data to build a model before utilizing it to predict future observations. Technically, it can be defined as “A time series is a set of observations x_t , each one being recorded at a specific time” (Brockwell & Davis, 2002). Least squares support vector machine termed as LSSVM is a machine learning technique that is widely used in forecasting. LSSVM is different from support vector machine (SVM); SVM offers quadratic programming with inequality constraints, in contrast, LSSVM exhibits a system of a linear equations with equality constraints (Suykens, Gestel, Brabanter, Moor, & Vandewalle, 2002). The performance of LSSVM is related to two important parameters: regularization parameter (γ) and kernel parameter (σ^2). LSSVM is susceptible to a problem (of over fitting or under fitting) when the selection of parameters is inadequate. In literature, optimizing LSSVM hyper parameters encompasses two approaches: cross validation (CV) and theoretical technique. The first approach is incompetent due to the comprehensiveness of the parameters search whereas, the second approach embraces meta-heuristic search algorithms that perform well in most cases (Mustaffa, Yusof, & Kamaruddin, 2014).

Based on the literature, meta-heuristic algorithms comprise two categories: local search-based meta-heuristic and population search-based meta-heuristic. Local search-based meta-heuristic algorithm focuses on a single solution and iterates to improve it, such as Simulated Annealing (Hu, Wang, & Ma, 2015) and Tabu Search (Yao, Hu, Zhang, & Jin, 2014). On the other hand, population search-based meta-heuristic algorithm randomly generates a set of solutions and chooses the optimal solution by evaluating them using

certain objective function(s). Examples include evolutionary programming (Xin Yao, Yong Liu, & Guangming Lin, 1999), Genetic Algorithm (McCall, 2005), and swarm algorithms (nature-inspired algorithms) such as particle swarm optimization (PSO) (Ho, Shiyong Yang, Guangzheng Ni, & Wong, 2006), Cuckoo Search (CS) (Hegazy, Soliman, & Abdul Salam, 2015) and Bat algorithm (Hegazy et al., 2015).

Meta-heuristic algorithms are efficient in providing current and global best solutions in search space, where they have proven to be successful in different fields (Sood, Verma, Panchal, & Kavita, 2018). For example, the CS algorithm has been successfully applied in numerous problems (Shehab, Khader, & Laouchedi, 2018; Shehab, Khader, Laouchedi, & Alomari, 2019). However, CS suffers from low convergence rate and global optimization problem. On the other hand, another swarm algorithm, that is the Bat algorithm, has a balanced capability in searching (i.e. global and local search). Thus, the reason of hybridizing CS with Bat algorithm is to increase the prediction accuracy for diabetes forecasting so that better diagnosis can be provided by medical/health officers. In addition, the dependency of LSSVM on the appropriate initialization of its hyper parameters puts the forecasting accuracy at risk. Hence, having an optimized LSSVM will be more beneficial.

In this study, a total of four algorithms are introduced; hybrid Bat algorithm with LSSVM (BAT-LSSVM), hybrid CS with LSSVM (CUCKOO-LSSVM), hybrid Bat algorithm with CS to optimize LSSVM (BAT-CUCKOO-LSSVM), and hybrid CS with Bat algorithm to optimize LSSVM (CUCKOO-BAT-LSSVM). Comparisons between the algorithms are made by analyzing the performance metrics for the outcome of diabetes forecasting. This paper is structured as follows: related work of LSSVM and meta-heuristic algorithms (i.e., Bat algorithm, CS and PSO) utilized in forecasting, technical sections that present the proposed algorithms, experimental setup, results and discussion and the conclusion including some future directions of the study.

RELATED WORK

Forecasting can be defined as a utilization of historical data to predict or estimate future events or trends and it is useful in making decisions. Forecasting methods are classified into three approaches: statistical, artificial intelligence and machine learning (Chan, Dillon, & Chang, 2013). The statistical approach is a traditional method which can be further categorized into univariate and multivariate. Univariate statistical approach predicts y from trends alone, while multivariate statistical approach predicts y from trends and other variables. The univariate approach includes the use of filtering techniques such as Kalman filtering (Louka et al., 2008) and exponential filtering (Chan, Dillon,

Singh, & Chang, 2012) whereas Autoregressive Integrated Moving Average (Lee & Tong, 2011), K-nearest neighbor (Fan, Guo, Zheng, & Hong, 2019) and Bayesian (Thompson & Miller, 1986) are examples of the multivariate approach. Despite the statistical approach yielding acceptable estimates, they do not address the nonlinear characteristics of forecasting (Chan et al., 2013). Thus, artificial intelligence approaches such as Neural Network (NN) has been presented to overcome this shortcoming. NN has been employed in many fields and proven successful in generating high accuracy. Despite this, NN is very complicated as it requires estimation and may be trapped into local minima. To overcome this problem, there has been studies on integrating NN with metaheuristic algorithm (Ozerdem, Olaniyi, & Oyedotun, 2017).

On the other hand, machine learning offers the ability for learning computers without explicit programming (Samuel, 1959). Among the methods offered by machine learning are the SVM and LSSVM. SVM is a large margin algorithm that works to separate training samples into two classes by maximum margin hyper plane between classes. It is a powerful tool for large dimensional data. Since SVM is very effective, it has been used in many application domains. In SVM, two important parameters (i.e. the regularization and the kernel parameters) need to be correctly determined in order to minimize generalization error. Different parameter settings will affect the results of the prediction model (Mustaffa et al., 2014). The computational process of standard SVM relies on quadratic programming solvers which are difficult to perform and requires high computational cost. On the other hand, LSSVM approach solves a set of linear equations without quadratic programming solvers (Mustaffa, Sulaiman, Ernawan, & Noor, 2018; Mustaffa et al., 2014). LSSVM has good convergence and produces high accuracy which has led to its use in forecasting. However, LSSVM relies on the initialization of its parameters; regularization and kernel parameters, like SVM, so as to minimize generalization errors. Determining the best values for the parameters in order to generate the least error can be formulated as an optimization problem. Luo et al. (2008) proposed to tune the parameters of LSSVM by quantum-inspired evaluation algorithm (QEA) which is an example of evaluation algorithm used in various optimization applications. QEA can speed up the evolution by mutation steps that lead to diverse solutions. Hybrid models generate good accuracy as compared with LSSVM tuned by cross validation method with wavelet kernel and Gaussian kernel (Luo et al., 2008). Mustaffa et al. (2014) proposed LSSVM that is tuned by using Improved Artificial Bee Colony (IABC) for gasoline price forecasting. The MAPE and RMSPE results of the proposed method are better than LSSVM tuned by Artificial Bee Colony (ABC) and LSSVM tuned by Back-Propagation Neural Network.

There were also studies on the use of Bat algorithm to optimize LSSVM (Hegazy et al., 2015; Soliman & Salam, 2014; Wu & Peng, 2015). Soliman

and Salam (2014) proposed weekly stock price forecasting and the results indicated that their proposed method achieved better performance in terms of root mean squared error (RMSE), mean absolute error (MAE), SMAPE and percent mean relative error (PMRE) (Soliman & Salam, 2014). This is similar to Wu and Peng (2015) who proposed China wind power forecasting. Their results demonstrated higher accuracy and lower mean squared error (MSE), RMSE and MAE (Wu & Peng, 2015). A similar trend in results was obtained by Hegazy et al. (2015) where they compared five swarm algorithms namely Bat algorithm, Artificial Bee Colony, Flower Pollination algorithm, modified CS, and PSO to optimize LSSVM using stock historical data. The results showed that the Flower Pollination algorithm with LSSVM was a better method in terms of error rate (Hegazy et al., 2015).

In previous work, researchers (Mustaffa et al., 2018; Ong & Zainuddin, 2019; Sun & Sun, 2017) proved the success of the CS in different fields. Sun and Sun (2017) demonstrated the success of a hybrid model for concentration forecasting using Principle Component Analysis and LSSVM optimized by CS against LSSVM. Mustaffa et al. (2018) introduced a hybrid LSSVM with four swarm algorithms namely CS, Genetic Algorithm, Differential Evolution and Grey Wolf Optimizer for water level forecasting. All proposed algorithms were able to produce small MSE, RMSPE and high Theil's U value. Ong and Zainuddin (2019) proposed an improved CS algorithm for optimizing wavelet Neural Network. Zheng et al. (2018) developed PSO with mutation operation to optimize the parameters of constructed wavelet LSSVM and used it for forecasting dissolved gas. Zhang, Tan and Yang (2012) proposed a hybrid method for predicting electricity price by combining wavelet transformed with auto-regressive integrated moving average (ARIMA) while LSSVM was optimized by PSO.

Due to the success of Bat and CS algorithms in earlier studies, investigations on how they can help to tune LSSVM parameters is worth conducting. Nevertheless, the slow convergence rate of CS algorithm needs to be resolved and since Bat algorithm offers a balanced search, integration of the two optimization algorithms may benefit machine learning methods such as LSSVM.

Least Squares Support Vector Machine

In LSSVM, suppose a set of M points $\{x^j, y^j\}$, where x^j means the input values while y^j refers to the output values. LSSVM estimation function for nonlinear regression is shown in Equation 1 (Suykens et al., 2002).

$$y(x) = w^T \phi(x^j) + b + e_j \quad (1)$$

Where: w means the weight vector, $\varphi(x^i)$ means nonlinear function, B means the bias term, e_j means the error between the actual and the predicted output. The weight vector w and the bias term b can be realized using optimizing function which is displayed in the following Equation 2 (Suykens et al., 2002).

$$\min_{w,b,e} J(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{j=1}^M e_j^2 \quad (2)$$

This equation is subject to equality constraints as defined in the following:

Where: e_j means the error variable, $J(w,e)$ means the loss function, Γ means the adjustable constant. The Lagrangian multiplier function is applied to Equation 3 which yields,

$$L(w,b,e,\alpha) = J(w,e) - \sum_{j=1}^M \alpha_j \{w^T \varphi(x^j) + b + e_j - y^j\} \quad (3)$$

Where: α_j means the Lagrangian multiplier, Γ means the regularization parameters. The conditions for optimality upper function of this problem can be generated through all derivatives set to equal zero, which is formulated using Equation 4.

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 &\rightarrow w = \sum_{j=1}^M \alpha_j \varphi(x^j) \\ \frac{\partial L}{\partial b} = 0 &\rightarrow \sum_{j=1}^M \alpha_j = 0 \\ \frac{\partial L}{\partial e_j} = 0 &\rightarrow \alpha_j = \gamma e_j \\ \frac{\partial L}{\partial \alpha_j} = 0 &\rightarrow w^T \varphi(x^j) + b + e_j - y_j = 0 \end{aligned} \quad (4)$$

Where: $j=1, 2, \dots, M$.

After ignoring w and e_j , the Karush-Kuhn-Tucker (KKT) conditions for optimality are transformed into a set of linear equations as shown in Equation 5.

$$\begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} 0 & y^T \\ y & \Omega + I/\gamma \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (5)$$

The LSSVM model for function estimation is as shown in Equation 6.

$$y(x) = \sum_{j=1}^M a_j K(x, x_j) + b \quad (6)$$

Where: α_j , b are the solutions for a linear system in Equation 5, $K(x, x_j)$ is the kernel function. In this study, the kernel function is the radial basis function (RBF) kernel as defined in the following Equation 7.

$$k(x, x_j) = e^{-\frac{\|x-x_j\|^2}{2\sigma^2}} \quad (7)$$

In RBF, σ is a tuning parameter where such value and the value of γ regularization parameter in Equation 2 need to be optimized in order to minimize the generalization error. This study proposes the optimization of the two parameters using population meta-heuristic algorithms.

Bat Algorithm

Bat algorithm is a meta-heuristic algorithm. It was formulated by Yang (2010) and has been efficiently identified as a global solution. Bats utilize echolocation to sense distance; besides, they have positions (X_j) that represent the solutions, velocities (V_j), frequencies (F_j) that represent the objective function, loudness (A) and pulse rate (r). All bats move randomly and update their frequencies, velocities and positions based on Equations 8, 9 and 10, respectively as shown in the following (Yang, 2010).

$$F_j = F_{\min} + (F_{\max} - F_{\min}) \beta \quad (8)$$

$$V_j^t = V_j^{t-1} + (X_j^t - X^*) F_j \quad (9)$$

$$X_j^t = X_j^{t-1} + V_j^t \quad (10)$$

Where: β is a random number between 0 and 1. X^* is the best position (best solution) among all bats. The pseudo-code of basic Bat algorithm is shown in Algorithm 1.

Algorithm 1: Basic Bat algorithm (Yang, 2010)

Input : Determine the number of Bats (n),
 Randomly, determine the initialized position (X_n),
 Randomly, determine the initialized velocity (V_n).
 Determine the frequencies F_n , pulse rate r , loudness A .

Output : Return the best solution

Step 1 : Evaluate each Bat and find best

Step 2 : While not stopping criterion do

Step 3 : For $j = 1$ to n do

(continued)

Algorithm 1: Basic Bat algorithm (Yang, 2010)

- Step 4** : Generate new solution by adjusting frequency, update velocity, and update position using Equations 8, 9, and 10
Step 5 : If (rand > r)
Step 6 : Generate a solution around the best solution
Step 7 : End if
Step 8 : If (rand < A & F(X_j) < F(X*))
Step 9 : Replace the current solution with new solution
Step 10 : Replace the current fitness with new fitness
Step 11 : End if
Step 12 : End while
-

Cuckoo Search

Prior to Bat algorithm, Yang (2010) has also designed an optimization algorithm inspired by the cuckoo species, with their obligated brood parasitism by laying their eggs in the nests of other species or hosts. There are three rules in which the CS algorithm is based; the first rule is that the appropriate place (i.e. the nest) for the egg is chosen randomly. Second, the bird puts only one egg at a time and finally, one of the best nests is chosen for the second generation. The next step is to determine the best nest and the best eggs to be used as input data for future generations.

The available bird nests of hosts is ascertained by a pre-determined value while the detection rate of the cuckoo's egg(s) by host birds takes a probability [0; 1]. If an exotic egg is discovered, the host bird either eliminates the egg or leaves the nest and builds a new nest. The pseudo-code of basic CS is shown in Algorithm 2 (Hegazy et al., 2015; Shehab, Khader & Laouchedi, 2018; Yang, 2010).

Algorithm 2: Basic Cuckoo Search (Yang, 2010)

- Input** : Determine the initial population of (n), and the hosts' nests (X_j), objective function f(x)
Output: Return the best solution
Step 1 : **While** not stopping criterion do
Step 2 : For j = 1 to n do
Step 3 : Generate new solution by selecting a cuckoo randomly, generate a solution and then evaluate it by objective function
Step 4 : Randomly select a nest (i)
Step 5 : If (F_j > F_i)
-

(continued)

Algorithm 2: Basic Cuckoo Search (Yang, 2010)

Step 6 : Update the solution with new one **i 7:** End if
Step 8: A rate (pa) of poor-quality nests are ignored and new nests are generated
Step 9: Find the best solution by ranking the solutions
Step 10: End while

Particle Swarm Optimization

PSO is the behavior studied in bird flocking. PSO was first introduced by **Step** Kennedy and Ebehart (1995) where each single solution represents a bird (it is called a particle) in search space. A particle has initial random position vector (X_i) and velocity (V_i) in search area. The best position is determined by the objective function that measures the cost of position. In each iteration, each particle updates the velocity of the particle at time (t) based on Equation 11 and calculates a new position of the particle at time (t) depending on Equation 12. The pseudo code of a basic PSO is shown in Algorithm 3 (Yang, 2010).

$$V(t+1) = wV(t) + c1 \times r1 \times (X_{pbest} - X(t)) + c2 \times r2 \times (X_{gbest} - X(t))$$

$$X(t+1) = X(t) + V(t+1)$$

Algorithm 3: Basic PSO (Yang, 2010)

Input : Determine the number of particles (n). Randomly, determine the initialized position (X_n), Randomly; determine the initialized velocity (V_n).

Output: Return the best solution

Step 1 : Evaluate each particle and find pbest and gbest

Step 2 : While not stopping criterion do

Step 3 : For $i = 1$ to number of particles do

Step 4 : Calculate the fitness function f

Step 5 : Update personal best and global best of each particle

Step 6 : Update velocity of the particle by using Equation 11

Step 7 : Update the position of the particle using Equation 12

Step 8 : End for

Step 9 : End while

PROPOSED METHODS

This section includes the discussion on the three optimized LSSVMs; BAT-LSSVM, CUCKOO-LSSVM and PSO-LSSVM. The process flow and

performance comparisons among the three methods (i.e., BAT-LSSVM, CUCKOO-LSSVM and PSO-LSSVM) are presented and conducted, respectively. Then a detailed procedure in combining two algorithms (i.e., Bat and CS) to optimize LSSVM is presented. The purpose of this comparison is to evaluate and determine which combination yielded the highest accuracy. Performance is measured using five metrics; MAPE (Yusof et al., 2015) as shown in Equation 13, accuracy (Yusof, Kamaruddin, Husni, Ku-Mahamud, & Mustafa, 2013) as shown in Equation 14, SMAPE (Soliman & Salam, 2014) as shown in Equation 15 and RMSPE (Mustaffa et al., 2018) as shown in Equation 16, and best fitness as shown in Equation 17.

$$MAPE = \frac{1}{N} \left[\sum_{i=1}^N \left| \frac{y_i - p_i}{y_i} \right| \right] \quad (13)$$

$$Accuracy = 100\% - (MAPE \times 100) \quad (14)$$

$$SMAPE = \frac{1}{N} \left[\sum_{i=1}^N \frac{2|y_i - p_i|}{|y_i + p_i|} \right] \quad (15)$$

$$RMSPE = \sqrt[2]{MSPE} \quad (16)$$

$$MSPE = \frac{1}{N} \left[\sum_{i=1}^N \left[\frac{y_i - p_i}{y_i} \right]^2 \right]$$

$$BestFitness = f(x_j) = 1/(1 + MAPE_j) \quad (17)$$

BAT-LSSVM

In order to employ Bat algorithm to optimize the hyper-parameters of LSSVM (σ , γ), the value of each pair of parameters (σ , γ), is a potential solution in the search area with restricted boundaries, where, the frequency of each bat is represented by objective function, MAPE. The objective function evaluates the solutions that are generated randomly in the beginning, then, the bat with the minimum or maximum frequency (depending on the problem) will be identified. New solutions are generated by adjusting frequency, velocity and position. If a stopping criterion is satisfied, the best solution is obtained, where; it has the lowest MAPE. The process flow of proposed BAT-LSSVM is shown in Algorithm 4.

Algorithm 4: Proposed BAT-LSSVM

- Input :** Determine the number of Bats (n). Randomly, determine the initialized position (X_n), where, x_i represents the hyper-parameters of LSSVM (σ , γ). Randomly, determine the initialized velocity (V_n). Determine the frequencies F_n , pulse rate r , loudness A .
- Output:** Return the best solution
- Step 1 :** Initialize the LSSVM model using generated solution.
- Step 2 :** Train the LSSVM model.
- Step 3 :** Evaluate the LSSVM model using Equation 13
- Step 4 :** Frequencies, F_n , at x_i is determined by objective function, $f(x_i)$:
 $f(x_i) = 1/(1 + MAPE_i)$
- Step 5 :** Evaluate each Bat and find best $F(X^*)$
- Step 6 :** While not stopping criterion do
- Step 7 :** For $j = 1$ to n do
- Step 8 :** Generate new solution by adjusting frequency, update velocity, update position using Equations 8, 9, and 10
- Step 9 :** If ($\text{rand} > r$)
- Step 10:** Generate a solution around the best solution
- Step 11:** End if
- Step 12:** Train the LSSVM model with new solution
- Step 13:** Evaluate new solutions and update $F(X_j)$
- Step 14:** If ($\text{rand} < A$ & $F(X_j) > F(X^*)$)
- Step 15:** Replace the current solution with new solution
- Step 16:** Replace the best $F(X^*)$
- Step 17:** End if
- Step 18:** End while
-

CUCKOO-LSSVM

In optimizing the hyper-parameters of LSSVM (σ , γ) using Cuckoo Search algorithm, the solutions of algorithm are depicted by a pair of parameters (σ , γ), where they represent the nest. In this study, each nest evaluates by objective function, where the MAPE is used as an objective function. In the beginning process of the CUCKOO-LSSVM, initial solutions are generated randomly; then, the solutions are trained by the LSSVM model and the LSSVM model is then evaluated using MAPE. If a stopping criterion is not satisfied, the generation of new solutions and evaluation continues until the conditions are met. After that, the best solution is chosen which holds the minimum value of MAPE (i.e. maximum value of fitness). The process flow of the proposed CUCKOO-LSSVM is shown in Algorithm 5.

Algorithm 5: Proposed CUCKOO-LSSVM

Input : Determine the initial population of (n), and the hosts' nests (X_j), where, X_j represents the hyper-parameters of LSSVM (σ, γ). Determine the rate (pa), Randomly, generate a solution.

Output : Return the best solution

Step 1 : Initialize the LSSVM model using generated solution.

Step 2 : Train the LSSVM model.

Step 3 : Evaluate the LSSVM model using Equation 13

Step 4 : Determine the fitness, F_n , at x_j is determined by objective function, $f(x_j)$:

$$f(x_i) = 1/(1 + MAPE_i)$$

Step 5 : Evaluate each nest

Step 6 : While not stopping criterion do

Step 7 : Find best nest $F(X^*)$, which represents maximum value.

Step 8 : Choose a nest randomly x_i and avoid best nest.

Step 9 : Train the LSSVM model, evaluate new nest by objective function, and find fitness $F(x_i)$

Step 10 : If ($F(x_i) \geq F(X^*)$)

Step 11 : Replace the current solution with new solution

Step 12 : Replace the best $F(X^*)$

Step 13 : End if

Step 14 : If ($\text{rand} > pa$)

Step 15 : Replace some of the worse nests.

Step 16 : End if

Step 17 : End while

PSO-LSSVM

In order to employ PSO in optimizing the hyper-parameters of LSSVM (σ, γ), the value of each pair of parameters (σ, γ) is determined by the value of swarm position in the search area. The position of each swarm changes as it depends on the velocity of each swarm and the change of best swarm (pbest) and global swarm (gbest). The evaluation of each position is based on objective function that is formulated using MAPE. The process of LSSVM model training continues using the new solutions (positions) until the stopping conditions are met. Then, the best solution is chosen and stored. The process flow of the proposed PSO-LSSVM is shown in Algorithm 6.

Algorithm 6: Proposed PSO-LSSVM

Input : Determine the number of swarm (n). Randomly, determine the initialized position (X_n), where, x_i represents the hyper-parameters of LSSVM (σ, γ). Randomly, determine the initialized velocity (V_n).

Output: Return the best solution

Step 1 : Initialize the LSSVM model using generated solution.

Step 2 : Train the LSSVM model.

Step 3 : Evaluate the LSSVM model using Equation 13

Step 4 : Frequencies, F_n , at x_i is determined by objective function, $f(x_i)$:

$$f(x_i) = 1/(1 + MAPE_i)$$

Step 5 : Evaluate each swarm and find pbest and gbest

Step 6 : While not stopping criterion do

Step 7 : For $j = 1$ to n do

Step 8 : Update personal best and global best of each swarm

Step 9 : Update velocity of the particle by using Equation 11

Step 10 : Update the position of the particle using Equation 12

Step 11 : Train the LSSVM model with new solution.

Step 12 : Evaluate new solutions, and calculate the fitness function f

Step 13 : Update personal best and global best of each swarm

Step 14 : End while

BAT-CUCKOO-LSSVM

In this integration, two algorithms are used to optimize LSSVM hyper-parameters (σ, γ). First, the Bat algorithm is employed, and its outcome is used as the input for CS. The value of each pair of parameters (σ, γ) is a potential solution in the search area with restricted boundaries, where, the frequency of each bat is represented by objective function (i.e., MAPE). The objective function is used to evaluate solutions which are randomly generated. This is followed by determining the best bat (i.e., the one with minimum frequency). New solutions are produced by adjusting the frequency, velocity, and position of the bats. The process is repeated until the stopping criterion is fulfilled (i.e., the smallest MAPE). The solutions produced by the Bat algorithm are passed to CS as initial solutions, where each of the solution is considered as a nest. Each nest is evaluated using the objective function (also based on MAPE). Then, the solutions are trained by LSSVM where the model with the smallest MAPE is stored for prediction purposes. Similarly, it all relies on the achievement of the stopping criterion. Generations of new solutions and evaluations continue until the stopping condition is met. The process flow of the proposed BAT-CUCKOO-LSSVM is shown in Algorithm 7.

Algorithm 7: Proposed BAT-CUCKOO-LSSVM

Input : Determine the number of Bats (n). Randomly, determine the initialized position (X_n), where, x_i represents the hyper-parameters of LSSVM (σ, γ). Randomly, determine the initialized velocity (V_n). Determine the frequencies F_n , pulse rate r , loudness A .

Output : Return the solutions that include the best.

Step 1 : Initialize the LSSVM model using generated solution.

Step 2 : Train the LSSVM model.

Step 3 : Evaluate the LSSVM model using Equation 13

Step 4 : Frequencies, F_n , at x_i is determined by objective function, $f(x_i)$:

$$f(x_i) = 1/(1 + MAPE_i)$$

Step 5 : Evaluate each Bat and find the best $F(X^*)$

Step 6 : While not stopping criterion do

Step 7 : For $j = 1$ to n do

Step 8 : Generate new solutions by adjusting frequency, update velocity, update position using Equations 8, 9, and 10

Step 9 : If ($\text{rand} > r$)

Step 10 : Generate a solution around the best solution

Step 11 : End if

Step 12 : Train the LSSVM model with new solutions.

Step 13 : Evaluate new solutions and update $F(X_j)$

Step 14 : If ($\text{rand} < A$ & $F(X_j) > F(X^*)$)

Step 15 : Replace the current solution with new solution

Step 16 : Replace the best $F(X^*)$

Step 17 : End if

Step 18 : End while

Input : Determine the initial population of (n), and the host nests (X_j), where, X_j represents the hyper-parameters of LSSVM (σ, γ) that gained from the Bat algorithm. Determine the rate (pa)

Output : Return the best solution

Step 1 : Nests equal gained solutions.

Step 2 : Evaluate each nest

Step 3 : While not stopping criterion do

Step 4 : Find best nest $F(X^*)$, which represents maximum value.

Step 5 : Choose a nest randomly x_i and avoid best nest.

Step 6 : Train the LSSVM model, evaluate new nest by objective function, and find fitness $F(x_i)$

Step 7 : If ($F(x_i) \geq F(X^*)$)

Step 8 : Replace the current solution with new solution

Step 9 : Replace the best $F(X^*)$

Step 10 : End if

Step 11 : If ($\text{rand} > pa$)

Step 12 : Replace some of the worse nest.

Step 13 : End if

Step 14 : End while

CUCKOO-BAT-LSSVM

As for the CUCKOO-BAT-LSSVM, the operation starts with CS. Cuckoo Search solutions are depicted by a pair of parameters (σ, γ) , that indicates the nest. Each nest is evaluated using objective function (i.e., MAPE). Similar to the earlier process in BAT-CUCKOO-LSSVM, initial solutions are generated randomly. Then, they are trained by LSSVM and evaluated using MAPE. If the stopping criterion is not met, generations of new solutions and evaluations continue. After that, the solutions (i.e., the ones with the smallest MAPE) will be used to represent potential solutions in the search area of Bat algorithm. The standard Bat operation is then followed until the stopping criteria is fulfilled. The process flow of the proposed CUCKOO-BAT-LSSVM is shown in Algorithm 8.

Algorithm 8: Proposed CUCKOO-BAT-LSSVM

Input : Determine the initial population of (n) , and the host nests (X_j) , where, X_j represents the hyper-parameters of LSSVM (σ, γ) . Determine the rate (pa) . Randomly, generate a solution.

Output : Return the solutions that include the best one.

Step 1 : Initialize the LSSVM model using generated solution.

Step 2 : Train the LSSVM model.

Step 3 : Evaluate the LSSVM model using Equation 13

Step 4 : Determine the fitness, F_n , at x_j is determined by objective function, $f(x_j)$:

$$f(x_i) = 1/(1 + MAPE_i)$$

Step 5 : Evaluate each nest

Step 6 : While not stopping criterion do

Step 7 : Find best nest $F(X^*)$, which represents maximum value.

Step 8 : Choose a nest randomly x_i and avoid best nest.

Step 9 : Train the LSSVM model, evaluate new nest by objective function, and find fitness $F(x_i)$

Step 10 : If $(F(x_i) \geq F(X^*))$

Step 11 : Replace the current solution with new solution

Step 12 : Replace the best $F(X^*)$

Step 13 : End if

Step 14 : If $(rand > pa)$

Step 15 : Replace some of the worse nests.

Step 16 : End if

Step 17 : End while

(continued)

Algorithm 8: Proposed CUCKOO-BAT-LSSVM

- Input :** Determine the number of Bats (n). Determine the initialized position (X_n), where, xi represents the hyper-parameters of LSSVM (σ, γ) that is gained from the Cuckoo algorithm. Randomly, determine the initialized velocity (V_n). Determine the frequencies F_n , pulse rate r, loudness A.
- Output:** Return the best solution
- Step 1 :** Evaluate each Bat and find best $F(X^*)$
- Step 2 :** While not stopping criterion do
- Step 3 :** For $j = 1$ to n do
- Step 4 :** Generate new solution by adjusting frequency, update velocity, update position using Equations 8, 9, and 10
- Step 5 :** If (rand > r)
- Step 6 :** Generate a solution around the best solution
- Step 7 :** End if
- Step 8 :** Train the LSSVM model with new solution.
- Step 9 :** Evaluate new solutions and update $F(X_j)$
- Step 10 :** If (rand < A & $F(X_j) > F(X^*)$)
- Step 11 :** Replace the current solution with new solution
- Step 12 :** Replace the best $F(X^*)$
- Step 13 :** End if
- Step 14 :** End while

EXPERIMENTAL SETUP

Implementation of BAT-LSSVM, CUCKOO-LSSVM, PSO-LSSVM, BAT-CUCKOO-LSSVM, and CUCKOO-BAT-LSSVM are conducted using LSSVM lab toolbox (Pelkmans et al., 2002). The parameter setting of these experiments is presented in Table 1.

Table 1

Experimental Setup of the Proposed Methods

Bat algorithm		CS		PSO	
Number of bats (n)	100	Initial population (n)	100	Number of swarm (n)	100
Pulse rate (r)	0.5	Rate (pa)	0.25	w	1
Loudness (A)	0.25			c1, c2	1.5

Data Description

In this study, real datasets, named Diabetes Dataset (Dua & Graff, 2017), was obtained from UCI repository which was utilized to test the effectiveness of the forecasting. The samples covered were from 10 October 1989 to 21 April 1991. Table 2 shows the descriptive statistics of the dataset which includes 1683 cases and four variables. The variables are: date, time, code and value of diabetes mellitus. Before utilizing the data, the dataset was divided into three sub-datasets, 70% for training, 15% for validation and 15% for testing.

Table 2

Descriptive Statistics of Data

Variable	N	Minimum	Maximum	Mean	Std. Deviation
Date	1683	14901	14902	-	-
Time	1683	3600	85500	49624.53	19334.380
Code	1683	33	65	44.64	12.777
Value	1683	0	393	82.82	97.012

Data Normalization

In order to simplify the training task and produce better results, data normalization was performed using min-max normalization (Yusof et al., 2015) and is expressed in Equation 18.

$$\bar{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (18)$$

Where, \bar{x} is the normalized data, x is the original data, x_{\min} refers to minimum value in dataset while x_{\max} refers to maximum value.

RESULTS

The forecasted values for the testing dataset (i.e., values for attribute diabetes) generated by BAT-LSSVM, CUCKOO-LSSVM, PSO-LSSVM, LSSVM, CUCKOO-BAT-LSSVM and BAT-CUCKOO-LSSVM were compared among each other against the actual values (Figure 1). It showed that the forecasted values of CUCKOO-BAT-LSSVM and BAT-CUCKOO-LSSVM were close to the original.

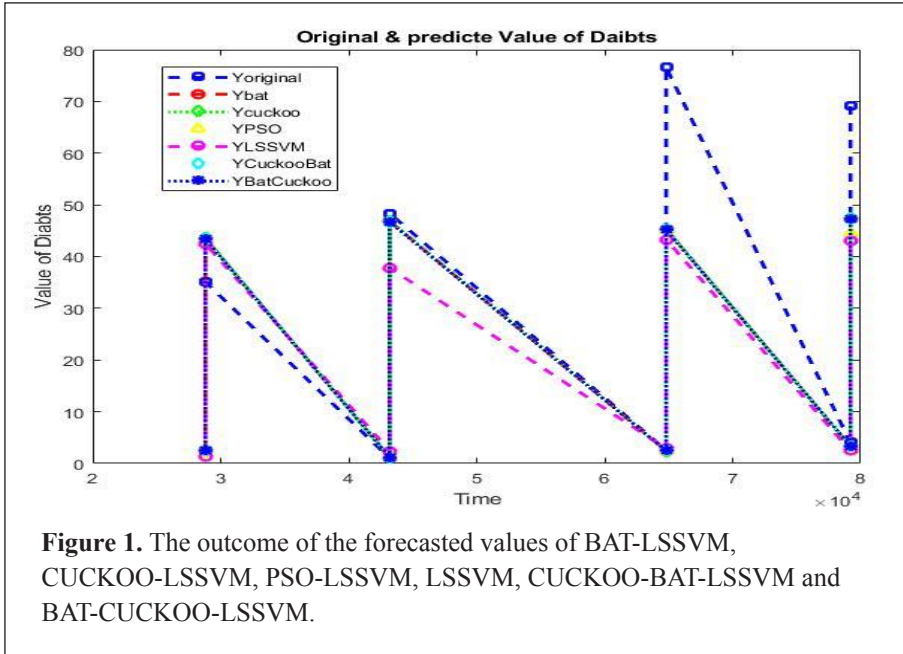


Table 3

Average Results of Three Optimized Methods BAT-LSSVM, CUCKOO-LSSVM, and PSO-LSSVM with LSSVM

Method	MAPE Test	Accuracy	SMAPE Test	RMSPE Test	Best Fitness
LSSVM	26.84089	73.15911	24.55659539	0.418034693	0.016658599
CUCKOO-LSSVM	21.36823	78.63177	18.20837469	0.413055088	0.035003629
PSO-LSSVM	24.7515	75.2485	22.76379234	0.408215635	0.019302678
BAT-LSSVM	21.26162	78.73838	18.26129948	0.407197641	0.03434636

A comparison of the results among the three optimized methods, BAT-LSSVM, CUCKOO-LSSVM, and PSO-LSSVM, and non-optimized LSSVM are presented in Table 3. It is noted that BAT-LSSVM generated the lowest MAPE i.e. 21.26162 (as highlighted in Table 3) which is smaller than CUCKOO-LSSVM which is 21.36823, PSO-LSSVM is 24.7515 and LSSVM is 26.84089. Figure 2 shows the change in MAPE for the four methods.

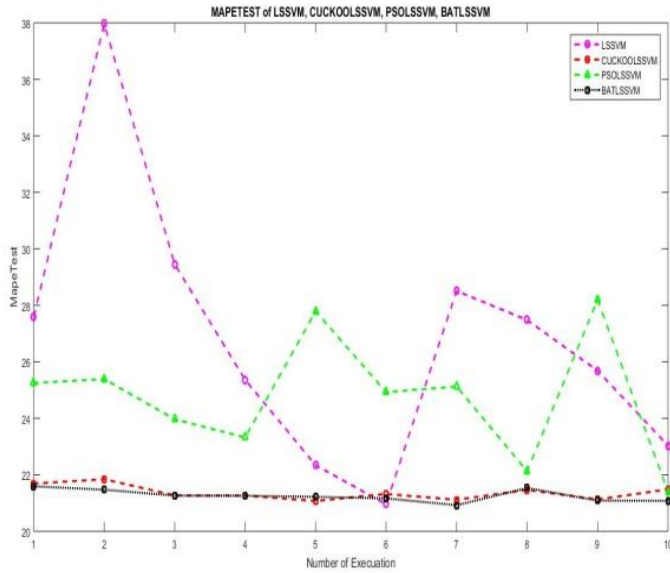


Figure 2. MAPE test of LSSVM, CUCKOO-LSSVM, PSO-LSSVM, & BAT-LSSVM.

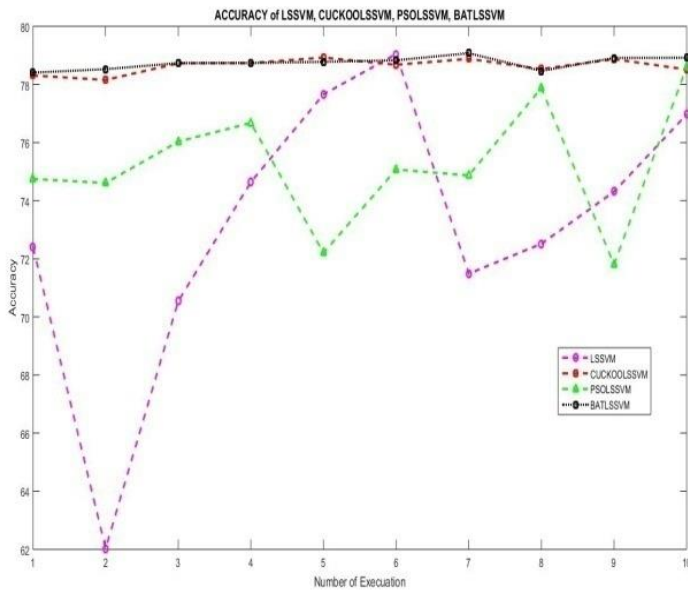
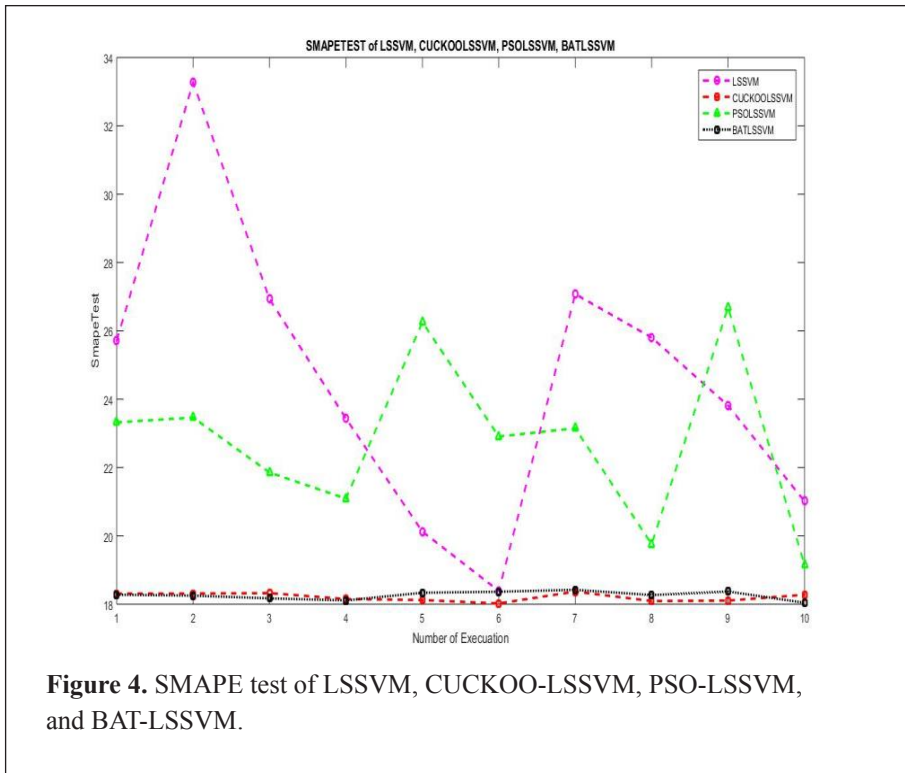


Figure 3. Accuracy of LSSVM, CUCKOOLSSVM, PSOLSSVM, and BATLSSVM.

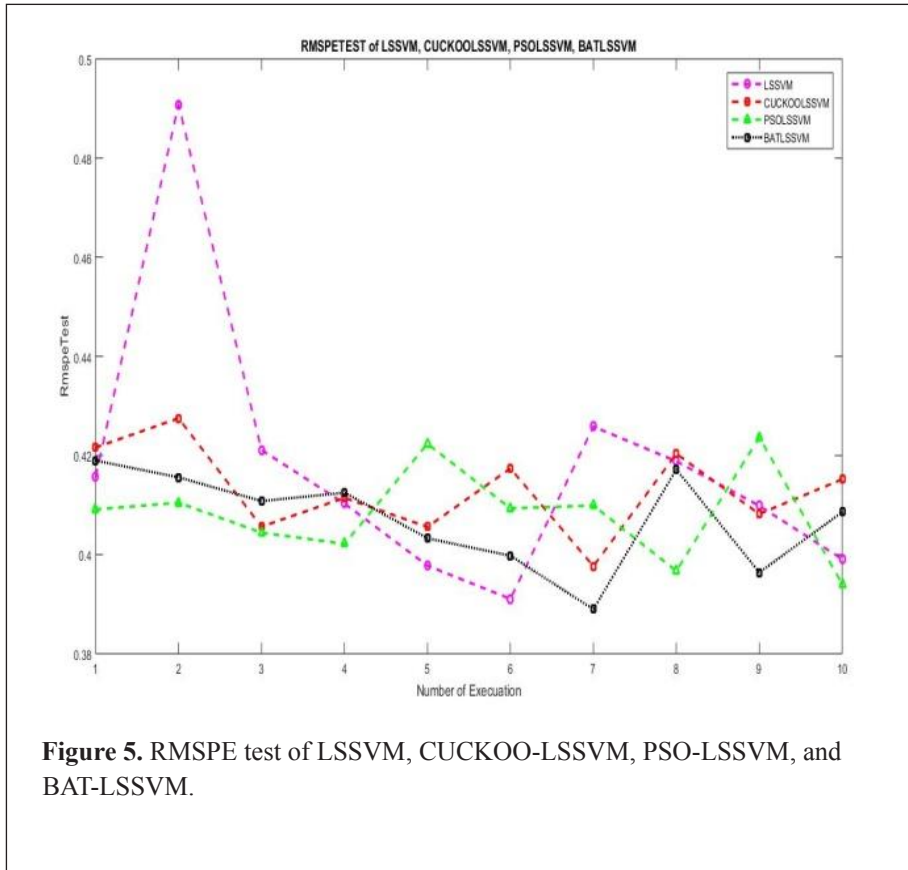
Further, the average Accuracy for the four methods: LSSVM, CUCKOO-LSSVM, PSO-LSSVM, and BAT-LSSVM are presented in Table 3. The highest score can be seen in the BAT-LSSVM (as highlighted in Table 3 which is 78.73838) whereas the lowest is produced by the non-optimized LSSVM. In contrast to the MAPE, the larger the value for accuracy, the better the method. Figure 3 illustrates the difference in the predicted accuracy.

In addition, the data in Table 3 also includes information on SMAPE for the four LSSVM methods. The average SMAPE test of BAT-LSSVM is 18.26129948, CUCKOO-LSSVM is 18.20837469, PSO-LSSVM is 22.76379234, and LSSVM is 24.55659539. The CUCKOO-LSSVM has the smallest SMAPE compared with the other methods. Illustration of the change is shown in Figure 4.



From Table 3, it can also be observed that the RMSPE value for BAT-LSSVM is the smallest. Figure 5 shows the change in RMSPE test values for all methods under analysis. Besides the error rate, Table 3 also shows the fitness value for the methods, LSSVM, CUCKOO-LSSVM, PSO-LSSVM, and BAT-LSSVM. The average fitness of BAT-LSSVM is 0.03434636 which is similar to CUCKOO-LSSVM (i.e., 0.035003629). While, the fitness for

PSO-LSSVM is 0.019302678, which is closer to the one obtained by LSSVM (i.e., 0.016658599). Nevertheless, CUCKOO-LSSVM is a better method as it generated the strongest fitness. Illustration of the results is provided in Figure 6 and based on these results (i.e. error rates, accuracy and fitness), it is noted that the two strong methods are BAT-LSSVM (i.e. the best in MAPE, accuracy and RMSPE) and CUCKOO-LSSVM (i.e. the best in terms of SMAPE and fitness).



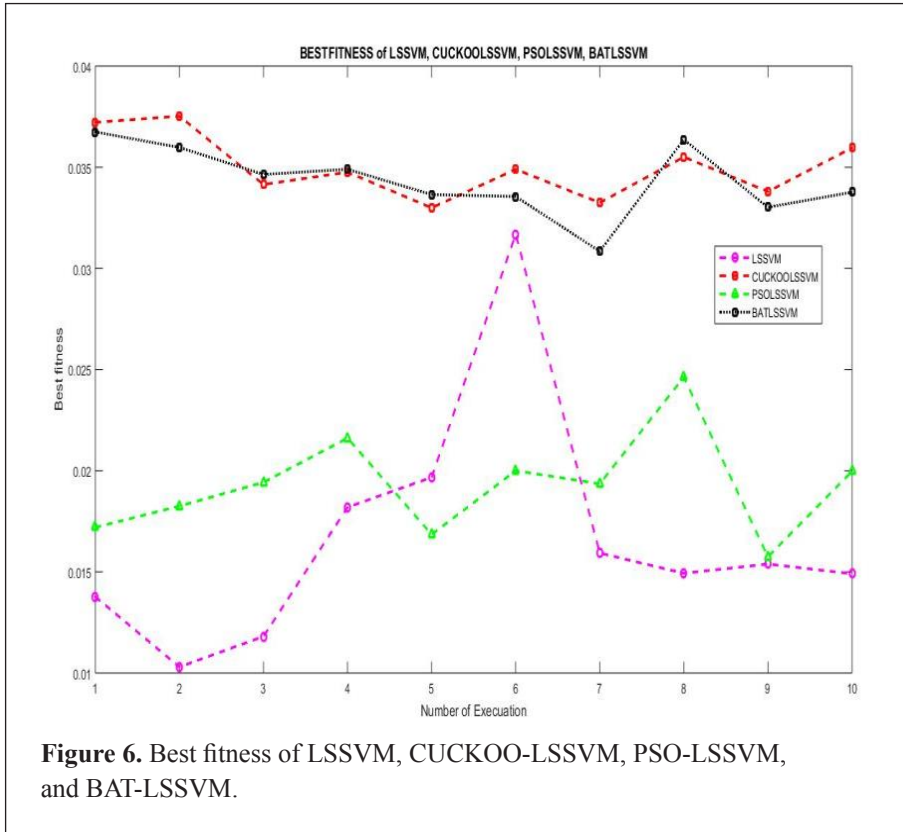
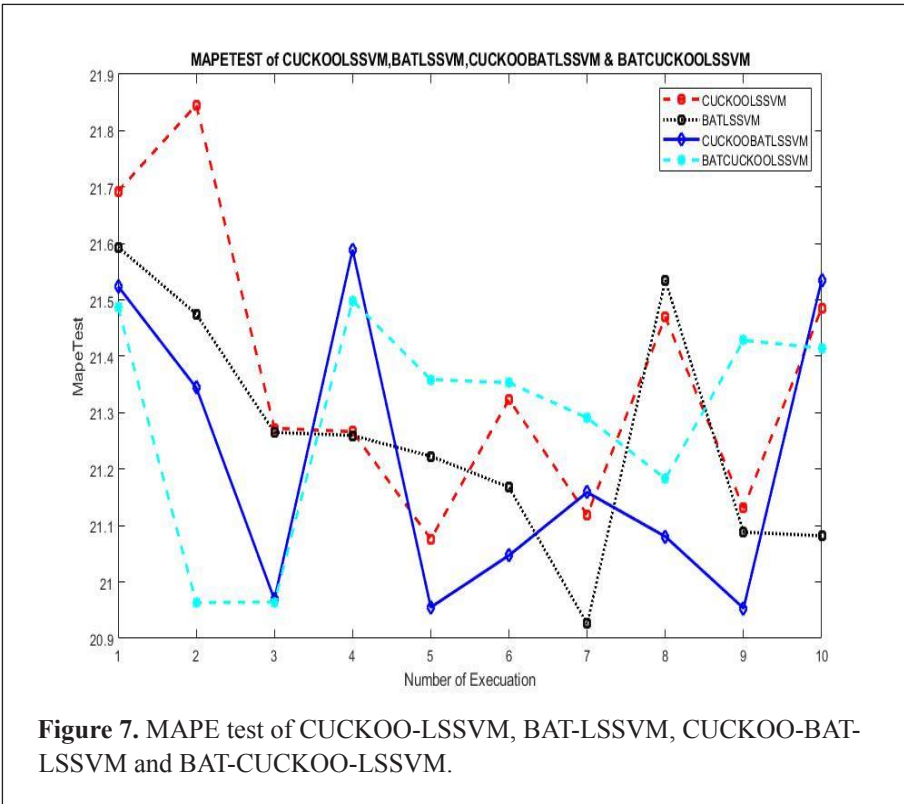


Table 4

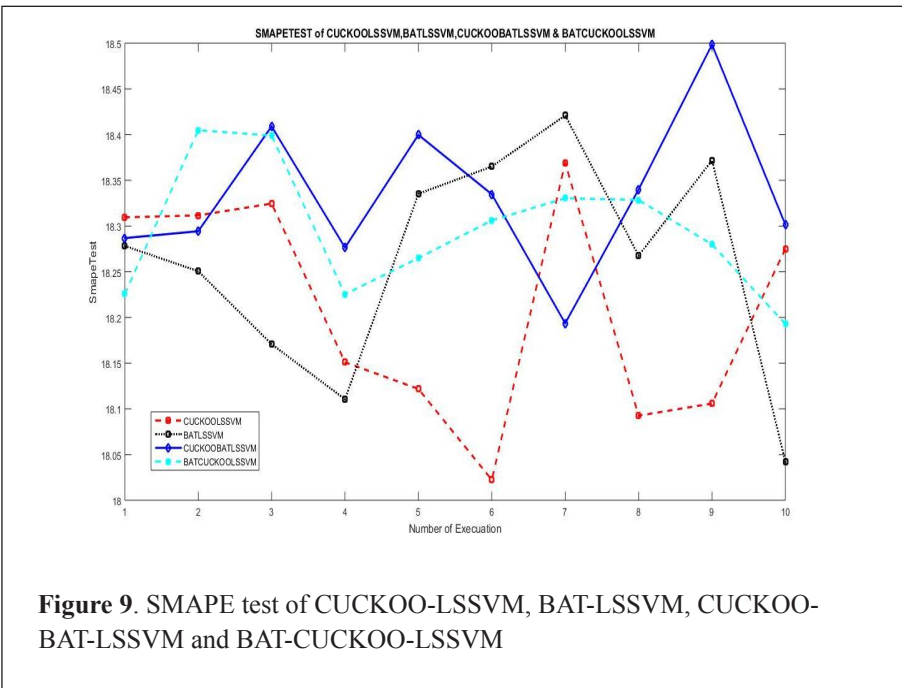
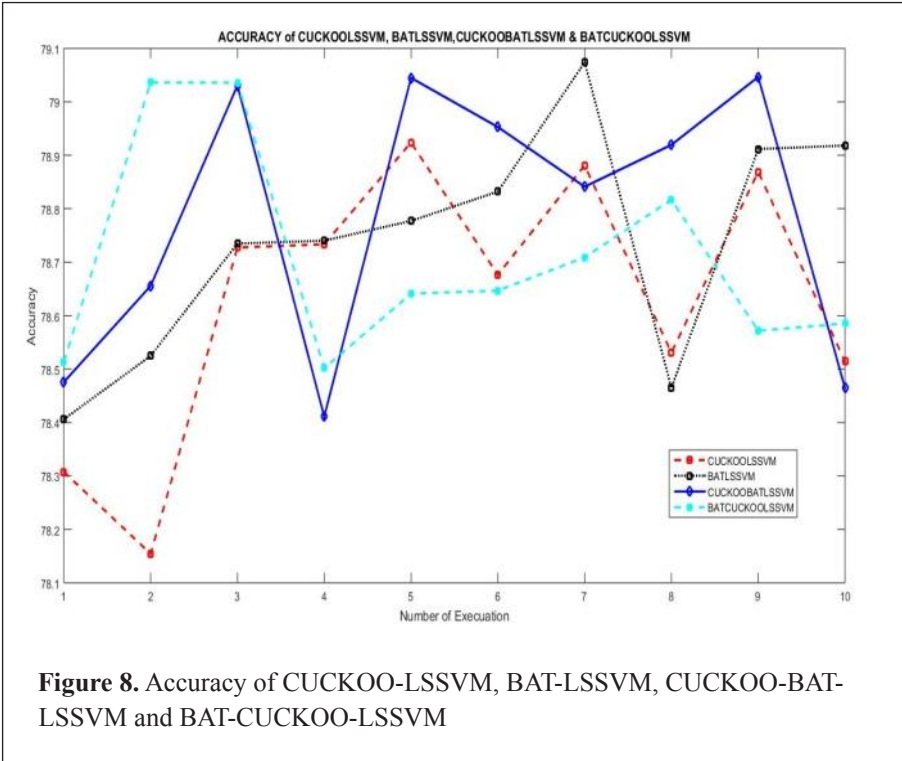
Average Results of CUCKOO-LSSVM, BAT-LSSVM, CUCKOO-BAT-LSSVM and BAT-CUCKOO-LSSVM

Method	MAPE Test	Accuracy	SMAPE Test	RMSPE Test	Best Fitness
CUCKOO-LSSVM	21.36823	78.63177	18.20837469	0.413055088	0.035003629
BAT-LSSVM	21.26162	78.73838	18.26129948	0.407197641	0.03434636
CUCKOO-BAT-LSSVM	21.2158	78.7842	18.33347682	0.40304565	0.033579709
BAT-CUCKOO-LSSVM	21.2943	78.7057	18.29575524	0.407233212	0.034490449

Experimental results using two algorithms to optimize the LSSVM are shown in Table 4. Data in Table 4 compares the outcome of four methods: CUCKOO-LSSVM, BAT-LSSVM, CUCKOO-BAT-LSSVM, and BAT-CUCKOO-LSSVM. Investigation on the effectiveness of using two optimizers has shown that CUCKOO-BAT-LSSVM is better than BAT-CUCKOO-LSSVM. The average MAPE for CUCKOO-BAT-LSSVM is 21.2158 in contrast to the BAT-CUCKOO-LSSVM which generated a higher value, i.e. 21.2943. Figure 7 shows the change in MAPE values for four of the methods.



As CUCKOO-BAT-LSSVM produced the smallest MAPE and RMSPE (Figure 10), hence, it also generated the highest forecasting accuracy (i.e., CUCKOO-BAT-LSSVM with 78.7842% and BAT-CUCKOO-LSSVM with 78.7057%). Figure 8 shows the change in accuracy values for four of the methods. However, the best fitness (on average) was produced by the CUCKOO-LSSVM which was 0.035003629 as compared with CUCKOO-BAT-LSSVM which produced the worst (i.e., 0.033579709). The change in the results is presented in Figure 11. Nevertheless, CUCKOO-BAT-LSSVM is still the best method as the difference is not significant.



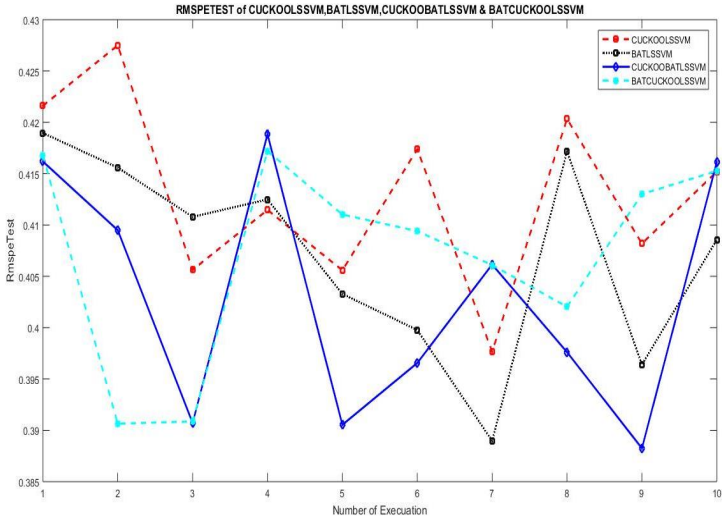


Figure 10. RMSPE test of CUCKOO-LSSVM, BAT-LSSVM, CUCKOOBAT-LSSVM and BATCUCKOO-LSSVM

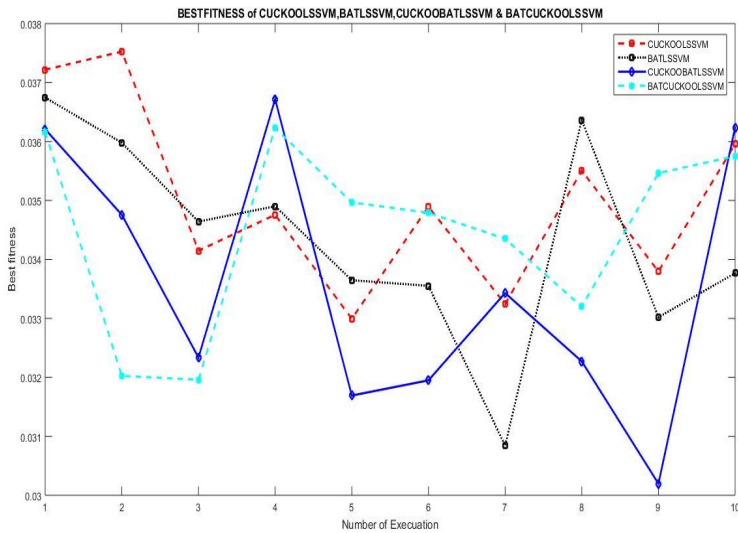


Figure 11. Best Fitness of CUCKOO-LSSVM, BAT-LSSVM, CUCKOOBAT-LSSVM and BATCUCKOO-LSSVM

CONCLUSION

This study focuses on how to optimize the hyper-parameters of LSSVM, one of the efficient models in machine learning. LSSVM was optimized using three swarm intelligence algorithms; Bat, CS and PSO. The methods were evaluated on medical data which was to predict the value of diabetes. The success of the CS algorithm in finding solutions for global optimization problems in different fields and the strength of the Bat algorithm in exploration and exploitation have yielded good results. The experimental results have demonstrated that BAT-LSSVM has proven to be a good forecasting model in terms of MAPE, accuracy, and RMSPE, whereas CUCKOO-LSSVM is better in terms of SMAPE and fitness value. In addition, the use of the two swarm algorithms (i.e., Bat and CS) to optimize LSSVM have increased LSSVM performance. A lower error rate and higher accuracy have been obtained by both the optimizers. The experimental results also showed that the hybrid CUCKOO-BAT-LSSVM produced better results than the hybrid BAT-CUCKOO-LSSVM in terms of accuracy, MAPE, RMSPE and fitness value. Besides, it also obtained better results compared with CUCKOO-LSSVM and BAT-LSSVM in terms of accuracy, MAPE and RMSPE. With this, the proposed CUCKOO-BAT-LSSVM can be acknowledged as an alternative solution for an accurate forecasting model. To further enhance the forecasting, an investigation of parameter selection for swarm algorithms is worth exploring.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

- Brockwell, P. J., & Davis, R. A. (2002). *Introduction to Time Series and Forecasting* (pp. XIV, 437). Springer-Verlag New York. doi:10.1007/b97391
- Chan, K. Y., Dillon, T. S., & Chang, E. (2013). An Intelligent Particle Swarm Optimization for Short-Term Traffic Flow Forecasting Using on-Road Sensor Systems. *IEEE Transactions on Industrial Electronics*, 60(10), 4714–4725. doi:10.1109/TIE.2012.2213556
- Chan, K. Y., Dillon, T. S., Singh, J., & Chang, E. (2012). Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg Marquardt Algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 644–654. doi:10.1109/TITS.2011.2174051

- Dua, D., & Graff, C. (2017). {UCI} Machine Learning Repository. Retrieved from <http://archive.ics.uci.edu/ml>
- Fan, G. F., Guo, Y. H., Zheng, J. M., & Hong, W. C. (2019). Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. *Energies*, 12. doi:10.3390/en12050916
- Hegazy, O., Soliman, O. S., & Abdul Salam, M. (2015). Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction, 5(18).
- Ho, S. L., Shiyong Yang, Guangzheng Ni, & Wong, H. C. (2006). A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices. *IEEE Transactions on Magnetics*, 42(4), 1107–1110. doi:10.1109/TMAG.2006.871426
- Hu, J., Wang, J., & Ma, K. (2015). A hybrid technique for short-term wind speed prediction. *Energy*, 81, 563–574. doi:<https://doi.org/10.1016/j.energy.2014.12.074>
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). doi:10.1109/ICNN.1995.488968
- Lee, Y.-S., & Tong, L.-I. (2011). Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Systems*, 24(1), 66–72. doi:<https://doi.org/10.1016/j.knosys.2010.07.006>
- Louka, P., Galanis, G., Siebert, N., Kariniotakis, G., Katsafados, P., Pytharoulis, I., & Kallos, G. (2008). Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(12), 2348–2362. doi:<https://doi.org/10.1016/j.jweia.2008.03.013>
- Luo, Z., Wang, P., Li, Y., Zhang, W., Tang, W., & Xiang, M. (2008). Quantum-inspired evolutionary tuning of SVM parameters. *Progress in Natural Science*, 18, 475–480. doi:10.1016/j.pnsc.2007.11.012
- McCall, J. (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, 184(1), 205–222. doi:<https://doi.org/10.1016/j.cam.2004.07.034>
- Mustaffa, Z., Sulaiman, M. H., Ernawan, F., & Noor, N. M. (2018). Short Term Forecasting Based on Hybrid Least Squares Support Vector Machines. In *Advanced Science Letters* (Vol. 24, pp. 7455–7460). American Scientific Publishers. doi:<https://doi.org/10.1166/asl.2018.12958>
- Mustaffa, Z., Yusof, Y., & Kamaruddin, S. S. (2014). Gasoline Price Forecasting: An Application of LSSVM with Improved ABC. *Procedia - Social and Behavioral Sciences*, 129, 601–609. doi:<https://doi.org/10.1016/j.sbspro.2014.03.718>
- Ong, P., & Zainuddin, Z. (2019). Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series

- prediction. *Applied Soft Computing*, 80, 374–386. doi:<https://doi.org/10.1016/j.asoc.2019.04.016>
- Ozerdem, O. C., Olaniyi, E. O., & Oyedotun, O. K. (2017). Short term load forecasting using particle swarm optimization neural network. *Procedia Computer Science*, 120, 382–393. doi:10.1016/j.procs.2017.11.254
- Pelkmans, K., Suykens, J. A. K., Gestel, T. V., Barbanter, J. D., Lukas, L., Hamer, B., et al. (2002). LSSVM: A Matlab/c toolbox for least squares support vector machines. From www.esat.kuleuven.be/sista/lssvmlab
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. doi:10.1147/rd.33.0210
- Shehab, M., Khader, A. T., & Laouchedi, M. (2018). A hybrid method based on Cuckoo search algorithm for global optimization problems. *Journal of Information and Communication Technology*, 17(3), 469–491.
- Shehab, M., Khader, A. T., Laouchedi, M., & Alomari, O. A. (2019). Hybridizing cuckoo search algorithm with bat algorithm for global numerical optimization. *The Journal of Supercomputing*, 75(5), 2395–2422. doi:10.1007/s11227-018-2625-x
- Soliman, O. S., & Salam, M. A. (2014). A Hybrid BA-LS-SVM Model and Financial Technical Indicators for Weekly Stock Price and Trend Prediction.
- Sood, M., Verma, D. S., Panchal, V. K., & Kavita. (2018). Optimal path planning using Hybrid Bat Algorithm and Cuckoo Search. *International Journal of Engineering and Technology(UAE)*, 7, 30–33. doi:10.14419/ijet.v7i4.12.20987
- Sun, W., & Sun, J. (2017). Daily PM2.5 concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm. *Journal of Environmental Management*, 188, 144–152. doi:<https://doi.org/10.1016/j.jenvman.2016.12.011>
- Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. De, & Vandewalle, J. (2002). *Least Squares Support Vector Machines* (p. 308). World Scientific Pub Co Inc.
- Thompson, P. A., & Miller, R. B. (1986). Sampling the Future: A Bayesian Approach to Forecasting From Univariate Time Series Models. *Journal of Business & Economic Statistics*, 4(4), 427–436. doi:10.1080/07350015.1986.10509542
- Wu, Q., & Peng, C. (2015). Wind power grid connected capacity prediction using LSSVM optimized by the bat algorithm. *Energies*, 8, 14346–14360.
- Xin Yao, Yong Liu, & Guangming Lin. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2), 82–102. doi:10.1109/4235.771163
- Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.

- Yao, B., Hu, P., Zhang, M., & Jin, M. (2014). A Support Vector Machine with the Tabu Search Algorithm for Freeway Incident Detection. *Int. J. Appl. Math. Comput. Sci.*, 24(2), 397–404. doi:10.2478/amcs-2014-0030
- Yusof, Y., Ahmad, F. K., Kamaruddin, S. S., Omar, M. H., & Mohamed, A. J. (2015). Short Term Traffic Forecasting Based on Hybrid of Firefly Algorithm and Least Squares Support Vector Machine. In M. W. Berry, A. Mohamed, & B. W. Yap (Eds.), *Soft Computing in Data Science* (pp. 164–173). Singapore: Springer Singapore.
- Yusof, Y., Kamaruddin, S. S., Husni, H., Ku-Mahamud, K. R., & Mustaffa, Z. (2013). Forecasting model based on LSSVM and ABC for natural resource commodity. *International Journal of Computer Theory and Engineering*, 5(6), 906.
- Zhang, J., Tan, Z., & Yang, S. (2012). Day-ahead electricity price forecasting by a new hybrid method. *Computers & Industrial Engineering*, 63(3), 695–701. doi:https://doi.org/10.1016/j.cie.2012.03.016
- Zheng, H., Zhang, Y., Liu, J., Wei, H., Zhao, J., & Liao, R. (2018). A novel model based on wavelet LS-SVM integrated improved PSO algorithm for forecasting of dissolved gas contents in power transformers. *Electric Power Systems Research*, 155, 196–205. doi:https://doi.org/10.1016/j.epr.2017.10.010