International Journal of Advanced Computer Research
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
Volume-5 Issue-18 March-2015

Comparative Study between FPA, BA, MCS, ABC, and PSO Algorithms in Training and Optimizing of LS-SVM for Stock Market Prediction

Osman Hegazy[1*], Omar S. Soliman[1] and Mustafa Abdul Salam[2]
Faculty of Computers and Information, Cairo University, Egypt[1]
Higher Technological Institute (H.T.I), Tenth of Ramadan city, Egypt[2]

Received: 16-February-2015; Revised: 22-March-2015; Accepted: 28-March-2015

Abstract

In this Paper, five recent natural inspired algorithms are proposed to optimize and train Least Square- Support Vector Machine (LS-SVM). These algorithms are namely, Flower Pollination Algorithm (FPA), Bat algorithm (BA), Modified Cuckoo Search (MCS), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO). These algorithms are proposed to automatically select best free parameters combination for LS-SVM. Six financial technical indicators derived from stock historical data are used as inputs to proposed models. Standard LS-SVM and ANN are used as benchmarks for comparison with proposed models. Proposed models tested with six datasets representing different sectors in S&P 500 stock market. Proposed models were used to predict daily, weekly, and monthly stock prices. Results presented in this paper showed that the proposed models have quick convergence rate at early stages of the iterations. They achieved better accuracy than compared methods in price and trend prediction. They also overcame over fitting and local minima problems found in ANN and standard LS-SVM.

Keywords

Least Square- Support Vector Machine ,Flower Pollination Algorithm, Bat algorithm, Modified Cuckoo Search, Artificial Bee Colony, Particle Swarm Optimization, and stock market prediction.

1. Introduction

Stock market prediction is the process of attempting to specify the future value of company stock based on its historical data. It has been at focus since the good prediction can maximize investor's gains. Prediction of stock market is not an easy task, because the nature of stock market data is variable, nonlinear, volatile, and close to random-walk. Also choosing a suitable training and prediction method is still very critical problem [1].

Technical indicators were from the first methods used to predict stock market trend and price [2].They are mathematical functions that use stock historical data to determine the future price. They are classified in two classes, oscillators or leading indicators, and lagging indicators [2]. Leading indicators are designed to lead price movements. The lagging indicators follow the price action and are referred to as trend-following indicators.

Artificial Neural Networks (ANNs) are considered one of the most commonly machine learning techniques used in stock market prediction. In most cases ANNs suffer from over-fitting problem due to the large number of parameters to fix, and the little prior user knowledge about the relevance of the inputs in the analysed problem [3].

Support vector machines (SVMs) have been developed as an alternative that avoid ANNs limitations. SVMs compute globally optimal solutions, unlike those obtained with ANNs, which tend to fall into local minima [4].

Least squares support vector machine (LS-SVM) method which is presented in [5], is a reformulation of the traditional SVM algorithm. Although LS-SVM simplifies the SVM procedure, the regularization parameter and the kernel parameters play an important role in the regression system. Therefore, it is necessary to establish a methodology for properly selecting the LS-SVM free parameters.

*Author for correspondence

The perceived advantages of evolutionary strategies as optimization methods motivated the authors to consider such stochastic methods in the context of optimizing SVMs. A survey and overview of evolutionary algorithms (EAs) is found in [6].
EAs or natural inspired algorithms used in this work are Flower Pollination Algorithm (FPA), Bat algorithm (BA), Cuckoo Search (CS), Artificial Bee Colony (ABC), and Particle Swarm Optimization (PSO).

Flower Pollination Algorithm (FPA) was proposed by Yang in 2012[7]. It is inspired by the pollination process of flowers [7]. The main purpose of a flower is ultimately reproduction via pollination. Flower pollination is typically associated with the transfer of pollen, and such transfer is often linked with pollinators such as insects, birds, bats and other animals [8]. In [9] FPA was applied successively for different economic load dispatch problems. In [10] FPA was applied in nonlinear algebraic systems with multiple solutions. In [11] binary FPA was applied to feature selection application. In [12] FPA algorithm was used for multiobjective optimization. In [13] Study of FPA algorithm for continuous optimization is presented. In [14] FPA algorithm with dimension improvement is introduced.

Bat algorithm (BA) was proposed by Yang in 2010 [15]. It is considered a new meta-heuristic algorithm for continuous optimization. BA is based on the fascinating capability of microbats (echolocation) to find their prey and discriminate different types of insects even in complete darkness. BA has demonstrated to outperform some well-known nature-inspired optimization techniques like GA, and PSO algorithms [15].BA is applied in continuous optimization in the context of engineering design optimization .BA can deal with highly nonlinear problem efficiently and can find the optimal solutions accurately [16].Case studies include pressure vessel design, car side design, spring and beam design, truss systems, tower and tall building design and others.. BA can handle multiobjective problems effectively [17]. In [18], a detailed study of combined economic load and emission dispatch problems using bat algorithm is presented. They concluded that bat algorithm is easy to implement and much superior to ABC and GA algorithms in terms of accuracy and efficiency. In [19] a comparison study of bat algorithm with PSO, GA, and other algorithms in the context for e-learning is presented. They concluded that bat algorithm has clearly some advantages over other algorithms. In [20] A New Bat Based Back-Propagation (BAT-BP) Algorithm is presented. They suggested that the computational efficiency of BPNN training process is highly enhanced when combined with BA algorithm.

Cuckoo Search (CS) algorithm is proposed by Yang and Deb in 2009 [21]. It is considered a nature-inspired meta-heuristic algorithm for continuous optimization [21]. CS is based on the brood parasitism of some cuckoo species. CS is enhanced by the Levy flights [22], rather than by simple isotropic random walks. CS algorithm was applied to engineering design applications; it has superior performance over other algorithms for a range of continuous optimization problems such as spring design and welded beam design problems [23, 24, and 25]. Zheng and Zhou [26] provided a variant of cuckoo search using Gaussian process.
Modified Cuckoo Search (MCS) algorithm is proposed by Walton proposed in 2011 [27]. MCS improved standard CS algorithm especially in terms of convergence to global minimum in real world applications.

Artificial Bee Colony (ABC) is proposed by D. Karaboga in 2005 for real parameter optimization [28]. It is inspired by the intelligent behaviour of honey bees. Karaboga and Basturk have investigated the performance of the ABC algorithm on unconstrained numerical optimization problems which is found in [29], [30], [31] and its extended version for the constrained optimization problems in [32]. Hybrid artificial bee colony-based approach for optimization of multi-pass turning operations is used in [33]. A study on Particle Swarm Optimization (PSO) and ABC algorithms for multilevel thresholding is introduced in [34]. ABC optimization was used for multi-area economic dispatch in [35].

Particle Swarm Optimization algorithm (PSO) is proposed by James Kennedy and Russell Eberhart in 1995 [36]. PSO is one of the most used EAs. It is motivated by social behaviour of organisms such as bird flocking and fish schooling [36]. The PSO algorithm, while making adjustment towards "local" and "global" best particles, is similar to the crossover operation used by genetic algorithms [37]. In [38], [39] authors proved that SVM optimized by PSO gave better accuracy than standard SVM model. Jui Y. et al. [40] concluded that hybrid CI approaches

AIA-BPNN and ASA-BPNN are better than single CI technique BPNN-SCG and recommended for stock price.

This paper proposes five hybrid models. These are FPA-LS-SVM, BA-LS-SVM, MCS-LS-SVM, ABC-LS-SVM, and PSO-LS-SVM models. These models are hybridizing optimization algorithms (FPA, BA, MCS, ABC, and PSO) respectively, financial technical indicators, and LS-SVM model. The performance of LS-SVM is based on the selection of hyper parameters C (cost penalty), ϵ (insensitive-loss function) and γ (kernel parameter). Optimization algorithms were used to find the best parameter combination for LS-SVM.

The rest of paper is organized as follows: Section 2 presents the Least square support vector machine (LS-SVM) model; Section 3 presents the FPA algorithm; Section 4 presents the BA algorithm; Section 5 presents the MCS algorithm; Section 6 presents the ABC algorithm; Section 7 presents the PSO algorithm; Section 8 is devoted for the proposed models and its implementation in daily, weekly, and monthly stock price prediction; In Section 9 the results are discussed. The main conclusions of the work are presented in Section 10.

2. Least Square Support Vector Machine (LS-SVM)

Least squares support vector machines (LS-SVM) are least squares versions of support vector machines (SVM), which are a set of related supervised learning methods that analyse data and recognize patterns, and which are used for classification and regression analysis. In this version one finds the solution by solving a set of linear equations instead of a convex quadratic programming (QP) problem for classical SVMs. LS-SVM classifiers, were proposed by Suykens and Vandewalle [41]. Let X is input data matrix and is output vector. Given the training data set, where, the LS-SVM goal is to construct the function, which represents the dependence of the output on the input. This function is formulated as

$$f(x) = W^T \varphi(x) + b \qquad (1)$$

Where W and $\varphi(x)$: $R^p \to R^n$ are $n \times 1$ column vectors, and $b \in R$. LS-SVM algorithm computes the function (1) from a similar minimization problem found in the SVM method [4]. However the main difference is that LS-SVM involves equality constraints instead of inequalities, and it is based on a least square cost function. Furthermore, the LS-SVM method solves a linear problem while conventional SVM solves a quadratic one. The optimization problem and the equality constraints of LS-SVM are defined as follows:

$$\min_{w,e,b} j(w,e,b) = \frac{1}{2} w^T w + C \frac{1}{2} e^T \qquad (2)$$

$$y_i = w^T \varphi(x_i) + b + e_i \qquad (3)$$

Where e is the $n \times 1$ error vector, 1 is a $n \times 1$ vector with all entries 1, and $C \in R^+$ is the tradeoff parameter between the solution size and training errors. From Eq. (2) a Lagranian is formed, and differentiating with respect to w, b, e, a (a is Largrangian multipliers), we obtain

$$\begin{bmatrix} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -1^T \\ 0 & 0 & CI & -I \\ Z & 1 & I & 0 \end{bmatrix} \begin{bmatrix} W \\ b \\ e \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ y \end{bmatrix} \qquad (4)$$

Where I represents the identity matrix and

$$Z = [\varphi(x1), \varphi(x2), \ldots, \varphi(x_n)]^T$$

From rows one and three in Eq. (4) $w = Z^T a$ and $Ce = a$

Then, by defining the kernel matrix $K = ZZ^T$, and the parameter $\lambda = C^{-1}$, the conditions for optimality lead to the following overall solution

$$\begin{bmatrix} 0 & 1^T \\ 1 & K + \lambda I \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \qquad (5)$$

Kernel function K types are as follows:

- Linear kernel
  $$K(x, x_i) = x_i^T x \qquad (6)$$
- Polynomial kernel of degree d:
  $$K(x, x_i) = (1 + x_i^T x / c)^d \qquad (7)$$
- Radial basis function RBF kernel :
  $$K(x, x_i) = \exp(-\|x - x_i\|^2 / \sigma^2) \qquad (8)$$

- MLP kernel :
  $$K(x, x_i) = \tanh(k x_i^T x + \theta) \quad (9)$$

3. Flower Pollination Algorithm (FPA)

Flower Pollination Algorithm (FPA) is a novel algorithm inspired by the pollination process of flowers [7]. Pollination can be achieved by self-pollination or cross-pollination. Cross-pollination, or allogamy, means pollination can occur from pollen of a flower of different plant, while self-pollination is the fertilization of one flower, such as peach flowers, from pollen of the same flower or different flowers of the same plant, which often occurs when there is no reliable pollinator available. Biotic, cross-pollination may occur at long distance, and the pollinators such as bees, bats, birds and flies can fly a long distance, thus they can considered as the global pollination. In addition, bees and birds may behave as Levy flight behavior [42], with jump or fly distance steps obey a Levy distribution. Furthermore, flower constancy can be used an increment step using the similarity or difference of two flowers.
The characteristics of pollination process, flower constancy and pollinator behavior, were idealized in the following rules [7]:

1. Biotic and cross-pollination is considered as global pollination process with pollen carrying pollinators performing Levy flights.
2. Abiotic and self-pollination are considered as local pollination.
3. Flower constancy can be considered as the reproduction probability is proportional to the similarity of two flowers involved.
4. Local pollination and global pollination is controlled by a switch probability$p \in [0, 1]$.

Due to the physical proximity and other factors such as wind, local pollination can have a significant fraction p in the overall pollination activities. In the global pollination step, flower pollens are carried by pollinators such as insects, and pollens can travel over a long distance. This ensures the pollination and reproduction of the fittest, and thus we represent the fittest as g_*. The first rule can be formulated as bee is updated using the Eq. (10):

$$X_i^{t+1} = X_i^t + L(X_i^t - g*) \quad (10)$$

Where X_i^t is the solution vector i at iteration t and $g*$ is the current best solution. The parameter L is the strength of the pollination which is a step size. Since insects may move over a long distance with various distance steps, a Levy flight can be used to mimic this characteristic efficiently [17]. L > 0 will be drawn from a Levy distribution as shown in Eq. (11)

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0) \quad (11)$$

Here$\Gamma(\lambda)$ is the standard gamma function, and this distribution is valid for large steps s > 0.
The local pollination (Rule 2) and flower constancy can be represented as in Eq. (12):

$$X_i^{t+1} = X_i^t + \varepsilon\left(X_j^t - g_k^t\right) \quad (12)$$

Where X_i^t and X_j^t are solution vectors drawn randomly from the solution set. The parameter ε is drawn from uniform distribution in the range from 0 to 1.

4. Bat Algorithm (BA)

Bat Algorithms (BA), which is a new nature inspired algorithm for continuous optimization is proposed by Yang in 2010 [15]. Yang developed the bat algorithm with the following three idealized rules:

a. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way.
b. Bats fly randomly with velocity v_i at position x_i with a frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission$r \in [0,1]$, depending on the proximity of their target.
c. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant valueA_{min}.

First, the initial position x_i, velocity v_i and frequency f_i are initialized for each bat b_i. For each time step t, the movement of the virtual bats is given by updating their velocity and position using Eq. (13), Eq. (14) and Eq. (15) respectively, as follows:

$$f_i = f_{min} + (f_{max} + f_{min})\beta, \qquad (13)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} + x*)f_i, \qquad (14)$$

$$x_i^t = x_i^{t-1} + v_i^t, \qquad (15)$$

Where β denotes a randomly generated number within the interval [0, 1]. Recall that x_i^tdenotes the value of decision variable j for bat i at time step t .The result of f_i in Eq. (13) is used to control the pace and range of the movement of the bats. The variable $x*$ represents the current global best location (solution) which is located after comparing all the solutions among all the n bats. In order to improve the variability of the possible solutions, Yang [15] has employed random walks. Primarily, one solution is selected among the current best solutions for local search and then the random walk is applied in order to generate a new solution for each bat;

$$x_{new} = x_{old} + \varepsilon A_t \qquad (16)$$

Where, A_t stands for the average loudness of all the bats at time t, and $\varepsilon \in [-1,1]$ is a random number. For each iteration of the algorithm, the loudness A_i and the emission pulse rate r_i are updated, as follows:

$$A_i^{t+1} = \alpha A_i^t \qquad (17)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t) \qquad (18)$$

Where α and γ are constants. At the first step of the algorithm, the emission rate, $\mathrm{r_i^0}$ and the loudness, $\mathrm{A_i^0}$ are often randomly chosen. Generally, $\mathrm{A_i^0} \in [1,2]$ and $\mathrm{r_i^0} \in [0,1]$.

5. Modified Cuckoo Search (MCS)

The cuckoo search algorithm (CS) can easily find the optimum [16] but, as the search relies entirely on random walks, a fast convergence cannot be guaranteed. Modified Cuckoo Search (MCS) made two modifications to the original CS algorithm to increase the convergence rate.

The first modification is made to the size of the Lévy flight step size α. In CS, α is constant and the value α = 1 is employed [21]. In the MCS, the value of α decreases as the number of generations increases. This is done for the same reasons that the inertia constant is reduced in the PSO, i.e. to encourage more localized searching as the individuals, or the eggs, get closer to the solution. An initial value of the Lévy flight step size A = 1 is chosen and, at each generation, a new Lévy flight step is calculated using α=A/√G, where G is the generation number. This exploratory search is only performed on the fraction of nests to be abandoned.

The second modification is to add information exchange between the eggs to speed up convergence to a minimum. In the CS, there is no information exchange between individuals and, essentially, the searches are performed independently.

6. Artificial Bee Colony (ABC)

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga in 2005 for real parameter optimization [28]. It is inspired by the intelligent behavior of honey bees. The colony of artificial bees consists of three groups of bees: employed, onlooker and scout bees. Half of the colony composed of employed bees and the rest consist of the onlooker bees. The number of food sources/nectar sources is equal with the employed bees, which means one nectar source is responsible for one employed bee. The aim of the whole colony is to maximize the nectar amount. The duty of employed bees is to search for food sources (solutions). Later, the nectars' amount (solutions' qualities/fitness value) is calculated. Then, the information obtained is shared with the onlooker bees which are waiting in the hive. The onlooker bees decide to exploit a nectar source depending on the information shared by the employed bees. The onlooker bees also determine the source to be abandoned and allocate its employed bee as scout bees. For the scout bees, their task is to find the new valuable food sources. They search the space near the hive randomly [43]. In ABC algorithm, suppose the solution space of the problem is D-dimensional, where D is the number of parameters to be optimized.

The fitness value of the randomly chosen site is formulated as follows:

$$\mathrm{fit_i} = \frac{1}{(1+\mathrm{obj.fun_i})} \qquad (19)$$

The size of employed bees and onlooker bees are both SN, which is equal to the number of food sources. There is only one employed bee for each food source whose first position is randomly generated. In each iteration of ABC algorithm, each employed bee determines a new neighbouring food source of its currently associated food source and computes the nectar amount of this new food source by

$$v_{ij} = x_{ij} + \theta(x_{ij} - x_{kj}) \qquad (20)$$

Where;
$i = 1,2, \ldots, SN,$
$j = 1,2, \ldots, D,$
$\theta =$ random number in range [0,1].

If the new food source is better than that of previous one, then this employed bee moves to new food source, otherwise it continues with the old one.

After all employed bees complete the search process; they share the information about their food sources with onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount by Equation:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \qquad (21)$$

Where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees.

Later, the onlooker bee searches a new solution in the selected food source site, the same way as exploited by employed bees. After all the employed bees exploit a new solution and the onlooker bees are allocated a food source, if a source is found that the fitness hasn't been improved for a predetermined number of cycles (limit parameter), it is abandoned, and the employed bee associated with that source becomes a scout bee. In that position, scout generates randomly a new solution by:

$$x_i^j = x_{min}^j + r(x_{max}^j - x_{min}^j) \qquad (22)$$

Where;
r is random number in range [0 , 1].

x_{min}^j , x_{max}^j are the lower and upper borders in the jth dimension of the problem space.

7. Particle Swarm Optimization Algorithm (PSO)

PSO is a heuristic search method which is derived from the behavior of social groups like bird flocks or fish swarms [36]. PSO moves from a set of points to another set of points in a single iteration with likely improvement using a combination of deterministic and probabilistic rules. The PSO has been popular because of its ease of implementation, and the ability to effectively solve highly nonlinear, mixed integer optimization problems that are typical of complex engineering systems. Optimization is achieved by giving each individual in the search space a memory for its previous successes, information about successes of a social group and providing a way to incorporate this knowledge into the movement of the individual [36].

Therefore, each individual (called particle) is characterized by its position $\vec{x}_i$, its velocity $\vec{v}_i$, its personal best position $\vec{p}_i$ and its neighborhood best position $\vec{p}_g$.

The elements of the velocity vector for particle i are updated as

$$\upsilon_{ij} \leftarrow \omega\upsilon_{ij} + c_1 q(x_{ij}^{pb} - x_{ij}) + c_2\gamma(x_j^{sb} - x_{ij}) \qquad (23)$$

Where $j = 1,..,n$, w is the inertia weight, x_i^{pb} is the best variable vector encountered so far by particle i, and x^{sb} is the swarm best vector, i.e. the best variable vector found by any particle in the swarm, so far c_1 and c_2 are constants, and q and r are random numbers in the range (0, 1).

Once the velocities have been updated, the variable vector of particle i is modified according to

$$x_{ij} \leftarrow x_{ij} + \upsilon_{ij}. \qquad (24)$$

The cycle of evaluation followed by updates of velocities and positions (and possible update of x_i^{pb} and x^{sb}) is then repeated until a satisfactory solution has been found.

8. The proposed models

The proposed models are based on the study and gathering of stock historical data (High, Low, Open, Close, and Volume) of S&P 500 stock market. The gathered data are sampled into three categories (daily, weekly, and monthly). Then financial technical indicators are extracted and calculated from the collected historical data. These technical are namely RSI, MFI, MACD, EMA, PMO, and SO. The selected indicators are from all types of indicators to make the system more robust and more accurate.

Indicators are used as inputs to proposed model. After extracting the indicators features, five models were constructed by optimizing and trained LS-SVM with five different optimization algorithms (FPA, BA, MCS, ABC, and PSO). These models are used in the prediction of daily, weekly, and monthly stock prices. Standard LS-SVM, and ANN are used as benchmarks for comparison with proposed model.

The proposed models are evaluated by four different criterions. Evaluation criterions are RMSE, MAE, SMAPE, and PMRE. The proposed models architecture contains seven inputs vectors represent the historical data and six derived technical indicators from raw datasets, and one output represents next price. The proposed models are summarized in Fig. 1.

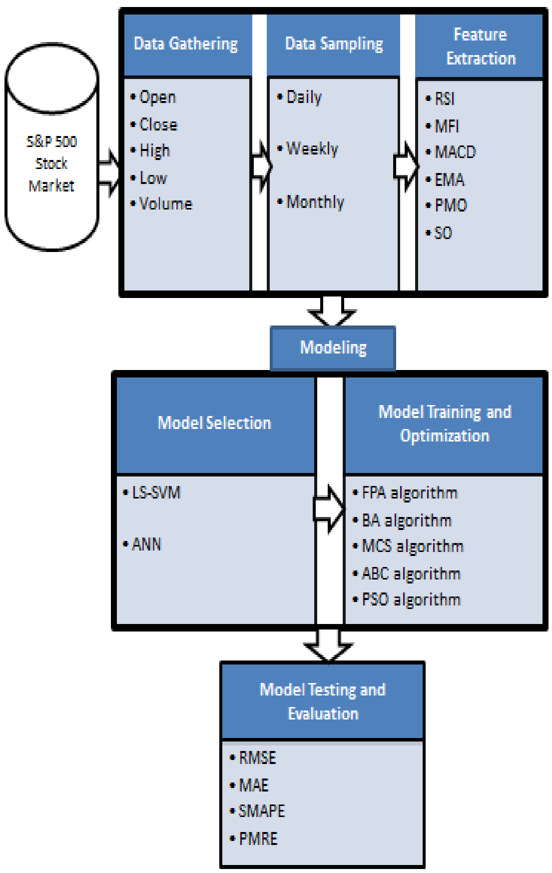

Fig.1: The proposed models steps

The financial technical indicators, which are calculated from the raw datasets, are calculated as follows:

- **Price Momentum Oscillator (PMO)**

PMO is an oscillator based on a Rate of Change (ROC) calculation that is exponentially smoothed twice. Because the PMO is normalized, it can also be used as a relative strength tool. Stocks can thus be ranked by their PMO value as an expression of relative strength.
TC = today's close price
TDAC = close price ten days ago
The following was used to calculate PMO:

$$PMO = TC - TDAC \quad (25)$$

- **Relative Strength Index (RSI)**

A technical momentum indicator that compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset. The formula for computing the Relative Strength Index is as follows.

$$RSI = 100 - [100 / (1 + RS)] \quad (26)$$

Where RS = Avg. of x days' up closes divided by average of x days' down closes.

- **Money Flow Index (MFI)**

This one measures the strength of money in and out of a security. The formula for MFI is as follows.

$$\text{Money Flow (MF)} = TP * V \quad (27)$$

Where, TP is typical price, and V is money Vol.
Money Ratio (MR) is calculated as:

$$MR = (\text{Positive MF} / \text{Negative MF}) \quad (28)$$

$$MFI = 100 - (100/ (1 + MR)) \quad (29)$$

- **Exponential Moving Average (EMA)**

This indicator returns the exponential moving average of a field over a given period of time. EMA formula is as follows.

$$EMA = [\alpha * T\,Close] + [1 - \alpha * Y\,EMA] \quad (30)$$

Where T is Today's close and Y is Yesterday's close.

- **Stochastic Oscillator (SO)**

The stochastic oscillator defined as a measure of the difference between the current closing price of a security and its lowest low price, relative to its highest high price for a given period of time. The formula for this computation is as follows.

$$\%K = [(CP - LP) / (HP - LP)] * 100 \quad (31)$$

Where, CP is Close price, LP is Lowest price, HP is Highest Price, and LP is Lowest Price.

- **MovingAverage Convergence/Divergence (MACD)**

This function calculates difference between a short and a long term moving average for a field. The formulas for calculating MACD and its signal as follows.

$$\mathrm{MACD} = [0.075 * \mathrm{E}] - [0.15 * \mathrm{E}] \quad (32)$$

Where, E is EMA (CP)

$$\text{Signal Line} = 0.2 * \text{EMA of MACD} \quad (33)$$

9. Results and Discussions

In this section FPA, MCS, BA, ABC, and PSO algorithms are compared in optimizing LS-SVM. The number of flowers, nests, bats, bees, and particles are 50 and trained for 100epochs. These models were trained and tested with daily datasets for six companies cover different sectors in S&P 500 stock markets. These companies are Adobe, Bank of America (BAC), Exxon mobile, General Electric (GE), Pepsi, and Pfizer. Datasets periods are as follows which available in [44]. Simulation results are done using matlab 2012b. Daily datasets period are from Feb. 2011 to Feb. 2014. Weekly datasets period are from Feb. 2004 to Feb. 2014. Monthly datasets period are from Feb. 2000 to Feb. 2014. These datasets are divided into training part (70%) and testing part (30%). In table 1 the RMSE of daily stock market prediction is summarized for all compared algorithms. From table one can notice that FPA algorithm achieved lowest sum of RMSE value with little advance over BA, ABC, MCS, and PSO algorithms. In table 2 the RMSE of weekly stock market prediction for all compared models is presented. FPA algorithm also achieved lowest sum of RMSE. In table 3 the RMSE of monthly stock market prediction is outlined. BA algorithm achieved lowest sum of RMSE value with little advance over FPA, ABC, MCS, and PSO algorithms.

Table 1: RMSE of daily results

RMSE	FPA	BA	ABC	MCS	PSO	SVM	ANN
Adobe	0.826133	0.826224	**0.825102**	0.838449	0.826444	9.037474	10.396
BAC	**0.903589**	0.903669	0.903957	1.002392	0.903613	5.910966	7.163308
Exxon	0.754428	0.754315	0.754324	**0.752006**	0.754432	1.567004	1.627606
GE	**0.224023**	0.22499	0.224346	0.224243	0.225191	0.757544	0.854772
Pepsi	0.616275	**0.615191**	0.616553	1.279864	0.61653	1.216737	1.320679
Pfizer	**0.278108**	0.2782	0.278299	0.332281	0.278305	0.583551	0.825387
Sum of RMSE	**3.602556**	3.602589	3.602581	4.429235	3.604515	19.07328	22.18775

Table 2: RMSE of weekly results

RMSE	FPA	BA	ABC	MCS	PSO	SVM	ANN
Adobe	**1.179148**	1.179282	1.179255	1.262344	1.179158	1.430817	2.146679
BAC	**0.737473**	0.737553	0.737491	0.748036	0.737545	2.302298	4.105002
Exxon	1.784019	1.784304	1.784122	**1.778899**	1.784019	2.098448	2.182685
GE	0.50962	0.510526	0.509886	**0.507277**	0.50962	0.665406	1.251016
Pepsi	1.093282	1.09303	**1.09267**	1.249822	1.093204	2.554887	2.158931
Pfizer	**0.50771**	0.507847	0.508029	0.516966	0.50772	0.712614	0.746991
Sum of RMSE	**5.811252**	5.812542	5.811453	6.063344	5.811266	9.76447	12.591304

Table 3: RMSE of monthly results

RMSE	FPA	BA	ABC	MCS	PSO	SVM	ANN
Adobe	4.140293	4.149244	4.149244	**4.04825**	4.14021	5.40894	6.192689
BAC	2.064775	**1.65135**	1.651355	4.160996	2.058583	3.654201	8.925551
Exxon	**3.570947**	3.571598	3.571598	3.593616	3.570951	4.200242	8.993002
GE	1.259394	1.261313	1.261313	1.500346	**1.259376**	1.473693	3.755476
Pepsi	2.100864	2.10393	2.10398	**2.062812**	2.100835	2.651189	8.99892
pfizer	1.07678	1.060424	**1.060422**	2.216248	1.066994	2.180261	3.193931
Sum of RMSE	14.21305	**13.79786**	13.79791	17.58227	14.19695	19.568526	40.059569

10. Conclusions

In this paper five bio inspired algorithms were proposed to train and optimize LS-SVM model. From results we can notice that standard LS-SVM and ANN models cannot overcome the overfitting problem and have slow convergence speed. Also these algorithms still suffer from falling in local minima problem. Using natural inspired algorithms or global search techniques help in overcoming the problems of traditional learning algorithms. FPA, ABC, BA, PSO, MCS algorithms in most cases convergences to a global minimum while traditional LS-SVM and ANN models failed. Proposed hybrid FPA-LS-SVM method achieved lowest error values (RMSE) in daily, weekly, and weekly stock price prediction. All proposed models (FPA-LS-SVM, BA-LS-SVM, ABC-LS-SVM, MCS-LS-SVM, and PSO-LS-SVM) could easily manipulate the fluctuation of stock time series while LS-SVM and ANN models failed to cope with these fluctuations. Convergence speed of proposed models to global minimum is very fast compared to classical methods.

References

[1] C. Olivier, Blaise Pascal University: “Neural network modeling for stock movement prediction, state of art”. 2007.
[2] P.Fernández-Blanco1, D.Bodas-Sagi1, F.Soltero1, J.I.Hidalgo1, Technical market indicators optimization using evolutionary algorithms, Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation, ACM New York, NY, USA ,2008, pp. 1851-1858.
[3] X. Leng, and H. Miller, Input dimension reduction for load forecasting based on support vector machines, IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies (DRPT2004), 2004, pp.510 - 514.
[4] V. Cherkassky, and Y. Ma, Practical Selection of SVM Parameters and Noise Estimation for SVM regression, Neural Networks, 17(1), pp.113-126, 2004.
[5] J. Suykens, V. Gestel, and J. Brabanter, Least squares support vector machines, World Scientific, 2002.
[6] A. Carlos , B. Gary , and A. David, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007.
[7] X.S. Yang, Flower pollination algorithm for global optimization: Unconventional Computation and Natural Computation 2012, Lecture Notes in Computer Science, Vol. 7445, pp. 240-249 (2012).
[8] B. J. Glover, Understanding Flowers and Flowering: An Integrated Approach, Oxford University Press, (2007).
[9] R. Prathiba , M. Balasingh Moses , S. Sakthivel, Flower Pollination Algorithm Applied for Different Economic Load Dispatch Problems, International Journal of Engineering and Technology (IJET), Vol 6 No 2 Apr-May 2014, pp. 1009-1016.
[10] G.M. platt, Application of Flower Pollination Algorithm in nonlinear algebraic systems with multiple solutions, Engineering Optimization IV.
[11] D Rodrigues, X.S. Yang, A.N. de Souza, J.P. Papa, Binary Flower Pollination Algorithm and Its Application to Feature Selection , Springer, Recent Advances in Swarm Intelligence and Evolutionary Computation Studies in Computational Intelligence Volume 585, 2015, pp 85-100.
[12] X.S Yang, M. Karamanoglu , X. He , Flower pollination algorithm: A novel approach for multiobjective optimization , Engineering Optimization, Volume 46, Issue 9, 2014.
[13] S. Łukasik, P. Kowalski, Study of Flower Pollination Algorithm for Continuous Optimization, Intelligent Systems'2014, Advances in Intelligent Systems and Computing Volume 322, 2015, pp 451-459.
[14] R. Wang and Y.Zhou, “Flower Pollination Algorithm with Dimension by Dimension Improvement,” Mathematical Problems in Engineering, vol. 2014, Article ID 481791, 9 pages, 2014. doi:10.1155/2014/481791.
[15] X. S., Yang, A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. Cruz, C.; Gonzalez, J. R.; Pelta, D. A.; Terrazas, G), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65–74, 2010.
[16] X. S., Yang and A. H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, Engineering Computations, Vol. 29, No. 5, pp.464–483, 2012.
[17] X. S., Yang, Bat algorithm for multi-objective optimisation, Int. J. Bio-Inspired Computation, Vol. 3, No. 5, pp. 267–274, 2011.
[18] B., Ramesh, V. C. J., Mohan, V. C. V., Reddy, Application of bat algorithm for combined economic load and emission dispatch, Int. J. of Electrical Engineering and Telecommunications, Vol. 2, No. 1, pp. 1–9,2013.
[19] K., Khan, A. Sahai, A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context, Int. J. Intelligent Systems and Applications (IJISA), Vol. 4, No. 7, pp. 23–29, 2012.

[20] N.M. Nawi, M. Z. Rehman, A. Khan, A New Bat Based Back-Propagation (BAT-BP) Algorithm, Springer, Advances in Intelligent Systems and Computing Vol. 240, pp. 395-404, 2014.
[21] X. S., Yang, S.Deb, Cuckoo search via Lévy flights. Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009, India), IEEE Publications, USA 2009; p. 210–214.
[22] I.Pavlyukevich, Le´vy flights, non-local search and simulated annealing. J Comput Phys, 2007, 226:1830–1844.
[23] A.H. Gandomi, X.S. Yang, A.H. Alavi (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35. doi:10.1007/s00366-011-0241-y.
[24] A.H. Gandomi, X.S. Yang, S. Talatahari, S. Deb (2012) Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. Comput Math Appl 63(1):191–200.
[25] X.S. Yang, S. Deb (2010) Engineering optimization by cuckoo search. Int J Math Modell Num Opt 1(4):330–343.
[26] H.Q. Zheng, Y Zhou (2012) A novel cuckoo search optimization algorithm based on Gauss distribution. J Comput Inf Syst 8: 4193–4200.
[27] S.Walton, O.Hassan, K.Morgan and M.R.Brown "Modified cuckoo search: A new gradient free optimisation algorithm" Chaos, Solitons & Fractals Vol 44 Issue 9, Sept 2011 pp. 710-718 DOI:10.1016/j.chaos.2011.06.004.
[28] D. Karaboga, An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[29] B. Basturk, and D. Karaboga, An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization, IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, 2006.
[30] D. Karaboga, and B. Basturk, A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, Journal of Global Optimization, 39(3), 2007, 459–471.
[31] D. Karaboga, and B. Basturk, On The Performance Of Artificial Bee Colony (ABC) Algorithm, Applied SoftComputing journal, 8(1), 2008, 687–697.
[32] D. Karaboga, and B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, Advances in SoftComputing: Foundations of Fuzzy Logic and Soft Computing, 4529, 2007, 789–798.
[33] A.R. Yildiz, Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach, Information Sciences: an International Journal, 220, 2013, 399-407.
[34] A. Gupta "Stock market prediction using Hidden Markov Models", IEEE Engineering and Systems (SCES), 2012 Students Conference on, pp.1-4, 2012.
[35] M. Basu, Artificial bee colony optimization for multi-area economic dispatch, International Journal of Electrical Power & Energy Systems, (49), 2013, 181–187.
[36] D. N. Wilke. "Analysis of the particle swarm optimization algorithm", Master's Dissertation, University of Pretoria, 2005.
[37] A.S. Khalil "An Investigation into Optimization Strategies of Genetic Algorithms and Swarm Intelligence". Artificial Life (2001).
[38] Z. Guo, H. Wang, Q. Liu: "Financial time series forecasting using LPP and SVM optimized by PSO" © Springer, Soft Computing Methodologies and Applications, December 2012.
[39] G. Xie: "The Optimization of Share Price Prediction Model Based on Support Vector Machine", International Conference on Control, Automation and Systems Engineering (CASE), pp.1-4., 30-31 July 2011.
[40] Y. Jui: "Computational Intelligence Approaches for Stock Price Forecasting", IEEE International Symposium on Computer, Consumer and Control (IS3C), pp. 52 - 55, 2012.
[41] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters, Vol. 9 (3), 293-300, 1999.
[42] I. Pavlyukevich, Levy flights, non-local search and simulated annealing, J. Computational Physics, 226, 1830-1844 (2007).
[43] P. Mansouri, B. Asady, and N. Gupta, A Novel Iteration Method for solve Hard Problems (Nonlinear Equations) with Artificial Bee Colony Algorithm, World Academy of Science, Engineering and Technology, 5(11), 2011, 389 - 392.
[44] http://finance.yahoo.com, last accepted 11 Jan. 2015.

Osman Hegazy received his Ph.D. in Computer Engineering in 1977 from Leicester University, UK. Currently, he is Professor at the Department of Information Systems, Faculty of Computers and Information, Cairo University, Egypt. He has more than 100+ publications in the areas of Data Mining, Data Engineering, Cloud Computing, and Computational Intelligence in international journals and international conferences. He has supervised over 50

M.Sc.'s and over 30 Ph.D. in computer science and information systems.
Email: Osman.hegazy@gmail.com

Omar Soliman received his PhD in Operations Research department, Faculty of Computers and Information, Cairo University, Egypt in 2001. He has more than 50+ publications in the areas of Computational Intelligence and Optimization in international journals and international conferences.

Mustafa Abdul Salam received the B.S from faculty of Computers and Information, Zagazig University, Egypt in 2003, and obtained M.Sc.'s degree in information system from faculty of Computers and Information, Menufiya University, Egypt in 2009. He is Ph.D. student at faculty of Computers and Information, Cairo University. He is teaching assistant at information system department, Higher Technological Institute. He has contributed more than 10+ technical papers in the areas of Data Mining, Computational Intelligence, and Natural-Inspired algorithms in international journals and international conferences.