ARTICLE IN PRESS

Swarm and Evolutionary Computation xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms

Vadlamani Ravi[a,*], Dadabada Pradeepkumar[a,b], Kalyanmoy Deb[c]

[a] Center of Excellence in Analytics, Institute for Development and Research in Banking Technology (IDRBT), Castle Hills, Masab Tank, Hyderabad 500057, India
[b] School of Computer and Information Sciences (SCIS), University of Hyderabad, Hyderabad 500046, India
[c] Koenig Endowed Chair Professor, Electrical and Computer Engineering, Michigan State University, East Lansing, MI 4882, USA

ARTICLE INFO

Keywords:
Financial time series prediction
Multi-objective optimization
Chaos theory
MLP
MOPSO
NSGA-II

ABSTRACT

Financial Time Series Prediction is a complex and a challenging problem. In this paper, we propose two 3-stage hybrid prediction models wherein Chaos theory is used to construct phase space (Stage-1) followed by invoking Multi-Layer Perceptron (MLP) (Stage-2) and Multi-Objective Particle Swarm Optimization (MOPSO) / elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) (Stage-3) in tandem. In both of these hybrid models, Stage-3 improves the prediction yielded by stage-2. The effectiveness of the proposed models is tested on financial datasets including the exchange rates data of US Dollar (USD) versus Japanese Yen (JPY), British Pound (GBP), Euro (EUR), and Gold price in terms of USD. From the results, it is concluded that Chaos+MLP+NSGA-II hybrid yielded better predictions than the other three-stage hybrid models: Chaos+MLP+MOPSO and Chaos+MLP+PSO, and Two-stage hybrid models: Chaos+PSO, Chaos+MOPSO and Chaos+NSGA-II in terms of both Mean Squared Error (MSE) and Directional Change Statistic (Dstat). Theil's inequality coefficient computed also confirms the superiority of the Chaos+MLP+NSGA-II hybrid over the Chaos+MLP+MOPSO across all datasets. Finally, Diebold-Mariano test indicates that the performance of Chaos+MLP+NSGA-II hybrid is statistically significant than the Chaos+MLP+MOPSO and other hybrids across all datasets. The results of these models are also compared with the two-stage hybrids found in literature [1,2] (Pradeepkumar and Ravi, 2014, 2017). These results are encouraging and suggest further application of these hybrids to other financial and scientific time series prediction problems in the future.

1. Introduction

A Time series is a set of chronologically recorded observations of one variable (Univariate) or multiple variables (Multivariate). In general, the Time series data is continuous, substantial in its size and high dimensional [3]. *Time series prediction* is a popular Time series data mining task [4]. In univariate time series forecasting, the problem is to predict the value of a variable at discrete times. It involves predicting the future given the past and present values. Note that the terms *prediction* and *forecasting* will be used interchangeably in the paper. Mista [5] describes various diverse applications of prediction found in Finance (e.g., exchange rate prediction, gold price prediction, stock price prediction, demand forecasting and volatility forecasting), Medicine (e.g., disease prediction, heart diagnosis prediction) and Business (e.g., demand forecasting, monthly sales forecasting).

Financial time series is a collection of chronologically recorded observations of the financial variable(s). For example, the daily exchange rate of a currency pair is a univariate financial time series. The financial time series are non-stationary and chaotic [6]. A time series is said to be chaotic if and only if it is nonlinear, deterministic and sensitive to initial conditions [7]. The prediction of a chaotic time series engages with the prediction of future behavior of the chaotic system by utilizing the current and past states of that system.

In addition to these, the Financial Time Series Prediction is a highly complicated task as a typical financial time series exhibits the following characteristics:

1. Financial time series often behave nearly like a random-walk process, making the prediction impossible (from a theoretical point of view) [8].
2. Financial time series are usually very noisy and chaotic [9,6].
3. Statistical properties of the financial time series are different at different points in time as the process is time-varying [8].

* Corresponding author.
E-mail addresses: rav_padma@yahoo.com (V. Ravi), dpradeepphd@gmail.com (D. Pradeepkumar), kdeb@egr.msu.edu (K. Deb).

http://dx.doi.org/10.1016/j.swevo.2017.05.003
Received 31 January 2017; Received in revised form 10 April 2017; Accepted 13 May 2017
2210-6502/

The challenging task in Financial Time Series Prediction is building the right forecasting model that captures the subtle and obvious changes in the data. Literature abounds with various traditional statistical methods such as Moving Averages, Exponential Smoothing, Auto-regressive Integrated Moving Averages (ARIMA) [10–12], etc. for financial time series prediction. While these methods are statistically powerful, they failed to yield accurate predictions on the test data. Later, various computational intelligent techniques such as Artificial Neural Networks, Fuzzy Systems, Swarm Intelligence based model, etc. have been proposed for time series prediction. Often, they tended to yield more accurate predictions. However, these are not without demerits [5]. In this context, the hybrid forecasting models (in short, hybrids) in the paradigm of Soft Computing, which combine different prediction models in various permutations and combinations found traction by demonstrably yielding superior predictions compared to their stand-alone counterparts.

Several researchers demonstrated that hybrid or ensemble models do yield better results compared to stand-alone models. Reid [13] and Bates and Granger [14] laid the foundation for proposing various hybrid time series models. Bates and Granger [14] concluded that suitably combining different forecasting methods can yield better predictions than the stand-alone methods. Similarly, Makridakis et al. [15] reported that a hybrid or an ensemble of several models is commonly needed to improve forecasting accuracy. Pelikan et al. [16], and Ginzburg and Horn [17] reported that the combination of several ANNs improved time series forecasting accuracy. An excellent comprehensive review of various hybrid prediction models and annotated bibliography can be found in [18]. Usually, a good hybrid prediction model can:

1. Improve the forecasting performance.
2. Overcome deficiencies of stand-alone models.
3. Reduce the model uncertainty [19].

Uncertainty and nonlinearity in a time series need not be caused by randomness alone always. It can be due to deterministic phenomena, labeled as chaos, which is highly sensitive to initial conditions. A financial time series can be affected by economical, social, industrial and geo-political factors. It is uncertain, noisy and incomplete. Despite this, the prediction of financial time series has tremendous practical potential in terms of huge financial gains. Visually, a chaotic and non-chaotic time series look alike and hence, traditionally, chaotic time series such as financial time series have been modeled to account for the inherent randomness. Therefore, prediction of financial time series, being complex and important, it calls for the development of more sophisticated and powerful hybrid techniques.

Chaos theory, pioneered by Poincare [20,21] offers a new way to model the underlying nonlinear dynamic behavior of a deterministic complex system by embedding a given scalar financial time series in its corresponding phase space using parameters such as lag and embedding dimension, where lag is the time delay and embedding dimension signifies the number of variables needed to represent the nonlinear dynamics of the chaotic system.

In general, Artificial Neural Networks (ANNs) including Multi-Layer Perceptron (MLP) can capture complex nonlinear relationships very well. They can generalize well and are good universal approximators [22]. However, in MLP, convergence is slow, local minima can affect the training process, and it is hard to scale. Multi-objective evolutionary algorithms (MOEAs) including Multi-objective Particle Swarm Optimization (MOPSO) and elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) are good at solving Multi-objective optimization problems [23]. NSGA-II improves the nondominated sorting algorithm and reduces the computational complexity. It sorts the combination of parents and children population with elitist strategy, introduces the crowded comparison operator to improve diversity of solutions, and avoids the use of niched operators.

This paper proposes two 3-stage hybrid financial time series prediction models. Both of these models check for the presence of chaos at both stages. In the Hybrid Model-1 (Chaos+MLP+MOPSO), chaos in the dataset is first modeled using minimum lag and minimum embedding dimension (Stage-1). The resultant multivariate time series is fed to MLP in Stage-2. Stage-3 tests for the presence of chaos in the residuals. If chaos is present, then it is suitably modeled and the resultant multivariate time series of the residuals is fed to MOPSO-trained auto-regression model; otherwise, polynomial regression is employed to model the residuals. The predicted values in the Stage-3 and the predicted values in the Stage-2 are algebraically summed up to obtain the final optimal predictions. The Hybrid Model-2 (Chaos+MLP+NSGA-II) is the same as Hybrid Model-1, except that the MOPSO-trained auto-regression is replaced with NSGA-II-trained auto-regression model in Stage-3.

The contributions of this paper are:

1. Two 3-stage hybrids including Chaos+MLP+MOPSO and Chaos+MLP+NSGA-II are proposed and these are applied to predict financial time series including exchange rates of JPY/USD, GBP/USD, EUR/USD and Gold price (USD).
2. The financial time series prediction problem is formulated as a bi-objective optimization problem in the Stage-3. Apart from MSE value, Directional change statistic (Dstat) [6] was also accorded importance, thereby considering both of them as two objective functions for the problem.
3. Thiel's inequality coefficient [24] was computed to determine the closeness of the predictions yielded by the hybrids. The results of proposed hybrids are tested whether they are statistically significant or not using Diebold-Mariano (DM) test [25].

The rest of this paper is structured as follows: Section 2 presents various hybrid models proposed for financial time series prediction. Later, various techniques used are described in Section 3. The proposed hybrid models are presented in Section 4. The Section 5 highlights the experimental methodology and the results obtained are discussed in detail in Section 6. Finally, the paper is concluded in Section 7.

2. Literature review

It is well known that hybrids of time series forecasting models yield better predictions than a single time series model [26]. In the following, a few relevant hybrid financial time series prediction models are reviewed. First, we present various Chaos-based hybrid forecasting models, second, we present various hybrid forecasting models involving MLP and PSO and finally, we present the forecasting models involving MOEAs.

The chaos-based hybrid forecasting models reviewed are as follows. Pavlidis et al. [27] proposed a hybrid technique for predicting financial time series involving pre-processing to account for chaos and a neural network trained by PSO/Differential Evolution (DE). The hybrid yielded promising results for both DE and PSO than the stand-alone neural network on the daily exchange rate data of JPY/USD and GBP/USD in terms of mean accuracy. Then, Huang et al. [28] proposed a hybrid model that models the Chaos in the data followed by prediction by support vector regression to predict FOREX Rates. They concluded that the hybrid outperformed the stand-alone techniques on the daily exchange rate data of EUR/USD, GBP, NZD, AUD, JPY and RUB in terms of Root Mean Squared Error (RMSE), MSE and Mean Absolute Error (MAE). Pradeepkumar and Ravi [1] proposed 3-stage hybrid models in the paradigm of soft computing for FOREX Rate prediction. They are Chaos+ANN (MLP/General Regression Neural Network (GRNN)/ Group Method of Data Handling (GMDH)) + PSO/Polynomial Regression (PR) and Chaos+PSO+ANN/PR. In these models, the Stage-3 refines the initial predictions yielded by Stage-2.

In both models, phase spaces of both characteristic information and residual information are reconstructed. The systematic modeling of chaos present in the data helped the proposed models in yielding better predictions compared to the stand-alone forecasting models after experimenting with daily exchange rates of JPY/USD, GBP/USD and EUR/USD in terms of both MSE and MAPE. Most recently, Pradeepkumar and Ravi [2] proposed 2-stage hybrid models including Chaos+CART (Classification and Regression Tree), Chaos+CART-EB (CART Ensembles and Bagger), Chaos+TreeNet, Chaos+MARS (Multivariate Adaptive Regression Splines), Chaos+LASSO (Least Absolute Shrinkage and Selection Operator) and Chaos+RFTE (Random Forest Tree Ensemble). The authors concluded that Chaos +MARS could outperform the other models and Pradeepkumar and Ravi [1] after experimenting with the daily exchange rates of JPY/USD, GBP/USD and EUR/USD in terms of both MSE and MAPE.

Various hybrid forecasting models involving either MLP or PSO or both but without accounting for chaos are briefly reviewed as follows. Zhang [29] proposed ARIMA+MLP model and reported that it outperformed both ARIMA and MLP on 3 datasets including Wolf's sunspot data, the Canadian lynx data and the GBP/USD exchange rate data. Chen and Leung [30] proposed a two-stage hybrid model that is composed of a time series model and GRNN in tandem. The authors concluded that the hybrid predicted exchange rates better than single stage models. Ince and Trafalis [31] also proposed a two-stage hybrid model in which stage-1 meant for input selection and co-integration analysis and stage-2 meant for obtaining predictions using ANN/SVR. They concluded that hybrid yielded better predictions. Khashei et al. [32] proposed a novel hybrid viz., ANN+ Fuzzy ARIMA for forecasting time series. The proposed hybrid obtained accurate results even under incomplete data conditions also. It worked better than Fuzzy model and ARIMA in situations when there are little historical data available. It also provided both of the finest and the worst possible situations to make good decisions.

Yu et al. [33] proposed an ensemble of multistage nonlinear Radial Basis Function (RBF) networks and concluded that it yielded improved accuracy on the FOREX data of GBP, EUR, DEM and JPY. Chang et al. [34] proposed the PSOBPN architecture to predict exchange rates, where PSO was employed to select optimal number of input nodes. They the whole dataset into six time based sliding windows. Then, superior variables are selected using PSO followed by BPN based forecasting. They concluded that PSOBPN could yield better forecast ability. They directed the readers to research on non-linear PSO model and further adoption of other evolutionary methods for comparative study. Chaudhuri and De [35] proposed Neuro-fuzzy regression model which yielded accurate predictions with fewer observations and incomplete datasets. Chang and Lee [36] proposed GA-based BPN and PSO-based BPN for forecasting exchange rate, where the whole data divided into six time-based sliding windows. Later, the superior variables were selected by the PSO and GA before BPN was invoked for prediction with selected variables. They concluded that the proposed linear model of PSO and GA could not perform well and suggested the readers to use nonlinear PSO model coupled with other evolutionary methods such as ACO. Gheyas and Smith [37] proposed novel neural network ensemble and tested it on 30 real datasets and concluded that it was effective in forecasting time series with seasonal patterns. Chang and Hsieh [38] proposed PSOBPN, where PSO was employed to select optimal hidden neurons to predict to NTD/USD rates. The authors concluded that the PSOBPN yielded better predictions and recommended the use of nonlinear PSO model. Hadavandi et al. [39] proposed PSO for estimating the coefficients of the AR model to forecast gold price, where only last two days gold price was considered to predict todays gold price. Li et al. [40] proposed a novel hybrid learning algorithm using PSO and Recursive Least Squares Estimator (RLSE) namely PSO+RLSE+PSO was proposed for the purpose of learning. The parameter space was divided into two subspaces of the premise parameters and consequent parameters respectively. The PSO and the RLSE were used in hybrid way to search for the optimal solution. The PSO-RLSE-PSO method is proposed to train the predictor, the neuro-fuzzy self-organizing system (NFS). The authors concluded that NFS neuro-fuzzy self-organizing system involving hybrid learning algorithm performed extremely well in terms of fast learning convergence. Aladag et al. [41] proposed a time-invariant fuzzy model to forecast based on PSO. The fuzzy C-means clustering method was used for fuzzification of time series. They concluded that the proposed method yielded accurate predictions. Then, Ravi et al. [42] presented a several FOREX forecasting models using a number of computational intelligent techniques such as Back-Propagation Neural Network (BPNN), Wavelet Neural Network (WNN), Multivariate Adaptive Regression Splines (MARS), Support Vector Regression (SVR), Dynamic evolving neuro-fuzzy Inference System (DENFIS), Group Method of Data Handling (GMDH) and Genetic Programming (GP) and concluded that ensemble of prediction models yielded better predictions after experimenting with USD versus JPY, GBP and DEM. Sermpinis et al. [43] introduced a hybrid Neural Network architecture of PSO and Adaptive Radial Basis Function (ARBF-PSO) to forecast the EUR/GPB exchange rates returns. The authors concluded that the proposed model outperformed MLP, Recurrent Neural Network (RNN) and Psi Sigma Neural Network (PSI). Sermpinis et al. [44] extended their work Sermpinis et al. [43] by proving the efficiency of hybrid neural network architecture after experimenting with more datasets and compared the results with more benchmarks. Donate et al. [45] proposed a weighted cross-validation evolutionary artificial neural network (EANN) ensemble that achieved better results compared to a non-weighted version of the same model. Pulido et al. [46] proposed PSO based structure optimization of ANN together with type-1 and type-2 fuzzy integration of responses. The authors concluded that hybrid outperformed other fuzzy integrations.

Rout et al. [47] proposed hybrid prediction model that combined an adaptive ARMA architecture and DE-based training of its feed-forward and feed-back parameters. The DE optimization is used to obtain optimal coefficients for training. This optimization strategy helped the hybrid prediction model to give better prediction accuracy than other hybrid models. Pradyot et al. [48] developed a novel knowledge guided artificial neural network (KGANN) model for exchange rate prediction. In the proposed model trained least mean squared output is added to an adaptive Fuzzy logic ANN to provide accurate predictions. The model yielded accurate predictions than LMS and FLANN models. Shen et al. [49] forecasted exchange rates using deep belief network (DBN). The authors applied conjugate gradient method to accelerate the learning of DBN. The authors concluded that the proposed method yielded accurate predictions than Feed Forward Neural Network (FFNN) after experimenting with the weekly data of GBP/USD, BRL/USD and INR/USD in terms of RMSE, MAE and MAPE. Hussain et al. [50] forecasted financial time series including exchange rates of USD/EUR, USD/GBP and JPY/USD using a novel regularized dynamic self-organized neural network inspired by the immune algorithm. The simulation results showed that the proposed neural network architecture yielded accurate predictions than SVM, RBFNN and BPNN. Sermpinis et al. [51] introduced a hybrid rolling genetic algorithm-support vector regression (RG-SVR) model and applied to forecasting and trading the EUR/USD, EUR/GBP and EUR/JPY exchange rates. The authors concluded that the proposed hybrid outperformed genetically and non-genetically optimized SVRs and SVMs. Svitlana [52] explored the application of neural networks to predict exchange rates of EUR/USD, GBP/USD and USD/JPY in daily, monthly and quarterly steps. The author concluded that the short-term prediction method provides good accuracy of the prediction and can be used in practical systems to predict one-step ahead exchange rates.

Various hybrid forecasting models involving MOEAs are as reviewed follows. Seng et al. [53] proposed non-linear time series prediction method based on Lyapunov theory based fuzzy neural network and Multi-objective GA. Then, Mishra et al. [54] proposed a prediction strategy and improved portfolio optimization by combining Multi-objective evolutionary optimization algorithms including NSGA-II and MOPSO. Later, Hassan et al. [55] introduced Hidden Markov Model (HMM), Fuzzy Logic and Multi-objective Evolutionary Algorithm (EA) based new hybrid fuzzy model to predict non-linear time series. Finally, Ronay et al. [56] proposed NSGA-II to train an NN for short-term forecasting of wind speed. Even though this work is not in the finance domain, it is mentioned here to show the utility of NSGA-II in time series forecasting.

3. Overview of techniques used

In these hybrids, Saida's method is used for checking the presence of chaos, Akaike Information Criterion (AIC) is used for obtaining minimum lag and Cao's method is used for obtaining minimum embedding dimension. A stand-alone prediction model MLP is used as part of hybrid models. An Evolutionary Algorithm namely PSO and various MOEAs such as MOPSO and NSGA-II are used to obtain near optimal coefficients of auto-regressive error model.

3.1. Chaos theory

Chaos theory is an area of deterministic dynamics proposing that seemingly random events can result from normal equations because of the complexity of the systems involved. Chaos in a time series is modeled by constructing the corresponding phase space from the time series using both lag (l) and embedding dimension (m). This process converts the given nonlinear single-dimensional time series into its equivalent multi-dimensional representation. This approach was pioneered by Packard et al. [57] and mathematically explained by Takens [58]. Let $Y = \{y_1, y_2, \ldots, y_k, y_{k+1}, \ldots, y_N\}$ be a scalar time series. It can be fully embedded in m-dimensional phase space represented by the vector $P_j = \{y_j, y_{j+l}, y_{j+2\,l}, \ldots, y_{j+(m-1)l}\}$ where $j = 1, 2, \ldots, N - (m-1)l/\Delta t$, m is called the embedding dimension ($m \geq \theta$, where θ is the dimension of the attractor), l is the time delay and Δt is the sampling time [57,58].

3.2. Saida's method

This method [59] estimates Lyapunov exponent [60] λ as in 1.

$$\lambda = \lim_{t \to \infty} \frac{1}{t} ln \frac{|\Delta g(y_0, t)|}{|y_0|} \tag{1}$$

where y_0 is a point in the state space that will produce orbit in that space using a system of equations. A value of $\lambda \geq 0$ implies that chaos is present; otherwise, it is absent in the time series.

3.3. Akaike Information Criterion (AIC)

Akaike Information Criterion [61] is one of the methods to select the right lag in modeling the auto regressive aspect of an auto correlated time series data. It identifies a lag j that minimizes $\frac{log(SSR(j))}{N} + (j+1)\frac{c(N)}{N}$ where SSR(j) is the sum of squared residuals for the Vector Auto Regression (VAR) with j lags, where N is the number of observations and c(N)=2.

3.4. Cao's method

Cao proposed a method [62] to determine minimum embedding dimension for a given set. Let $Y = \{y_1, y_2, \ldots, y_k, y_{k+1}, \ldots, y_N\}$ be a time series. It can be reconstructed as time delay vectors in the phase space as in Eq. (2):

$$Y_i^m = \{y_i, y_{i+l}, y_{i+2\,l}, \ldots, y_{i+(m-1)l}\};\ i = 1, 2, \ldots, n - (m-1)l \tag{2}$$

Here, Y_i^m is the ith reconstructed vector. The corresponding nearest neighbors of the Y_i^m, denoted as ψY_i^m, are given as in Eq. (3).

$$\psi Y_i^m = \{\psi y_i, \psi y_{i+l}, \psi y_{i+2\,l}, \ldots, \psi y_{i+(m-1)l}\};\ i = 1, 2, \ldots, n - (m-1)l \tag{3}$$

Similar to the idea of FNN method, they define Eq. (4).

$$\phi_2(i, m) = \frac{\|y_i^{m+1} - \psi y_i^m\|}{\|y_i^m - \psi y_i^m\|} \tag{4}$$

Here, $\|.\|$ is maximum norm that results in the Euclidian distance, y_i^{m+1} is the ith reconstructed vector and ψy_i^m is its nearest neighbor in embedding dimension m+1. To avoid the problems of FNN, they define the mean value of all $\phi_2(i, m)$ s as in Eq. (5).

$$E(m) = \frac{1}{n - ml} \sum_{i=1}^{n-ml} \phi_2(i, m) \tag{5}$$

$E(m)$ is only dependent of both l and m. To investigate its variation from m to m+1, we define Eq. (6):

$$E_1(m) = \frac{E(m+1)}{E(m)} \tag{6}$$

Whenever $E_1(m)$ stops changing for $m \geq m_0$ for some m_0, then m_0 is the required minimum embedding dimension what we look for.

3.5. Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron [63] is by far the most popular feed-forward network architecture. It is very powerful and versatile and found many applications in diverse fields. Typically, it comprises an input layer, a hidden layer, and an output layer. It is trained by the popular back-propagation algorithm where the weights connecting the layers are updated in an iterated manner. Once trained, the network can be used to make prediction on new and unseen data. It is one of the versatile networks that solves both classification and regression problems.

3.6. Multi-objective evolutionary algorithms

Multi-objective evolutionary algorithms (MOEAs) are used to solve multi-objective optimization problems. The MOEAs preserve non-dominated solutions and continue to progress algorithmically towards the Pareto optimal front, maintain diversity of solutions in Pareto optimal front and provide the decision maker a limited number of Pareto solutions for selection by decision maker (Coello Coello et al. [23]). A good review of MOEAs can be found in Deb [64], Coello Coello et al. [23] and Mukhopadhyay et al. [65,66]. The two Multi-objective optimization algorithms used in this paper are:

3.6.1. Particle Swarm Optimization for Multi-objective optimization (MOPSO) using scalar optimization

PSO, developed by Kennedy and Eberhart [67], gained popularity for its simplicity, ease of implementation and existence of very few user-defined parameters. The heuristics of PSO involve initialization and updating of the position and the corresponding velocity. It progresses towards the solution by mutual sharing of knowledge of every particle collectively.

In PSO, the population of particles with velocity ${V_{id}}^{old}$ is initially randomly generated. Each particle's velocity gets updated with respect to its corresponding old position ${x_{id}}^{old}$ using neighborhood best p_{id} See Eqs. (7), (8) and global best particle p_{gd} until the convergence criterion is satisfied. After the convergence criterion is satisfied, the global best particle is the optimal solution.

$$V_{id}^{New} = \gamma * V_{id}^{Old} + C_1 * rand * (p_{id} - X_{id}^{old}) + C_2 * rand * (p_{gd} - X_{id}^{old}) \tag{7}$$

$$X_{id}^{New} = X_{id}^{Old} + V_{id}^{New} \tag{8}$$

where ${V_{id}}^{Old}$ is old velocity, ${V_{id}}^{New}$ is updated velocity, ${X_{id}}^{Old}$ is old particle, ${X_{id}}^{New}$ is updated particle, p_{id} is a local best particle, p_{gd} is the global best particle, C_1 and C_2 are two positive constants, γ is the inertia weight and finally, *rand* is a random number between 0 and 1.

The PSO can also handle multiple objectives [68]. One approach is the conventional scalar optimization for handling multiple objectives. Accordingly, a new objective function is maximized that is formulated by summing up the product of the objectives (O_i, i=1,2…,n) with their corresponding user-defined non-negative weights (w_i, i=1,2…,n such that $\sum_{i=1}^{n} w_i = 1$) resulting in $\sum_{i=1}^{n} w_i O_i$. These weights can be fixed or dynamically adopted during the process of optimization.

3.6.2. Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

NSGA-II [69] is a modified version of NSGA procedure [70], which uses a better sorting algorithm, incorporates elitism, and does not require niching or any other parameter. In NSGA-II, a population is randomly initialized based on the domain of the decision variables. Then, the population is sorted based on non-domination of solutions resulting in various hierarchical fronts. Once the non-dominated sorting is completed, a crowding distance value is assigned to each solution depending on its closeness to other population members in the objective space. The individuals are selected based on their non-dominated rank and the crowding distance, in that order. Thereafter, recombination and mutation operators are applied to the selected parent solutions to create the new offspring population. This process continues until a termination criterion is reached. At every subsequent generation t, the offspring population Q_t and the current generation population P_t are combined and selection is invoked to select the solutions for the next generation. Elitism is taken care of because all the previous and current best solutions are passed on to the next-generation. Then, the population is sorted based on non-domination. Thereafter, new generation is formed by each front until the population size exceeds the current population size. In case the addition of all the solutions in the front F_i results in the population P_{t+1} exceeding N, then solutions in front F_i are selected based on their crowding distance in the descending order until the population size is N. The process is repeated to form subsequent generations. This process is depicted in Fig. 1.

4. Proposed hybrid models

4.1. Formulating multi-objective problem

Financial Time series prediction is formulated as a multi-objective optimization problem as follows:

Minimize O_1=Mean Squared Error (MSE) and

Maximize O_2=Directional change statistic (Dstat)

where MSE and Dstat are defined in Eqs. (15) and (16) respectively.

In financial time series prediction, minimizing the average prediction error is very much important. In addition to this, it is also very important to predict the movement of direction of financial time series. So, we selected the two objectives mentioned above.

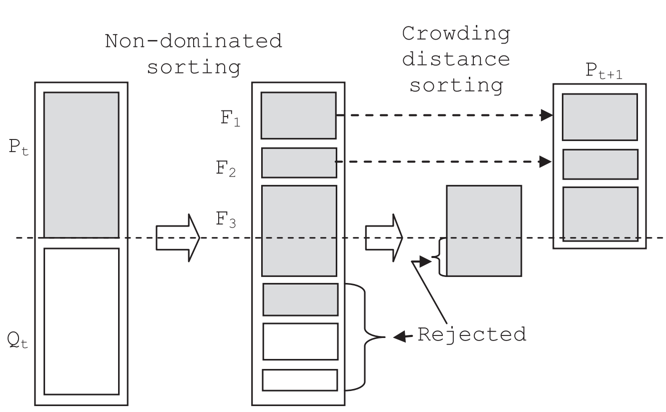

Fig. 1. Non-dominated Sorted Gentic Algorithm-II (NSGA-II)(adopted from [69]).

Table 1
Notations used in Three-stage hybrid model and their interpretations.

Notation	Interpretation
l_1,m_1	Lag and embedding dimension values used in Stage-1.
l_2,m_2	Lag and embedding dimension values used in Stage-2.
e_t	Error at time t.
$\hat{e}_t$	Predicted error at time t.
α_0, α_1, α_2…	Coefficients to be determined.
g1(.)	A Non-linear function to yield predictions using MLP.
g2(.)	A linear function to yield predictions using Polynomial Regression.
y_t	Actual data point at time t.
$\hat{y}_t$	Predicted data point at time t after Stage-1.
$\hat{\hat{y}}_t$	Predicted data point at time t after Stage-2.

NSGA-II solves this Multi-objective optimization problem directly. However, in order to solve it using MOSPO, it should be converted to a scalar optimization problem with equal weights accorded to MSE and Dstat, as follows:

Minimize $O_1 - O_2$, where O_1 and O_2 are defined as above.

4.2. Description of three-stage hybrid model

Table 1 presents the notations used in the hybrid models. The detailed description of Three-stage prediction model is as follows. Let $Y = \{y_1, y_2, \ldots, y_k, y_{k+1}, \ldots, y_N\}$ be a financial time series of N data points at $t = 1, 2, \ldots, k, k+1, \ldots, N$ respectively. The Three-stage prediction, as depicted by 2, proceeds as follows:

Stage-1: Phase space reconstruction

1. Check Y for the presence of chaos. If chaos is present, then reconstruct phase space from Y by using minimum lag (l_1) and minimum embedding dimension (m_1).
2. Partition rebuild Y into $Y_{Train} = \{y_t;\ t = l_1 m_1 + 1, l_1 m_1 + 2, \ldots, k\}$ and $Y_{Test} = \{y_t;\ t = k+1, k+2, \ldots, N\}$.

I. Training Phase
 (a) Stage-2: Using MLP
  i. Train MLP using the training set Y_{Train}
  ii. Obtain initial predictions using Eq. (9) and corresponding errors using Eq. (10) as follows.

$$\begin{aligned} \hat{y}_t &= g_1(y_{t-l_1}, y_{t-2l_1}, \ldots, y_{t-m_1 l_1}) \\ t &= l_1 m_1 + 1, l_1 m_1 + 2, \ldots, k \end{aligned} \tag{9}$$

$$\begin{aligned} e_t &= y_t - \hat{y}_t \\ t &= l_1 m_1 + 1, l_1 m_1 + 2, \ldots, k \end{aligned} \tag{10}$$

 (b) Stage-3: Using MOPSO/NSGA-II based Auto-regression model
  i. Check error set of all e_t obtained above for the presence of the chaos. Rebuild the set using l_2 and m_2 and obtain $\hat{e}_t$ using MOPSO/NSGA-II based auto-regression model as in Eq. (11), if chaos is present; otherwise, apply Polynomial Regression (PR) $g_2(x)$ with the functional form $Y_{out} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n$ in order to obtain $\hat{e}_t$ as in Eq. (12).

$$\begin{aligned} \hat{e}_t &= \alpha_0 + \alpha_1 e_{t-l_2} + \alpha_2 e_{t-2l_2} + \cdots + \alpha_{m_2} e_{t-m_2 l_2} \\ t &= l_1 m_1 + l_2 m_2 + 1, l_1 m_1 + l_2 m_2 + 2, \ldots, k \end{aligned} \tag{11}$$

$$\begin{aligned} \hat{e}_t &= g_2(e_t) \\ t &= l_1 m_1 + 1, l_1 m_1 + 2, \ldots, k \end{aligned} \tag{12}$$

  ii. Obtain final training set predictions after Stage-2 using Eq.

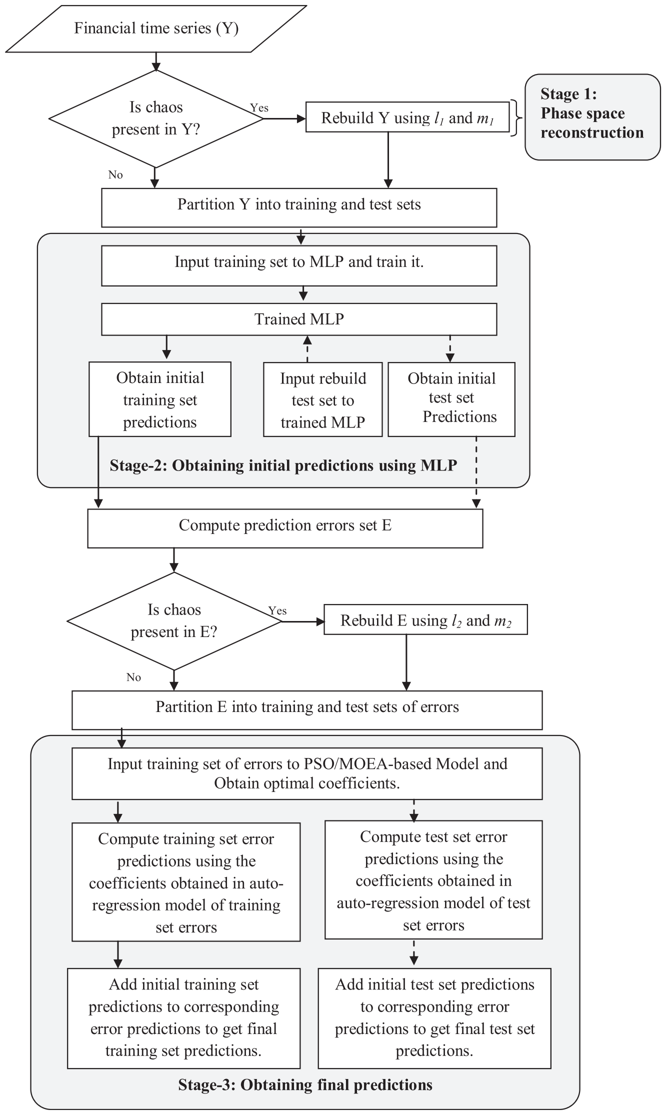

Fig. 2. Prediction using Three-Stage Hybrid Model.

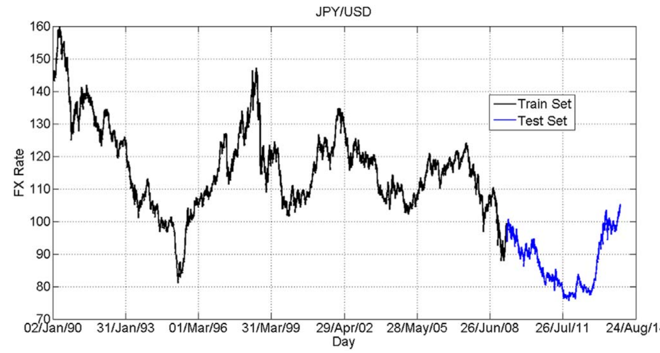

Fig. 3. Daily Forex Rates of JPY/USD.

(13) in the presence of chaos and using Eq. (14) in the absence of chaos.

$$\begin{aligned}\hat{\hat{y}}_t &= \hat{y}_t + \hat{e}_t \\ t &= l_1m_1 + l_2m_2 + 1,\ l_1m_1 + l_2m_2 + 2, \ldots, k\end{aligned} \tag{13}$$

$$\begin{aligned}\hat{\hat{y}}_t &= \hat{y}_t + \hat{e}_t \\ t &= l_1m_1 + 1,\ l_1m_1 + 2, \ldots, k\end{aligned} \tag{14}$$

II. Test Phase: In this phase, obtain test set predictions after going through the two stages by replacing the observations with $t = k+1, k+2, \ldots, N$ in Eqs. (9), (10), (11), (12), (13) and (14) respectively.

4.3. Hybrid model-1 (Chaos+MLP+MOPSO)

The proposed Hybrid Model-2 consists of three stages. In this hybrid, Chaos Theory is used to construct phase space from financial time series in Stage-1, MLP is involved in Stage-2 and MOPSO/PR is involved in Stage-3 (if chaos present or absent as the case may be). Here, when chaos is present, MOPSO is used to obtain optimal values of $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$ within Eq. (11).

The algorithm of obtaining optimal $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$ using MOPSO proceeds as follows.

1. Initialize the particles with random values within specified range. Each particle is a vector of length $m_2 + 1$.
2. Evaluate and minimize the objective function of the scalar optimiza-

Table 2
Descriptive Statistics of all Financial time series used in Three-stage hybrids.

Data	No. of Observations	Mean	Median	Min	Max	SD	Skewness	Kurtosis
JPY/USD								
All data	6036	110.55	110.27	75.72	159.90	16.93	0.03	−0.17
Training Set	4829	116.30	116.00	81.12	159.90	13.28	0.46	0.54
Test Set	1207	87.52	86.20	75.72	105.20	7.99	0.27	−1.28
GBP/USD								
All data	6036	1.655	1.615	1.366	2.110	0.16	0.75	−0.3
Training Set	4829	1.674	1.635	1.366	2.110	0.17	0.48	−0.79
Test Set	1207	1.579	1.586	1.397	1.698	0.05	−0.63	0.23
EUR/USD								
All data	3772	1.219	1.269	0.827	1.601	0.19	−0.44	−0.74
Training Set	3018	1.189	1.220	0.827	1.601	0.20	−0.12	−0.97
Test set	754	1.336	1.326	1.206	1.488	0.06	0.38	−0.3
Gold Price (USD)								
All data	7602	575.79	385.85	252.9	1896.5	404.93	1.65	1.43
Training Set	6081	391.95	371.15	252.9	1023.5	118.42	2.31	6.38
Test Set	1521	1310.75	1316	692.5	1896.5	296.89	−0.12	−1.09

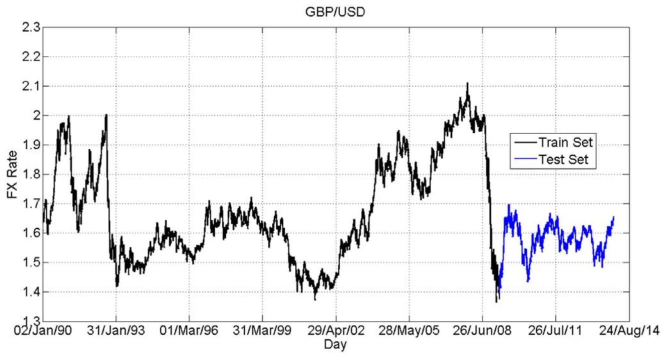

Fig. 4. Daily Forex Rates of GBP/USD.

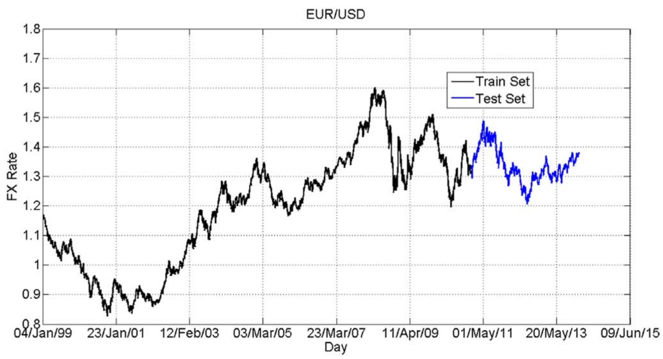

Fig. 5. Daily Forex Rates of EUR/USD.

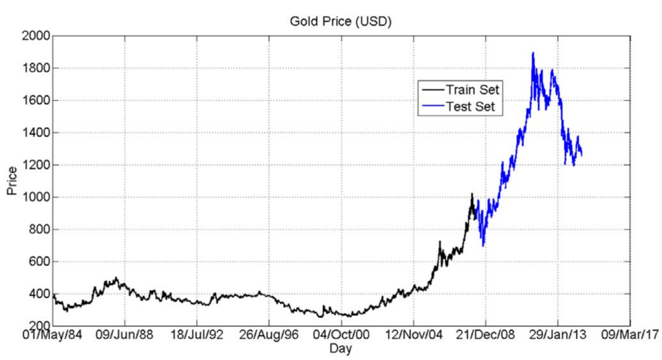

Fig. 6. Daily prices of Gold (USD).

tion problem of each particle, $w_1O_1 - w_2O_2$, where O_1=MSE, O_2=Dstat, w_1 and w_2 are weights, using initial prediction set and prediction errors.
3. Update individual and global best fitnesses and positions.
4. Update velocity and position of each particle as in Eqs. (7) and (8).
5. Repeat steps 2, 3 and 4 until all iterations are finished.

The coordinates of the global best particle are the optimized $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$.

4.4. Hybrid model-2 (Chaos+MLP+NSGA-II)

The proposed Hybrid Model-3 consists of three stages. In this hybrid, Chaos Theory is used to construct phase space from financial time series in Stage-1, MLP is involved in Stage-2 and NSGA-II/PR is involved in Stage-3 (if chaos present or absent as the case may be). It is similar to Hybrid Model-2 except that NSGA-II is used to obtain optimal $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$. within Eq. (11).

The algorithm of obtaining optimal $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$ using NSGA-II is as follows:

1. Generate a random population of n individuals. Each individual is a vector of length $m_2 + 1$.
2. Evaluate the fitness functions namely O_1=MSE and O_2=Dstat of each individual in the population using initial predictions and errors.
3. Rank the population according to the following steps:
   (a) Rank the population using Non-dominated sorting algorithm.
   (b) Calculate the crowding distance
4. Create new population by repeating the following steps:
   (a) Select two parents from the population based on the crowding selection operator.
   (b) Crossover the parents using crossover probability to form the offspring.
   (c) Mutate the new offspring using mutation probability.
   (d) Combine the parents and offspring.
   (e) Select n number of best individuals for the next generation and discard the others.
5. Repeat Steps 2-4 with the new population obtained from the previous generation. Repeat this until all generations are completed.
6. After the completion of all generations, pick up the particle best fitness values, i.e., minimal MSE and maximal Dstat. The coordinates of the best particle are the optimized coefficients $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{m_2}$.

Table 3
Tools and Techniques used for Three-Stage Prediction.

Technique Used	Used for	Tool used
MLP (http://www.neuroshell.com/)	Obtaining predictions	*Neuroshell®*
Saida's Method (http://www.mathworks.in/matlabcentral/fileexchange/22667-chaos-test)	Checking for presence of chaos	*MATLAB®*
Akaike Information Criterion (AIC) (http://gretl.sourceforge.net/)	Obtaining optimal lag	*Gretl®*
Cao's Method (http://www.mathworks.in/matlabcentral/fileexchange/36935-minimum-embedding-dimension/content/cao_deneme.m)	Obtaining predictions	*MATLAB®*
PSO	Obtaining coefficients of model	Java
MOPSO	Obtaining coefficients of model	Java
NSGA-II	Obtaining coefficients of model	*MATLAB®*

Table 4
Parameters used for different datasets.

Technique Used	Parameters	JPY/USD	GBP/USD	EUR/USD	Gold Price (USD)
MLP	Learning Rate	0.6	0.5	0.6	0.6
	Momentum Rate	0.9	0.7	0.9	0.9
	No.of Hidden Nodes	10	30	20	32
MOPSO-based Model	Particles	50	60	60	60
	Intertia Weight	0.8	0.6	0.8	0.8
	C_1, C_2	2,2	2,2	2,2	2,2
	Iterations	40,000	40,000	40,000	40,000
NSGA-II-based Model	Population	50	50	50	50
	Crossover rate	0.9	0.9	0.9	0.9
	Mutation rate	0.3	0.3	0.3	0.3
	Generations	2000	2000	2000	2000

5. Experimental design

5.1. Datasets used

The datasets collected are of daily US Dollar (USD) exchange rates with respect to three currencies- Japanese Yen (JPY), Great Britain Pound (GBP), Euro (EUR), Gold price in terms of USD. These four financial datasets are used for testing the effectiveness of the presented prediction models in the chapter. The foreign exchange data are obtained from (http://www.federalreserve.gov/releases/h10/hist/) and Gold price data is obtained from (http://www.quandl.com/LBMA/GOLD-Gold-Price-London-Fixing).

The daily exchange rates of JPY/USD, GBP/USD from 2nd January 1990 to 31st December 2013 and EUR/USD from 4th January 1999 to 31st December 2013 and Gold price data collected from 1st May 1984 to 30th May 2014 are used as datasets. Each dataset is partitioned into 80% of the dataset as a training set and 20% of the dataset as a test set as presented in Table 2 and is depicted by Fig. 3 through 6 respectively.

From Table 2, the observations made are as follows. For JPY/USD dataset (see Fig. 3, training set is with peaked distribution and test set is with flat distribution. The tail of both training set and test set distributions are more stretched on the side above mean. For GBP/USD dataset (see Fig. 4), training set is with flat distribution and the tail of training set distribution is more stretched above the mean. Test set is with peaked distribution and the tail of test set distribution is more stretched below the mean. For EUR/USD dataset (see Fig. 5), both training set and test sets are with flat distribution. The tail of training set distribution is more stretched below the mean and The tail of test set distribution is more stretched above the mean. Finally, for Gold Price (USD) (See Fig. 6)dataset, training set is with peaked distribution and test set is with flat distribution. The tail of the training set distribution is more stretched above the mean and of the test set distribution is more stretched below the mean.

5.2. Tools and techniques used

The execution of the proposed prediction models is carried out using Windows 7 *Professional*® platform. However, it can also be carried out on other platforms. It is carried out under the system with the specifications of 8 GB RAM and 500 GB HDD. Various tools employed in numerous experiments with the datasets are mentioned in Table 3.

5.3. Performance measures used

Three performance measures including MSE (Mean Squared Error), Dstat (Directional change stastic) and Theil's Inequality Coefficient (U) are used to measure the fitness of proposed financial time series prediction models.

The MSE see Eq. (15) is a good measure of how accurately the model predicts the response. It measures the average of the squares of the errors. MSE is useful when we are concerned about significant errors whose negative consequences are proportionately much bigger

Table 5
Results of Three-stage Hybrid Models for test set of JPY/USD.

Hybrid Model	Technique (Chaos Parameters)	MSE	Dstat	Theil's U	Rank
Chaos+MLP	–	10.785	48.053	0.018661	11
Chaos+PSO	–	1.8508	50.87	0.0077519	6
Chaos+MOPSO	–	16.69	52.941	0.02334	13
Chaos+NSGA-II	–	36.434	51.616	0.034715	15
Chaos+GRNN [1]	–	33.4999	–	–	14
Chaos+GMDH [1]	–	0.3560	–	–	3
Chaos+CART [2]	–	10.22168	–	–	10
Chaos+CART-EB [2]	–	14.16271	–	–	12
Chaos+MARS [2]	–	0.34555	–	–	2
Chaos+TreeNet [2]	–	9.07775	–	–	9
Chaos+LASSO [2]	–	3.33874	–	–	7
Chaos+RFTE [2]	–	5.92272	–	–	8
Chaos+MLP+PSO	Stage-1: Chaos Present MLP (l1=4, m1=20) Stage-2: Chaos Present PSO-based Model (l2=10, m2=11)	0.55246	47.722	0.0042271	4
Chaos+MLP+MOPSO	Stage-1: Chaos Present MLP (l1=4, m1=20) Stage-2: Chaos Present MOPSO-based Model (l2=10, m2=11)	0.58723	47.556	0.0043568	5
Chaos+MLP+NSGA-II	Stage-1: Chaos Present MLP (l1=4, m1=20) Stage-2: Chaos Present NSGA-II-based Model (l2=10, m2=11)	**2.00E−05**	**99.917**	**2.55E−05**	**1**

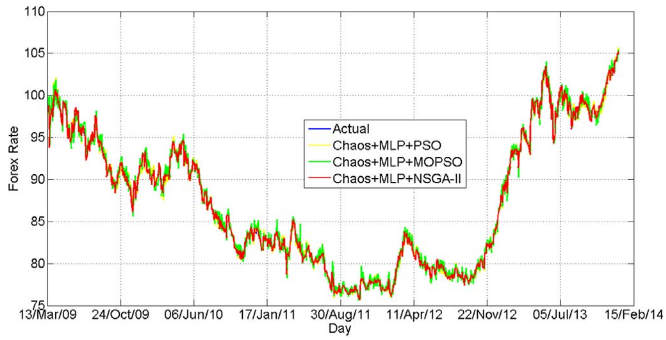

Fig. 7. Predictions of Three-stage Hybrid Models for test set of JPY/USD.

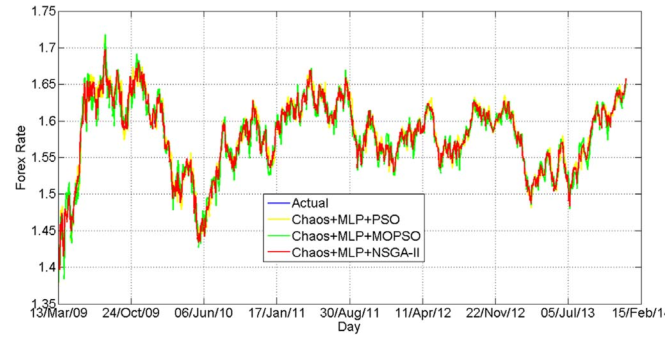

Fig. 8. Predictions of Three-stage Hybrid Models for test set of GBP/USD.

than equivalent smaller one [71]. An MSE value closer to 0 indicates a fit that is more useful for prediction.

$$MSE = \frac{\sum_{t=1}^{N} (y_t - \hat{\hat{y}}_t)^2}{N} \tag{15}$$

In predicting financial time series, it is not only important to measure the accuracy of predictions but also the directional change of time series. For this purpose, Yao and Tan [6] developed a measure (expressed in percentages) namely Dstat as in Eq. (16). It is also used in thesis.

$$Dstat = \frac{1}{N}\sum_{t=1}^{N} a_t * 100\% \tag{16}$$

$$where a_t = \begin{cases} 1, & \text{if}(y_{t+1} - y_t) * (\widehat{\hat{y}_{t+1}} - \hat{\hat{y}}_t) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

It is well known that Theil's Inequality Coefficient (see Eq. (17)) measures how well a forecasted time series is closer to actual time series [24,71]. Generally, the value of U lies in between 0 and 1. $U=0$ means that $y_t = \hat{\hat{y}}_t$ for all observations and there is a perfect fit and $U=1$ means that the performance is bad.

$$U = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N} (y_t - \hat{\hat{y}}_t)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N} (y_t)^2} + \sqrt{\frac{1}{N}\sum_{t=1}^{N} (\hat{\hat{y}}_t)^2}} \tag{17}$$

In Eqs. (15), (16) and (17), N is the number of predictions obtained, y_t is the actual observation at time t and $\hat{\hat{y}}_t$ is final predicted value at time t respectively.

6. Results and discussion

While conducting experiments over the datasets, different user-defined parameters are finetuned in order to obtain the best performance from the techniques. These are presented in Table 4. In MLP, the learning rate $(0 < \eta < 1)$ controls the size of weight and bias changes in learning of the training algorithm and momentum $(0 < \alpha < 1)$ simply adds a fraction α of the previous weight update to the current one. In this paper, we have chosen different learning rates, momentum rates and number of hidden nodes for each dataset. The parameters used in MOPSO for all datasets are as follows. The inertia weight (γ), controls the momentum of a particle. We have chosen $\gamma = 0.6$ for GBP/USD and $\gamma = 0.8$ for other three datasets. In PSO literature, it is quite a common practice to limit the swarm size to the range 20–60 [72–74]. In the current work, we selected the number of particles as 50 for JPY/USD and 60 for other three datasets after

Table 6
Results of Three-stage Hybrid Models for Test set of GBP/USD.

Hybrid Model	Technique (Chaos Parameters)	MSE	Dstat	Theil's U	Rank
Chaos+MLP	–	0.002737	52.196	0.016571	13
Chaos+PSO	–	0.000244	49.793	0.004947	11
Chaos+MOPSO	–	0.010106	50.539	0.032068	15
Chaos+NSGA-II	–	0.009934	47.639	0.031555	14
Chaos+GRNN [1]	–	0.000540	–	–	12
Chaos+GMDH [1]	–	0.0000856	–	–	3
Chaos+CART [2]	–	0.00010	–	–	5
Chaos+CART-EB [2]	–	0.00010	–	–	5
Chaos+MARS [2]	–	0.00008	–	–	2
Chaos+TreeNet [2]	–	0.00009	–	–	4
Chaos+LASSO [2]	–	0.00015	–	–	8
Chaos+RFTE [2]	–	0.00024	–	–	10
Chaos+MLP+PSO	Stage-1: Chaos Present MLP (l1=5, m1=16) Stage-2: Chaos Present PSO-based Model (l2=5, m2=16)	0.000119	49.213	0.003453	7
Chaos+MLP+MOPSO	Stage-1: Chaos Present MLP (l1=5, m1=16) Stage-2: Chaos Present MOPSO-based Model (l2=5, m2=16)	0.000152	49.71	0.003911	9
Chaos+MLP+NSGA-II	Stage-1: Chaos Present MLP (l1=5, m1=16) Stage-2: Chaos Present NSGA-II-based Model (l2=5, m2=16)	**2.16E−08**	**99.917**	**4.65E−05**	**1**

Table 7
Results of Three-stage Hybrid Models for Test set of EUR/USD.

Hybrid Model	Technique (Chaos Parameters)	MSE	Dstat	Theil's U	Rank
Chaos+MLP	–	0.002389	49.867	0.018277	15
Chaos+PSO	–	8.05E−05	47.878	0.003139	6
Chaos+MOPSO	–	0.000183	51.326	0.005067	10
Chaos+NSGA-II	–	0.00051	49.204	0.008458	13
Chaos+GRNN [1]	–	0.00010758	–	–	8
Chaos+GMDH [1]	–	0.000064089	–	–	3
Chaos+CART [2]	–	0.0008	–	–	14
Chaos+CART-EB [2]	–	0.00009	–	–	7
Chaos+MARS [2]	–	0.00006	–	–	2
Chaos+TreeNet [2]	–	0.00007	–	–	4
Chaos+LASSO [2]	–	0.00027	–	–	12
Chaos+RFTE [2]	–	0.00008	–	–	5
Chaos+MLP+PSO	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present PSO-based Model (l2=10, m2=16)	0.000169	49.204	0.004859	9
Chaos+MLP+MOPSO	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present MOPSO-based Model (l2=10, m2=16)	0.000366	50.928	0.007145	11
Chaos+MLP+NSGA-II	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present NSGA-II-based Model (l2=10, m2=16)	**7.09E−06**	**75.995**	**0.00315**	**1**

thorough experimentation which is the same as the one we carried out in [1]. Usually, the acceleration coefficients C_1(Self-confidence) and C_2(Swarm confidence) are within the range of [0,4] [75,74]. In the current work, we selected $C_1 = C_2 = 2$ which is the same as the one we carried out in [1], and the maximum number of iterations is 40,000 so that global best particle is converged to near optimal solution. The parameters commonly used in NSGA-II for all datasets are as follows. Population size is chosen as 50. Usually, both crossover rate and mutation rate are in [0,1]. So, we have chosen Crossover rate as 0.9 and Mutation rate as 0.3. We have chosen the Number of generations as 2000. These parameters are chosen after rigorous experimentation.

It is important to note that the models we used for comparitive study are: Chaos+GRNN and Chaos+GMDH [1], Chaos+CART, Chaos+CART-EB, Chaos+MARS, Chaos+TreeNet, Chaos+LASSO and Chaos+RFTE [2]. In their work, the authors utilized the hybrids only for predicting JPY/USD, GBP/USD and EUR/USD and they reported corresponding MSE values only. Based on the rank of minimum MSE, the results revealed that the hybrid Chaos+MLP+NSGA-II outperformed all of these models on all three datasets. The results of each dataset are described as follows.

6.1. JPY/USD

Table 5 presents the MSE, Dstat and Theil's U values using Two-stage hybrids including Chaos+MLP, Chaos+PSO, Chaos+MOSPO and Chaos+NSGA-II for test set of JPY/USD and chaos-based Three-stage models: Chaos+MLP+PSO, Chaos+MLP+MOPSO and Chaos+MLP+NSGA-II along with corresponding lag and embedding dimension. It also presents the reported MSE values of Chaos+GRNN, Chaos+GMDH, Chaos+CART, Chaos+CART-EB, Chaos+MARS, Chaos+TreeNet, Chaos+LASSO and Chaos+RFTE. The results, based on the rank of minimum MSE, revealed that Chaos+MLP+NSGA-II outperformed all other models. Fig. 7 depicts the predicted values of three-stage models along with the actual values and reveals that Chaos+MLP+NSGA-II could yield very closer predictions such that its curve overlapped the curve of actual values.

6.2. GBP/USD

Table 6 presents the MSE, Dstat and Theil's U values using Two-stage hybrids including Chaos+MLP, Chaos+PSO, Chaos+MOSPO and Chaos+NSGA-II for test set of GBP/USD and Three-stage models: Chaos+MLP+PSO, Chaos+MLP+MOPSO and Chaos+MLP+NSGA-II along with corresponding lag and embedding dimension. It also presents the reported MSE values of Chaos+GRNN, Chaos+GMDH, Chaos+CART, Chaos+CART-EB, Chaos+MARS, Chaos+TreeNet, Chaos+LASSO and Chaos+RFTE. The results, based on the rank of minimum MSE, revealed that Chaos+MLP+NSGA-II outperformed all other models. Fig. 8 depicts the predicted values of three-stage models along with the actual values and reveals that Chaos+MLP+NSGA-II could yield very closer predictions such that its curve overlapped the curve of actual values.

6.3. EUR/USD

Table 7 presents the MSE, Dstat and Theil's U values using Two-stage models including Chaos+MLP, Chaos+PSO, Chaos+MOSPO and Chaos+NSGA-II for test set of EUR/USD and Three-stage models: Chaos+MLP+PSO, Chaos+MLP+MOPSO and Chaos+MLP+NSGA-II along with corresponding lag and embedding dimension. It also

Fig. 9. Predictions of Three-stage Hybrid Models for test set of EUR/USD.

Table 8
Results of Three-stage Hybrid Models for test set of gold price (USD).

Hybrid Model	Technique (Chaos Parameters)	MSE	Dstat	Theil's U	Rank
Chaos+MLP	–	3023	50.756	0.070013	7
Chaos+PSO	–	425.93	56.189	0.007667	4
Chaos+MOPSO	–	1768.6	52.178	0.015637	5
Chaos+NSGA-II	–	437.38	54.317	0.007798	6
Chaos+MLP+PSO	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present PSO-based Model (l2=10, m2=16)	408.8	55.112	0.007528	3
Chaos+MLP+MOPSO	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present MOPSO-based Model (l2=10, m2=16)	371.52	59.244	0.007174	2
Chaos+MLP+NSGA-II	Stage-1: Chaos Present MLP (l1=1, m1=10) Stage-2: Chaos Present NSGA-II-based Model (l2=10, m2=16)	**0.01153**	**99.869**	**3.99E−05**	**1**

Fig. 10. Predictions of Three-stage Hybrid Models for test set of Gold Price (USD).

presents the reported MSE values of Chaos+GRNN, Chaos+GMDH, Chaos+CART, Chaos+CART-EB, Chaos+MARS, Chaos+TreeNet, Chaos +LASSO and Chaos+RFTE. The results, based on the rank of minimum MSE, revealed that Chaos+MLP+NSGA-II outperformed all other models. Fig. 9 depicts the predicted values of three-stage models along with the actual values and reveals that Chaos+MLP+NSGA-II could yield very closer predictions such that its curve overlapped the curve of actual values.

6.4. Gold Price (USD)

Table 8 presents the MSE, Dstat and Theil's U values using Two-stage models including Chaos+MLP, Chaos+PSO, Chaos+MOSPO and Chaos+NSGA-II for test set of Gold Price (USD) and Three-stage models: Chaos+MLP+PSO, Chaos+MLP+MOPSO and Chaos+MLP +NSGA-II along with corresponding lag and embedding dimension. The results, based on the rank of minimum MSE, revealed that Chaos +MLP+NSGA-II outperformed all other models. Fig. 10 depicts the predicted values of three-stage models along with the actual values and reveals that Chaos+MLP+NSGA-II could yield very closer predictions such that its curve overlapped the curve of actual values.

6.5. Discussion

Since it is very difficult to get identified Pareto Optimal Fronts for all the 30 runs of a Multi-Objective Evolutionary Algorithm, Fonseca suggested to plot Empiritcal Attainment Function (EAF) which we constructed and the same are depicted for each dataset. Generally, the EAF describes the probabilistic distribution of outcomes obtained by the stochastic algorithm in objective space [76,77]. The figures from Fig. 11–Fig. 14 depict the EAF plots, depicting the best, the median and the worst attainment surfaces of proposed MOEA-based hybrid models including Chaos+MLP+NSGA-II and Chaos+MLP+MOPSO. These plots are used for exploring the performance of stochastic local search algorithms for biobjective optimization

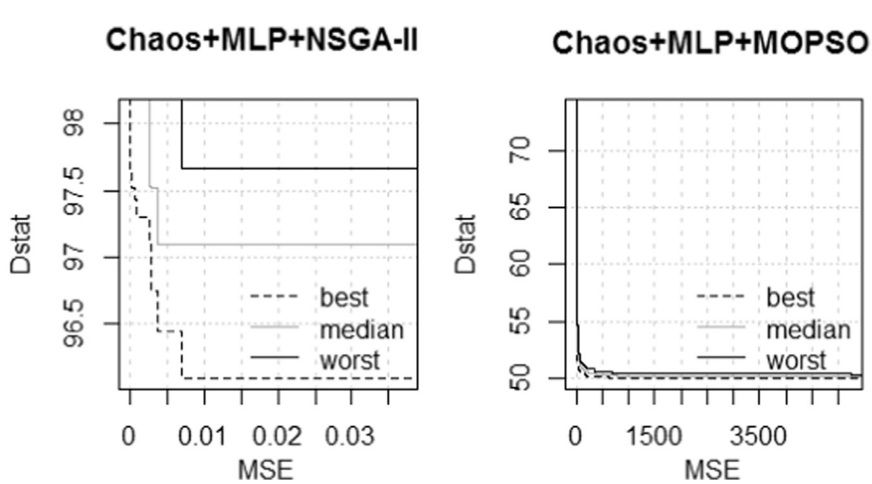

Fig. 11. EAF plot of both MOEA-based hybrids for JPY/USD.

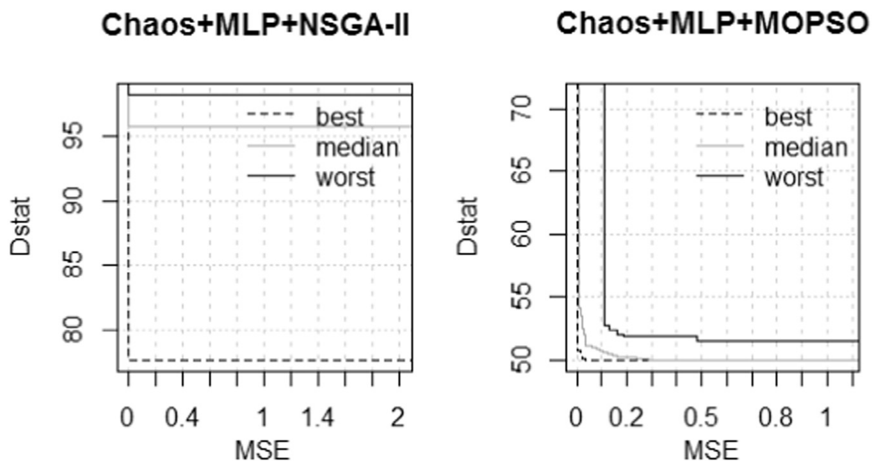

Fig. 12. EAF plot of both MOEA-based hybrids for GBP/USD.

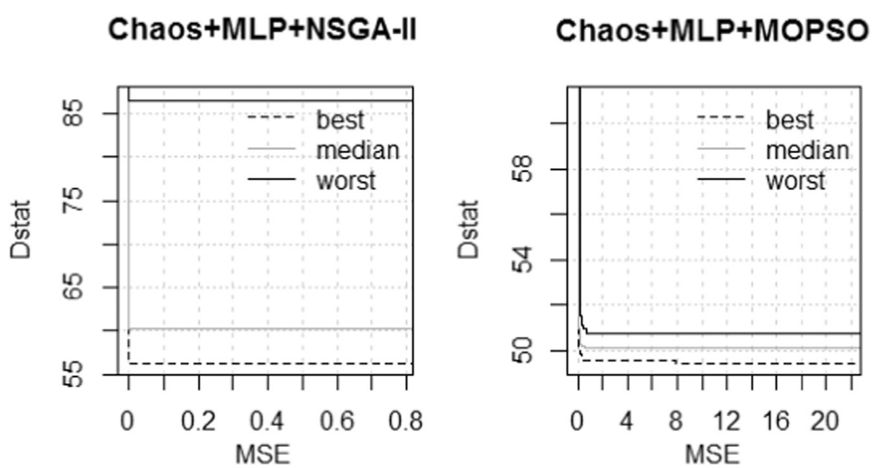

Fig. 13. EAF plot of both MOEA-based hybrids for EUR/USD.

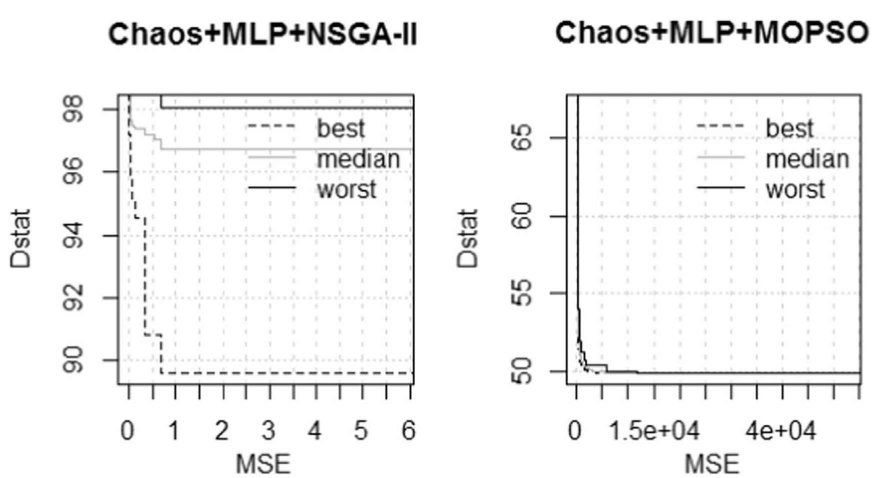

Fig. 14. EAF plot of both MOEA-based hybrids for Gold Price (USD).

problems and help in identifying certain algorithmic behaviors in a graphical way. The EAF plot (http://artax.karlin.mff.cuni.cz/r-help/library/eaf/html/eaf-package.html) depicts empirical k% attainment surfaces out of many runs. An empirical k% attainment surface is the line delimiting the objective space attained by at least k% of the runs of a multi-objective optimization algorithm. The best attainment surface is the region attained by at least one run, the median attainment surface is 50% attainment surface and the worst attainment surface is 100% attainment surface (https://eden.dei.uc.pt/paquete/papers/gecco2010.pdf).

Finally, the above forecasting accuracy measures including MSE, Dstat, and Theil's U do not formally test whether one method is statistically significantly different from another method or not. For this purpose, there is one popular test proposed by Diebold and Mariano [25]. We employed this test statistic that is implemented as a part of 'forecast' package from archieves of R (https://cran.r-project.org/web/packages/forecast/) to check whether Chaos+QRRF is performing statistically significantly different from other Chaos-based Forex rate forecasting models on average or not. Table 9 presents the results of

Table 9
Diebold Mariano Test results of test sets of all datasets.

Forecasting Model	JPY/USD	GBP/USD	EUR/USD	Gold Price
Chaos+MLP+NSGA-II Vs				
Chaos+MLP	26.29157	23.20075	24.375	23.675
Chaos+PSO	19.22925	17.79041	18.68052	17.79041
Chaos+MOPSO	19.7009	19.82007	19.76009	19.3557
Chaos+NSGA-II	18.69182	21.95429	19.7652	20.8796
Chaos+MLP+PSO	14.81932	20.93763	19.8445	19.95635
Chaos+MLP+MOPSO	14.2127	20.01934	18.87687	20.95845

the Diebold-Mariano test on three datasets. If the test statistic is less than or equal to 0.05, then the corresponding model is performing equally accurate with Chaos+MLP+NSGA-II. From the table, it is clear that Chaos+MLP+NSGA-II is superior to all models on four datasets.

7. Conclusion

The paper proposed novel three-stage financial time series forecasting models namely Chaos+MLP+MOPSO and Chaos+MLP+NSGA-II. The results in terms of MSE and Dstat on test datasets indicated that the Chaos+MLP+NSGA-II outperformed the hybrids including Chaos+MLP+MOPSO, Chaos+MLP+PSO, Chaos+PSO, Chaos+MOPSO, Chaos+NSGA-II, Chaos+GRNN, Chaos+GMDH, Chaos+CART, Chaos+CART-EB, Chaos+MARS, Chaos+TreeNet, Chaos+LASSO and Chaos+RFTE. Systematic modeling of chaos present in the datasets using chaos theory and later by a popular and powerful neural network along with the application of NSGA-II for predicting the residuals is the single most advantage of the current research since this combination yielded excellent MSE and Dstat values on the test data across the four datasets. Thiels inequality coefficient values and DM Test statistic also suggest the superiority of the Chaos+MLP+NSGA-II hybrid. The results are encouraging and we suggest further use of the proposed models in similar other financial and non-financial data.

References

[1] D. Pradeepkumar, V. Ravi, FOREX Rate Prediction Using Chaos, Neural Network and Particle Swarm Optimization, in: Proceedings of the 5th International Conference on Swarm Intelligence, ICSI 2014, volume LNCS 8795, Springer International Publishing, Hefei, China, 17–20 October 2014, pp. 363–375.

[2] D. Pradeepkumar, V. Ravi, Forex rate prediction: A hybrid approach using chaos theory and multivariate adaptive regression splines, in: Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Springer, Singapore, Bhubaneshwar, Odisha, India, 16-17 September 2016, 2017, pp. 219–227.

[3] T.-C. Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (2011) 164–181.

[4] P. Esling, C. Agon, Time-series data mining, ACM Comput. Surv. 45 (2012) 1–34.

[5] T. Mitsa, Temporal Data Mining, Chapman & Hall/ CRC Data mining and knowledge discovery series, CRC Press, US, 2010.

[6] J. Yao, C. Tan, A case study on using neural networks to perform technical forecasting of forex, Neurocomputing 34 (2000) 79–98.

[7] C. Dhanya, D. Nagesh Kumar, Nonlinear ensemble prediction of chaotic daily rainfall, Adv. Water Resour. 33 (2010) 327–347.

[8] T. Hellstrom, K. Holmstrom, Predicting the Stock Market, Technical Report, Technical Report IMa-TOM-1997-07, Center of Mathematical Modeling, Department of Mathematics and Physis, Malardalen University, Vasteras, Sweden, 1998.

[9] M. Magdon-ismail, A. Nicholson, Y. Abu-mostafa, Financial Markets: Very Noisy Information Processing, Proc. IEEE 86 (1998).

[10] G. Box, G. Tiao, Intervention analysis with applications to economic and environmental problems, JASA 70 (1975) 70–79.

[11] G. Box, G. Jenkins, Time series analysis: forecasting and control, Holden-Day, San Francisco, 1976.

[12] S. Makridakis, S. Wheelwright, R. Hyndman, Forecasting: methods and applications, edition 3 ed., New York: John Wiley & Sons, 1998.

[13] D.J. Reid, Combining three estimates of gross domestic product, Economica 35 (1968).

[14] J.M. Bates, C.W.J. Granger, The combination of forecasts, Oper. Res. Q. 20 (1969) 451–468.

[15] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, R. Winkler, The accuracy of extrapolation (time series) methods: Results of a forecasting competition, J. Forecast. 1 (1982) 111–153.

[16] E. Pelikan, C. De Groot, D. Wurtz, Power consumption in West-Bohemia: improved forecasts with decorrelating connectionist networks, Neural Netw. World 2 (1992) 701–712.

[17] I. Ginzburg, D. Horn, Combined neural networks for time series analysis, in: Advances in Neural Information Processing Systems 6, [Proceedings of the 7th NIPS Conference,Denver, Colorado, USA, 1993], 1993, pp. 224–231.

[18] R.T. Clemen, Combining forecasts: A review and annotated bibliography, Int. J. Forecast. 5 (1989) 559–583.

[19] C. Chatfield, Model uncertainty and forecast accuracy, J. Forecast. 15 (1996) 495–508.

[20] J. Poincaré, Sur le problème des trois corps et les équations de la dynamique. Divergence des séries de M. Lindstedt, Acta Math. 13 (1890) 1–270.

[21] G.P. Williams, Chaos Theory Tamed, Joseph Henry, Press, Washington, D.C, 1997.

[22] G. Zhang, B. Patuwo, M.Y. Hu, Forecasting with artificial neural networks:: The state of the art, Int. J. Forecast. 14 (1998) 35–62.

[23] C. A. Coello Coello, G. B. Lamont, D. A. Van Veldhuisen, Evolutionary algorithms for solving multi-objective problems, 2 ed., Springer series of Genetic Algorithms and Evolutionary Computation., 2007.

[24] H. Theil, Applied economic forecasting, North-Holland Pub. Co., Amsterdam, 1966.

[25] F. Diebold, R. Mariano, Comparing predictive accuracy, J. Bus. Econ. Stat. 13 (1995) 253–263.

[26] T. Temizel, M. Casey, A comparitive study of autoregressive neural network hybrids, Neural Netw. 19 (2005) 781–789.

[27] N.G. Pavlidis, D.K. Tasoulis, M.N. Vrahatis, Financial forecasting through unsupervised clustering and evolutionary trained neural networks, in: Proceedings of 2003 Congress on Evolutionary Computation, CEC'03, volume 4, Canberra, Australia, 8–12 Dec, 2003, pp. 2314–2321.

[28] S.-C. Huang, P.-J. Chuang, C.-F. Wu, H.-J. Lai, Chaos-based support vector regressions for exchange rate forecasting, Expert Syst. Appl. 37 (2010) 8590–8598.

[29] G. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159–175.

[30] A.-S. Chen, M.T. Leung, Regression neural network for error correction in foreign exchange forecasting and trading, Comput. Oper. Res. 31 (2004) 1049–1068.

[31] H. Ince, T.B. Trafalis, A hybrid model for exchange rate prediction, Decis. Support Syst. 42 (2006) 1054–1062.

[32] M. Khashei, S. Reza Hejazi, M. Bijari, A new hybrid artificial neural networks and fuzzy regression model for time series forecasting, Fuzzy Sets Syst. 159 (2008) 769–786.

[33] L. Yu, K.K. Lai, S. Wang, Multistage, RBF neural network ensemble learning for exchange rates forecasting, Neurocomputing 71 (2008) 3295–3302.

[34] J.-F. Chang, C.-W. Chang, W.-Y. Tzeng, Forecasting Exchange Rates Using Integration of Particle Swarm Optimization and Neural Networks, in: Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC), Kaohsiung, Taiwan, 07–09 Dec, 2009, pp. 660–663.

[35] A. Chaudhuri, K. De, Time Series Forecasting Using Hybrid Neuro-Fuzzy Regression Model, in: Proceedings of the 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, RSFDGrC 2009, Delhi, India, LNCS 5908, December 15-18, 2009, pp. 369–381.

[36] J.-F. Chang, Y.-J. Lee, Performance comparison between Genetic Algorithm and Particle Swarm Optimization based on back propagation network in forecasting exchange rate, J. Stat. Manag. Syst. 13 (2010) 805–821.

[37] I. Gheyas, L. Smith, A novel neural network ensemble architecture for time sereis forecasting, Neurocomputing 74 (2011) 3855–3864.

[38] J.-F. Chang, P.-Y. Hsieh, Particle Swarm Optimization based on BackPropagation Network Forecasting Exchange Rates, ICIC Int. 7 (2011) 6837–6847.

[39] E. Hadavandi, A. Ghanbari, S. Abbasian-Naghneh, Developing a Time Series Model Based On Particle Swarm Optimization for Gold Price Forecasting, in: Proceedings of the 2010 Third International Conference on Business Intelligence and Financial Engineering, Hong Kong, China, 2010, pp. 337–340.

[40] C. Li, C. W. Lin, H. Huang, Neural Fuzzy Forecasting of the China Yuan to US Dollar Exchange Rate – A Swarm Intelligence Approach, in: Proceedings of the Second International Conference, ICSI 2011, LNCS 6728, Chongqing, China, June 12-15, 2011, pp. 616–625.

[41] C.H. Aladag, U. Yolcu, E. Egrioglu, A.Z. Dalar, A new time invariant fuzzy time series forecasting method based on particle swarm optimization, Appl. Soft Comput. 12 (2012) 3291–3299.

[42] V. Ravi, R. Lal, N. Kiran, Foreign Exchange Rate Prediction using Computational Intelligence Methods, Int. J. Comput. Inf. Syst. Ind. Manag. Appl. 4 (2012) 659–670.

[43] G. Sermpinis, K. Theofilatos, A. Karathanasopoulos, E. Georgopoulos, C. Dunis, A Hybrid Radial Basis Function and Particle Swarm Optimization Neural Network Approach in Forecasting the EUR/GBP Exchange Rates Returns, in: Proceedings of the 13th International Conference on Engineering Applications of Neural Networks, EANN 2012, CCIS 311, London, UK, 20-23 September 2012, pp. 413–422.

[44] G. Sermpinis, K. Theofilatos, A. Karathanasopoulos, E.F. Georgopoulos, C. Dunis, Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and Particle Swarm Optimization, Eur. J. Oper. Res. 225 (2013) 528–540.

[45] J. Donate, P. Cortez, G.G. Sanchez, A. Migue, Time sereis forecasting using a weighted cross-validation evolutionary artificial neural network ensemble, Neurocomputing 109 (2013) 27–32.

[46] M. Pulido, P. Melin, O. Castillo, Optimization of ensemble neural networks with

fuzzy integration using the particle swarm algorithm for the US Dollar/MX time series prediction, in: 2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW), Boston, MA, USA, 24-26 June 2014, pp. 1–7.
[47] M. Rout, B. Majhi, R. Majhi, G. Panda, Forecasting of currency exchange rates using an adaptive ARMA model with differential evolution based training, J. King Saud. Univ. - Comput. Inf. Sci. 26 (2014) 7–18.
[48] P.R. Jena, R. Majhi, B. Majhi, Development and performance evaluation of a novel knowledge guided artificial neural network (KGANN) model for exchange rate prediction, J. King Saud. Univ. - Comput. Inf. Sci. 27 (2015) 450–457.
[49] F. Shen, J. Chao, J. Zhao, Forecasting exchange rate using deep belief networks and conjugate gradient method, Neurocomputing 167 (2015) 243–253.
[50] A.J. Hussain, D. Al-Jumeily, H. Al-Askar, N. Radi, Regularized dynamic self-organized neural network inspired by the immune algorithm for financial time series prediction, Neurocomputing 188 (2016) 23–30.
[51] G. Sermpinis, C. Stasinakis, K. Theofilatos, A. Karathanasopoulos, Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms-Support vector regression forecast combinations, Eur. J. Oper. Res. 247 (2015) 831–846.
[52] G. Svitlana, Neural networks performance in exchange rate prediction, Neurocomputing 172 (2016) 446–452.
[53] K. Seng, K. Tse, Non-linear time series prediction based on Lyapunov theory-based fuzzy neural network and multiobjective genetic algorithm, in: AI 2003, LNAI Vol. 2903, 2003, pp. 890–898.
[54] S.K. Mishra, G. Panda, B. Majhi, R. Majhi, Improved Portfolio Optimization Combining Multiobjective Evolutionary Computing Algorithm and Prediction Strategy, in: Proceedings of the World Congress on Engineering 2012, volume I, London, U.K., 4-6 July 2012, pp. 4–8.
[55] M. Rafiul Hassan, B. Nath, M. Kirley, J. Kamruzzaman, A hybrid of multiobjective Evolutionary Algorithm and HMM-Fuzzy model for time series prediction, Neurocomputing 81 (2012) 1–11.
[56] A. Ronay Ak, A. Ronay Ak, A. Li, A. Vitelli, A. Zio, Multi-objective Genetic Algorithm Optimization of a Neural Network for Estimating Wind Speed Prediction Intervals, Technical Report, hal 00864850, ver 1, 2013.
[57] N. Packard, J. Crutcheld, J. Farmer, R. Shaw, Geometry from time series, Phys. Rev. Lett. 45 (1980) 712–716.
[58] F. Takens, Detecting strange attractors in turbulence, in: Dynamical Systems and Turbulence, Volume 898 of the series Lecture Notes in Mathematics, Springer Berlin Heidelberg, 1981, pp. 366–381.
[59] A. Saïda, Using the Lyapunov Exponent as a Practical Test for Noisy Chaos, SSRN Electronic Journal (2007).
[60] A. Lyapunov, Probl`eme ǵeneral de la stabilit´e du mouvement, Ann. Math. Stud. 17 (1977).
[61] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control 19 (1974) 716–723.
[62] L. Cao, Practical method for determining the minimum embedding dimension of a scalar time series, Phys. D: Nonlinear Phenom. 110 (1997) 43–50.
[63] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 318–362.
[64] K. Deb, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, New York, NY, USA, 2001.
[65] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C.A. Coello Coello, A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I, IEEE Trans. Evolut. Comput. 18 (2014) 4–19.
[66] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C.A. Coello Coello, Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II, IEEE Trans. Evolut. Comput. 18 (2014) 20–35.
[67] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, Perth, Australia, 27 Nov.-1 Dec., 1995, pp. 1942–1948.
[68] C. Coello Coello, G. Pulido, M. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evolut. Comput. 8 (2004) 256–279.
[69] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: nsga-ii, IEEE Trans. Evolut. Comput. 6 (2002) 182–197.
[70] N. Srinivas, K. Deb, Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolut. Comput. 2 (1994) 221–248.
[71] S. Makridakis, M. Hibon, Evaluating accuracy (or error) measures, Insead (1995) 1–41.
[72] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: IEEE World Congress on Computational Intelligence, Anchorage, Alaska, 4-9 May 1998, pp. 69–73.
[73] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Congress on Evolutionary Computation-CEC99, IEEE, Washington D.C., USA, 6–9 July 1999, pp. 1931–1938.
[74] S. Das, A. Abraham, A. Konar, Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives, Stud. Comput. Intell. 116 (2008) 1–38.
[75] G. Venter, J. Sobieszczanski-Sobieski, Particle Swarm Optimization, in: Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics, Denver, Colorado, 22–25 April 2002, pp. 1–9.
[76] C.M. Fonseca, V.G. da Fonseca, L. Paquete, Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function, in: Evolutionary Multi-Criterion Optimization. in: Proceedings of the Third International Conference, EMO 2005, LNCS 3410, Berlin: Springer, 2005, pp. 250–264.
[77] C.M. Fonseca, A.P. Guerreiro, M. López-Ibánez, L. Paquete, On the computation of the empirical attainment function, in: Proceedings of the Sixth International Conference on Evolutionary Multi-Criterion Optimization, EMO 2011, LNCS 6576, Berlin: Springer, 2011, pp. 106–120.