Neural Comput & Applic (2019) 31:577–592
https://doi.org/10.1007/s00521-017-3089-2

ORIGINAL ARTICLE

Stock price prediction using hybrid soft computing models incorporating parameter tuning and input variable selection

Mustafa Göçken[1] · Mehmet Özçalıcı[2] · Aslı Boru[1] · Ayşe Tuğba Dosdoğru[1]

Received: 24 September 2016 / Accepted: 15 June 2017 / Published online: 3 July 2017

Abstract Over the years, high-dimensional, noisy, and time-varying natures of the stock markets are analyzed to carry out accurate prediction. Particularly, speculators and investors are understandably eager to accurately predict stock price since millions of dollars flow through the stock markets. At this point, soft computing models have empowered them to capture the data patterns and characteristics of stock markets. However, one of the open problems in soft computing models is how to systematically determine architecture of models for given applications. In this study, Harmony Search is utilized to optimize the architecture of Neural Network, Jordan Recurrent Neural Network, Extreme Learning Machine, Recurrent Extreme Learning Machine, Generalized Linear Model, Regression Tree, and Gaussian Process Regression for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead stock price prediction. The experimental results show worthy findings of stock market behavior over different prediction terms and stocks. This study also helps researchers understand which prediction model performed the best and how different conditions affect the prediction accuracy of the models. Proposed hybrid models can be successfully used by speculators and investors to make the investment or to hedge against potential risk in stock markets.

Keywords Jordan Recurrent Neural Network · Recurrent Extreme Learning Machine · Generalized Linear Model · Regression Tree · Gaussian Process Regression · Stock price prediction

1 Introduction

Stock market can be defined as a complex nonlinear dynamic system. Many factors have an influence on the stock market, and also complex and subtle interrelation are available between these factors [1, 2]. Therefore, analysis of the stock market is one of the interesting points of stock market research over the years. Especially with the stock market continued to heat up, accurate stock market prediction has become one of the key issues for countries all over the world. One of the main ideas to remarkable stock market research is providing the best prediction results using the minimum relevant input data [3]. For these purposes, a great deal of attention has been focused on the idea of using soft computing models for stock market prediction since correctly designed models can easily converge on the optimal result. However, their performance depends on parameter tuning and input variable selection. Particularly, variable selection is a central issue for model developer. It improves the performance of the proposed model. In addition, variable selection not only provides faster and more cost-effective predictors but also ensures a better understanding of the fundamental system [4]. Variable

✉ Mustafa Göçken
mgocken@adanabtu.edu.tr

Mehmet Özçalıcı
mozcalici@kilis.edu.tr

Aslı Boru
aboru@adanabtu.edu.tr

Ayşe Tuğba Dosdoğru
adosdogru@adanabtu.edu.tr

[1] Industrial Engineering Department, Adana Science and Technology University, Yeşiloba Yerleşkesi, Yeşiloba mah., Ögretmenler Bulvarı, 46278 sok., No: 3, 01180 Seyhan, Adana, Turkey

[2] International Trade and Logistics Department, Kilis 7 Aralık University, Kilis, Turkey

selection models are essentially divided into filter models, wrapper models, and embedded models. Filter models choose subsets of input variables as a preprocessing step. Wrapper models use the learning machine of interest as a black box to evaluate input variable subsets considering their predictive power. Embedded models perform input variable selection in the process of training and are normally specific to given learning machines [4]. It should be emphasized that not a single variable selection model is universally superior to the others. To specify which variable selection model is best to use and how to use it, one should first determine the purpose of variable selection and then study the characteristics of the dataset at hand [5].

To determine optimal settings of input variables and to tune parameters for stock price prediction, Harmony Search (HS) is used for Neural Network (NN), Jordan Recurrent Neural Network (JRNN), Extreme Learning Machine (ELM), Recurrent Extreme Learning Machine (RELM), Generalized Linear Model (GLM), Regression Tree (RT), Gaussian Process Regression (GPR) in this study. The selection of input variables strongly influences complexity and performance of models, but the performance of models is also affected by different factors such as the number of hidden neurons and the transfer function type. Too few or too many hidden neurons can lead to underfitting or overfitting issues, respectively, that greatly degrade the generalization capability to lead with significant deviation in prediction. In addition, the determination of transfer functions can strongly affect complexity and performance of the models. Hence, what kinds of transfer functions should be selected is very important.

In literature, many variants of the soft computing based architecture were created to improve the performance of the models. For example, Zhu et al. [6] proposed an evolutionary ELM model using differential evolution (DE), which is utilized to find the optimal input weights and hidden biases, and Moore–Penrose (MP)-generalized inverse that is utilized to define the output weights. Evolutionary ELM provides an opportunity to reduce the hidden neurons and to improve the generalization performance. Suresh et al. [7] used a real coded genetic algorithm (RCGA) to determine proper values for the free parameters in ELM whose input weights, bias values, and the number of hidden neurons are specified automatically. RCGA-based ELM has the ability to find small network to approximate the classifier function, but the time taken to receive the results is more. Lan et al. [8] proposed a constructive model for hidden neurons selection to become stable and to handle the architectural design of ELM network. Basically, proposed model is to identify the significance of each hidden neuron and to determine the optimal subset of hidden neurons achieving comparable generalization performance. Saraswathi et al. [9] used a particle swarm optimization (PSO), integer-coded genetic algorithm, and ELM for accurate gene selection and sparse data classification. In the study, the optimal input weights are determined by PSO algorithm and integer-coded genetic algorithm, which reduce the gene numbers, and are used to reduce the computational effort. The proposed model provides high classification accuracy with just twelve genes. Lahoz et al. [10] assigned the biases and hidden weights of the single-hidden layer feedforward neural network (SLFN) using bi-objective microgenetic algorithm that gives reasonable results while maintaining the time of execution. Huang and Lai [11] employed the PSO to optimize the structural risk minimization function for the optimal number of hidden neurons in ELM. To improve ELM classifier's generalization performance, Xue et al. [12] proposed a variable-length PSO algorithm to determine the number of neurons in layer as well as corresponding input weights and hidden biases. Bazi et al. [13] applied the DE to analyze the model selection problem of ELM. Also, orthogonal crossover is used to improve the search ability of the DE. The experimental result showed that DE-based ELM is faster than the DE-based support vector machine with accurate and efficient classification. Hegazy et al. [14] used a flower pollination algorithm that selects input weights and hidden biases to create more compact network structure than traditional ELM model for monthly stock price prediction. The proposed model has the ability to achieve lowest error value with little advance since its parameter is few and can be easily tuned. To improve the balance between explorative and exploitive power, Yang et al. [15] proposed a differential evolution coral reef optimization (DECRO), where the DE is utilized to perform the broadcast spawning. It is determined that DECRO-based ELM is prone to improve the prediction accuracy of DE-based ELM and coral reef optimization-based ELM. Furthermore, it enhances the prediction speed of the ELM. In recent years, RELM is also used to improve the prediction performance of ELM [16, 17].

Ruxanda and Badea [18] built various configurations of NNs to make predictions on Bucharest Stock Market Index and then evaluated NN models in terms of prediction errors. In the study, the number of hidden neurons is altered between 2 and 6, logistic sigmoid and hyperbolic tangent sigmoid are used as a type of activation function, and gradient descent and Broyden–Fletcher–Goldfarb–Shanno method are applied as a training algorithm. Developing more realistic models basically depends on NN parameters. To increase the performance of NN models, a significant amount of studies (especially, hybrid NN models) have been done in this field. Guresen et al. [19] reviewed artificial NN and hybrid models for time series forecasting in a detailed way and evaluated the performance of dynamic artificial NN, hybrid NNs, and multilayer perceptron in forecasting time series used in market values. Hsieh et al. [20] presented an integrated system for stock price prediction that used wavelet transforms and Recurrent Neural Network (RNN) whose weights and biases are determined by artificial bee colony algorithm. To ensure that the application of the model is sufficiently robust, proposed model is applied to four stock markets. Wei and Cheng [21] used RNN to build a prediction model and proposed three refined processes in the hybrid model where essential technical indicators are selected from popular indicators by a correlation matrix, stepwise regression, and a decision tree. The study supported that selecting the important technical indicators reduces forecasting errors more effectively. Zahedi and Rounaghi [22] used NN to predict stock price and applied 20 accounting variables in principal component analysis model to evaluate their effects on stock price and to identify effective factors in stock price by using real values. To reduce the computational complexity, Anish and Majhi [23] combined the variable selection with a feedback type of the functional link artificial NN model trained with recursive least square algorithm for efficient prediction of stock prices. Dash et al. [24] presented a hybrid feedforward functional link dynamic neural network model to ensure a trade-off between speed and accuracy.

To summarize, nowadays one is equipped with a multitude of models, but the choice of the models that allow for successful and efficient solution of a particular problem is usually not trivial. The main problem is parameter tuning since no definite and explicit model is available to determine optimal parameter setting for prediction model. In this case, metaheuristics can be used to systematically assign the optimal setting of parameters. In this study, HS is used to tune optimal parameter for proposed models. The reason of selecting HS can be summarized as follows: (1) there is no need to use initial values for the decision variables; (2) the derivative information is unnecessary since a stochastic random search is used by HS; (3) HS is able to adapt easily to a broad class of optimization problems due to its simplicity [25]. In the light of previous studies, it can also be said that HS does not require parameter fine-tuning to reach high-quality solutions. Thus, it is less sensitive to the values of the chosen parameters [26]. It is hoped that the present paper may help find the best model, or at least filter out the ones that are not promising in stock price prediction. The main aims of this study are summarized as follows:

- Select the relevant technical indicators for NN, JRNN, ELM, RELM, GLM, RT, and GPR models,
- Determine the proper number of hidden neurons for NN, JRNN, ELM, RELM models,
- Determine the proper number of context neurons for JRNN and RELM models,
- Determine the appropriate transfer function type for NN, JRNN, ELM, RELM models,
- Create HS-NN, HS-JRNN, HS-ELM, HS-RELM, HS-GLM, HS-RT, and HS-GPR models for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction,
- Testing of the models with three stocks that are selected among the BIST-100 companies, and their names are Eregli Iron and steel company (EREGL), Eczacıbası Pharmaceutical industry (ECILC), and Afyon cement industry (AFYON),
- To compare the forecasting performance of models, five performance measures namely Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Theil's Inequality Coefficient (TheilU), and Directional Prediction Statistics (DS) are utilized.

The remainder of this paper is created as follows. The basic structure of the proposed models is described in Sect. 2. In Sect. 3, a detailed analysis of the models is given. Section 4 is devoted to conclusions.

2 Proposed models

The lack of certain architectures in a particular model is still a controversial issue. For soft computing models, the performance of the proposed models is highly dependable not only on input variable selection but also on many architectural parameters. Therefore, we firstly used HS to tune the optimum parameters to provide the desired output. Then, we synthesized determined parameters and constructed the hybrid models for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction, and three stocks are used.

Overall, dataset is divided into training and testing datasets. Training dataset covers the period from April 17, 2013, up to September 11, 2015, and there are 1200 trading sessions. Testing dataset cover the period from September 14, 2015, up to November 30, 2015, and there are 60 trading sessions. Original dataset is transformed to a normalized dataset to increase convergence ability and decrease the noise. In this study, sigmoid normalization is applied as normalization technique using the following equation:

$$\mathrm{x}_{\mathrm{new}} = \frac{1}{1+e^{-\lambda x_{\mathrm{old}}}} \tag{1}$$

where x_{new} represents the value after normalization and x_{old} denotes the value before normalization. $\lambda = 1/x_{\max}$, and $x_{\max}$ is the maximum value of a dataset.

In stock price prediction, determination of relevant technical indicators is one of the critical steps. A large number of technical indicators can cause irrelevant information, while a small number of technical indicators could be insufficient for proposed models. Since the chosen relevant variables subset could signify better the original character of dataset, prediction with these input variables could increase the accuracy and efficiency. In this study, relevant technical indicators are selected from initial variable pool (Table 1). Note that initial variable pool is the same as with the study of Göçken et al. [27], but only today's close–previous close price was extracted.

For particular stock or index, the fitness of a technical indicator is dynamic. Stocks can also demonstrate conflicting trends on different time. The optimization of technical indicators should be adaptive to changes in the fitness and to generate new instances of the indicator as required [28]. Relevant technical indicators can change according to the proposed models. In this paper, the number of candidate variables to be selected is fixed at five for each model. When input variables (technical indicators) are higher than five, the computational time increases. In addition, the basic idea to effective forecasting in stock market is ensuring the best results using the least complex model with the minimum required input data [3]. Therefore, the number of selected technical indicators is restricted to be five in HS to improve performance of the proposed hybrid models. The complete set of different transfer functions for each model is given in Tables 2 and 3.

It is known that the selection of transfer function strongly influences the complexity, but how to determine a suitable transfer function is still an unresolved problem. Actually, the choice of transfer functions depends on the model type and the problem type. Hence, this study is to be useful for researchers working on transfer functions.

What's more, we used HS as a simple and efficient model to set the proper number of hidden and context neurons since the number of neurons directly influences the generalization performance of the models.

2.1 Harmony Search (HS)

HS mimics the improvisation process of musicians. To find a best harmony, each musician plays a note. In same manner, the decision variables of the cost (or profit) function can be typically represented by musicians in optimization problems. HS attempts to determine an optimal solution vector in this function where each decision variable (musician) produces a value (note) [29]. In general optimization problem, basic steps of the HS can be summarized as follows [30]:

Table 1 Initial variable pool

Index	Variable name	Index	Variable name
1	Previous close	23	Close price accelerator
2	Previous high	24	Opening price momentum
3	Previous low	25	High price momentum
4	Previous open	26	Low price momentum
5	5-day simple moving average	27	Close price momentum
6	6-day simple moving average	28	Chaikin volatility
7	10-day simple moving average	29	K% stochastic
8	20-day simple moving average	30	D% stochastic
9	5-day exponential moving average	31	Slow K% stochastic
10	6-day exponential moving average	32	Slow D% stochastic
11	10-day exponential moving average	33	William %R
12	20-day exponential moving average	34	Relative strength index
13	5-day triangular moving average	35	Middle Bollinger band
14	6-day triangular moving average	36	Upper Bollinger band
15	10-day triangular moving average	37	Lower Bollinger band
16	20-day triangular moving average	38	Highest high value
17	Accumulation distribution oscillator	39	Lowest low value
18	Closing price moving average convergence divergence	40	Median price
19	9-day moving average of close moving average convergence divergence	41	Price rate of change
20	Opening price accelerator	42	Typical price
21	High price accelerator	43	Weighted close
22	Low price accelerator	44	William's accumulation/distribution line

Table 2 List of available transfer function types in HS-NN/HS-JRNN models

Transfer functions		Formula
1	Hyperbolic tangent sigmoid transfer function	$\mathrm{tansig}(n) = 2/(1 + \exp(-2 * n)) - 1$
2	Elliot sigmoid transfer function	$\mathrm{elliotsig}(n) = n/(1 + \mathrm{abs}(n))$
3	Log-sigmoid transfer function	$\mathrm{logsig}(n) = 1/(1 + \exp(-n))$
4	Radial basis transfer function	$\mathrm{radbas}(n) = \exp\left((-n)^2\right)$
5	Pure linear transfer function	$\mathrm{purelin}(n) = n$
6	Normalized radial basis transfer function	$\mathrm{radbasn}(n) = \exp\left((-n)^2\right)/\mathrm{sum}\left(\exp\left((-n)^2\right)\right)$
7	Soft max transfer function	$\mathrm{softmax}(n) = \exp(n)/\mathrm{sum}(\exp(n))$
8	Triangular basis transfer function	$\mathrm{tribas}(n) = \begin{cases} 1 - \mathrm{abs}(n), & \text{if } -1 \le n \le 1 \\ 0, & \text{otherwise} \end{cases}$
9	Net inverse transfer function	$\mathrm{netinv}(n) = 1/n$

Table 3 List of available transfer function types in HS-ELM/HS-RELM

Transfer functions		Formula
1	Sigmoid function	$G(a, b, \mathbf{x}) = 1/(1 + \exp(-(a\mathbf{x} + b)))$
2	Sine function	$G(a, b, \mathbf{x}) = \sin(a \times \mathbf{x} + b)$
3	Hard limit function	$G(a, b, \mathbf{x}) = \begin{cases} 1, & \text{if } a \times x + b \le 0 \\ 0, & \text{otherwise} \end{cases}$
4	Triangular basis function	$G(a, b, x) = \begin{cases} 1 - \lvert a \times x + b \rvert, & \text{if } \lvert a \times x + b \rvert \le 1 \\ 0, & \text{otherwise} \end{cases}$
5	Radial basis function	$G(a, b, \mathbf{x}) = \exp\left(-(a\mathbf{x} + b)^2\right)$

Step 1 The optimization problem is briefly described.

$$\begin{aligned} &\mathit{Minimize}\ (\text{or } \mathit{Maximize})\ f(\vec{x}) \\ &\text{subjected to } x_i \in X_i, \quad i = 1, 2, \ldots, N. \end{aligned} \tag{2}$$

where $f(\cdot)$ denotes a scalar objective function; $\vec{x}$ represents a solution vector including decision variables x_i; X_i denotes the set of possible range of values for each decision variable x_i and changes between the lower (Lx_i) and upper bounds (Ux_i) for each decision variable; and N represents the number of decision variables. In this step, harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and the number of improvisations (NI) are also initialized.

Step 2 HM is initialized using Eq. (3).

$$x_i^j = Lx_i + rand()(Ux_i - Lx_i) \tag{3}$$

where $j = 1, 2, 3, \ldots, HMS$ and $rand()$ is a uniformly distributed random number between 0 and 1.

Step 3 This step is responsible for creating a new potential variation. The new harmony vector $(\vec{x}' = x_1', x_2', \ldots, x_N')$ is created using memory consideration, pitch adjustment, and random selection. The value of the decision variables for the new vector is selected from any of the values already existing in the current HM.

$$x_i' \in \begin{cases} x_i' \in x_i^1, x_i^2, \ldots \ldots x_i^{\mathrm{HMS}}, & \text{with probability HMCR} \\ x_i' \in X_i, & \text{with probability } 1 - \text{HMCR} \end{cases} \tag{4}$$

Then, parameter PAR is applied by using following equation:

$$x_i' = \begin{cases} x_i' \pm rand() \times bw & \text{with probability PAR} \\ x_i' & \text{with probability } (1 - \text{PAR}) \end{cases} \tag{5}$$

where bw represents an arbitrary distance bandwidth.

For JRNN and RELM models, HM includes five parts. The number of hidden neurons, the number of context neurons, and transfer function type are represented by first, second, and third parts, respectively. The fourth part denotes the information on whether the considered technical indicator is selected or not. The last part of the HM represents the objective function. Note that similar representation of the HM matrix can be found in [27]. For NN and ELM models, HM includes four parts. The number of hidden neurons and transfer function type are represented by first and second parts, respectively. The third part denotes the input variable selection. The final part of the HM represents the objective function. For GLM, RT, and GPR models, HM only includes two parts. First one

Table 4 HS parameter values

Parameter	Value
HMS	50
HMCR	0.95
PAR	0.3
Maximum number of iteration	50

denotes input variables, and second part represents the objective function.

Step 4 A new harmony vector ($\vec{x}' = x'_1, x'_2, \ldots . x'_N$) is improvised in HM. Then, HM is updated and the generated solution is evaluated. If its fitness value is better than the worst harmony in the HM, it will replace the worst harmony with the new one. Otherwise, it is eliminated.

Step 5 HS terminates after a certain maximum number of iterations. Otherwise, HS will repeat steps 3 and 4. HS parameter values for proposed models are given in Table 4. Note that these values were determined by considering upper bound of parameters in [31] where HMS value ranges between 10 and 50, value of HMCR ranges between 0.7 and 0.95, and PAR is set between 0.01 and 0.3.

2.2 Hybrid models

Although various models and algorithms have been presented in literature for input variable selection, determining the transfer function type is still controversial. However, the selection of transfer functions can strongly affect complexity and performance of models and has been said to play a significant role in the convergence of the models. Hence, we used HS for variable selection and parameter tuning including the determination of the transfer function type, and the optimal number of hidden and context neurons. After HS-NN, HS-JRNN, HS-ELM, HS-RELM, HS-GLM, HS-RT, and HS-GPR are created using all optimal solutions that are generated by HS. Thus, proposed models calculate an output to the problem based on the parameters specified by HS that takes a number of iterations for convergence. Note that the number of iteration in hybrid models is same with HS.

The general structure of proposed hybrid models is seen in Fig. 1. Initially, the dataset is divided randomly into four subsets of equal size. Then, each subset is divided to 80% of data as training and 20% of data as testing set. However, the percent distributions of training and testing dataset can be any other ratios such as 50/50, 60/40, 70/30, and 95/5. In subsequent sections, each step of the models is described to give the details of the hybrid modes.

2.2.1 Harmony Search-based Neural Network (HS-NN)

NNs are formed in layer by layer basis. Each layer can include one or more computational neurons. The jth neuron of a NN can be expressed as:

$$y_j = \varphi_j\left(\sum_i w_{ji}x_i - b_j\right) \tag{6}$$

where y_j is the output of jth neuron, w_{ji} is the connection weight between ith neuron and jth neuron, x_i is ith input, b_j is the bias at the jth neuron, and $\varphi_j(\cdot)$ denotes the transfer function at jth neuron. Since x represents the input (known) and variable y specifies the output (to be computed), we need information on the other remaining variables using a training process that may help determine the optimal values. More details can be found in [32].

The selected variables and optimized parameter values of HS-NN model are given in Table 5. Hyperbolic tangent sigmoid transfer function (1) and elliot sigmoid transfer function (2) are generally used in HS-NN. Also, the most commonly used number of hidden neurons is 20 neurons. Typical price (42) is the most commonly used technical indicator as input variables.

2.2.2 Harmony Search-based Jordan Recurrent Neural Network (HS-JRNN)

JRNN can be considered as an extended version of the NN models. It includes input, hidden, output, and context layers. Note that temporal information is extracted from the input data by means of context layer. In JRNN, connections between the context layers and the output layer are fixed with a weight of one. Weights are set to carry out a specific application. Then, learning algorithm is performed in JRNN and input data are used to calculate the output of the network. In proposed model, data from lower layer neurons are propagated forward to upper layer neurons through feedforward connection network (see Hikawa and Araga [33] for details).

For HS-JRNN, the number of input variables, the transfer function type, and the number of hidden and context neurons are optimized with HS algorithm (Table 6).

In HS-JRNN, two most commonly used transfer function types are pure linear transfer function and normalized radial basis transfer function. The most commonly used number of hidden layer neurons is 9 neurons, while the most commonly used number of neurons in the context layer is 10 neurons. K % stochastic (29) is the most commonly used technical indicator.

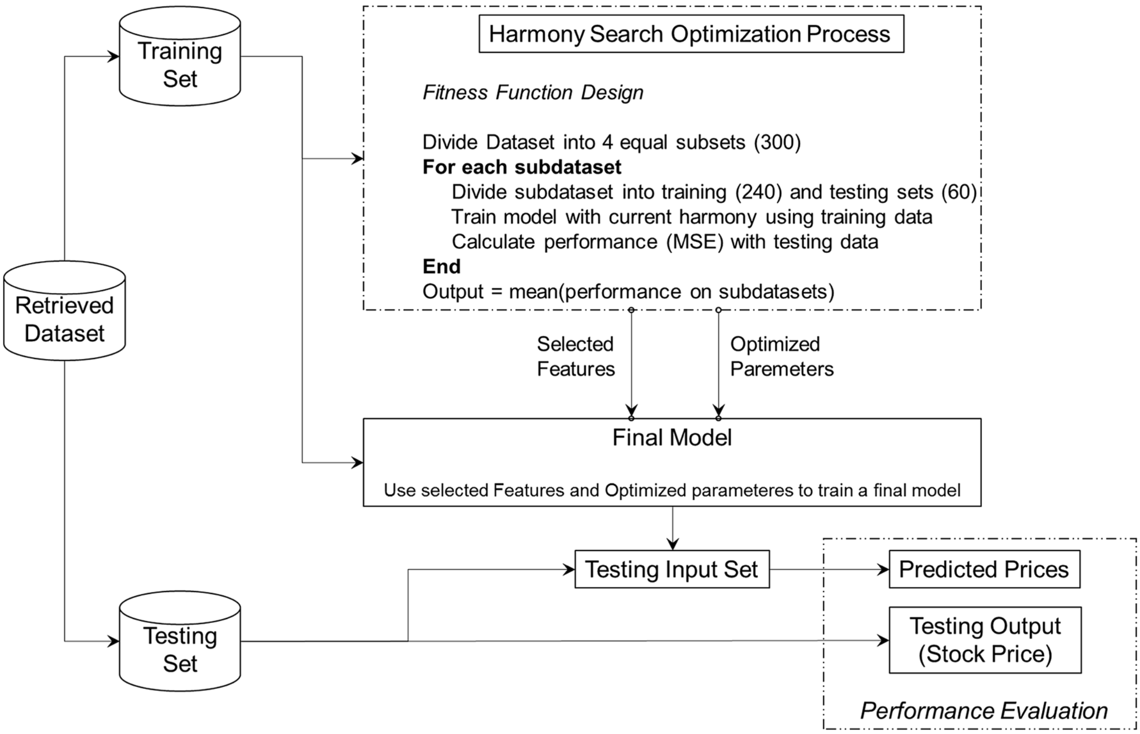
Training Set
Retrieved Dataset
Testing Set
Harmony Search Optimization Process
Fitness Function Design
Divide Dataset into 4 equal subsets (300)
For each subdataset
Divide subdataset into training (240) and testing sets (60)
Train model with current harmony using training data
Calculate performance (MSE) with testing data
End
Output = mean(performance on subdatasets)
Selected Features
Optimized Paremeters
Final Model
Use selected Features and Optimized parameteres to train a final model
Testing Input Set
Predicted Prices
Testing Output (Stock Price)
Performance Evaluation

Fig. 1 Block diagram of proposed hybrid models

Table 5 Selected variables and optimized parameter values for HS-NN

Prediction term	Stocks	Transfer function type	Number of hidden neurons	Selected variables' index				
1-day-ahead	ECILC	2	11	14	16	20	22	42
	EREGL	1	20	4	11	35	42	43
	AFYON	2	18	10	20	21	25	33
2-day-ahead	ECILC	1	20	4	11	35	42	43
	EREGL	1	20	4	11	35	42	43
	AFYON	2	18	10	20	21	25	33
3-day-ahead	ECILC	1	20	4	11	35	42	43
	EREGL	1	20	4	11	35	42	43
	AFYON	2	18	10	20	21	25	33
5-day-ahead	ECILC	1	20	4	11	35	42	43
	EREGL	4	20	15	24	25	26	28
	AFYON	5	7	18	26	30	37	41
7-day-ahead	ECILC	2	11	14	16	20	22	42
	EREGL	2	11	14	16	20	22	42
	AFYON	5	7	18	26	30	37	41
10-day-ahead	ECILC	8	10	10	16	23	29	43
	EREGL	6	3	13	15	22	28	44
	AFYON	8	10	10	16	23	29	43

2.2.3 *Harmony Search-based Extreme Learning Machine (HS-ELM)*

A new learning algorithm for the SLFN called the ELM overcomes the aforementioned disadvantages of NN. ELM avoids being stuck at local optimization and is reasonably faster than the traditional gradient-based algorithms. In addition, it ensures NN models with better generalization performance [15]. ELM has an ability to reach the smallest training error and obtain the smallest

Table 6 Selected variables and optimized parameter values for HS-JRNN

Prediction term	Stocks	Transfer function type	Number of hidden neurons	Number of context neurons	Selected variables' index				
1-day-ahead	ECILC	5	9	10	2	12	32	36	39
	EREGL	5	14	14	23	24	29	32	43
	AFYON	8	16	9	4	8	10	11	44
2-day-ahead	ECILC	5	11	2	6	29	30	31	32
	EREGL	6	9	10	2	6	11	29	38
	AFYON	8	16	9	4	8	10	11	44
3-day-ahead	ECILC	5	9	10	6	23	29	32	39
	EREGL	2	11	13	2	6	21	25	29
	AFYON	6	16	12	4	9	18	37	43
5-day-ahead	ECILC	5	9	10	6	23	29	32	39
	EREGL	2	11	13	2	6	21	25	29
	AFYON	6	29	12	7	16	26	29	36
7-day-ahead	ECILC	5	19	4	23	30	31	36	39
	EREGL	6	19	8	16	26	38	42	44
	AFYON	6	29	12	7	16	26	29	36
10-day-ahead	ECILC	6	10	1	9	14	15	35	39
	EREGL	4	10	15	10	19	20	22	23
	AFYON	8	9	7	2	12	18	23	25

norm of weights [34]. On the other hand, without the proper input layer weights and bias, more hidden layer neurons are required to enhance the performance of ELM [15]. In ELM, the input weights and hidden biases are randomly determined and MP-generalized inverse is utilized to define the output weights [6]. Details can be found in [34]. Briefly, the steps of the ELM are:

Given a training set $\Phi = \{(x_k, t_k) | (x_k) \in \mathbb{R}^n,$ $t_k \in \mathbb{R}, \quad k = 1, 2, \ldots, N\}$, hidden neuron transfer function type $h(x)$, and the number of hidden neurons, L,

1. Randomly initiate considering any continuous sampling distribution hidden neuron parameters.
2. Calculate the hidden layer output matrix H:

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \tag{7}$$

   where $h_i(x)$ denotes a transfer function of the ith neuron. The ith column of H represents the ith hidden neuron output vector with respect to inputs $x_1, x_2, \ldots, x_N$.
3. Calculate the output weights β_i:

$$\beta = H^{\dagger} T \tag{8}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_L]^T$ specifies the vector of the output weights, $T = [t_1, t_2, \ldots, t_N]^{\mathrm{T}}$ represents the training data output matrix, and $H^{\dagger}$ denotes the MP-generalized inverse of matrix H. β is a result of the minimization of approximation error:

$$\min \| H\beta - T \| \tag{9}$$

The output function of ELM is defined as (one output case):

$$f_L(x) = \sum_{i=1}^{L} f_i(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \tag{10}$$

where $f_i(x) = \beta_i h_i(x)$ is the weighted output of the ith hidden neuron.

The output function $f_L(x)$ is a linear combination of the transfer functions $h_i(x)$ [35].

In HS-ELM, triangular basis is used as transfer function type for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction models except 10-day-ahead prediction models for EREGL that is used sigmoidal transfer function type. The number of hidden neurons is usually 16 neurons. Previous high (2), 10-day simple moving average (7), 10-day triangular moving average (15), high price accelerator (21), and opening price momentum (24) are generally selected as technical indicators (Table 7).

2.2.4 Harmony Search-based Recurrent Extreme Learning Machine (HS-RELM)

RELM has been shown a novel training approach for a single-hidden layer JRNN whose output can be defined by Ertugrul [17]:

$$T = \sum_{j=1}^{m} \beta_j g\left(\sum_{i=1}^{n} w_{i,j} x_i + \sum_{i=n+1}^{n+r} w_{i,j} \delta(t - i + n) + b_j \right) \tag{11}$$

where m is the neuron numbers in the hidden layer and n is the neuron numbers in the input layers. δ denotes delay, t shows the instance order, r represents the employed context neuron numbers that are backward connections from the output layer to the input layer.

The feedbacks in RELM are taken as new inputs with delay and attached to the H matrix. Note that the major difference between RELM and ELM is that RELM is built for sequential (time ordered) datasets, but the order of the data in the dataset is not important in ELM. Details can be found in [17]. In this paper, the number of input variables, transfer function type, number of hidden, and context neurons, are optimized. In HS-RELM, sigmoidal transfer function type is applied for all models. The number of hidden layer neurons is generally 21 neurons, and the number of context neurons is usually 16 neurons for proposed models. In general, previous open (4), 10-day exponential moving average (11), William %R (33), lower Bollinger band (37), and median price (40) are selected as technical indicators (Table 8).

2.2.5 Harmony Search-based Generalized Linear Model (HS-GLM)

In GLMs, the predictor variables $X_j (j = 1, \ldots, p)$ are combined to generate a linear predictor (LP) that is related to the expected value $\mu = E(Y)$ of the response variable Y through a link function g(), such as:

$$g(E(Y)) = LP = \alpha + X^T \beta \tag{12}$$

where Y is the response variable, α is a constant denoted the intercept, $X = (X_1, \ldots, X_p)$ denotes a vector of p predictor variables, $\beta = \{\beta_1, \ldots, \beta_p\}$ represents the vector of p regression coefficients [36]. In HS-GLM, close price accelerator (23) is found to be the most commonly used technical indicator as seen in Table 9.

2.2.6 Harmony Search-based Regression Tree (HS-RT)

RT is an automatic classifier and creates a binary tree structure classifier. The repeated splits of subsets into two descendant subsets are used to produce the tree. Every split is an inquiry about the input variables, and the answers of "yes" and "no" lead to the left and right descendant subsets, respectively. Details of RT can be found in [37]. In

Table 7 Selected variables and optimized parameter values for HS-ELM

Prediction term	Stocks	Transfer function type	Number of hidden neurons	Selected variables' index				
1-day-ahead	ECILC	4	16	2	7	15	21	24
	EREGL	4	16	2	7	15	21	24
	AFYON	4	16	2	7	15	21	24
2-day-ahead	ECILC	4	16	2	7	15	21	24
	EREGL	4	16	2	7	15	21	24
	AFYON	4	16	2	7	15	21	24
3-day-ahead	ECILC	4	16	2	7	15	21	24
	EREGL	4	16	2	7	15	21	24
	AFYON	4	16	2	7	15	21	24
5-day-ahead	ECILC	4	16	2	7	15	21	24
	EREGL	4	16	2	7	15	21	24
	AFYON	4	16	2	7	15	21	24
7-day-ahead	ECILC	4	16	2	7	15	21	24
	EREGL	4	16	2	7	15	21	24
	AFYON	4	16	2	7	15	21	24
10-day-ahead	ECILC	4	22	3	10	17	18	28
	EREGL	1	18	5	11	21	25	37
	AFYON	4	16	2	7	15	21	24

Table 8 Selected variables and optimized parameter values for HS-RELM

Prediction term	Stocks	Transfer function type	Number of hidden neurons	Number of context neurons	Selected variables' index				
1-day-ahead	ECILC	1	21	16	4	11	33	37	40
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	21	16	4	11	33	37	40
2-day-ahead	ECILC	1	21	16	4	11	33	37	40
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	21	16	4	11	33	37	40
3-day-ahead	ECILC	1	21	16	4	11	33	37	40
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	21	16	4	11	33	37	40
5-day-ahead	ECILC	1	21	16	4	11	33	37	40
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	21	16	4	11	33	37	40
7-day-ahead	ECILC	1	21	16	4	11	33	37	40
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	21	16	4	11	33	37	40
10-day-ahead	ECILC	1	17	9	11	16	19	38	42
	EREGL	1	21	16	4	11	33	37	40
	AFYON	1	17	9	11	16	19	40	42

HS-RT model, 6-day simple moving average (6), 5-day triangular moving average (13), low price accelerator (22), close price accelerator (23), and K % stochastic (29) are selected as input variables for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction models and three stocks (ECILC, EREGL, and AFYON).

Table 9 Optimal variable subset for HS-GLM

Prediction term	Stocks	Selected variables' index				
1-day-ahead	ECILC	6	13	22	23	29
	EREGL	6	13	22	23	29
	AFYON	13	20	26	33	41
2-day-ahead	ECILC	6	13	22	23	29
	EREGL	18	23	35	38	40
	AFYON	13	20	26	33	41
3-day-ahead	ECILC	6	13	22	23	29
	EREGL	6	13	22	23	29
	AFYON	6	13	22	23	29
5-day-ahead	ECILC	6	13	22	23	29
	EREGL	6	13	22	23	29
	AFYON	5	19	21	23	39
7-day-ahead	ECILC	6	13	22	23	29
	EREGL	6	13	22	23	29
	AFYON	5	19	21	23	39
10-day-ahead	ECILC	8	21	26	35	37
	EREGL	6	13	22	23	29
	AFYON	6	13	22	23	29

2.2.7 *Harmony Search-based Gaussian Process Regression (HS-GPR)*

GPR is a flexible model to handle the problems of non-linear regression. In GPR, gaussian process is a collection of random variables $t = (t(x_1), t(x_2), \ldots)$ which have a gaussian joint distribution,

$$P(t|C, x_n) = \frac{1}{Z}\exp\left(-\frac{1}{2}(t-\mu)^T C^{-1}(t-\mu)\right) \tag{13}$$

for any input sets $\{x_n\}$. C represents the covariance matrix denoted by the covariance function $C(x_n, x_m; \Theta)$ which is parametrized by hyperparameters Θ, and μ denotes the mean function. GPR predict $\tilde{t}$ on the new data point $\tilde{x}$ giving predictive mean and variance of the posterior distribution.

$$\tilde{y}(\tilde{x}) = \mathbf{k}(\tilde{x}) C_N^{-1} t \tag{14}$$

$$\sigma_{\tilde{y}}^2(\tilde{x}) = C(\tilde{x}, \tilde{x}) - k(\tilde{x}) C_N^{-1} k(\tilde{x}) \tag{15}$$

where $\mathbf{k}(\tilde{x}) = (C(x_1, \tilde{x}), \ldots, C(x_N, \tilde{x}))$ represents the covariance between the training data and $\tilde{x}$, and C_N denotes the $N \times N$ covariance matrix of training data points given the covariance function $C.C_N^{-1}\mathbf{t}$ specifies independent of the new data. Details of GPR can be found in [38]. In this study, 6-day simple moving average (6), 5-day triangular moving average (13), low price accelerator (22), close price accelerator (23), and K % stochastic (29) are determined as the input variables for HS-GPR model.

3 Results and discussion

There exists ample evidence that stock markets serve as one of the leading indicator for the economy, and hence, stock price prediction at higher frequencies can provide early indication to the direction the economy might be headed, besides providing information about rising imbalances and risks. We presented how hybrid soft computing models are applied effectively to the problem of analyzing stock market and to prognosticate future price in a useful and applicable manner. We aimed at encouraging researchers to seek and embrace new opportunities of soft computing models in stock markets. In this study, HS is utilized to construct seven robust hybrid models for stock price prediction. To ensure that proposed hybrid models can be applicable in other stock markets, our work is also applied to three stocks (ECILC, EREGL, and AFYON). To clarify which prediction model performed the best under different conditions, five different statistical performance measures including MAPE, MAE, RMSE, TheilU, and DS are used. MAPE is a relative measure that denotes errors as a percentage of the actual data. It is simple to calculate and easy to understand [39]. MAPE is calculated as follows:

$$\mathrm{MAPE} = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{X_t - F_t}{X_t}\right| \tag{16}$$

F_t is predicted stock price, X_t is actual stock price, and N denotes total number of tests for all performance measures. The MAPE values for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction models are given in Table 10 for three stocks. It is clearly seen that although the prediction accuracy of the models changes according to prediction terms and stocks, HS-JRNN is the best model according to the MAPE values. Also, HS-RELM model for ECILC and AFYON has good prediction performance in comparison with other models. The MAPE performances of AFYON are quite worse than the other stocks (ECILC and EREGL).

MAE is dependent on the scale of the dependent variable, but it is less sensitive to large deviations than the usual squared loss. Its formula:

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|X_t - F_t| \tag{17}$$

RMSE formula is given in Eq. (18).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - F_t)^2} \tag{18}$$

HS-JRNN is the best model according to MAE and RMSE values (Tables 11 and 12). Note that the smaller difference between RMSE and MAE means the smaller variance in the individual errors of the sample.

TheilU is scale invariant and lies between 0 and 1. Note that 0 shows a perfect fit, and hence, low values are the ideal case of perfect forecast. TheilU is calculated as follows:

$$\mathrm{TheilU} = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t - F_t)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(X_t)^2} + \sqrt{\frac{1}{N}\sum_{t=1}^{N}(F_t)^2}} \tag{19}$$

The assigned values by DS are assumed to be between 0 and 100. The larger values of DS mean better performance in stock market prediction. DS is calculated using Eqs. 20 and 21. The TheilU and DS values for 1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction models are given in Tables 13 and 14.

$$\mathrm{DS} = \frac{100}{N}\sum_{t=1}^{N}a_t \tag{20}$$

$$a_t = \begin{cases} 1, & (X_t - X_{t-1})(F_t - X_{t-1}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Although the prediction performances of models vary according to the prediction terms and stocks, HS-JRNN is reasonably better than the other models considered in this study. Furthermore, the comparison of RMSE and/or MAPE values showed that especially HS-JRNN model for 3-day-ahead prediction of AFYON stocks generally gives more promising forecasting results than many other studies [20, 21, 23, 40–42]. The findings supported that HS-JRNN has the ability to extract remarkable forecasting results from selected technical indicators. Also, Dematos et al. [43] presented that recurrent networks can be helpful for forecasting financial prices since they use information from the sequence or time dependence of the inputs as well as the inputs themselves. In this direction, we can say that HS-JRNN can be successfully used for stock market forecasting due to the advantage of time dependence.

One of the most important contributions is that significant improvements are made through the proposed hybrid models. They have proven to be useful in variable selection and parameter determination that is an active and fruitful field of research in soft computing models. It is noteworthy that according to the results of performance measures, our proposed models can serve desirable approach in the stock price prediction. The results also indicate that one should first determine the characteristics of the stocks to determine

Table 10 MAPE values for proposed models

Prediction term	Stock	HS-NN	HS-JRNN	HS-ELM	HS-RELM	HS-GLM	HS-RT	HS-GPR
1-day-ahead	ECILC	9.48	0.02	13.31	1.93	2.32	3.81	2.33
	EREGL	7.69	0.02	74.26	5.25	4.94	6.00	4.71
	AFYON	98.75	0.42	88.17	12.42	20.91	31.62	21.71
2-day-ahead	ECILC	3.57	0.01	2.97	2.55	2.88	4.23	3.00
	EREGL	18.32	0.01	73.19	6.58	6.16	7.69	6.13
	AFYON	98.62	0.30	88.96	19.20	27.61	40.10	28.05
3-day-ahead	ECILC	9.65	0.01	13.67	3.19	3.41	5.28	3.53
	EREGL	14.74	0.02	72.76	7.60	7.55	9.19	7.44
	AFYON	98.50	0.0002	89.24	25.41	34.47	49.00	34.21
5-day-ahead	ECILC	11.35	0.01	13.25	4.14	4.33	7.19	4.58
	EREGL	14.63	0.06	75.27	9.34	10.19	11.27	10.01
	AFYON	125.1	0.01	88.56	35.54	45.87	76.99	45.11
7-day-ahead	ECILC	12.43	0.01	13.31	4.76	4.78	7.71	5.18
	EREGL	17.57	0.0005	77.65	11.82	11.99	13.25	11.88
	AFYON	126.4	0.01	88.84	35.86	58.56	124.1	57.11
10-day-ahead	ECILC	11.03	0.0013	7.71	5.17	5.48	7.91	5.44
	EREGL	12.80	0.73	16.14	13.73	14.41	12.88	13.98
	AFYON	75.66	0.03	92.92	40.25	78.98	118.7	79.17

Table 11 MAE values for proposed models

Prediction term	Stock	HS-NN	HS-JRNN	HS-ELM	HS-RELM	HS-GLM	HS-RT	HS-GPR
1-day-ahead	ECILC	0.0071	0.000014	0.0100	0.0015	0.0018	0.0029	0.0018
	EREGL	0.0015	0.000003	0.0144	0.0010	0.0010	0.0012	0.0009
	AFYON	0.0044	0.000017	0.0041	0.0003	0.0005	0.0007	0.0005
2-day-ahead	ECILC	0.0027	0.000006	0.0102	0.0019	0.0022	0.0032	0.0023
	EREGL	0.0035	0.000002	0.0143	0.0013	0.0012	0.0015	0.0012
	AFYON	0.0044	0.000010	0.0041	0.0005	0.0006	0.0008	0.0007
3-day-ahead	ECILC	0.0073	0.000006	0.0103	0.0024	0.0026	0.0040	0.0027
	EREGL	0.0029	0.000003	0.0142	0.0015	0.0015	0.0018	0.0014
	AFYON	0.0044	0.00000	0.0041	0.0006	0.0008	0.0010	0.0008
5-day-ahead	ECILC	0.0086	0.000005	0.0099	0.0032	0.0033	0.0054	0.0035
	EREGL	0.0030	0.000012	0.0146	0.0018	0.0020	0.0022	0.0019
	AFYON	0.0055	0.00000	0.0041	0.0008	0.0010	0.0015	0.0010
7-day-ahead	ECILC	0.0093	0.000006	0.0099	0.0036	0.0036	0.0058	0.0039
	EREGL	0.0036	0.00000	0.0150	0.0023	0.0023	0.0026	0.0023
	AFYON	0.0055	0.00000	0.0041	0.0008	0.0012	0.0022	0.0013
10-day-ahead	ECILC	0.0082	0.000001	0.0059	0.0039	0.0041	0.0060	0.0041
	EREGL	0.0025	0.000137	0.0030	0.0026	0.0028	0.0025	0.0027
	AFYON	0.0014	0.000001	0.0041	0.0008	0.0017	0.0020	0.0017

which model is the best to use. Also, architecture of the model is reasonably important to improve the prediction performance of the models. The prediction performance of the model directly depends on how the parameters are selected, but an almost unlimited number of variations are available to be used by models. The primary advantage of using HS-based hybrid model is to provide a generic framework that captures the underlying dynamics of noisy and dimensional stock market data. HS determines the most relevant technical indicators and simultaneously

Table 12 RMSE values for proposed models

Prediction term	Stock	HS-NN	HS-JRNN	HS-ELM	HS-RELM	HS-GLM	HS-RT	HS-GPR
1-day-ahead	ECILC	0.0087	0.000015	0.0118	0.0019	0.0022	0.0044	0.0022
	EREGL	0.0020	0.000005	0.0184	0.0012	0.0011	0.0014	0.0011
	AFYON	0.0049	0.000018	0.0046	0.0005	0.0008	0.0012	0.0008
2-day-ahead	ECILC	0.0036	0.000012	0.0118	0.0025	0.0028	0.0037	0.0029
	EREGL	0.0042	0.000003	0.0183	0.0015	0.0014	0.0018	0.0014
	AFYON	0.0049	0.000011	0.0046	0.0007	0.0010	0.0014	0.0010
3-day-ahead	ECILC	0.0084	0.000007	0.0119	0.0032	0.0034	0.0052	0.0035
	EREGL	0.0039	0.000006	0.0183	0.0018	0.0018	0.0023	0.0018
	AFYON	0.0049	0.00000	0.0046	0.0009	0.0012	0.0016	0.0012
5-day-ahead	ECILC	0.0100	0.000006	0.0116	0.0040	0.0042	0.0072	0.0045
	EREGL	0.0037	0.000013	0.0185	0.0022	0.0024	0.0027	0.0024
	AFYON	0.0060	0.00000	0.0046	0.0012	0.0015	0.0020	0.0015
7-day-ahead	ECILC	0.0121	0.000006	0.0118	0.0045	0.0046	0.0074	0.0048
	EREGL	0.0042	0.00000	0.0187	0.0028	0.0029	0.0033	0.0029
	AFYON	0.0060	0.00000	0.0046	0.0012	0.0018	0.0030	0.0018
10-day-ahead	ECILC	0.0090	0.000001	0.0078	0.0049	0.0051	0.0072	0.0052
	EREGL	0.0030	0.000140	0.0038	0.0032	0.0034	0.0034	0.0033
	AFYON	0.0021	0.000001	0.0045	0.0012	0.0023	0.0029	0.0023

Table 13 TheilU values for proposed models

Prediction term	Stock	HS-NN	HS-JRNN	HS-ELM	HS-RELM	HS-GLM	HS-RT	HS-GPR
1-day-ahead	ECILC	0.0594	0.000099	0.0747	0.0124	0.0148	0.0288	0.0147
	EREGL	0.0524	0.000116	0.3437	0.0300	0.0286	0.0360	0.0281
	AFYON	0.9738	0.001793	0.8313	0.0471	0.0774	0.1113	0.0795
2-day-ahead	ECILC	0.0242	0.000080	0.0751	0.0162	0.0183	0.0242	0.0190
	EREGL	0.1156	0.000079	0.3419	0.0382	0.0354	0.0450	0.0364
	AFYON	0.9735	0.001080	0.8384	0.0694	0.0973	0.1329	0.0988
3-day-ahead	ECILC	0.0584	0.000049	0.0756	0.0209	0.0221	0.0339	0.0229
	EREGL	0.1054	0.000158	0.3422	0.0449	0.0459	0.0594	0.0455
	AFYON	0.9736	0.000001	0.8375	0.0905	0.1158	0.1546	0.1152
5-day-ahead	ECILC	0.0697	0.000038	0.0733	0.0266	0.0276	0.0468	0.0292
	EREGL	0.1003	0.000321	0.3456	0.0561	0.0607	0.0691	0.0602
	AFYON	0.9917	0.000048	0.8349	0.1116	0.1390	0.1865	0.1421
7-day-ahead	ECILC	0.0827	0.000043	0.0748	0.0292	0.0299	0.0482	0.0315
	EREGL	0.1162	0.000003	0.3498	0.0687	0.0724	0.0863	0.0714
	AFYON	0.9911	0.000050	0.8401	0.1129	0.1605	0.2697	0.1649
10-day-ahead	ECILC	0.0564	0.000008	0.0529	0.0320	0.0332	0.0469	0.0336
	EREGL	0.0788	0.003569	0.0921	0.0782	0.0839	0.0857	0.0811
	AFYON	0.1945	0.000066	0.7929	0.1207	0.1988	0.2565	0.1984

searching the most appropriate transfer function type and neuron numbers in hidden and context layer. The interactions of the parameters clearly influence the prediction performance, and therefore, determining critical parameters of the model increases the prediction speed and accuracy. However, not all parameters affecting the hybrid model were taken into account in this paper. In the ongoing research, it can be interesting to consider all other parameters such as training functions, iteration number in the designing of HS-based hybrid model. In this study,

Table 14 DS values for proposed models

Prediction term	Stock	HS-NN	HS-JRNN	HS-ELM	HS-RELM	HS-GLM	HS-RT	HS-GPR
1-day-ahead	ECILC	35	59	30	39	35	35	36
	EREGL	26	59	34	24	27	31	29
	AFYON	28	59	28	33	30	33	29
2-day-ahead	ECILC	41	59	32	35	35	31	32
	EREGL	33	59	32	23	24	35	25
	AFYON	28	59	28	32	28	38	29
3-day-ahead	ECILC	34	59	34	32	35	34	35
	EREGL	32	59	33	26	27	30	26
	AFYON	28	59	28	32	30	31	31
5-day-ahead	ECILC	34	59	33	33	34	34	35
	EREGL	33	59	26	26	25	27	25
	AFYON	28	59	28	32	32	28	32
7-day-ahead	ECILC	32	59	36	35	35	34	35
	EREGL	32	59	27	26	28	32	30
	AFYON	28	59	28	35	32	31	33
10-day-ahead	ECILC	34	59	36	39	35	35	38
	EREGL	31	59	32	28	29	32	31
	AFYON	31	59	28	34	34	38	33

following problems are solved in order to carry out accurate prediction.

(i) How many input variables (technical indicators) should be used in for NN, JRNN, ELM, RELM, GLM, RT, and GPR models?
(ii) How many neurons can be used in hidden layer?
(iii) How many neurons can be used in context layer?
(iv) Which transfer function should be used?

Our work can be helpful to determine robust model architecture. Also, the prediction accuracy of our proposed hybrid models will empower speculators and investors to decide investment strategy in stock markets.

4 Conclusion

It is important to keep in mind that determination of model parameters is one of the most important factors in prediction. However, the process of determining optimal architecture of models for a stock market problem is still a controversial issue. Particularly, how to select relevant input variables to predict stock price accurately is one of the most important issues to make an investment decision. Accordingly, the emphasis of previous studies is generally on input variable selection and therefore neglecting the importance of transfer functions. However, the determination of transfer functions can strongly affect complexity and performance of model. The uniqueness of our proposed models comes from the fact that HS is used for variable selection and parameter tuning. Thus, transfer function type, and number of hidden and context neurons are successfully optimized with HS algorithm. Findings of analyses indicate that close price accelerator (23) and K % stochastic (29) are two mostly used technical indicators for proposed prediction models. HS-NN and HS-JRNN models commonly used elliot sigmoid transfer function and pure linear transfer function, while sigmoidal transfer function type is generally used for HS-ELM and HS-RELM. The number of hidden neurons for HS-NN, HS-JRNN, HS-ELM, and HS-RELM varies between 3 neurons and 29 neurons. The most commonly used number of hidden neurons is 16 neurons. For HS-JRNN and HS-RELM, the number of context neurons varies between 1 neuron and 16 neurons. The most commonly used number of context neurons is 16 neurons.

To create further comparison, proposed hybrid models are applied for six different prediction terms (1-, 2-, 3-, 5-, 7-, and 10-day-ahead prediction) and three stocks (ECILC, EREGL, and AFYON). According to the obtained MAPE, MAE, RMSE, TheilU, and DS values, HS-JRNN outperforms the others. However, other proposed models can also be very suitable for prediction with highly complex stock market data. HS-JRNN gives a promising direction to the study of stock price prediction and other stock markets. Although proposed hybrid models have a satisfactory prediction performance, some insufficiencies should be considered to be enhanced. For instance, it can be better to use improved HS for architecture optimization in soft computing models. Moreover, different metaheuristics such as genetic algorithm and PSO can be utilized to compare the performance of the hybrid models. Finally, fundamental indicators such as inflation rates, foreign exchange rates, money supply, import–export figures, employment rate, financial ratios of companies, and other stock index series can be added as initial input variables pool.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Agrawal S, Murarka PD (2013) Stock price forecasting: comparison of short term and long term stock price forecasting using various techniques of artificial neural networks. Int J Adv Res Comput Sci Softw Eng 3(6):154–170
2. Park K, Shin H (2013) Stock price prediction based on a complex interrelation network of economic factors. Eng Appl Artif Intell 26(5–6):1550–1561
3. Novak MG, Velušček D (2016) Prediction of stock price movement based on daily high prices. Quant Financ 16(5):793–826
4. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
5. Huang SH (2015) Supervised feature selection: a tutorial. Artif Intel Res 4(2):22–37
6. Zhu Q-Y, Qin AK, Suganthan PN, Huang G-B (2005) Evolutionary extreme learning machine. Pattern Recogn 38(10):1759–1763
7. Suresh S, Saraswathi S, Sundararajan N (2010) Performance enhancement of extreme learning machine for multi-category sparse data classification problems. Eng Appl Artif Intell 23 (7):1149–1157
8. Lan Y, Soh YC, Huang G-B (2010) Constructive hidden nodes selection of extreme learning machine for regression. Neurocomputing 73:3191–3199
9. Saraswathi S, Sundaram S, Sundararajan N, Zimmermann M, Nilsen-Hamilton M (2011) ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. IEEE ACM T Comput Bi 8(2):452–463
10. Lahoz D, Lacruz B, Mateo PM (2011) A bi-objective micro genetic extreme learning machine. In: 2011 IEEE workshop on hybrid intelligent models and applications, pp 68–75
11. Huang Y-W, Lai D-H (2012) Hidden node optimization for extreme learning machine. AASRI Proc 3:375–380
12. Xue B, Ma X, Gu J, Li Y (2013) An improved extreme learning machine based on variable-length particle swarm optimization. In: Proceeding of the IEEE international conference on robotics and biomimetics (ROBIO) Shenzhen, China, pp 1030–1035
13. Bazi Y, Alajlan N, Melgani F, AlHichri H, Malek S, Yager RR (2014) Differential evolution extreme learning machine for the classification of hyperspectral images. IEEE Geosci Remote Sens Lett 11(6):1066–1070
14. Hegazy O, Soliman OS, Salam MA (2015) FPA-ELM model for stock market prediction. Int J Adv Res Comput Sci Softw Eng 5 (2):1050–1063
15. Yang Z, Zhang T, Zhang D (2016) A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training. Cogn Neurodyn 10:73–83
16. Janakiraman VM, Nguyen X, Assanis D (2015) Nonlinear model predictive control of a gasoline HCCI engine using extreme learning machines. In: Review, special issue on neurodynamic systems for optimization and applications, pp 1–15. https://arxiv.org/pdf/1501.03969v1.pdf. Accessed 28 June 2017
17. Ertugrul ÖF (2016) Forecasting electricity load by a novel recurrent extreme learning machines approach. Int J Electr Power Energy Syst 78:429–435
18. Ruxanda G, Badea LM (2014) Configuring artificial neural networks for stock market predictions. Technol Econ Dev Econ 20 (1):116–132
19. Guresen E, Kayakutlu G, Daim TU (2011) Using artificial neural network models in stock market index prediction. Expert Syst Appl 38(8):10389–10397
20. Hsieh T-J, Hsiao H-F, Yeh W-C (2011) Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm. Appl Soft Comput 11:2510–2525
21. Wei L-Y, Cheng C-H (2012) A hybrid recurrent neural networks model based on synthesis features to forecast the Taiwan stock market. Int J Innov Comput Inf Control 8(8):5559–5571
22. Zahedi J, Rounaghi MM (2015) Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran Stock Exchange. Phys A 438:178–187
23. Anish CM, Majhi B (2016) Hybrid nonlinear adaptive scheme for stock market prediction using feedback FLANN and factor analysis. J Korean Stat Soc 45:64–76
24. Dash SK, Bisoi R, Dash PK (2016) A hybrid functional link dynamic neural network and evolutionary unscented Kalman filter for short-term electricity price forecasting. Neural Comput Appl 27:2123–2140
25. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933
26. Yang XS (2009) Harmony search as a metaheuristic algorithm, in music-inspired harmony search algorithm: theory and applications. In: Geem ZW (ed) Studies in computational intelligence. Springer, Berlin, pp 1–14
27. Göçken M, Özçalıcı M, Boru A, Dosdoğru AT (2016) Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Syst Appl 44:320–331
28. Fernández-Blanco P, Bodas-Sagi DJ, Soltero FJ, Hidalgo JI (2008) Technical market indicators optimization using evolutionary algorithms. In: Proceedings of the 10th annual conference companion on Genetic and evolutionary computation, pp 1851–1858
29. Diao R, Shen Q (2012) Feature selection with harmony search. IEEE Trans Syst Man Cybern B Cybern 42(6):1509–1523
30. Chakraborty P, Roy GG, Das S, Jain D, Abraham A (2009) An improved harmony search algorithm with differential mutation operator. Fundam Inform 95:1–26
31. Geem ZW, Cho Y-H (2011) Optimal design of water distribution networks using parameter-setting-free harmony search for two major parameters. J. Water Resour Plan Manag 137:377–380
32. Ojha VK, Abraham A, Snášel V (2014) Simultaneous optimization of neural network weights and active nodes using metaheuristics. In: 14th international conference on hybrid intelligent systems, pp 248–253
33. Hikawa H, Araga Y (2011) Study on gesture recognition system using posture classifier and Jordan recurrent neural network. In: Proceedings of international joint conference on neural networks, San Jose, California, USA, pp 405–412

34. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
35. Dudek G (2016) Extreme learning machine as a function approximator: Initialization of input weights and biases. In: Proceedings of the 9th international conference on computer recognition systems CORES 2015. Springer, Berlin, pp 59–69
36. Guisan A, Edwards TC, Hastie T (2002) Generalized linear and generalized additive models in studies of species distributions: setting the scene. Ecol Model 157:89–100
37. Yang J, Stenzel J (2006) Short-term load forecasting with increment regression tree. Electr Power Syst Res 76:880–888
38. Seo S, Wallat M, Graepel T, Obermayer K (2000) Gaussian process regression: active data selection and test point rejection. In: Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks, pp 241–246
39. Makrıdakıs S, Hıbon M (1995) Evaluating accuracy (or error) measures, insead. http://www.insead.edu/facultyresearch/research/doc.cfm?did=46875. Accessed 23 Aug 2016
40. Lu C-J (2013) Hybridizing nonlinear independent component analysis and support vector regression with particle swarm optimization for stock index forecasting. Neural Comput Appl 23:2417–2427
41. Dai W, Shao YE, Lu C-J (2013) Incorporating feature selection method into support vector regression for stock index forecasting. Neural Comput Appl 23:1551–1561
42. Thenmozhi M, Chand GS (2016) Forecasting stock returns based on information transmission across global markets using support vector machines. Neural Comput Appl 27:805–824
43. Dematos G, Boyd MS, Kermanshahi B, Kohzadi N, Kaastra I (1996) Feedforward versus recurrent neural networks for forecasting monthly Japanese yen exchange rates. Financ Eng Jpn Mark 3:59–75