J. Inst. Eng. India Ser. B (October 2023) 104(5):1103–1118
https://doi.org/10.1007/s40031-023-00912-0

ORIGINAL CONTRIBUTION

Improving Time Series Prediction with Deep Belief Network

Soumya Das[1] · Monalisa Nayak[2] · Manas Ranjan Senapati[3]

Received: 30 December 2020 / Accepted: 7 August 2023 / Published online: 25 August 2023

Abstract In this paper, the time series data prediction is done using Deep Belief Network (DBN). The time series data chosen are stock price data, exchange rate data, and electricity consumption data. DBN predicts these three datasets. Particle Swarm Optimization and Local Linear Wavelet Neural Network are also used for prediction of these three datasets. The Root Mean Square Error and Mean Absolute Percentage Error parameters are used to validate the performance of the algorithm. DBNs are more efficient than other machine learning algorithms because they generate less error. They are fault tolerant and use parallel processing. They avoid over fitting and increase the model generalization.

Keywords Deep belief network (DBN) · Particle swarm optimization (PSO) · Local linear wavelet neural network (LLWNN)

✉ Manas Ranjan Senapati
manassena@gmail.com

Soumya Das
aug10.soumya@gmail.com

Monalisa Nayak
monalisa.nayak11@gmail.com

[1] Government College of Engineering, Kalahandi, India

[2] Department of Electronics and Telecommunication Engineering, Indira Gandhi Institute of Technology, Sarang, India

[3] Department of Information Technology, Veer Surendra Sai University of Technology, Burla, India

Introduction

Time series data are a collection of statistics that are placed at equal time intermissions [1, 2]. In current years, research on time series modeling has witnessed multiple increase. As a result of extensive research, the change in behavior of the time series data can be predicted [3, 4]. Time series comprises of following components: (1) Trend: it is the chief component which is an outcome of a long-term moment of various factors [1, 5]. When time series show an upward or downward on steady movement for a long duration of time, then, we name it as a trend. (2) Cyclic: it is the component that occurs when the time series show a increase and decrease in an uneven period. This component usually stretches over long intervals. This cyclical variation is observed by most financial time series and economic categories. (3) Seasonal: it is the type of component in which time series are partial by periodic factors that repeats regularly at a periodic interval of time, i.e., weekly, monthly, quarterly. There are many kinds of factors like traditional events, weather conditions, climate, like the sale of tea during winter and sale of ice-cream during summer, Durga Puja sale. (4) Irregular component: it is unpredictable. They are random variations that can be caused due to many factors like war, earthquake, flood.

The stock market is a community marketplace that exists to issue, sell, and buy stocks [1, 6, 7]. A stock is a partial possession in an industry to share its profit. History says that prices of other assets and stock prices dynamically impact on the economic activity. If stock price increases, then, the economy also rises [1].

Another instance of a time series dataset is currency exchange data. It is a multivariable nonlinear system. Due to the erratic nature of the exchange rate market, it has been a extremely challenging task. So, researchers have developed various neural network techniques to control the multivariable nonlinear systems. Neural network techniques can adapt extensively and learn from past data [8].

In a complex commercial building, the consumption of electricity is inherently nonlinear and transient in nature. Intermediate to long term forecast is needed for usage of electricity in a housing and profitable building at an hourly basis to care demand response strategies, decision making about operations, and distributed generation system's installation. Due to progress in smart metering forecasting of sub-meter usages in the household level, it is also getting widespread on-demand response program and smart building control [9].

In time series, the developed dataset and the produced dataset are not from the same distribution and the real-world time series data are non-stationary, and the statistical properties of the distribution will be shifting as new model enters. The only way is to retrain the model every time whenever new data came in. It is not as continuous learning where we need to update already trained model whenever new data enter. So, here every time a new model is retrained when we generate a new forecast. With the growth of business, the time series forecasting will become harder as the amount of data increases. Now a days, stock market is built by combining various technologies like machine learning, big data and expert systems that communicates with each other to get accurate decisions. The connectivity of user in global environment on the internet has led to decrease in stability, susceptibility of customer sentiments and prone to mischievous attack. So, many hybrid approaches like combination of statistical and machine learning are developed that are better in stock market prediction [10].

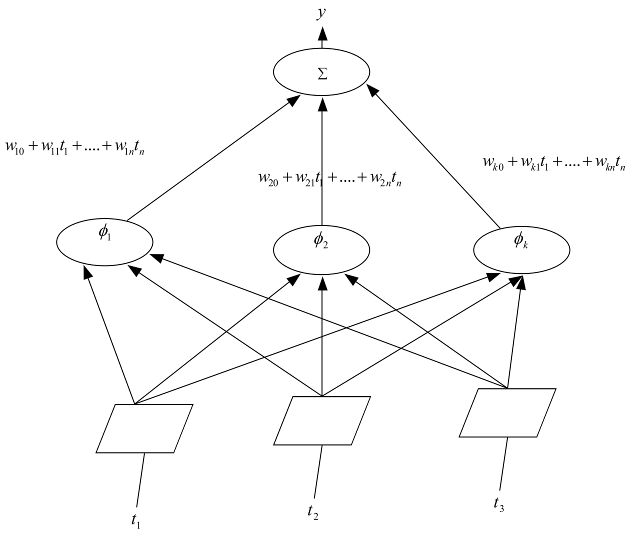

y
Σ
$w_{10}+w_{11}t_1+....+w_{1n}t_n$
$w_{20}+w_{21}t_1+....+w_{2n}t_n$
$w_{k0}+w_{k1}t_1+....+w_{kn}t_n$
ϕ_1
ϕ_2
ϕ_k
t_1
t_2
t_3

Fig. 1 A local linear wavelet neural network

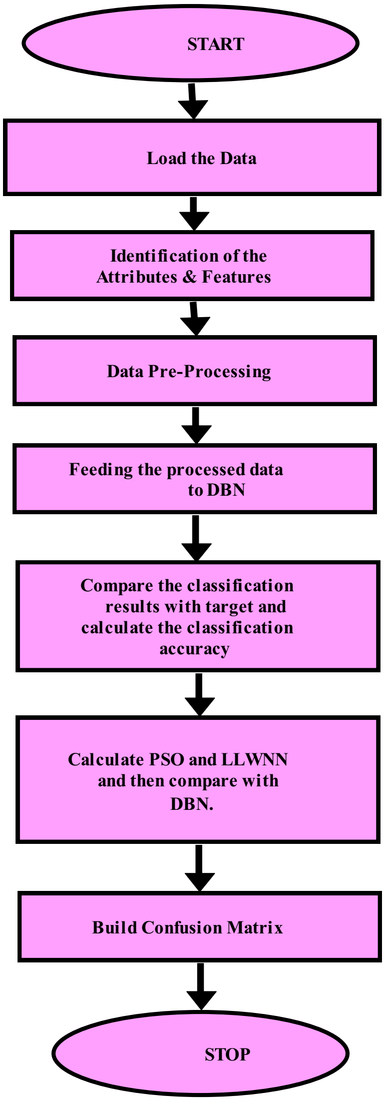

START
Load the Data
Identification of the Attributes & Features
Data Pre-Processing
Feeding the processed data to DBN
Compare the classification results with target and calculate the classification accuracy
Calculate PSO and LLWNN and then compare with DBN.
Build Confusion Matrix
STOP

Fig. 2 Flowchart of DEEPNN model

As, DBN is a generative model that creates samples according to the features which the model learns at the time of training [11]. DBN is a type of deep learning algorithm. It is an effective algorithm that helps to solve the problems like low velocity and the phenomenon of overfitting in learning [12]. The features extraction by the DBN has higher separability and robustness that brings higher classification accuracy increasing the classifier performance [13].

In this paper, DBN is used for the analysis of three time series data like stock price data, exchange rate data, and electricity consumption data. First, the data are preprocessed that removes all the missing values. Then, the data are normalized using min–max normalization method. Finally, the processed data is fed to the DBN where classification is executed. Classification accuracy is calculated with the help of Root Mean Square Error (RMSE), Mean

Absolute Percentage Error (MAPE) and time of execution. Then, the results are compared with other machine learning algorithms like Particle Swarm Optimization (PSO) and Local Linear Wavelet Neural Network (LLWNN).

This paper is summarized as: after the introduction to the problem, "Related Work" section discusses the related work. "Deep Belief Network (DBN)" section discusses about DBN. Other predictive models are discussed in "Predictive Models" section. Findings and analysis of DBN as well as other predictive models are discussed in "Finding and Analysis" section. Finally, the conclusion is drawn in "Conclusion" section.

Related Work

A deep Long Short-Term Memory Neural Network (LSTMNN) was proposed with entrenched layer and automatic encoder for predicting the stock market. The experimental result shows that LSTM with an embedded layer is better than the other with 57.2% accuracy [14]. The performances of these two models are verified using Sinopec and Shanghai A-share composite index. The Shanghai A-share composite index shows better predictive accuracy [14]. A Convolution Neural Network algorithm was developed to predict stock price dataset, i.e., historical data Nifty from January 5, 2015, to December 27, 2019. Here, 8 regression and 8 classification methods are used from which CNN shows the best result with 1.09 Root Mean Square Error (RMSE). The result shows that neural network-based models are highly capable to extract and learn the training dataset features. Moreover, multivariate analysis enables higher accuracy than univariate models [15].

An adaptive prediction model was developed with knowledge guided artificial neural network (KGANN) to predict exchange rate efficiently [16]. The new method is having two parallel systems, i.e., the first one is LMS trained Adaptive Linear Combiner, and the second one is the Functional Link Artificial Neural Network (FLANN) model to deliver an additional precise exchange rate than that foretold by LMS or FLANN model independently. KGANN provides a more effective model to predict the exchange rate than that of LMS and FLANN individually [16]. MLFNN and NARX with 0.001 Mean Square Error (MSE) show better forecasting efficiency than other techniques according to the MSE plot [17]. An Improved Shuffled Frog Leaping-based Learning Strategy (ISFL) was integrated with Pi-Sigma Neural Network (PSNN) to predict the exchange rate, i.e., US dollar against three different currencies like Canadian Dollar (CAD), Swiss Franc (CHF) and Japanese Yen (JPY) from Jan 2014 to Nov 2017. The result shows that the Pi-Sigma ISFL with USD/CAD shows about 0.0197 RMSE which is much better than other techniques. The presentation of the anticipated model can be detected with other time series data. A hybrid robust learning method can be developed for PSNN [18].

A Recurrent Neural Network (RNN) algorithm was developed to support demand response strategies decision making about processes and connection of distributed generation systems [19]. These outcomes are associated with 3-layered Multi-Layered Perceptron (MLP) and show better performance when done in an hourly basis, but when it is compared in yearly basis than MLP shows better performance over RNN, this model helps to get surrogates for missing transient variables that affect the load profile in profitable structures [19]. Cost-Effective Firefly-based Algorithm (CEFA) was developed to initialize population, encoding problem and evaluate fitness in order to provide optimized execution of workflow and cost-effective within a time limit. CEFA is compared with different algorithms like IC-PCP,

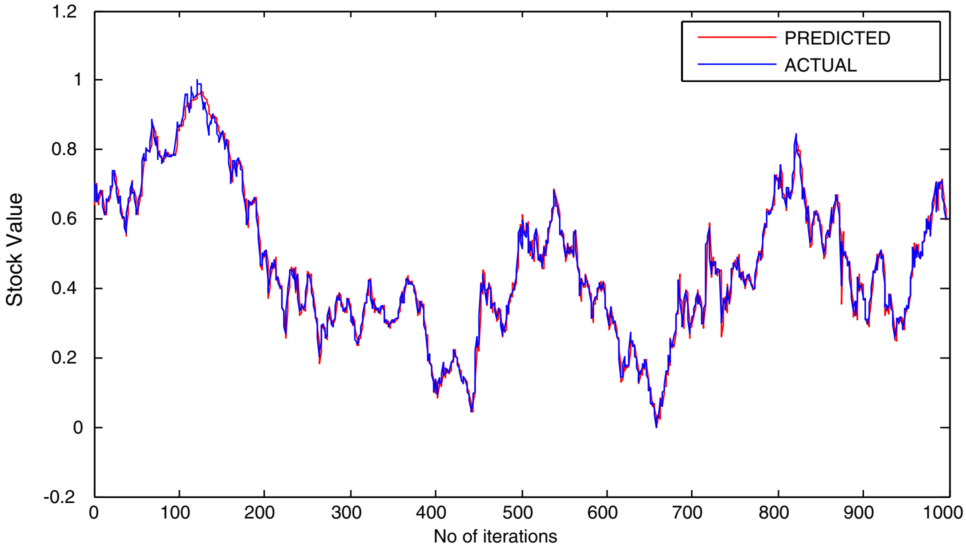


Fig. 3 Real versus forecast during training for DBN in stock market

RWO, PSO, Robustness-cost time etc. CEFA uses Cloud Sim tool for simulation. This work can be extended in real-cloud environment [20]. Soft-Margin Complex Polyhedron Classifier (SM-CPC) was developed to provide better accuracies with the well-known classifier. They are helpful in providing noise lenience and arrangement with the instance where two classes share shared illustrations. They are not suitable for high dimensional and large-scale classification problems [21].

Deep Belief Network (DBN)

DBN is a class of unsupervised network [22, 23]. RBM consists of a observable layer (input layer) and a hidden layer. This can be expressed as:

$$P(H/V) = \frac{P(H/V)}{P(V)} \tag{1}$$

$$P(V/H) = \frac{P(V,H)}{P(H)} \tag{2}$$

$P(V/H)$ = conditional probability, where H = hidden vector, V = input vector

Energy-based RBM model:

$$P(H,V) = P(V,H) = \frac{1}{Z}e^{-E(V,H)} \tag{3}$$

$E(V, H)$ is the energy of hidden and visible unit joint configuration.

$$Z = \sum\nolimits_{V,H} e^{-E(V,H)} \tag{4}$$

RBMs training process:

“*Step 1* Initialize

The weights are established with normal distribution for individually input data $X_t, t \in [1,Z], V = X_t$

Step 2 $P(H/V)$, the probability of hidden layer is computed

$$P(H/V) = \sigma(wV)$$

Step 3 $P(V/H)$, the probability of reconstructed input layer is computed

$$P(V/H) = \sigma(wH)$$

Step 4 Find Δw, the reconstruction error

$$\Delta w = <VH>_{\text{data}} - <VH>_{\text{reconstruction}}$$

Step 5 Calculate $E(V, H)$, energy function and update ‘w’, weights

Table 1 Parameter settings

Sl No	Parameter details	Description
1	W	Weight
2	B_i	Bias
3	c_j	Bias
4	V	Input vector unit
5	H	Hidden vector unit
6	Z	Joint configuration
7	$\Delta \boldsymbol{w}$	Construction error
8	$P(V/H)$	Conditional probability
9	$E(V, H)$	Energy of hidden and visible unit joint configuration

Table 2 Specifics of datasets

Dataset details	Attributes	Features	Date
Yahoo Inc	1500	7	1st Jan 2007 to 1st Jan 2013
Currency exchange	900	7	19th July 2016 to 1st Jan 2020
Household electricity consumption	1500	9	16/12/2006 time 5.24 PM to 02/03/2008 time 06.23 AM

Table 3 Details of stock market during testing

Prediction method	RMSE	Time taken	MAPE
DBN	0.003214	0.8643	0.7713
PSO	0.05621	0.6532	0.8638
LLWNN	0.09762	0.9542	0.9011

Table 4 Specifics of exchange rate through testing

Prediction method	RMSE	Time taken	MAPE
DBN	0.005421	0.7659	0.6593
PSO	0.0953	0.8705	0.7430
LLWNN	0.0879	0.9073	0.6909

$$w \leftarrow w + \varepsilon(\Delta w)$$

$$E(V,H) = -\sum\nolimits_i V_i B_i - \sum\nolimits_i H_j C_j - \sum\nolimits_{i,j} V_i H_j w_{ij}$$

where $i = 1\ldots\ldots..n, j = 1\ldots\ldots..m$, B_i and C_j are bias.

Step 6 Step2 and Step5 are repeated till $E(V, H)$ converge.

Table 5 Details of household electricity consumption during testing

Prediction method	RMSE	Time taken	MAPE
DBN	0.0008906	0.7642	0.7802
PSO	0.005971	0.6091	0.8973
LLWNN	0.009851	0.8903	0.8979

Table 6 Presentation assessment between diverse representations (Regular remoteness during testing between forecasted & actual price) for the stock market

Number of epochs	DBN	PSO	LLWNN
1–50	0.0027025554	0.0273211160	0.0124231488
51–100	0.0002150634	0.0269125310	0.0164289039
101–150	0.0178241505	0.0300148672	0.0432504503
151–200	0.0022996022	0.0272497203	0.0132889795
201–250	0.0025140053	0.0285487102	0.0210680971
251–300	0.0151517494	0.0295753959	0.0406018735
301–350	0.0080093765	0.0287500862	0.0311798943
351–400	0.0311035165	0.0199460962	0.0277621475
401–450	0.0119040209	0.0312712527	0.0374597926
450–500	0.0281073337	0.0158772913	0.0607405078
501–550	0.0014010568	0.0028077096	0.0038982155

Table 7 Presentation assessment between diverse representations (Regular remoteness during testing between forecasted and actual price for exchange rate

Number of epochs	DBN	PSO	LLWNN
1–50	0.0087425744	0.0279226560	0.0167950488
51–100	0.0006759634	0.0269198710	0.0180789039
101–150	0.0178542565	0.0300112348	0.0748904503
151–200	0.0022996132	0.0277489719	0.0785845595
201–250	0.0025146653	0.0284467048	0.0275331671
251–300	0.0152513657	0.0295768599	0.0406886535
301–350	0.0086893872	0.0249510662	0.0311894843
351–400	0.0311034674	0.0199490819	0.0274908475
401–450	0.0119079559	0.0250789527	0.0377656926
450–500	0.0281876437	0.0165789613	0.0608897078
501–550	0.00723050568	0.0081140962	0.0938453155

Fine Tuning

Step 1 Train the 1st RBM with the data *X*.

Step 2 Fix '*w*' and B_i & C_j of 1st RBM. The state of hidden units is used as visible data for the 2nd RBM.

Step 3 After training the 2nd, RBM is stacked on the top of 1st RBM.

Table 8 Presentation assessment between diverse representations (Regular remoteness during testing between forecasted and actual price for household electricity consumption

Number of epochs	DBN	PSO	LLWNN
1–50	0.0034025744	0.0256721006	0.0127850488
51–100	0.0002585663	0.0543915210	0.0179785039
101–150	0.0178242505	0.0300107854	0.0446254103
151–200	0.0022995622	0.0277497985	0.0132889895
201–250	0.0025111153	0.0284467422	0.0210631671
251–300	0.0151113494	0.0295766519	0.0406075635
301–350	0.0080893765	0.0287510032	0.0311756943
351–400	0.0312334265	0.0199461259	0.0274349075
401–450	0.0187541259	0.0250719457	0.0375697926
450–500	0.0281832337	0.0165773473	0.0606725078
501–550	0.0044112568	0.0078018946	0.0066872155

Step 4 Repeat step 2 and 3 for the preferred quantity of layers, every time by proliferating skyward either mean values or sample.

Step 5 Fine-tuning each parameter in this model concerning a proxy for DBN log-likelihood."

As per the name suggests, DNN is a feed forward neural network with many layers, but DBN is a generative probabilistic model that has undirected connections between the top layers like in RBM [24].

Advantages of DBN over other networks are: It has a higher modeling capacity per parameter and also has well-organized training process that chains unsupervised knowledge for feature recognition with a succeeding stage of supervised knowledge which tunes the features to improve the discernment [22, 25].

Predictive Models

Particle Swarm Optimization (PSO)

In this algorithm, each particle in the population adapts recurrently to previous positive areas. Velocity update and position update are two main operators. In each generation, every particle is rushing to the particle's global best position. Each particle velocity is measured. Henceforward, operating the particles according to given equations [26]:

Velocity is determined as:

$$x_{ij} = p(x_{ij} + k_1 \times r_1 \left(T_i - b_{ij}\right) + (k_2 \times r_2 \left(G_{\text{best}} - b_{ij}\right)) \tag{5}$$

Each particle location is updated as:

Table 9 External validation results

Types of datasets	Techniques	RMSE	Source
Stock market	"DBN"	**0.003214**	DBN
	"LSTM-NN"	0.00428	[10]
	"CNN"	1.09	[11]
Currency exchange rate	"DBN"	**0.005421**	DBN
	"MLFNN + NARX"	0.008324	[13]
	"Pi-Sigma ISFL"	0.0197	[14]
Household electricity consumption	"DBN"	**0.0008906**	DBN
	"ANN + MLP"	0.022365	[15]

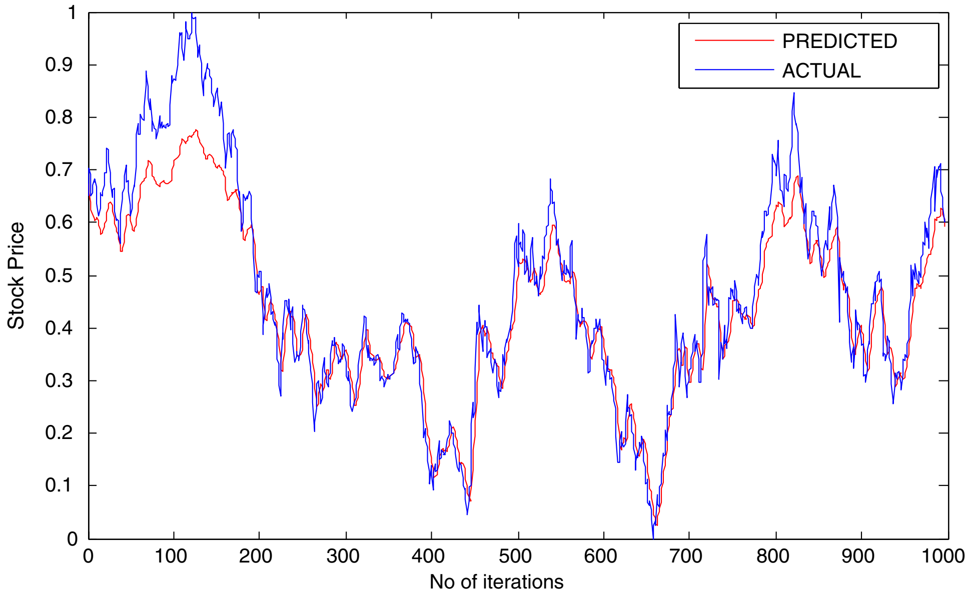


Fig. 4 Real versus forecast during testing for DBN in stock market

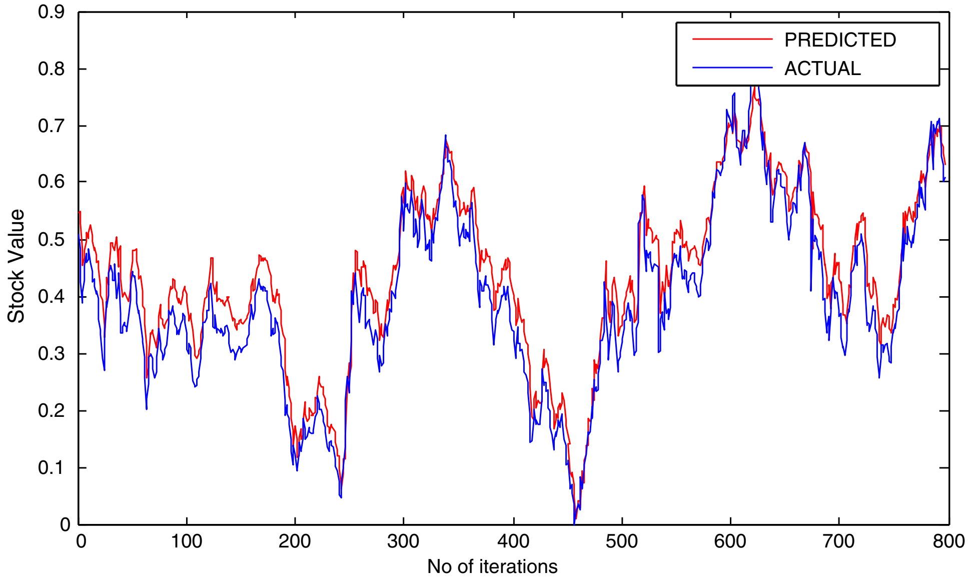


Fig. 5 Real versus forecast during training for PSO stock market

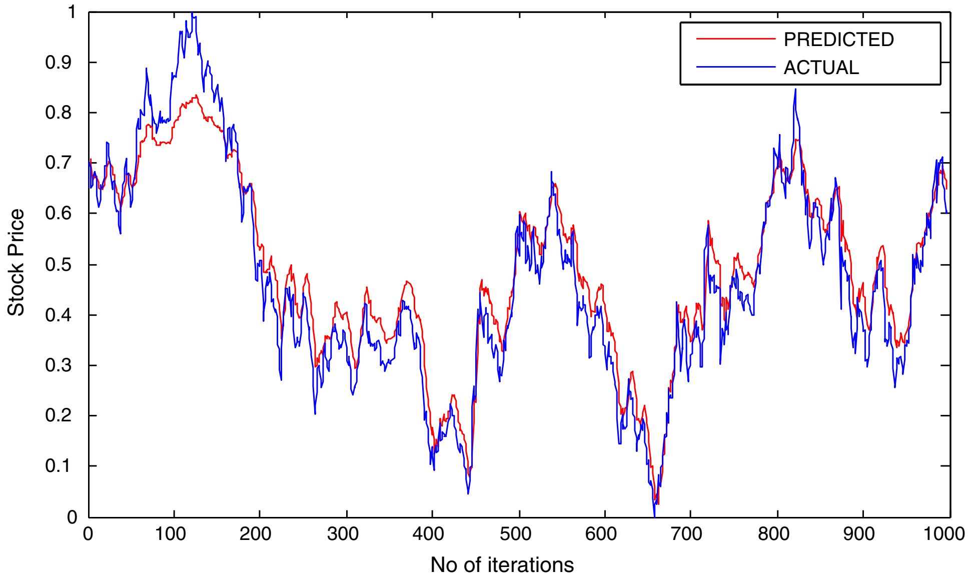


Fig. 6 Real versus forecast during testing for PSO in stock market

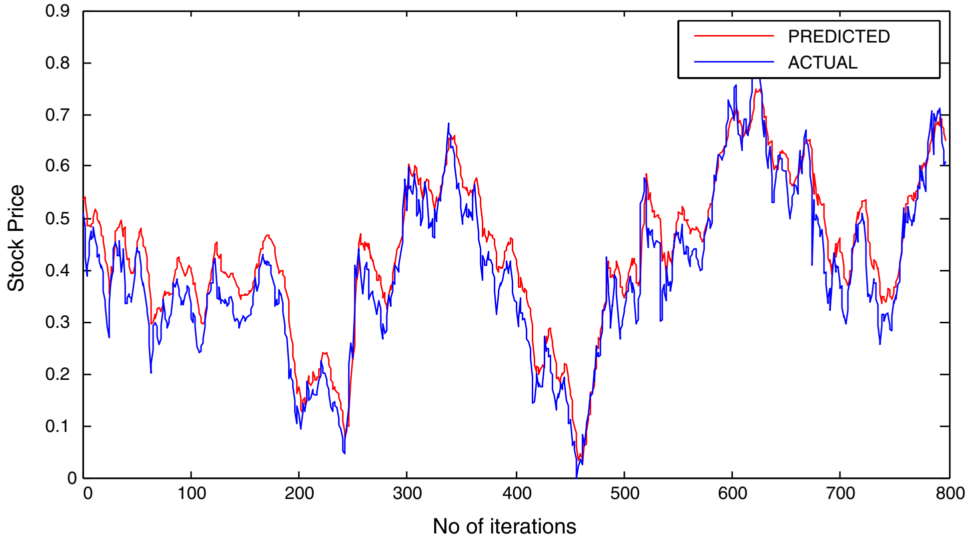


Fig. 7 Real versus forecast during training for LLWNN in stock market

$$b_{ij} = b_{ij} + x_{ij} \tag{6}$$

where x = velocity of the particle, b = random variable, T_i = local best value, G_{best} = global best value, p = inertia weight, k_1 & k_2 = acceleration constants, r_1 & r_2 = random variables between 0 and 1.

PSO algorithm [27]:

1. Particles are randomly initialized.
2. Fitness value with objective function is computed and deliberated as P_{best}
3. Position and speed of every particle is updated.
4. Velocity of each particle is updated with Eq. (5).
5. Location of each particle is updated with Eq. (6).
6. The G_{best} and P_{best} are updated until stopping criteria is met.

Local Linear Wavelet Neural Network (LLWNN)

Wavelet Neural Network (WNN) delivers improved learning effectiveness and structure transparency as compared to a multilayer perceptron (Fig. 1). Limitation of WNN is that when it is used for multidimensional problems, then, many hidden layers are needed [28, 29]. Then, LLWNN is developed which is a modification of WNN. Figure 2 shows the architecture of LLWNN. The productivity of the output layer is determined as:

Fig. 8 Real versus forecast during testing for LLWNN in stock market

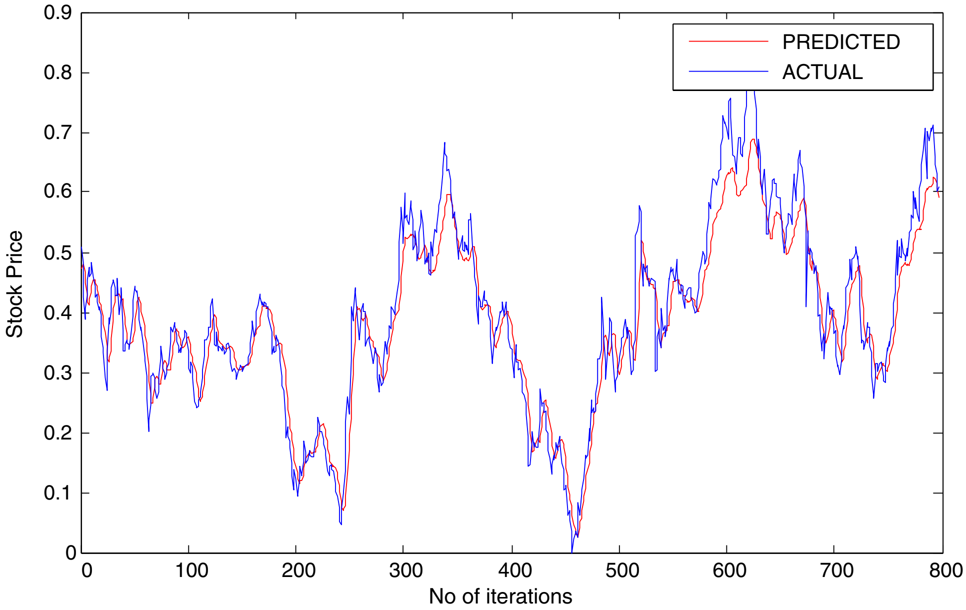
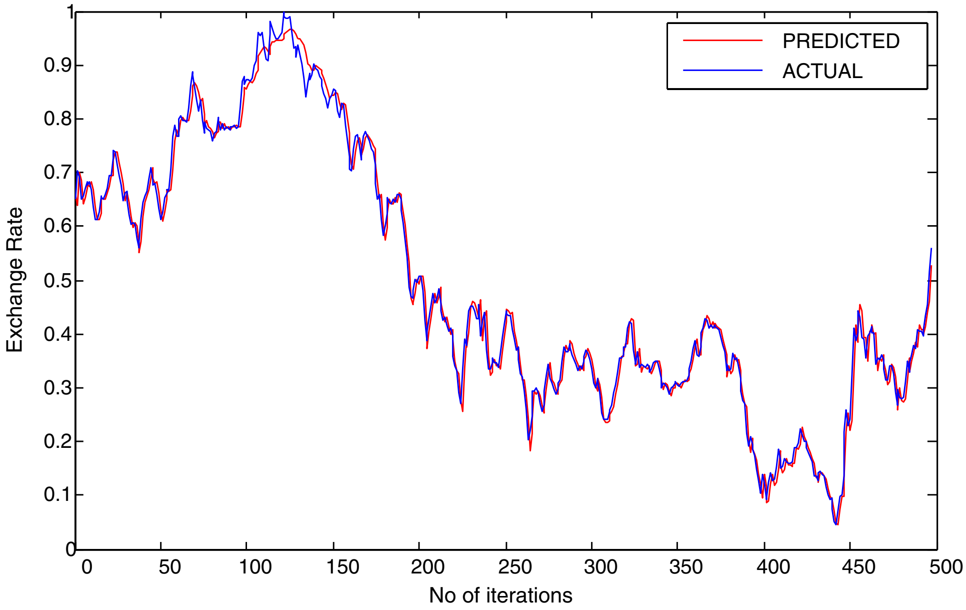


Fig. 9 Real versus forecast during training for DBN in exchange rate



$$y = \sum_{k=1}^{N} \left(w_{k0} + w_{k1}t_1 + \cdots + w_{kn}t_n\right)\varphi_k(t) \tag{7}$$

y= $\sum_{k=1}^{N} \left(w_{k0} + w_{k1}t_1 + cdots + w_{kn}t_n\right)\left|p_k\right|^{-\frac{1}{2}}\varphi\left(\frac{t-q_k}{p_k}\right)$where $T=[t_1, t_2,\ldots, t_n]$.

In LLWNN, piecewise constant weight is used, i.e., a linear model is presented as

$$v_k = w_{k0} + w_{k1}t_1 + \cdots + w_{kn}t_n \tag{8}$$

Direct models (v_k) ($K=1, 2\ldots N$) activities are found by active wavelet functions $\varphi_k(t)(K = 1, 2, \ldots ..N)$ that are associated locally. The translation, space, and local linear parameters are initialized arbitrarily at the commencement and then enhanced, by RLS (Recursive Least Square) algorithm.

Finding and Analysis

Different types of time series datasets are used to check the outcome of the DBN algorithm. At the start, the features are selected from the training data. Then, the relevant features are provided to the multilayer perceptron where the DBN update the factors. The performance validation of the classifier is checked by three datasets. 70% of the information

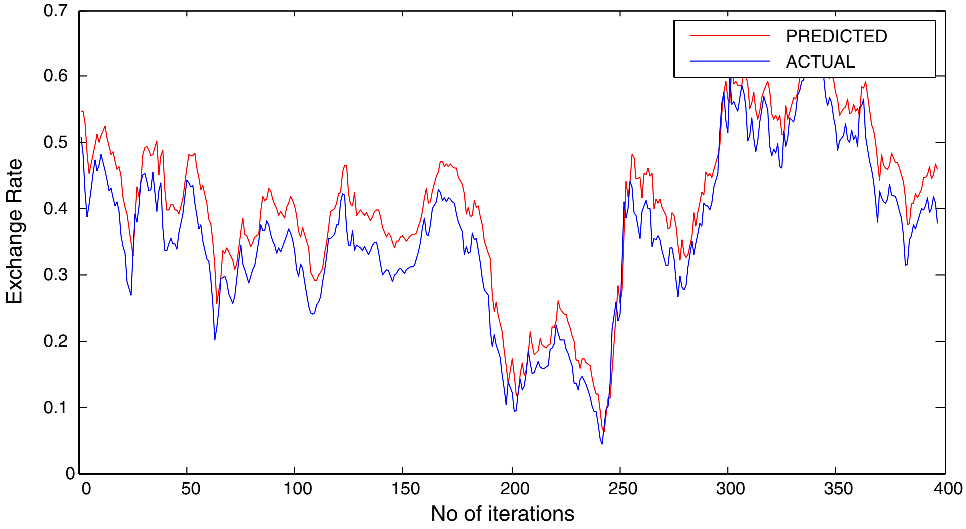


Fig. 10 Real versus forecast during testing for DBN in exchange rate

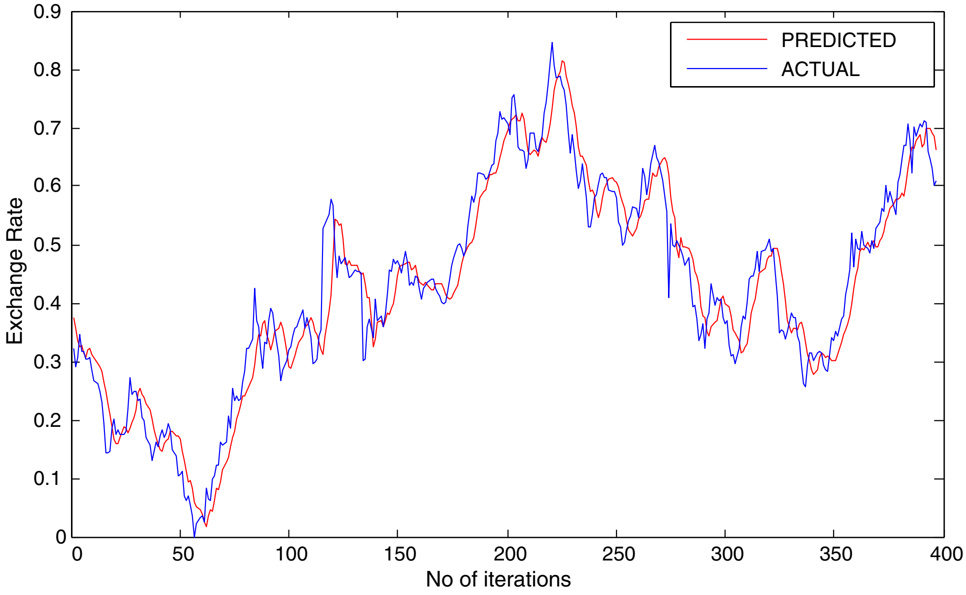


Fig. 11 Real versus forecast during training for PSO in exchange rate dataset

is taken for training purpose and 30% for testing. Data preprocessing is carried out where the misplaced values are discarded before the statistics is checked with the model. After that, data were normalized by min–max normalization. The min–max normalization method is specified as:

$$X_{\text{normalized}} = \frac{X_{\text{original}} - X_{\text{minimum}}}{X_{\text{maximum}} - X_{\text{minimum}}} \tag{9}$$

Summary of the complete effort done is provided in Fig. 3 and describe as:

Step 1 Loading the time series data, i.e., Currency Exchange rate, Household Electricity Consumption, and Yahoo Inc. datasets.

Step 2 Identification of the attributes and features.

Step 3 Statistics Pre-Processing is added where the raw statistics is transformed into an understandable format. The original data may contain some errors and are mostly inconsistent and incomplete.

Step 4 Classification is started after the processed data is fed to DBN.

Step 5 Calculation of cataloguing correctness is executed using diverse constraints like RMSE, MAPE, and Time of Implementation.

Step 6 The calculation of cataloguing correctness of other techniques is further executed and then the comparison is done with DBN.

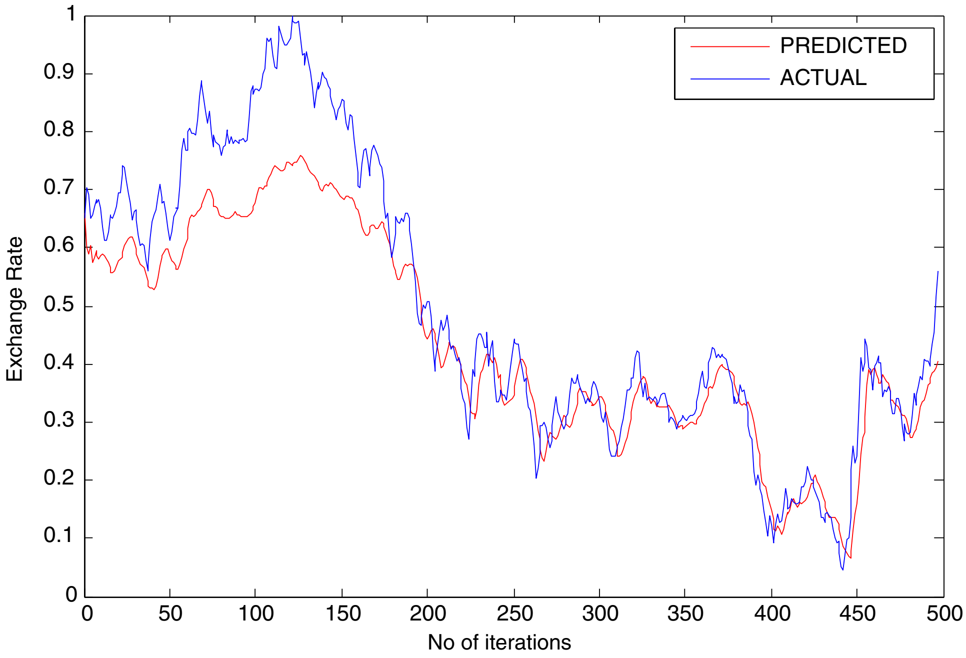


Fig. 12 Real versus forecast during testing for PSO in exchange rate

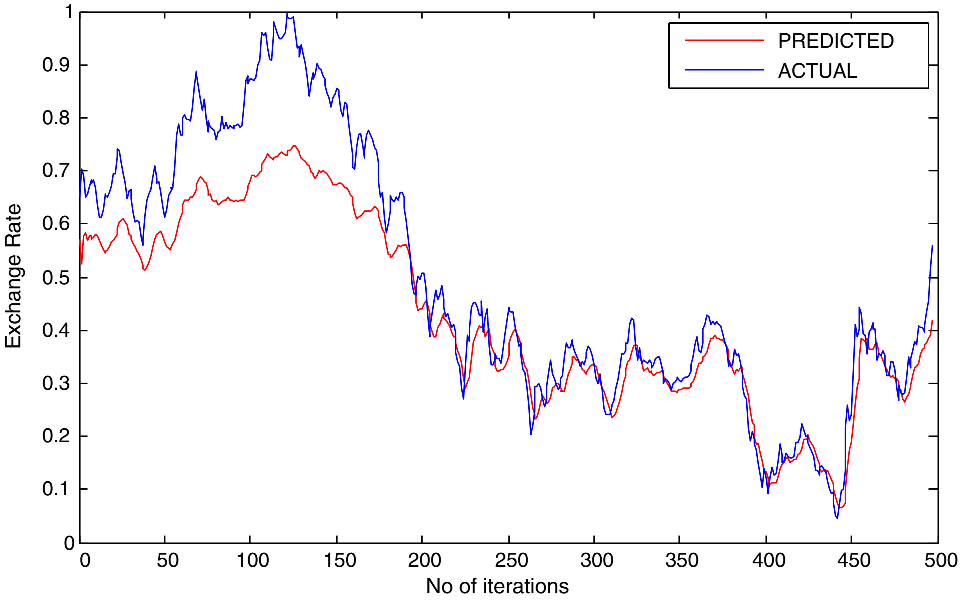


Fig. 13 Real versus forecast during training for LLWNN in exchange rate

Step 7 The confusion matrix of DBN is constructed for Currency Exchange rate, Household Electricity Consumption, and Yahoo Inc. datasets.

Step 8 End the process.

The DBN performance is equated with different methods like PSO and LLWNN. The DBN performance is evaluated by three parameters:

(1) Root Mean Square Error (RMSE). The RMSE formulae are given below:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} (Y_n - T_n)^2}{N}} \tag{10}$$

where Y_n = expected output, T_n = actual output (target) s, N = total data sample size.

(2) The Mean Absolute Percentage Error (MAPE) formulae is as follows:

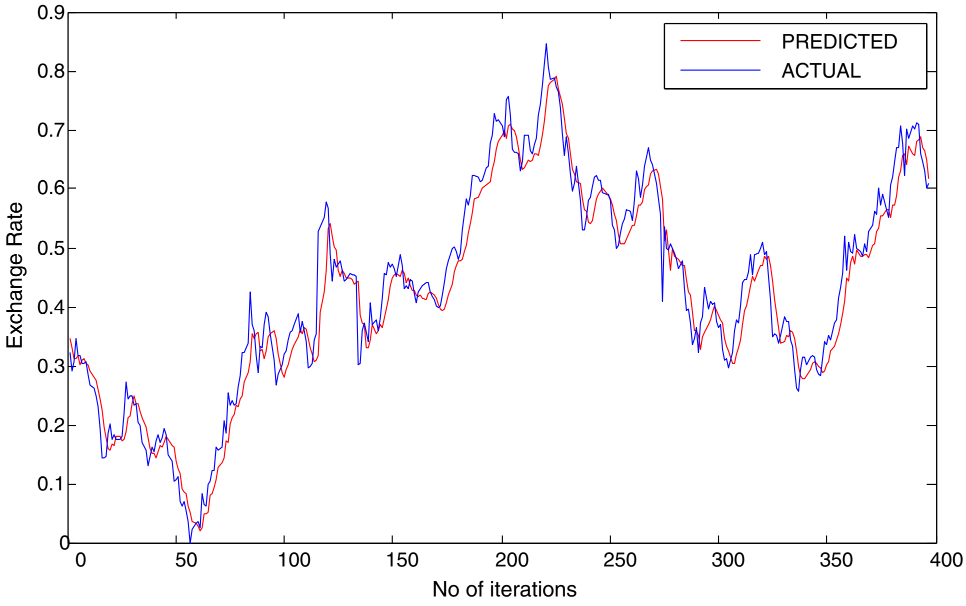


Fig. 14 Real versus forecast during testing for LLWNN in exchange rate

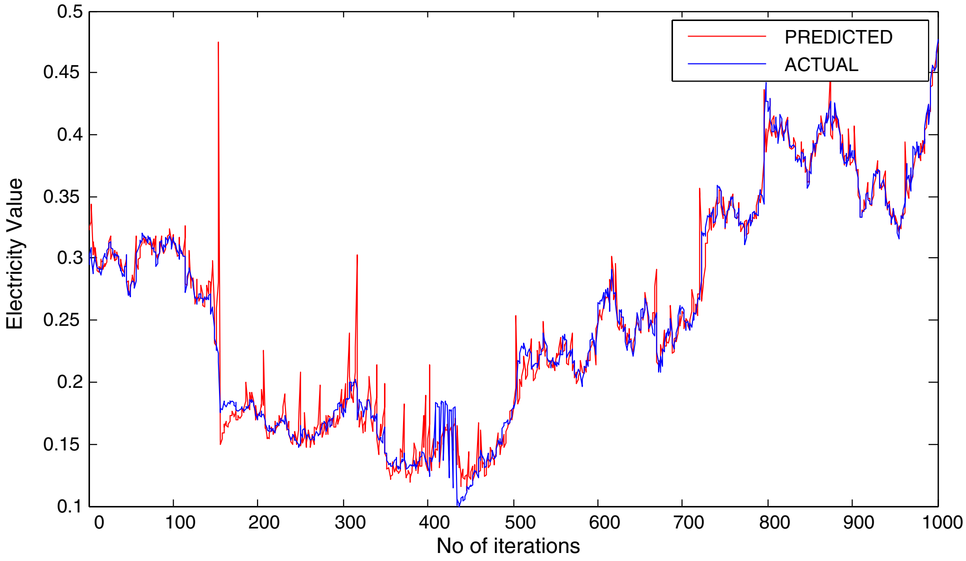


Fig. 15 Real versus forecast during training for DBN in electricity consumption

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_{t-F_t}}{A_t}\right| \tag{11}$$

where A_t is the real value, and F_t is the estimate value.

(3) Time of Execution

The MATLAB software is used here to get the simulation result. The normalized input variables are taken to free them from measurement units.

Specifics of Datasets

(i) The stock market for yahoo.inc [30] is used for analysis. The dataset consists of 7 columns. The dataset consists of 1500 samples. The yahoo data are taken from 1st January 2007 to 1 January 2011. The data till 1 October 2009 is taken as training data, and the rest is taken as testing data.

(ii) The currency exchange rate from INR to USD. The dataset consists of 2430 rows and 6 attributes [31]. The time gap from May 2010 to 2018 is considered as training data. May 2019 is considered for trying. The currency exchange dataset displays the currency exchange data of the US dollar to Indian National

Fig. 16 Real versus forecast during testing for DBN in electricity consumption

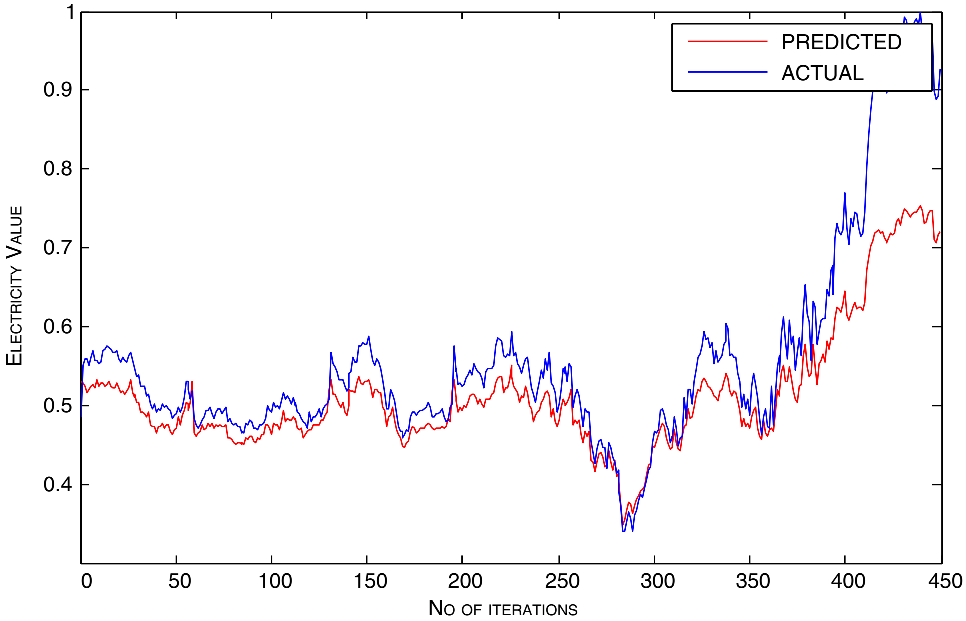
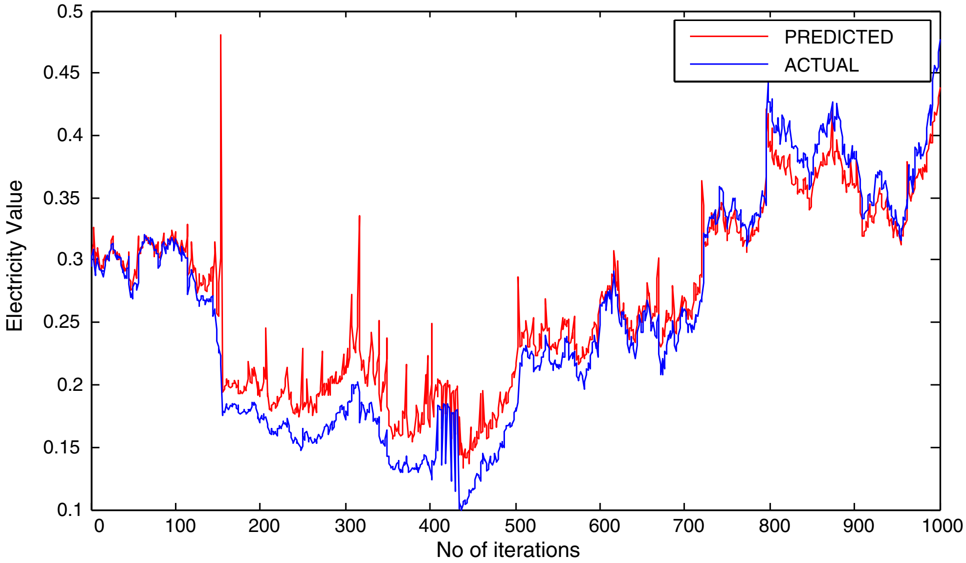


Fig. 17 Real versus forecast during training for PSO in electricity consumption



Rupees (INR). The dataset has 1500 samples and 7 columns namely date, price open, high low volume, and exchange %. Out of which price is taken as the target value. From December 31st, 2019 to January 31st, 2018 is taken as training value and the rest is taken as testing value.

(iii) The energy consumption data [32] are taken for 10 min intervals for 4.5 months. The ZigBee wireless sensor network checks the temperature and humidity circumstances. The energy usage of the preceding month is considered as for training, and the subsequent day energy usage is considered as testing data. The household electricity statistics records the electricity consumption value in 10 min interval. We have taken data of each day and taken the mean of total day's data (around 397 samples). In that way, we have taken data from 16/12/2006 at 5.24 PM to 02/03/2008 at 06.23 AM (1500 samples). The data till 18 September 2007 10.02 AM are taken as training statistics, and the surplus is taken as testing statistics. The column Global active power is taken as the target value [33].

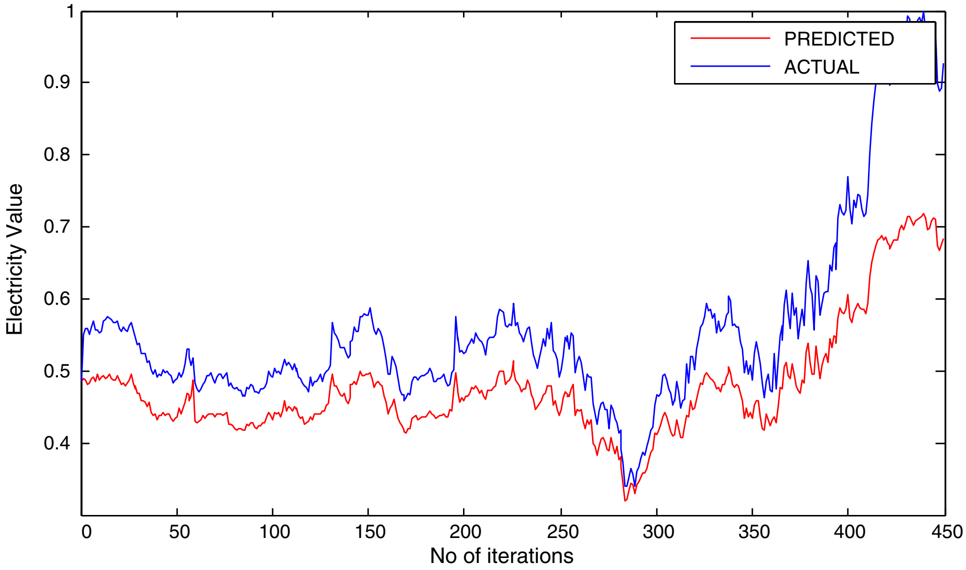


Fig. 18 Real versus forecast during testing for PSO in electricity consumption

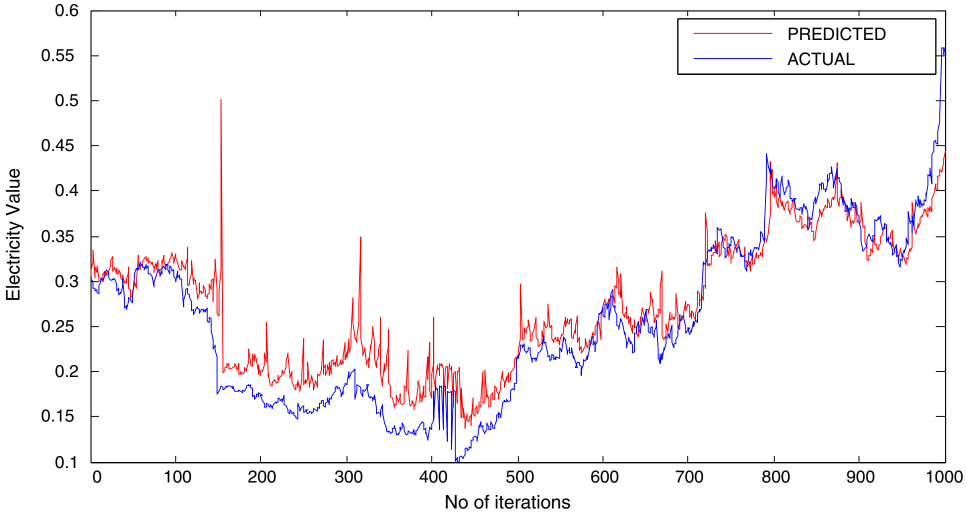


Fig. 19 Real versus forecast during training for LLWNN in electricity consumption

Results and Discussion

The superiority of a DBN is verified by figures and tables. Tables and figures are described below as:

Table 1 discusses parameter settings of deep belief networks.

Table 2 titles the details of diverse kinds of time series datasets.

Table 3 represents the stock market details during the time of testing. In 0.6532 s, PSO shows a MAPE of 0.8638 with 0.05621 RMSE, while DBN has a MAPE of 0.7713 with 0.003214 RMSE which is a better alternative in 0.8643 s. LLWNN within 0.9542 s shows RMSE of 0.09762 with 0.9011 MAPE which is worth mentioning.

Table 4 represents the exchange rate details during the time of testing. Here, LLWNN in 0.9073 s shows MAPE of 0.6909 and 0.0879 RMSE which is higher in the case of DBN having 0.6593 MAPE and 0.005421 RMSE in 0.7659 s. PSO is also having a better RMSE of 0.0953 in less time of 0.8703 s with a MAPE of 0.7430.

Table 5 represents the Household Electricity consumption details during the time of testing. Here, DBN shows better RMSE of 0.0008906 in 0.7642 s with 0.7802 MAPE than PSO having 0.005971 RMSE in 0.6091 s. The results of LLWNN are worth mentioning as it is having 0.009851 RMSE in 0.8903 s with 0.8979 MAPE.

Table 6 depicts the average distance during testing between the actual and forecasted prices for the stock market. It is apparent from the table that DBN has the bottommost variance between real and estimated price, i.e., 0001. Next to DBN, the better performance between PSO and LLWNN is PSO having the distance as 0.002. The LLWNN model is showing a distance of 0.003.

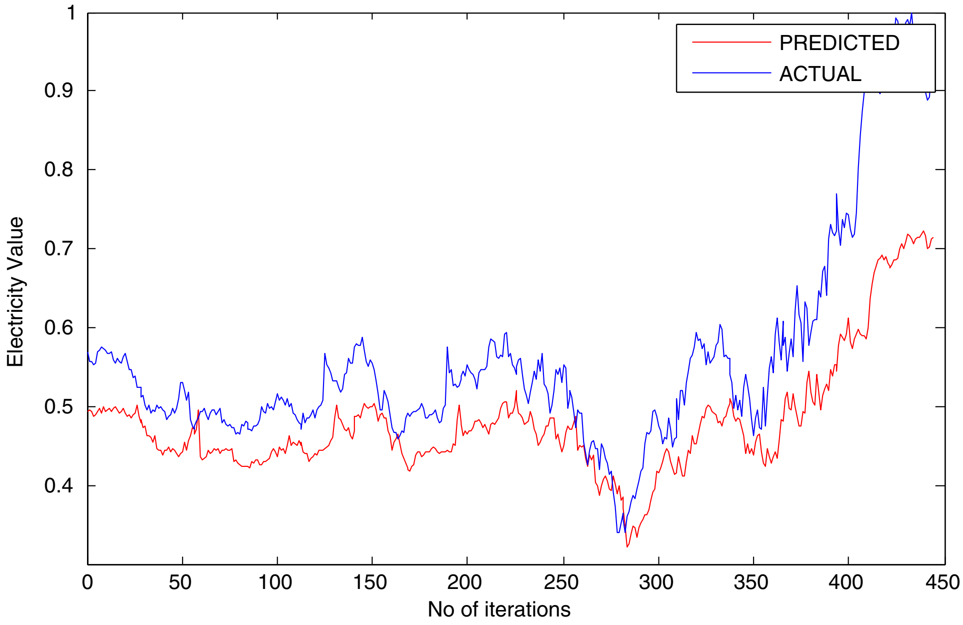


Fig. 20 Real versus forecast during testing for LLWNN in electricity consumption

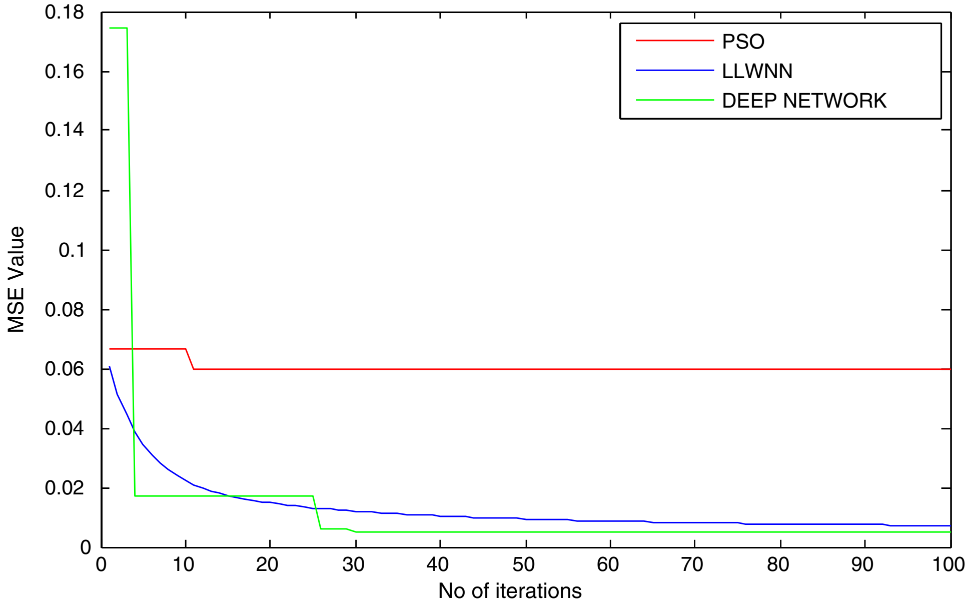


Fig. 21 MSE conjunction graph of stock market

Table 7 describes the distance between the actual and forecasted prices for the exchange rate. The DBN model provides the best results having a distance of 0.007. The PSO and LLWNN model provides a distance of 0.008 and 0.09, respectively.

Table 8 discusses the distance between actual and forecasted for household electricity consumption. The DBN model provides a distance of 0.004 whereas PSO and LLWNN model provide a distance of 0.007and 0.006, respectively.

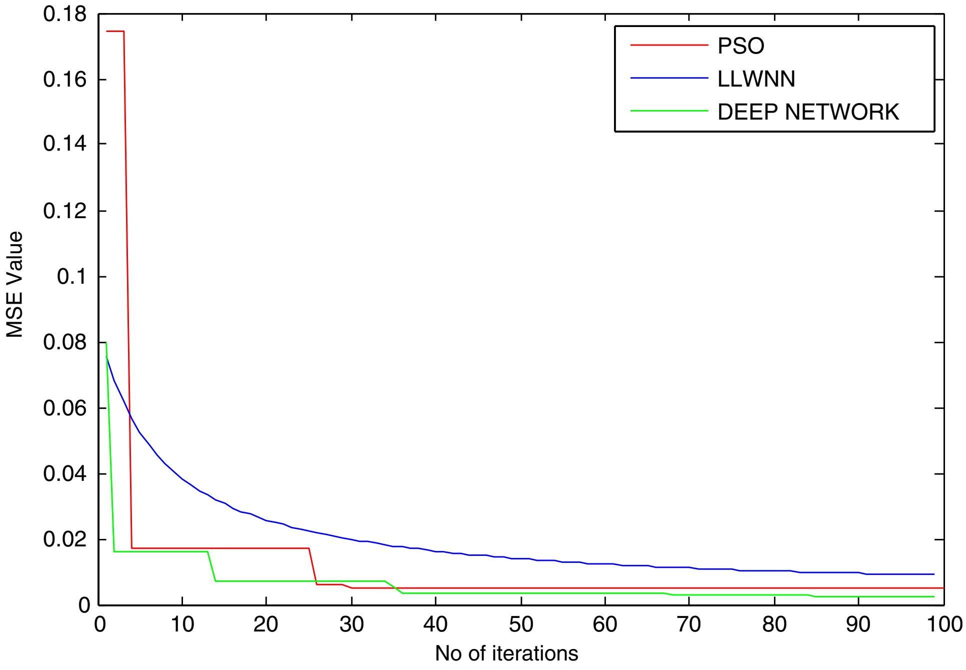
Table 9 depicts the external validation results where DBN is associated with additional different techniques. It shows better RMSE than other method.

Figures 3 and 4 depict the assessment between the real and predicted prices for the stock market using DBN for training and testing.

Figures 5 and 6 depict the assessment between the real and prediction prices for the stock market using PSO for training and testing.

Figures 7 and 8 depict the assessment between the real and predicted prices for the stock market using LLWNN for training and testing.

Fig. 22 MSE conjunction graph of exchange rate



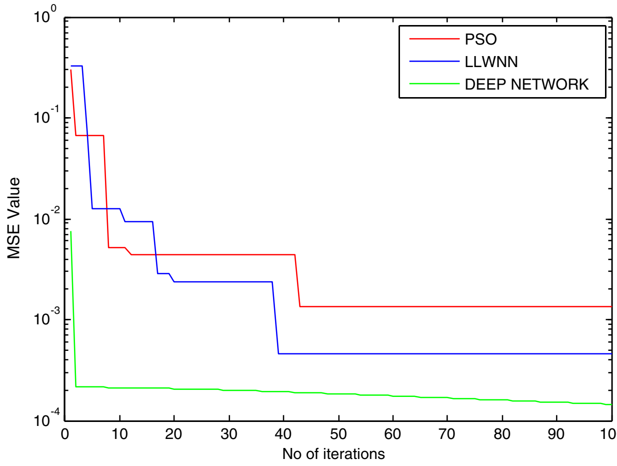


Fig. 23 MSE conjunction graph of electricity consumption

Figures 9 and 10 depict the assessment between the real and predicted prices for the exchange rate using DBN for training and testing.

Figures 11 and 12 depict the assessment between the real and predicted prices for the exchange rate using PSO for training and testing.

Figures 13 and 14 depict the comparison between the actual and forecasted prices for the exchange rate using LLWNN for training and testing.

Figures 15 and 16 depict the comparison between the actual and forecasted prices for electricity consumption using DBN for training and testing.

Figures 17 and 18 depict the comparison between the actual and forecasted price for electricity consumption using PSO for training and testing.

Figures 19 and 20 depict the comparison between the actual and forecasted price for electricity consumption using LLWNN for training and testing.

Figures 21, 22, and 23 present the comparison of MSE Convergence using different methods for the Stock market, Exchange rate, and Electricity consumption, respectively. It is evident from the graphs that DBN converges faster than other methods in all the datasets.

Conclusion

Analyzing the trends of time series data remain a challenging task even after a decade of extensive research. As the time series data does not follow any particular pattern, it proves difficult for the researchers to analyze it and use it conveniently. This paper presents a state of art predictive model using DBN to analyze the time series data. The data are also analyzed by PSO and LLWNN. The assessment of the model is validated with the benefit of RMSE and MAPE. The predictive model led by DBN provides an RMSE value of 0.0032, 0.0054, and 0.0089, respectively, for the Stock market data, Exchange rate data, and Electricity consumption data, respectively. The MAPE values are 0.7713, 0.6593 and 0.7802 for the Stock market data, Exchange rate data and Electricity consumption data, respectively.

In the future, more complex structures can be designed with the help of DBN. It may be applied using different feature selection algorithms for large datasets.

Funding No backing source for doing this research effort.

Declarations

Conflict of interest There is no conflict of interest**.**

References

1. S.M. Idrees, M.A. Alam, P. Agarwal, A prediction approach for stock market volatility based on time series data. IEEE Access (2019). https://doi.org/10.1109/ACCESS.2019.2895252
2. T. Raicharoen, C. Lursinsap, P. Sanguanbhokai (2003) Application of critical support vector machine to time series prediction, in *Circuits and Systems, ISCAS'03,* vol 5, IEEE (2003), pp. V–V
3. S. Das et al., A self-adaptive fuzzy-based optimized functional link artificial neural network model for financial time series prediction. Int. J. Bus. Forecast. Mark. Intell. **2**(1), 55–77 (2015)
4. S. Dash, M.R. Senapati, U.R. Jena, K-NN based automated reasoning using a bilateral filter based texture descriptor for computing texture classification. Egypt. Inf. J. (2018). https://doi.org/10.1016/j.eij.2018.01.003
5. C.W.J. Granger, P. Newbold, *Forecasting Economic Time Series* (Academic Press, Cambridge, 2014)
6. A. Khodabakhsh et al., Multivariate sensor data analysis for oil refineries and multi-mode identification of system behavior in real-time. IEEE Access **6**, 64389–64405 (2018)
7. Agarwal, *Introduction to the Stock Market*. Intelligent Economist. Date accessed December 18 (2017)
8. F. Shen, J. Chao, J. Zhao, Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing **167**, 243–253 (2015). https://doi.org/10.1016/j.neucom.2015.04.071
9. B. Donga, Z. Lia, S.M.M. Rahmana, R. Vega, A hybrid model approach for forecasting future residential electricity consumption. Energy Build. **117**, 341–351 (2016). https://doi.org/10.1016/j.enbuild.2015.09.033
10. D. Shah, H. Isah, F. Zulkernine, Stock market analysis: a review and taxonomy of prediction techniques. Int. J. Financ. Stud. **7**, 26 (2019). https://doi.org/10.3390/ijfs7020026
11. Walter H.L. Pinaya et al. Using deep belief network modelling to characterize differences in brain morphometry in schizophrenia. Sci. Rep. **6**, Article No: 38897 (2016)
12. Y. Hua, J. Guo, H. Zhao, Deep belief networks and deep learning (2015). https://doi.org/10.1109/ICAIOT.2015.7111524
13. X. Dai et al., Deep belief network for feature extraction of urban artificial targets. Math. Probl. Eng. **2020**, 1–13 (2020). https://doi.org/10.1155/2020/2387823
14. X. Pang et al., An innovative neural network approach for stock market prediction. J. Supercomput **76**, 2098–2118 (2020). https://doi.org/10.1007/s11227-017-2228-y
15. S. Mehtab, J. Sen, *Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Timeseries*, IEEE (2020). arXiv:2001.09769
16. 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**, 450–457 (2015). https://doi.org/10.1016/j.jksuci.2015.01.002
17. T.A. Chaudhuri, I. Ghosh, Artificial neural network and time series modeling based approach to forecasting the exchange rate in a multivariate framework. J. Insur. Financ. Manag. **1**(5), 92–123 (2016)
18. R. Dash, Performance analysis of a higher-order neural network with an improved shuffled frog leaping algorithm for currency exchange rate prediction. Appl. Soft Comput. **67**, 215–231 (2018). https://doi.org/10.1016/j.asoc.2018.02.043
19. A. Rahmana, V. Srikumar, A.D. Smitha, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Appl. Energy **212**, 372–385 (2018). https://doi.org/10.1016/j.apenergy.2017.12.051
20. K.K. Chakravarthi, L. Shyamala, V. Vaidehi, Cost–effective workflow scheduling approach on cloud under deadline constraint using firefly algorithm. Appl. Intell. (2020). https://doi.org/10.1007/s10489-020-01875-1
21. Q. Leng et al., A soft-margin convex polyhedron classifier for nonlinear task with noise tolerance. Appl. Intell. (2020). https://doi.org/10.1007/s10489-020-01854-6
22. M. Qin, Z. Li, Z. Du, Red tide time series forecasting by combining ARIMA and deep belief network. Knowl.-Based Syst. (2017). https://doi.org/10.1016/j.knosys.2017.03.027
23. G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief networks. Neural Comput. **18**(7), 1527–1554 (2006)
24. M. Saleem, *Deep Learning For Speech Classification And Speaker Recognition*, Thesis December (2014)
25. A. Mohamed, G.E. Hinton, G Penn, Understanding how deep belief networks perform acoustic modelling, in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4273–4276 (2012)
26. Y.Z. Hsieh, M.C. Su, P.C. Wang, A PSO-based rule extractor for medical diagnosis. J. Biomed. Inform. **49**, 53–60 (2014). https://doi.org/10.1016/j.jbi.2014.05.001
27. M.R. Senapati, S. Das, S. Mishra, A novel model for stock price prediction using hybrid neural network. J. Inst. Eng. India Ser. B **99**(6), 555–563 (2018). https://doi.org/10.1007/s40031-018-0343-7
28. M.R. Senapati, P.K. Dash, Local linear wavelet neural network-based breast tumor classification using firefly algorithm. Neural Comput. Appl. **22**, 1591–1598 (2013). https://doi.org/10.1007/s00521-012-0927-0. (**ISSN 0941–0643**)
29. Y. Chen, B. Yang, J. Dong, Time series prediction using a local linear wavelet neural network. NeuroComputing **69**, 449–465 (2006). https://doi.org/10.1016/j.neucom.2005.02.006
30. Yahoo finance, www.finance.yahoo.com
31. Investing.com (currency exchange dataset), www.investing.com/currency exchange
32. L.M. Candanedo, V. Feldheim, D. Deramaix, Data-driven prediction models of energy use of appliances in a low-energy house. Energy Build. **140**, 81–97 (2017). https://doi.org/10.1016/j.enbuild.2017.01.083
33. G. Hebrail, *UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems* (Individual household electric power consumption Data Set, 2012), https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption