Engineering Applications of Artificial Intelligence 66 (2017) 49–59

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A hybrid ETS–ANN model for time series forecasting

Sibarama Panigrahi, H.S. Behera *

Department of Computer Sc. Engineering and Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha, 768018, India

ARTICLE INFO

Keywords:
Time series forecasting
Exponential smoothing
ETS
ANN
Hybrid model

ABSTRACT

Over the past few decades, a large literature has evolved to forecast time series using various linear, nonlinear and hybrid linear–nonlinear models. Recently, hybrid models by suitably combining linear models like autoregressive integrated moving average (ARIMA) with nonlinear models like artificial neural network (ANN) have become popular due to superior performance than individual models. These models assume the time series to be a sum of a linear and a nonlinear component. However, a real world time series may be purely linear or purely nonlinear or often contains a combination of linear and nonlinear patterns. Motivated by this need, a new hybrid methodology is developed by combining linear and nonlinear exponential smoothing models from innovation state space (ETS) with ANN. The proposed hybrid ETS–ANN model glorifies the chances of capturing different combination of linear and/or nonlinear patterns in time series. This is because both ETS and ANN models have linear as well as nonlinear modeling capability. However, ANN cannot handle linear patterns equally well as nonlinear patterns. Therefore, in the proposed method, first ETS is applied to the given time series and predictions are obtained. This enhances the chances of capturing existing linear patterns (if any) well using linear ETS models. Then residual error sequence is calculated by subtracting the ETS-predictions from the original series. The residual error sequence obtained is modeled by ANN. Then final prediction is obtained by combining the ETS-predictions with ANN-predictions. Sixteen time series datasets are used for comparative performance analysis of the proposed methodology with ARIMA, ETS, multilayer perceptron(MLP) and some existing hybrid ARIMA–ANN models. Experimental results show that the proposed hybrid model shows statistically promising result for the datasets used.

1. Introduction

Time series forecasting (TSF) plays a key role in making better social, organizational, economical and individual strategic decision under uncertainty. For example, accurate bankruptcy prediction and credit scoring (Lin et al., 2012) assist financial institutions to avoid financial crisis, forecasting electricity load (Raza and Khosravi, 2015; Behera et al., 2010) helps in better power system planning and in achieving the concept of next generation power system such as smart grid, forecasting electricity price (Weron, 2014) assist energy companies to avoid over/under contracting and then selling/buying power in the balancing market, forecasting internet traffic (Meade and Islam, 2015) helps service providers to enhance their service, forecasting call volumes (Meade and Islam, 2015) in a call center assists scheduling staff, forecasting natural physical phenomena like whether (Maqsood et al., 2004), earthquake (Reyes et al., 2013) assist mankind to be prepared by taking necessary precautions. Latest review papers provide a summary of application of forecasting in several areas (Lin et al., 2012; Raza and Khosravi, 2015; Weron, 2014; Meade and Islam, 2015; Donkor et al., 2014; Fagiani et al., 2015; Chandra et al., 2013; Zhang, 2003). Future of most of these phenomena usually captures true relationships existing in the past observations. Therefore, future of a phenomenon can be predicted by systematically analyzing the past observations and such a process is known as time series forecasting. The past observations are recorded sequentially through time to form a time series which is represented by a vector $y = [y_{1,}y_{2,\ldots,}y_n]^T$. Based on the number of time series used in forecasting, the TSF method may be multivariate or univariate. In multivariate methods, in addition to the prediction variable or time series, forecasts depend on values of one or more additional time series, called predictor or explanatory variables. However, in univariate methods, an appropriate model is developed by carefully analyzing a single time series and then the model is used to predict the future values of the series. The univariate TSF method is particularly useful when little knowledge is available on the underlying data generating process or when there is no satisfactory explanatory model that relates prediction

* Corresponding author.
E-mail addresses: panigrahi.sibarama@gmail.com (S. Panigrahi), hsbehera_india@yahoo.com (H.S. Behera).

http://dx.doi.org/10.1016/j.engappai.2017.07.007
Received 12 June 2017; Received in revised form 8 July 2017; Accepted 10 July 2017
Available online 23 September 2017
0952-1976/

variable to other explanatory variables (Zhang, 2003). In this paper, an attempt has been made to develop a hybrid forecasting model using univariate forecasting method.

Traditionally, various statistical models like moving average, autoregressive integrated moving average (ARIMA), exponential smoothening are used predominantly for TSF. These models assume that the time series under study is generated from a linear process. However, most of the time series processes often exhibit temporal and spatial variability, and are suffered by issues of nonlinearity of physical processes (Zhang et al., 1988). To deal with the nonlinear patterns observed in real world time series, some nonlinear statistical models, such as the autoregressive heteroskedastic models (ARCH and GARCH family) are developed. However, the availability of more than hundred variations of these models (Enders, 1995; Hamilton, 1994) with each capturing only a specific nonlinear pattern makes it difficult to select an appropriate model for the time series under study. Thus, the benefits of these models are limited in general forecasting problem. Recently, artificial neural networks (ANNs) have found increasing consideration in forecasting theory due to its several advantageous features such as universal approximation capability (Hornik et al., 1989), data driven models and nonlinear modeling capability. Despite of several advantageous features, neural network model alone is not capable to handle linear as well as nonlinear patterns equally well (Zhang, 2003). Therefore, several hybrid models (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Babu and Reddy, 2014; de Oliveira and Ludermir, 2016; Khandelwal et al., 2015) were developed by suitably combining a linear model with a nonlinear model. These hybrid models demonstrated better overall forecast accuracy than individual models. However, despite exponential smoothing methods including nonlinear methods (ETS) are optimal forecasts from innovation state space models (Hyndman and Khandakar, 2008) and nonlinear exponential smoothing models have no equivalent ARIMA counterpart, the effectiveness of exponential smoothing models have not yet been explored in hybrid models. Motivated by this, an attempt is made to develop a new hybrid model by combining the linear and nonlinear exponential smoothing models from innovation state space (ETS) with ANN model.

The rest of this paper is organized as follows. Section 2 demonstrates a summary of some of the forecasting models including ARIMA, exponential smoothing, ANN and hybrid ARIMA–ANN. The proposed hybrid methodology is explained in Section 3. Section 4 gives the experimental set up and simulation results. Finally conclusions and future works are described in Section 5.

2. Preliminaries

2.1. ARIMA

ARIMA is a linear forecasting model for stationary time series. Therefore, in the first step, a time series is made stationary by differencing d times in conjunction with nonlinear transformations such as logging or deflating (if necessary). The consequential data are considered as a linear function of past p data values and q errors (as in Eq. (1)) i.e. modeled as an autoregressive moving average (ARMA) time series.

$$y_t = \varnothing_1 y_{t-1} + \varnothing_2 y_{t-2} + \cdots + \varnothing_p y_{t-p} + \Theta_1 \varepsilon_{t-1} + \Theta_2 \varepsilon_{t-2} + \cdots \Theta_q \varepsilon_{q-1} \quad (1)$$

where y_t represents actual value at tth time period, ε_t represents the error sequence which is assumed to be white noise and is Gaussian distributed i.e. with mean zero and constant variance of σ^2. $\varnothing_i\,(i = 1, 2, \ldots, p)$ are autoregressive (AR) coefficients and $\Theta_j\,(j = 1, 2, \ldots, q)$ are moving average (MA) coefficients. p and q integers are referred as model orders. Considering all, the time series model is denoted as ARIMA(p, d, q). Some unique cases can be attained by setting extreme values to the order of the model parameters. For instance, if $q = 0$, then the model reduces to an autoregressive model of order p and if $p = 0$, the model represents moving average model of order q. One of the most crucial parts in ARIMA modeling procedure is to identify the suitable order (p, q) of model.

Box and Jenkins (Khandelwal et al., 2015) developed a realistic method to build ARIMA model. The modeling procedure consists of three steps: (a) identification of model order (b) estimation of model coefficients; and (c) forecasting the data. In the first step, after the time series is made stationary, the model parameters p and q are identified using correlation analysis (Box and Jenkins, 1990). Once a tentative model is identified, in the second step, the model coefficients are estimated using nonlinear optimization procedures like Gaussian maximum likelihood estimation (GMLE) (Khashei and Bijari, 2006). The model is confirmed using Akaike information criterion (AIC). The most suitable ARIMA model has the minimum AIC value. Finally, using selected ARIMA model, estimated model coefficients and past values of series the future values are predicted.

2.2. Exponential smoothing

In exponential smoothing methods, the forecasts are made considering weighted averages of past observations. The latest observations are given exponentially more weight than older observations. Since its inception in 1950, a variety of exponential smoothing methods have been developed. All exponential smoothing methods were originally classified by Pegels (1969) which is extended by Gardner (1985). Later these methods were again modified by Hyndman et al. (2002), and again extended by Taylor (2003). Considering all modifications and extensions, a total of fifteen different exponential methods were developed. These fifteen methods are discriminated based on the nature of trend and seasonality component observed. The trend components may be absent or additive or additive damped or multiplicative or multiplicative damped. The seasonality component may be absent or additive or multiplicative. By considering additive and multiplicative error components along with trend and seasonality component, Hyndman et al. (2008b) described two possible innovations state space models for each of the fifteen models, resulting in thirty different models. In addition, they have developed an automatic forecasting method using these thirty models. To distinguish these models, a triplet (E,T,S) was used. ETS stands for error, trend and seasonality components. The general model for all these models involves a state vector $x_t = \left(l_t, b_t, s_t, s_{t-1}, \ldots, s_{t-m+1}\right)'$ and the state space equations (Hyndman and Khandakar, 2008) have the following form

$$y_t = w\left(x_{t-1}\right) + r\left(x_{t-1}\right)\varepsilon_t \quad (2)$$

$$x_t = f\left(x_{t-1}\right) + g\left(x_{t-1}\right)\varepsilon_t \quad (3)$$

where $\left\{\varepsilon_t\right\}$ is a Gaussian white noise process with mean zero and variance σ^2 and $\mu_t = w\left(x_{t-1}\right)$. The model with additive error has $r\left(x_{t-1}\right) = 1$, so that $y_t = \mu_t + \varepsilon_t$. The model with multiplicative errors has $r\left(x_{t-1}\right) = \mu_t$, so that $y_t = \mu_t\left(1 + \varepsilon_t\right)$. Thus, $\varepsilon_t = \left(y_t - \mu_t\right)/\mu_t$ is a relative error for the multiplicative model and any value of $r\left(x_{t-1}\right)$ will lead to identical point forecast for y_t (Hyndman and Khandakar, 2008). For a detailed description of state space models for exponential smoothing methods, the interested readers may go through (Hyndman and Khandakar, 2008; Hyndman et al., 2008b).

2.3. Artificial neural network

ANNs are data driven flexible models that are capable of approximating a large class of nonlinear problems to any desired level of accuracy. Thus, a variety of ANN models have been developed and applied in a wide range of applications across diversified fields. Out of various ANN models, single hidden layer multilayer perceptron are used widely in time series forecasting (Zhang et al., 1988). The network architecture (as in Fig. 1) of such a model consists of an input layer, single hidden layer and an output layer with adjacent interlayer nodes are connected by acyclic links. In forecasting problem, the number of inputs and number of neurons in the hidden layer is flexible whereas

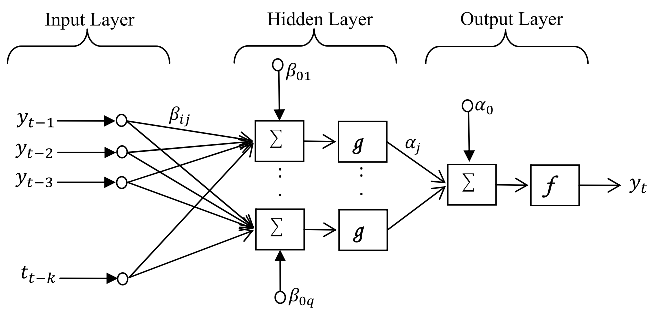

Fig. 1. Architecture of three-layer ANN.

the output layer has a single neuron. To model the TSF problem using such a network is considered as obtaining the relationship (as in Eq. (4)) between the output y_t and past k values of the series $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$, with k denoting the number of inputs of the ANN. Therefore, before modeling time series using ANN, the time series is first transformed to a pattern set of $n-k+1$ patterns, with each pattern consisting of k inputs $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$ and one target y_t. Usually, the first $x\%$ of the patterns is used for training, next $y\%$ is used for validation and rest is used for testing. The training set and validation set are used by training algorithms like back propagation, Levenberg–Marquardt (LM) algorithm or evolutionary training algorithms (Sahu et al., 2013; Panigrahi, 2017; Karali et al., 2013) to obtain the near optimal weights of the ANN. Then the performance of the model is evaluated using the test set.

$$y_t = f\left(\alpha_0 + \sum_{j=1}^{q} \alpha_j g\left(\beta_{0j} + \sum_{i=1}^{k} \beta_{ij} y_{t-1}\right)\right) \tag{4}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{5}$$

where $\alpha_j (j = 0, 1, 2, \ldots, q)$ is the weight between j th hidden neuron and output neuron, α_0 is the bias unit of output neuron, β_{0j} is the bias at j th hidden neuron, β_{ij} is the weight between i th input and j th hidden neuron, k is the number of inputs and q is the number of hidden neurons. Generally, sigmoid activation function (as in Eq. (5)) is used at hidden layer whereas linear activation function is used at output layer.

2.4. Zhang's hybrid ARIMA–ANN model

In 2003, Zhang (2003) presented a hybrid time series forecasting model using both ARIMA and ANN models. This hybrid model assumes every time series y_t is a combination of linear patterns L_t and nonlinear patterns N_t (as shown in Eq. (6)). First, an ARIMA model is fit directly to the time series y_t to capture the linear patterns and then residual error e_t (as shown in Eq. (7)) is obtained by subtracting the ARIMA predictions $\hat{L}_t$ from the actual data y_t. The residual error sequence e_t is assumed to be nonlinear and is used by ANN to obtain the predictions $\hat{N}_t$fore$_t$. Then the predictions from ARIMA $\hat{L}_t$ and predictions from ANN $\hat{N}_t$are combined to obtain the final predictions $\hat{y}_t$ (as shown in Eq. (8)).

$$y_t = L_t + N_t \tag{6}$$

$$e_t = y_t - \hat{L}_t \tag{7}$$

$$\hat{y}_t = \hat{L}_t + \hat{N}_t. \tag{8}$$

2.5. Babu and Reddy's hybrid ARIMA–ANN model

In 2014, Babu and Reddy (2014) proposed another hybrid ARIMA–ANN model for time series forecasting. In this technique, the time series is assumed to be a sum of low-volatility l_t and high-volatility components h_t (as shown in Eq. (9)). Unlike Zhang's model (Zhang, 2003), the time series is first decomposed into low-volatile l_t and high-volatile h_t components by using a moving average filter (as shown in Eq. (10)). The length of moving average filter q is adjusted each iteration until the kurtosis has a value of 3. This is because, according to Jarque–Bera normality test, a series with kurtosis (as shown in Eq. (11)) value 3 is Gaussian and is considered to be low-volatile which is appropriate for the application of ARIMA model. Once the suitable length of moving average filter is found, the data is decomposed into low-volatile and high-volatile components. After the decomposition, the low-volatile sequence l_t is modeled using ARIMA and the high-volatile sequence h_t is modeled using ANN. Then the final predictions (as shown in Eq. (12)) are obtained by combining the ARIMA predicted low-volatility component $\hat{l}_t$ and ANN predicted high-volatility component $\hat{h}_t$.

$$y_t = l_t + h_t \tag{9}$$

$$y_{ma} = \frac{1}{q} \sum_{i=t-q+1}^{t} y_i \tag{10}$$

$$kurtosis(y) = \frac{E((y-E(y)))^4}{\left(E\left((y-E(y))^2\right)\right)^2} \tag{11}$$

$$\hat{y}_t = \hat{l}_t + \hat{h}_t. \tag{12}$$

3. Proposed method

In recent years, several hybrid univariate forecasting models (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Babu and Reddy, 2014; de Oliveira and Ludermir, 2016) have been developed by suitably combining linear and nonlinear models. The hybrid models have used two approaches of presenting input data to individual models. In the first approach (Babu and Reddy, 2014; de Oliveira and Ludermir, 2016) the time series is first decomposed into two series with one series modeled by a linear model and other one modeled by a nonlinear model. In this approach, a moving average filter (Babu and Reddy, 2014) or exponential smoothing filter (de Oliveira and Ludermir, 2016) is usually used for the decomposition of original series. However, as the length of the filter increases more data values are lost in the process of averaging (Makridakis et al., 1998). The situation becomes even worse in case of a smaller time series with a larger length of moving average filter, since less data is available for the model building process. In contrast to the first approach, the second one (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Khandelwal et al., 2015) applies a linear model directly to the original series. Then the residual error sequence is obtained and is modeled by a nonlinear model. Finally, the final prediction is obtained by combining the predictions obtained from linear and nonlinear models.

All hybrid models (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Babu and Reddy, 2014; de Oliveira and Ludermir, 2016; Khandelwal et al., 2015) assume a time series to be a combination of linear or low-volatile and nonlinear or high-volatile components. The hybrid models used ARIMA (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Babu and Reddy, 2014; de Oliveira and Ludermir, 2016) or Wavelet Transform (DWT) (Khandelwal et al., 2015) to model linear/low-volatile components and ANN (Zhang, 2003; Faruk, 2010; Khashei and Bijari, 2011; Babu and Reddy, 2014; Khandelwal et al., 2015) or support vector machine (SVM) (de Oliveira and Ludermir, 2016) to handle the nonlinear/high-volatile components. These models perform well when a time series is composed of a linear and a nonlinear pattern. However, a real world time series may be purely linear or purely nonlinear or often contain a combination of linear and nonlinear patterns. This demands a new hybrid methodology that can handle a time series with different combination of linear and/or nonlinear patterns. Therefore in this study, instead of linear ARIMA models, exponential smoothing methods with error, trend and seasonality (ETS) component from innovation state space models (Hyndman and Khandakar, 2008) is hybridized with ANN. ETS has both linear and nonlinear models. The linear ETS models are special cases of ARIMA models whereas nonlinear ETS models have no ARIMA counterpart (Hyndman and Khandakar, 2008). The use of ETS not only retains some of the advantages of ARIMA but also provides some nonlinear models. The linear and nonlinear ETS models assist in capturing linear and nonlinear patterns respectively. Motivated by this, a hybrid ETS–ANN model is proposed to capture different combination of linear and/or nonlinear patterns existing in a time series.

$$y_t = C_t^1 + C_t^2 \tag{13}$$

$$e_t = y_t - \widehat{C_t^1} \tag{14}$$

$$\hat{y}_t = \widehat{C_t^1} + \widehat{C_t^2}. \tag{15}$$

Algorithm 1 presents the proposed methodology. The proposed model (as shown in Fig. 2) assumes the time series is a sum of two components C_t^1 and C_t^2 (as in Eq. (13)). The two components may be either linear or nonlinear, resulting in three different types of linear and nonlinear pattern combinations. Two models ETS and ANN are chosen to capture the two components since both ETS and ANN have linear as well as nonlinear modeling capability. In this model, first ETS is applied to the original series y_t and the predictions $\widehat{C_t^1}$ are obtained. This is because: (1) ETS has both linear and nonlinear models; (2) though ANN can handle linear patterns, it cannot handle linear patterns equally well like nonlinear patterns; (3) If the time series contains combination of a linear and a nonlinear pattern, more chances will be there to capture the linear patterns by ETS linear models and the resulting residual error sequence becomes nonlinear which again enhances the chances to be captured well by ANN model. The application of ETS require O(n) operations to obtain the predictions since in worst case the ETS uses all past observations to obtain the predictions. Then the residual error sequence e_t is obtained by subtracting first component predictions $\widehat{C_t^1}$ from original series y_t (as in Eq. (14)) which requires O(n) operations. The residual error sequence e_t is modeled by ANN to obtain the predictions $\widehat{C_t^2}$. Assuming x training patterns, y validation patterns for a single hidden layer one output neuron ANN model with n_{iter} iterations required for convergence, the complexity of the ANN model is $n_{iter} \times (x + y)$. For a time series $y = \left[y_1, y_{2,\ldots,} y_n\right]^T$ and k -input ANN the total number of train and validation patterns is$(n - k + 1)$. Thus the resulting number of operations is $n_{iter} \times (n - k + 1)$ which is equal to O(n) considering n_{iter} and k as constant. Finally, the final prediction is obtained by combining the predictions obtained by ETS model $\widehat{C_t^1}$ with the predictions obtained from ANN $\widehat{C_t^2}$ (as shown in Eq. (15)) which again requires O(n) operations. Considering all, the complexity of the proposed model is O(n).

The proposed model can give better prediction compared with the other hybrid models existing in the literature. This can be understood from the following reasons. Zhang (2003), Faruk (2010) and Khashei and Bijari (2011) applied ARIMA directly to the original series to capture the linear patterns. These models will give poor result in case the time series contains only nonlinear patterns. This is because a nonlinear sequence cannot be efficiently modeled by a linear model and vice versa (Babu and Reddy, 2014). Moreover, ideal ARIMA have two interesting properties such as stationarity and linearity. Thus ARIMA can only provide better fit if the given stationary time series is truly Gaussian. Thus, if a time series do not satisfy any of the conditions, ARIMA consequently the resulting hybrid model provide poor forecast. This problem is delicately neutralized in the proposed hybrid model, since instead of ARIMA, ETS is applied directly which have both linear and non-linear modeling capability. Recently, Babu and Reddy (2014) and de Oliveira and Ludermir (2016) first decomposed the time series into low and high volatile components using a moving average and exponential smoothing filter respectively. The low-volatile component is modeled by ARIMA and high volatile component is modeled by ANN. The performance of these models depends on the kurtosis value of low-volatile component. The low-volatile component must have to have a kurtosis value of 3 to be efficiently modeled by ARIMA. Most often two problems arise which are: (1) more data points are lost in the decomposition process (2) the low-volatile component may not have a kurtosis value 3. For both the cases these two models give poor forecast which is not a case in the proposed model. Moreover, the decomposition and repeated estimation of kurtosis require higher computational overhead which is again not a case in the proposed model. Hence the proposed model is not only simple but also provides better chance to achieve improved forecast than all hybrid models in the considered literature. This fact is verified by the extensive simulation and result analysis.

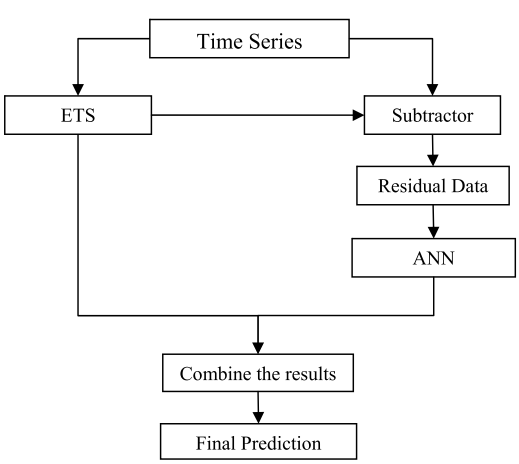

Fig. 2. Proposed methodology.

4. Experimental setup and results

In this study, to evaluate the performance of proposed model, three individual models such as ARIMA (Hyndman and Khandakar, 2008), ETS (Hyndman and Khandakar, 2008), multilayer perceptron(MLP) and two hybrid ARIMA–ANN models including Zhang's (Zhang, 2003) and Babu and Reddy (2014) were considered. The ARIMA and ETS models were implemented and optimized using the Forecast package for R (Hyndman and Khandakar, 2008) and the rest were simulated in MATLAB. For MLP and both the hybrid models, single hidden layer MLP was used as the ANN model and trained using Levenberg–Marquardt algorithm. The number of inputs is obtained by analyzing the autocorrelation function (ACF) of the time series. The number of hidden neurons h of ANN was determined by making simulations in the range one to twenty and the configuration that attain smallest mean SMAPE over 50 executions was presented in the result. The number of neurons in the output layer was fixed to one. The maximum length of moving average filter q for Babu and Reddy's method was set to half the length of training data and was selected based on the minimum value of $|3 - kurtosis\left(l_t\right)|$. This is because, for some time series the length of moving average filter is so large that all training data are lost in the process of averaging. Also, in some time series the kurtosis of low-volatile component l_t never achieved a value 3 even by considering the length of moving average filter up to the length of training data.

Sixteen datasets (described in Table 1) from the Time Series Data Library Hyndman (2010) were considered for comparative performance analysis. Each time series was divided into two subsets: in-sample and out-of-sample (test-samples). The in-samples were used to build the model whereas the test-samples were used to assess the performance of the models. The in-sample contained approximately 80% of the data and rest was used as out-of-samples. For all models excluding ARIMA and ETS, the in-sample data is again divided into train set (approximately 60% of original series) and validation set (approximately 20% of original series). The division of datasets, the number of significant lags, ARIMA and ETS models used in the simulations for all time series is presented

Algorithm 1: The proposed methodology

1: Given a time series$y = [y_1, y_2, \ldots, y_n]^T$.

2: Input the length of in-sample l_i[train (60%), validation (20%)] and out-of-sample l_o [test (20%)] data.

3: Normalize the time series using min-max normalization considering the minimum and maximum value of in-sample data.

4: Determine the best ETS (E, T, S) model and its parameters using the normalized in-sample data$[y_1', y_2', \ldots, y_{l_i}']^T$.

5: Obtain predictions using selected ETS model$[\widehat{y_1'}, \widehat{y_2'}, \ldots \widehat{y_n'}]^T$.

6:De-normalize the predictions to obtain prediction for the first component $\widehat{C}_t^1$.

7: Obtain the residual series by subtracting ETS-predictions from the original series which represents the second component$e_t = y_t - \widehat{C}_t^1$.

8: Perform lag selection using autocorrelation function on the in-sample data of residual series $[e_1, e_2 \ldots e_{l_i}]^T$

9: Normalize the residual series using min-max normalization considering the minimum and maximum value of in-sample residual data$[e_1', e_2', \ldots, e_{l_i}']^T$.

10: Obtain predictions using ANN model$[\widehat{e_1'}, \widehat{e_2'}, \ldots \widehat{e_n'}]^T$.

11: De-normalize the predictions to obtain the predictions for the second component$\widehat{C}_t^2$.

12: Final predictions are obtained by combining the ETS predictions $\widehat{C}_t^1$ with ANN predictions$\widehat{C}_t^2$.

Table 1
Datasets description.

Time series	Description
Accidental Death	Monthly accidental deaths in USA, 1973–1978
IBM	IBM common stock closing prices: daily, 17/05/1961–2/11/1962
Lake	Monthly Lake Erie Levels, 1921–1970
Lynx	Annual number of lynx trapped, MacKenzie River, 1821–1934
Pollution	Monthly shipment of pollution equipment, 01/1966–10/1976
Stock	Annual common stock price, US, 1871–1970
Sun Spot	Wolf's Sunspot Numbers. 1700–1987
Colorado River	Monthly Flows, Colorado River Lees Ferry. 1911–1972.
Passenger	International airline passengers: monthly totals in thousands. Jan 49–Dec 60.
European Internet Traffic	Hourly Internet traffic data (in bits) from a private ISP with centers in 11 European cities, 06:57 h on 7 June to 11:17 h on 31 July 2005.
UK Internet traffic	Hourly aggregated internet traffic data (in bits) from an ISP of United Kingdom academic network, 19 November 2004, at 09:30 h and 27 January 2005, at 11:11 h. Data
Temperature	Mean monthly air temperature (Deg. F) Nottingham Castle 1920–1939
Unemployment	Monthly Canadian total unemployment figures (thousands) 1956–1975
Milk	Monthly milk production: pounds per cow. 1962–1975, adjusted for month length
Mumps	Monthly reported number of cases of mumps, New York city, 1928–1972
Chickenpox	Monthly reported number of chickenpox, New York city, 1931–1972
standard and poor	Monthly returns of value-weighted S and P (standard and poor) 500 stock from 1926–1991
Traffic	Monthly traffic fatalities in Ontario 1960–1974

in Table 2. Note that for Babu and Reddy's method the number of significant lags and used ARIMA models are different than that of Table 2. In all the methods, data is normalized (as in Eq. (16)) using the minimum and maximum value of in-sample data. The normalized data is given as input to obtain normalized predictions. The normalized predictions are de-normalized and forecast accuracy is measured. Several measures have been proposed to evaluate the forecast accuracy. Hyndman and Koehler (2006) conducted a comparative study on different forecast accuracy measures. In this study, root mean square error (RMSE) (as in Eq. (17)) and symmetric mean absolute percentage error (SMAPE) (as in Eq. (18)) were considered.

$$y' = \frac{y - min_y}{max_y - min_y} \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \tag{17}$$

$$SMAPE = \frac{1}{n}\sum_{j=1}^{n}\frac{|y_j - \hat{y}_j|}{\left(\left|Y_j\right| + |T_j|\right)/2} \tag{18}$$

where y_j and $\hat{y}_j$ are actual and predicted values, respectively at j th time point and n is the number of previously known elements.

In this study, the models are evaluated based on one-step-ahead prediction. By making the aforementioned experimental setup, 50 independent simulations were carried out for each method on each dataset. However, to reduce the inconsistency arising due to ANN model parameters and training algorithms, best 30 results were used for

Table 2
Datasets division, number of lags, ARIMA and ETS models.

Time series	Train patterns	Validation patterns	Test patterns	Number of lags	ARIMA	ETS
Accidental death	43	14	15	12	ARIMA(2,0,2)	ETS(A,N,N)
IBM	221	74	74	10	ARIMA(3,2,1)	ETS(A,N,N)
Lake	360	120	120	12	ARIMA(2,1,3)	ETS(A,Ad,N)
Lynx	68	23	23	12	ARIMA(1,1,1)	ETS(A,N,N)
Pollution	78	26	26	12	ARIMA(2,1,0)	ETS(A,Ad,N)
Stock	59	20	20	10	ARIMA(2,1,1)	ETS(A,N,N)
Sun spot	172	58	58	11	ARIMA(2,0,1)	ETS(A,N,N)
Colorado river	446	149	149	12	ARIMA(2,1,1)	ETS(A,N,N)
Passenger	86	29	29	12	ARIMA(3,1,3)	ETS(A,N,N)
Internet traffic data of UK	994	331	332	24	ARIMA(1,1,2)	ETS(A,Ad,N)
Temperature	144	48	48	12	ARIMA(5,0,1)	ETS(A,N,N)
Unemployment	144	48	48	12	ARIMA(2,0,3)	ETS(A,Ad,N)
Milk	93	31	32	12	ARIMA(1,1,3)	ETS(A,Ad,N)
Mumps	320	107	107	12	ARIMA(2,0,2)	ETS(A,Ad,N)
Chickenpox	298	100	100	12	ARIMA(4,0,2)	ETS(A,Ad,N)
Traffic	108	36	36	12	ARIMA(0,1,0)	ETS(A,N,N)

performance analysis. The analysis is divided into two parts: (1) analysis based on RMSE (2) analysis based on SMAPE.

4.1. Analysis based on RMSE

In this section, the results are analyzed based on RMSE measure. The RMSE measure is scale dependent and sensitive to outlier data, thus it is not used to compare forecasting models across multiple time series. Therefore the comparisons are restricted to individual time series. Table 3 represents the mean RMSE over 30 best executions out of 50 executions. It can be observed that the proposed ETS–ANN methodology provides best RMSE on seven datasets. However, to compare the proposed model against other models, a Wilcoxon signed-rank test with 95% of confidence level was applied. The results are shown in Table 4 indicating which model is better (+), worse (−) and equivalent (≈) than the proposed model. The proposed model out-performed all models in six datasets (Sun Spot, Passenger, Unemployment, Milk, Chickenpox and Traffic). The proposed model achieved better RMSE when compared to: ETS in 13 datasets, ARIMA in 12 datasets, MLP in 14 datasets, Babu and Reddy (2014) in 10 datasets and Zhang (2003) in 9 datasets. To have a better idea regarding the performance of the proposed methodology, for each time series comparison graphs (Figs. 3–18) were plotted showing how close the forecasted values to that of actual values.

Table 3
Root mean square error (RMSE) for all datasets (best values in bold).

	Hyndman and Khandakar (2008) ETS	Hyndman and Khandakar (2008) ARIMA	MLP	Babu and Reddy (2014)	Zhang (2003)	Proposed ETS–ANN
Accidental death	685.21	584.06	490.13	557.44	379.57	**389.98**
IBM	**7.0076**	7.5304	9.0840	11.949	7.6040	7.2500
Lake	0.5116	0.3638	0.3826	0.3555	**0.3544**	0.3666
Lynx	47.213	43.437	1096.2	880.43	**848.86**	882.41
Pollution	872.96	771.91	996.93	736.28	**692.45**	742.32
Stock	**6.9745**	8.6031	32.319	23.196	8.6183	7.1804
Sun spot	31.884	21.558	23.651	21.829	20.536	**19.541**
Colorado river	0.5047	**0.4749**	0.5803	0.5489	0.5176	0.5143
Passenger	52.429	37.377	33.404	26.506	26.623	**21.709**
Internet traffic data of UK	4030.9	3933.6	**1924.6**	2038.5	2184.1	2165.6
Temperature	5.1422	2.6929	**2.3515**	2.4496	2.5496	2.5496
Unemployment	51.286	54.272	45.160	48.788	43.712	**33.424**
Milk	26.778	23.267	14.218	11.736	7.7676	**6.4125**
Mumps	108.15	109.99	99.515	**78.360**	94.317	88.481
Chickenpox	171.39	233.75	147.88	159.42	161.12	**124.79**
Traffic	31.743	31.358	24.648	21.759	21.061	**20.369**

Table 4
Individual Results for the Hypothesis test using the RMSE. The results indicate which method is better, worse or equivalent to the proposed method using '+', '− ' and '≈' respectively.

	Hyndman and Khandakar (2008) ETS	Hyndman and Khandakar (2008) ARIMA	MLP	Babu and Reddy (2014)	Zhang (2003)
Accidental death	−	−	−	+	+
IBM	+	−	−	−	−
Lake	−	+	−	+	+
Lynx	−	+	−	≈	≈
Pollution	−	−	−	≈	+
Stock	≈	−	−	−	−
Sun Spot	−	−	−	−	−
Colorado river	+	+	−	−	≈
Passenger	−	−	−	−	−
Internet traffic data of UK	−	−	+	−	≈
Temperature	−	+	+	+	+
Unemployment	−	−	−	−	−
Milk	−	−	−	−	−
Mumps	−	−	−	+	−
Chickenpox	−	−	−	−	−
Traffic	−	−	−	−	−

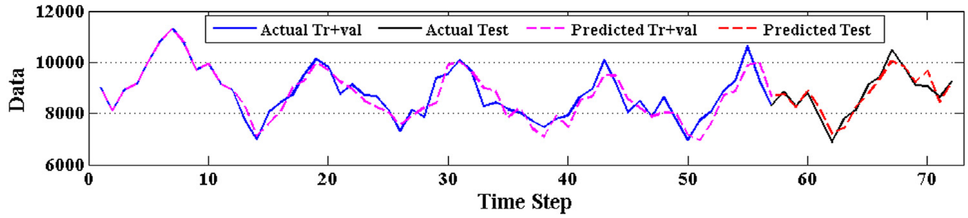

Fig. 3. Prediction of Accidental Death time series.

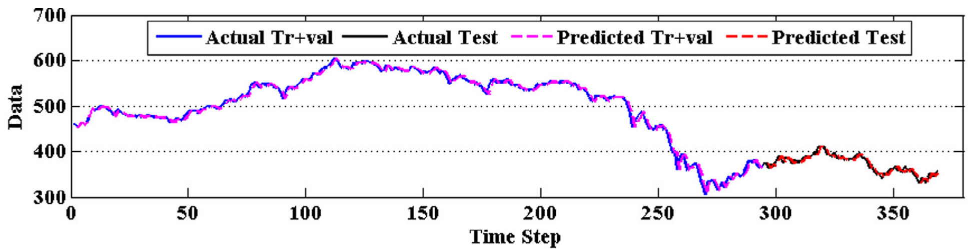

Fig. 4. Prediction of IBM time series.

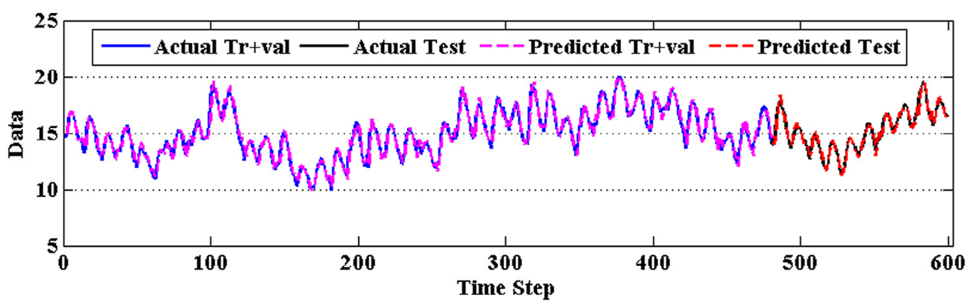

Fig. 5. Prediction of Lake time series.

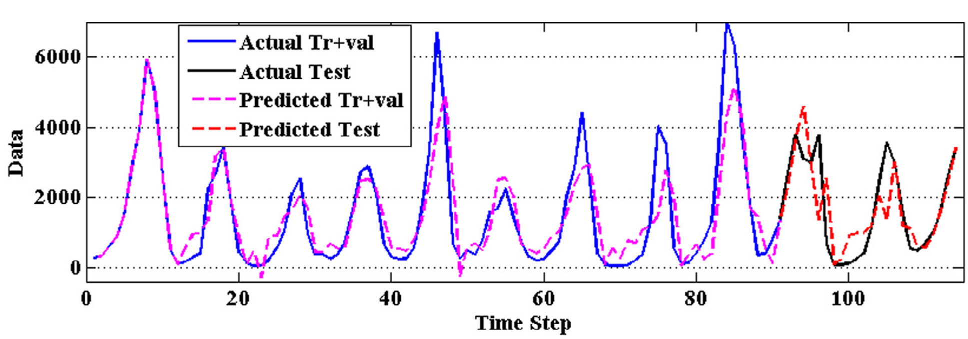

Fig. 6. Prediction of Lynx time series.

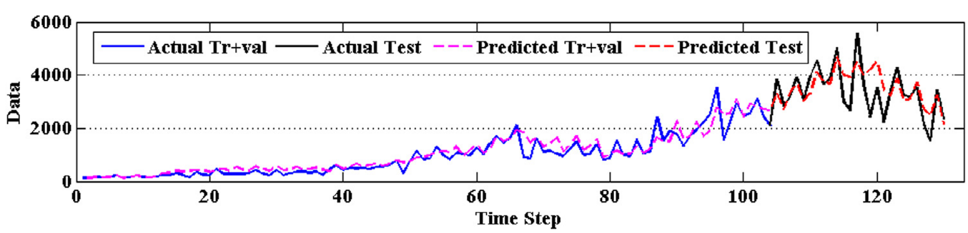

Fig. 7. Prediction of Pollution time series.

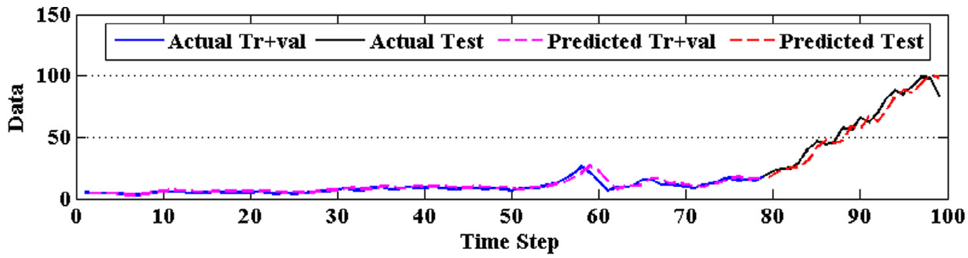

Fig. 8. Prediction of Stock time series.

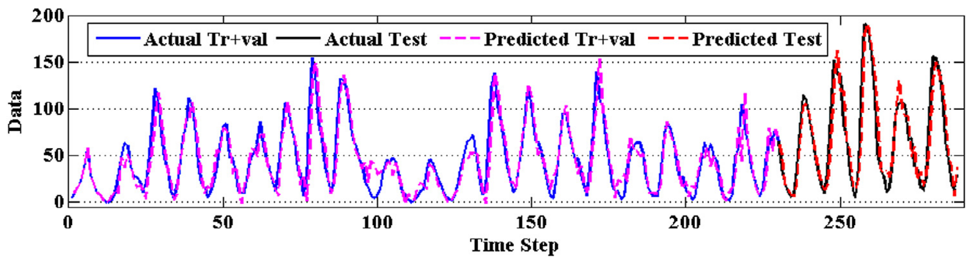

Fig. 9. Prediction of Sun Spot time series.

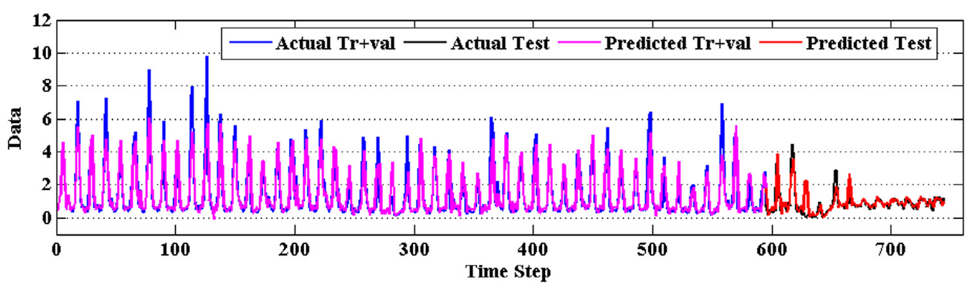

Fig. 10. Prediction of Colorado River time series.

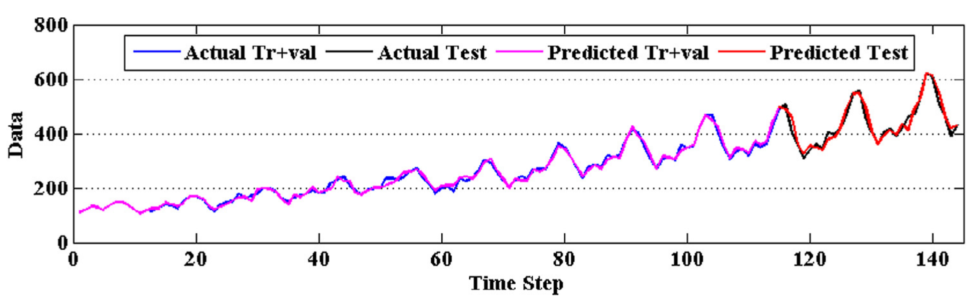

Fig. 11. Prediction of Passenger time series.

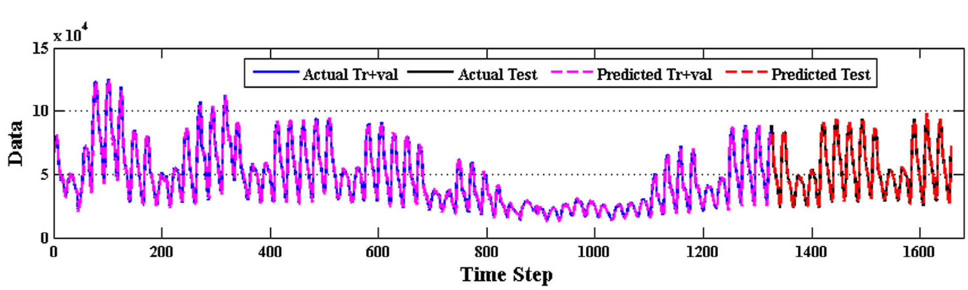

Fig. 12. Prediction of Internet traffic data of UK time series.

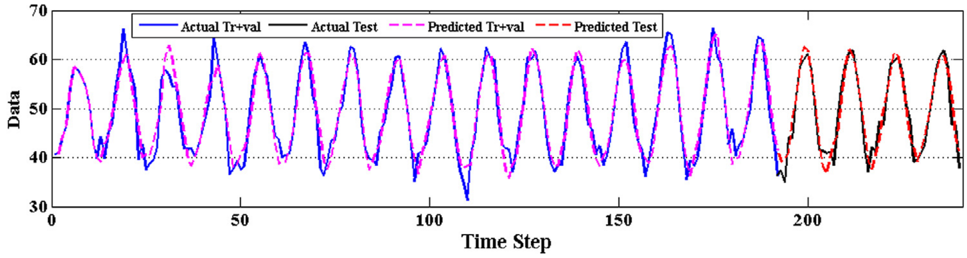

Fig. 13. Prediction of Temperature time series.

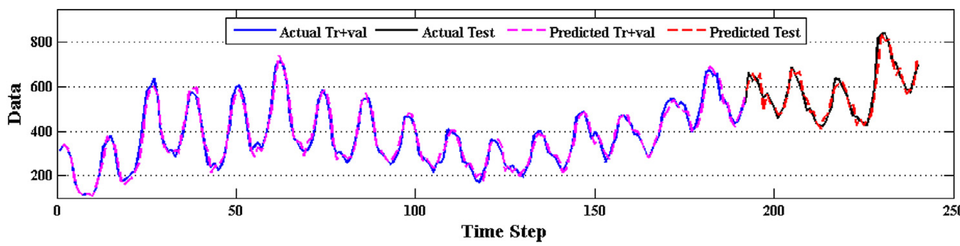

Fig. 14. Prediction of Unemployment time series.

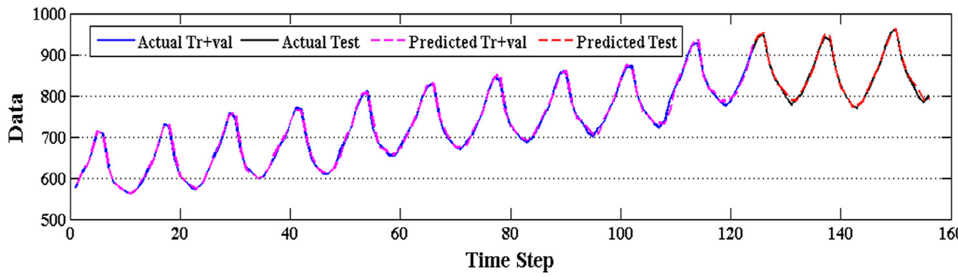

Fig. 15. Prediction of Milk time series.

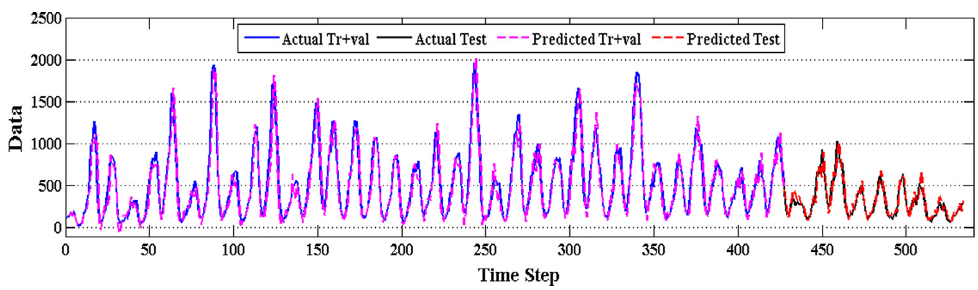

Fig. 16. Prediction of Mumps time series.

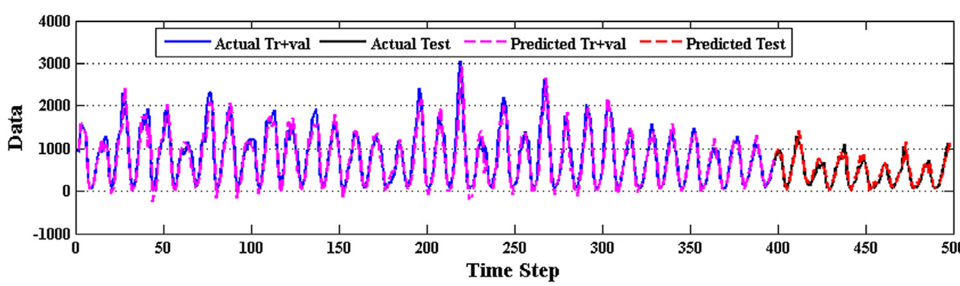

Fig. 17. Prediction of Chickenpox time series.

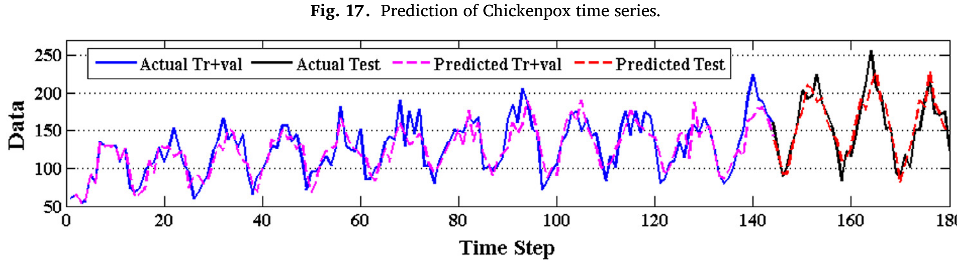

Fig. 18. Prediction of Traffic time series.

4.2. Analysis based on SMAPE

In this section, SMAPE measure is used to analyze the results. The SMAPE is a scale independent measure and can be used for model evaluation across multiple time series. Therefore comparisons are made individually on each time series and considering all the time series.

For evaluating the comparative performance of models on each individual time series, mean SMAPE over best 30 executions out of 50 executions were obtained and presented in Table 5. The proposed ETS–ANN model provided best SMAPE on seven datasets. For a careful analysis, a Wilcoxon signed-rank test with 95% of confidence level was applied and the obtained results are presented in Table 6. It can be clearly observed that the proposed model out-performed all models in seven datasets. Additionally, the proposed model achieved better SMAPE in Wilcoxon signed-rank test when compared to: ETS in 12 cases, ARIMA in 14 cases, MLP in 14 cases, Babu and Reddy (2014) in 11 cases and Zhang (2003) in 12 cases.

In order to evaluate the models across all datasets, Friedman and Nemenyi hypothesis test (Hollander et al., 1999) was conducted on the SMAPE measure. Friedman's test was used to rank the models for each dataset separately. If the ranks are significantly different i.e. the null hypothesis is rejected, a Nemenyi test was employed (Demsar, 2006) by making a pair wise comparison among all models. The results with respect to SMAPE are presented in Table 7. It can be clearly observed that the proposed model obtained the best rank among all models considered in this study.

Table 5
Symmetric mean absolute percentage error (SMAPE) for all datasets (best values in bold).

	Hyndman and Khandakar (2008) ETS	Hyndman and Khandakar (2008) ARIMA	MLP	Babu and Reddy (2014)	Zhang (2003)	Proposed ETS–ANN
Accidental death	7.0366	5.4168	4.6924	5.4511	3.6836	**3.4941**
IBM	**1.5427**	1.6486	1.9525	2.4890	1.6592	1.5955
Lake	2.8143	1.7767	1.8321	**1.7132**	1.7185	1.7607
Lynx	**48.781**	50.714	63.445	51.249	53.269	56.674
Pollution	21.329	**19.295**	25.445	18.318	**17.680**	18.537
Stock	**9.9226**	11.811	41.747	37.840	12.552	10.609
Sun Spot	49.039	37.141	35.005	37.141	35.537	**29.453**
Colorado River	**28.822**	37.023	39.228	43.850	36.758	36.059
Passenger	9.9966	6.8829	5.3276	4.7027	5.0594	**3.5187**
Internet traffic data of UK	4.9155	4.8438	**2.8952**	2.9430	3.1220	3.0218
Temperature	8.3699	4.8149	**4.1199**	4.2225	4.4404	4.4404
Unemployment	6.5909	7.3117	6.0006	6.2131	6.0949	**4.4097**
Milk	2.0752	1.9558	1.3403	1.0848	0.7258	**0.6175**
Mumps	29.112	31.333	23.912	**21.714**	24.494	22.449
Chickenpox	54.297	54.430	34.568	38.393	37.818	**28.391**
Traffic	18.138	17.669	13.036	11.630	11.406	**11.053**

Table 6
Individual Results for the Hypothesis test using the SMAPE. The results indicate which method is better, worse or equivalent to the proposed method using '+', '– ' and '≈' respectively.

	Hyndman and Khandakar (2008) ETS	Hyndman and Khandakar (2008) ARIMA	MLP	Babu and Reddy (2014)	Zhang (2003)
Accidental death	–	–	–	–	–
IBM	+	–	–	–	–
Lake	–	–	–	+	+
Lynx	+	+	–	+	+
Pollution	–	–	–	≈	+
Stock	+	–	–	–	–
Sun Spot	–	–	–	–	–
Colorado River	+	≈	–	–	≈
Passenger	–	–	–	–	–
Internet traffic data of UK	–	–	+	+	–
Temperature	–	–	+	–	–
Unemployment	–	–	–	–	–
Milk	–	–	–	–	–
Mumps	–	–	–	+	–
Chickenpox	–	–	–	–	–
Traffic	–	–	–	–	–

Table 7
Ranks of all methods using SMAPE.

Model	Rank SMAPE	Mean rank
Proposed ETS–ANN	54.2021	1
Zhang (2003)	74.2646	2
Babu and Reddy (2014)	93.7979	3
Hyndman and Khandakar (2008) ETS	100.125	4
Hyndman and Khandakar (2008) ARIMA	107.0625	5
MLP	113.5479	6

5. Conclusion

In this study, a new hybrid model for time series forecasting is proposed by combining linear and nonlinear exponential smoothing (ETS) models from innovation state space (ETS) with ANN. The hybrid model provides a better chance to capture different combination of linear and/or nonlinear relationship in the time series by using the linear and nonlinear modeling capability of both the models. The proposed ETS–ANN model assumes the time series to be a sum of two components which may be linear or nonlinear. First ETS is applied to the original time series to obtain the predictions for first component. Then the residual error sequence is obtained by subtracting the ETS-predictions from original series. The residual error sequence is used by ANN to obtain the predictions for second component. Finally, the final prediction is obtained by combining the ETS-predictions with ANN-predictions. Sixteen datasets and five different models from the literature were considered to evaluate the effectiveness of the proposed model. The results revealed that no model was best for all datasets. In few cases the individual models performed better than hybrid models. However, hybrid models out-performed individual models in most of the cases. The proposed methodology obtained the best overall forecast accuracy considering all the datasets analyzed. In addition, to have a robust evaluation of results Friedman and Nemenyi hypothesis tests were conducted. Results indicated the superiority of proposed model by achieving best rank among all the models. In order to improve the performance of the proposed model, one may use significant input variables selection for time series forecasting methods (Tran et al., 2015) instead of analyzing auto correlation function for lag selection of ANN models. Also, one may use deep learning methods (Lngkvist et al., 2014) for better selection and learning of features which will enhance the performance.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.engappai.2017.07.007.

References

Babu, C.N., Reddy, B.E., 2014. A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data. Appl. Soft Comput. 23, 27–38.

Behera, H.S., Dash, P.K., Biswal, B., 2010. Power quality time series data mining using S-transform and fuzzy expert system. Appl. Soft Comput. 10 (3), 945–955.

Box, G.E.P., Jenkins, G., 1990. Time Series Analysis, Forecasting and Control. Holden-Day Incorporated.

Chandra, D.R., Kumari, M.S., Sydulu, M., 2013. A detailed literature review on wind forecasting. In: International Conference on Power, Energy and Control, ICPEC, pp. 630-634.

Demsar, J., 2006. Statistical comparisons of classifiers over multiple datasets. J. Mach. Learn. Res. 7, 1–30.

de Oliveira, J.F., Ludermir, T.B., 2016. A hybrid evolutionary decomposition system for time series forecasting. Neurocomputing 180, 27–34.

Donkor, E., Mazzuchi, T., Soyer, R., Alan Roberson, J., 2014. Urban water demand forecasting: review of methods and models. J. Water Resour. Plann. Manage. 140, 146–159.

Enders, W., 1995. Applied Econometric Time Series, first ed. Wiley, New York.

Fagiani, M., Squartini, S., Gabrielli, L., Spinsante, S., Piazza, F., 2015. A review of datasets and load forecasting techniques for smart natural gas and water grids: analysis and experiments. Neurocomputing 170 (25), 448–465.

Faruk, D.O., 2010. A hybrid neural network and ARIMA model for water quality time series prediction. Eng. Appl. Artif. Intell. 23, 586–594.

2006. Gaussian maximum likelihood estimation for ARMA models. I. Time series. J. Time Ser. Anal. 27 (6), 857–875.

Hamilton, J.D., 1994. Time Series Analysis, first ed. Princeton University Press, Princeton.

Hollander, M., Wolfe, D.A., Chicken, E., 1999. Nonparametric Statistical Methods. John Wiley& Sons, Hoboken, NJ.

Hornik, K., Stinnchcombe, M., White, H., 1989. Multi-layer feed forward networks are universal approximators. Neural Netw. 2, 359–366.

Hyndman, R.J., TimeSeriesDataLibrary, 2010. URL http://data.is/TSDLdemo.

Hyndman, R.J., Khandakar, Y., 2008. Automatic time series forecasting: the forecast package for R. J. Statist. Softw. 27 (3), 1–22.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. Int. J. Forecast. 22 (4), 679–688.

Hyndman, R.J., Koehler, A.B., Ord, J.K., Snyder, R.D., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer-Verlag.

Hyndman, R.J., Koehler, A.B., Snyder, R.D., Grose, S., 2002. A state space framework for automatic forecasting using exponential smoothing methods. Int. J. Forecast. 18 (3), 439–454.

Jr Gardner, E.S., 1985. Exponential smoothing: the state of the art. J. Forecast. 4, 1–28.

Karali, Y., Panigrahi, S., Behera, H.S., 2013. a novel differential evolution based algorithm for higher order neural network training. J. Theor. Appl. Inf. Technol. 56 (3), 355–361.

Khandelwal, I., Adhikari, R., Verma, G., 2015. Time series forecasting using hybrid ARIMA and ANN models based on DWT decbomposition. Procedia Comput. Sci. 48, 173–179.

Khashei, M., Bijari, M., 2011. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Appl. Soft Comput. 11 (2), 2664–2675.

Lin, W.Y., Hu, Y.H., Tsai, C.F., 2012. Machine learning in financial crisis prediction: a survey. IEEE Trans. Syst. Man Cybern. C 42, 421–436.

Lngkvist, M., Karlsson, L., Loutfi, A., 2014. A review of unsupervised feature learning and deep learning for time series modeling. Pattern Recognit. Lett. 42 (0), 11–24.

Makridakis, S., Wheelwright, S., Hyndman, R., 1998. Forcasting: Methods and Applications, third ed. Wiley, Hoboken, NJ.

Maqsood, I., Khan, M.R., Abraham, A., 2004. An ensemble of neural networks for weather forecasting. Neural Comput. Appl. 13 (2), 112–122.

Meade, N., Islam, T., 2015. Forecasting in telecommunications and ICT-A review. Int. J. Forecast. 31 (4), 1105–1126.

Panigrahi, S., 2017. A novel hybrid chemical reaction optimization algorithm with adaptive differential evolution mutation strategies for higher order neural network Training. Int. Arab J. Inf. Technol. 14 (1), 18–25.

Pegels, C.C., 1969. Exponential forecasting: some new variations. Manage. Sci. 15 (5), 311–315.

Raza, M.Q., Khosravi, A., 2015. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renewable Sustainable Energy Rev. 50, 1352–1372.

Reyes, J., Morales-Esteban, A., Martnez-lvarez, F., 2013. Neural networks to predict earthquakes in Chile. Appl. Soft Comput. 13 (2), 1314–1328.

Sahu, K.K., Panigrahi, S., Behera, H.S., 2013. A novel chemical reaction optimization algorithm for higher order neural network training. J. Theor. Appl. Inf. Technol. 53 (3), 402–409.

Taylor, J.W., 2003. Exponential smoothing with a damped multiplicative trend. Int. J. Forecast. 19, 715–725.

Tran, H.D., Muttil, N., Perera, B.J.C., 2015. Selection of significant input variables for time series forecasting. Environ. Model. Softw. 64, 156–163.

Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future. Int. J. Forecast. 30, 1030–1081.

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

Zhang, G.P., Patuwo, B.E., Hu, M.Y., 1988. Forecasting with artificial neural networks: The state of art. Int. J. Forecast. 14, 35–62.