

# A Review on the Comparison of Box-Jenkins ARIMA and LSTM of Deep Learning

Stavelin Abhinandithe K\*, Madhu B<sup>1</sup>, Balasubramanian S<sup>2</sup>, Sahana C<sup>3</sup>

\*Assistant Professor, Division of Medical Statistics, Faculty of Life Sciences,

<sup>1</sup>Associate Professor, Department of Community Medicine JSS Medical College,

<sup>2</sup>Dean & Director of Research, Department of Water & Health,

<sup>3</sup>Division of Medical Statistics, Faculty of Life Sciences,

\*, 1, 2, 3 JSS AHER, Mysore, Karnataka, India

## ABSTRACT

A time-series is a set of events, sequentially calculated over time. Predicting the Time Series is mostly about predicting the future. The ability of a time series forecasting model is determined by its success in predicting the future. This is often at the expense of being able to explain why a specific prediction was made. The Box-Jenkins model implies that the time series is stationary, and thus suggests differentiating non-stationary series once or several times to obtain stationary effects. This generates a model for ARIMA, the "I" being the word for "Integrated". The LSTM networks, comparable to computer memory, enforce a gated cell for storing information. Such as the previously mentioned networks, the LSTM cells also recognize when to make preceding time-steps reads and writes information. Even though the work is new, it is obvious that LSTM architectures provide tremendous prospects as contenders for modeling and forecasting time series. The outcomes of the overall discrepancy in error indicate that in regards of both RMSE and MAE, the LSTM-model tended to have greater predictive accuracy than the ARIMA-model.

**KEYWORDS:** ARIMA, LSTM, Time series model, RMSE, MAE

**How to cite this paper:** Stavelin Abhinandithe K | Madhu B | Balasubramanian S | Sahana C "A Review on the Comparison of Box-Jenkins ARIMA and LSTM of Deep Learning" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-3, April 2021, pp.409-414, URL: www.ijtsrd.com/papers/ijtsrd39831.pdf



IJTSRD39831

Copyright © 2021 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



## I. INTRODUCTION

A time-series is a set of events, sequentially calculated over time. Such findings either are continuous over time, or recorded at discrete intervals of equal time.<sup>[1]</sup> Exploration of the time series is a rather useful field of data analysis, retrieving information from past findings to establish the progression of a current phenomenon and to promote its prediction into the long term. The study also illustrates definite associations that might seem obvious to the naked eye and describes certain characteristics required to recognize this phenomenon's current state of play. Continuous time series, such as brain activity assessment, are generally evaluated by measuring the series at intervals of equal time to offer a discrete time series. Observations in the series frequently correspond in various ways at different time-stages. The potential consequences can be estimated using observed data while succeeding observations are dependent. Analysis of the time series is the field of study where these associations and correlations are analyzed. The correlation might be the order of the data gathered, the linearity of the model, reiterating trends etc.<sup>[2]</sup> The research community had also raised a handful of attempts to enhance time series exploration and analysis.<sup>[3]</sup>

## A. Prediction in time series:

Predicting the Time Series is mostly about predicting the future. Although many data from the time series relies on its previous values. Near history attributes are strong measures of conduct in a factor. Lagged values of a variable as in a rate of exchange are regressed over one or more lagged values of themselves to determine the present and potential values of the variable. Forecasting in the time series is reliant on numerous models and methods, and is commonly included to predict different aspects of human behavior <sup>[4-6]</sup>. While forecasting time series, one primarily uses sequence preceding values to estimate an upcoming value. It's convenient to map the association of the y variable with the preceding y vector values because of the usage of past values. The intention of time series analysis is to research the time series path patterns and to construct a model to explain the data structure and then forecasting potential values of time series.<sup>[7]</sup>

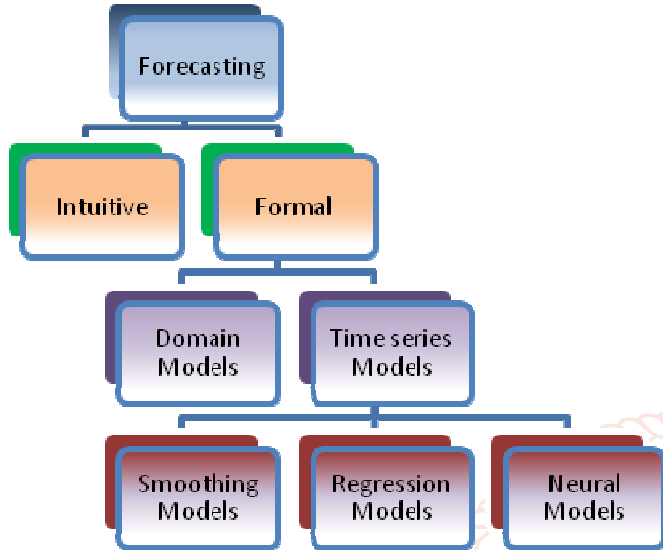
## B. Forecasting in time series:

Time series estimation includes utilizing time series observations, in accordance with the time series analysis, to build a predictive model illustrating the dependencies. This

model is then used in series estimation of future values. The forecast uses only past knowledge to estimate future values.<sup>[4]</sup> The ability of a time series forecasting model is determined by its success in predicting the future. This is often at the expense of being able to explain why a specific prediction was made.

**II. Forecasting methods in time series analysis.**

Methods of forecasting appear to be divided into two main categories. We are: intuitive and formalized.



**Fig.1. Classification of forecasting methods**

**Intuitive forecasting** Methods comprise forecasts and expert inferences. They are also used nowadays in market research, economics, politics and other fields, the action of which is very complicated or harder to predict with mathematical models<sup>[8]</sup>

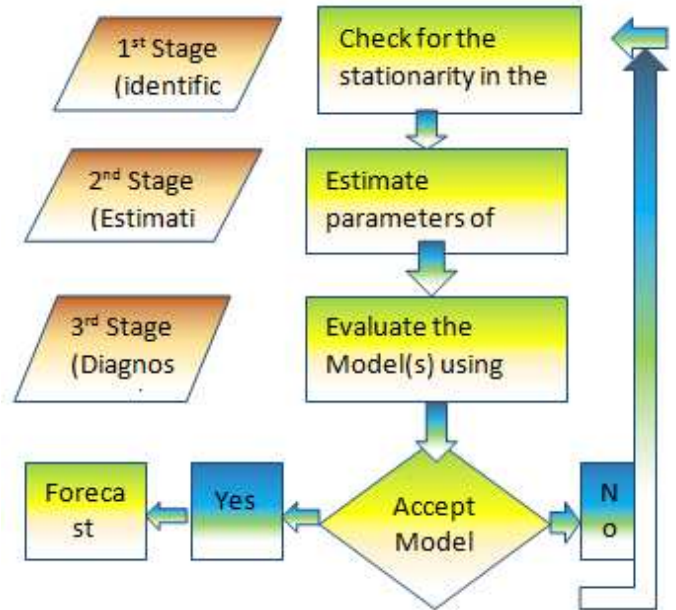
**Formalized methods** are the methods, which use mathematical models to predict future values. They are divided into domain models and time-series models.

**Domain models**–Models, focused on domain processes, laws, and mechanisms. Climate prediction model for instance includes an equation of fluid dynamics and thermodynamics. The most common solutions to software failures include the software reliability models. Their key disadvantage: they don't suit all different technology classes, since they focus on unique attributes of it<sup>[9]</sup>. A thorough comprehension of the processes, methodologies and technologies of software design and testing is needed to construct an appropriate model of software reliability and to be qualified to draw conclusions dependent on such a model.

**Time-series** models are mathematical forecasting models that aim to determine, throughout the framework itself, the reliance of the future value from the past value as well as compute the estimation relying on this dependence. Such models are identical for various contexts, i.e. their overall appearance does not alter according to the design of the time series<sup>[10]</sup>. Time series models<sup>[11]</sup> can be further divided into:

- Regression models;
- Smoothing models;
- Models based on neural networks.

**III. Model description: Basics of Box and Jenkins Time Series Models**



The Box-Jenkins Time Series models are titled after the statisticians George Box and Gwilym Jenkins (Box and Jenkins, 1970), these models generate projected estimates depending on the statistical criteria of the time series data obtained and these models have achieved considerable interest in the fields of organizational analysis, management science and statistics. The Box-Jenkins ARMA (Autoregressive Moving Average) model is a combination of the AR (Auto Regressive) and MA (Moving Average) models. The Box-Jenkins model implies that the time series is stationary, and thus suggests differentiating non-stationary series once or several times to obtain stationary effects. This generates a model for ARIMA, the "I" being the word for "Integrated". They are also known as the Autoregressive Integrated Moving Average (ARIMA) models (Box and Jenkins, 1976).

There are many ways to forecast using the time-series models, and the most commonly used of which is as follows

- Autoregressive Models (AR).
- Moving Average Models (MA).
- Mixed Models (ARMA).
- Integrated Mixed Models (ARIMA).

**Autoregressive Integrated Moving Averages Models:**

In 1976, Box and Jenkins first published ARIMA in a book which gained considerable coverage from the science establishment, based on statistical analysis at that period. Thus, this approach is implemented in a vast range of sectors and stays one of the best reliable data processing and operational prediction models<sup>[13]</sup>. ARIMA stands for Autoregressive (AR) Integrated (I) Moving Average (MA), otherwise regarded as the Box- Jenkins approach.<sup>13</sup> As indicated by the terminology, ARIMA (p, d, q) incorporates main components of the model:

AR: Auto regression. A regression model using the relationships between an observation and a number of observed lags (p).

I: Integrated. To render the time series stationary by calculating the observed variations at various times (d).

MA: Moving Average. A methodology that bears into consideration the dependence among the observations and

the terms of residual error while using a moving average model for lagging observations (q).

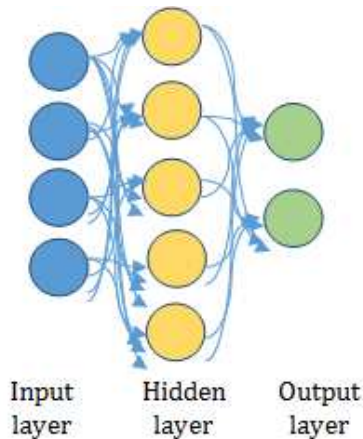
The Box-Jenkins method<sup>[14]</sup> summarizes the ARIMA process in three main steps:

**Identification:** the initial move is to split through the time series by the three processes: AR (autoregressive), I (integrated) and MA (moving average); this phase apparently allows the parameters p, d and q to be defined, whilst testing the series' stationarity first. Configuration of parameters p, q is accomplished owing to the functions of autocorrelation and partial autocorrelation that we would address in depth in the section of realization. The parameter d is differentiation order.

**Estimation:** The following phase in the Box-Jenkins method is to determine the parameters of the respective models by supplying the p d and q orders. Estimates are made using non-linear methodologies.

**Diagnosis:** the final stage in the Box-Jenkins process relates to checking the model's validity. That is, validate that the predicted model is suited to the accessible data. We refer to statistical testing for that.

## LSTM



## Neural Networks

**ANN:** ANNs are partially influenced by our brain's neural interactions, attempting to imitate the neuronal circuits and its behavioral patterns. ANNs, being resilient and self-adaptive, typically offer satisfactory remedies to implicitly inferred non-linear challenges, and solve challenges such as speech recognition, natural language processing, and forecasting<sup>20</sup>. The framework includes so-called layers across input and output, in which raising the amount of layers often raises network complexities. By loading identified data into the network, the network is educated by weight allocation in reference to the preferred input output. The weights are network-adjusted scalars to which the error within the output, which is inferred by the network, and the true expected output utilizing the weight gradient of the error. Nevertheless, those kinds of networks are not ideal for sequential results. Since the network seems to have no memory of earlier stages, it cannot model dependencies, leaving it difficult to interpret sequential data. Therefore a memory of some sort is desired.

**RNN:** RNNs, recurrent neural networks, string numerous layers of networks together, whereby, in connection to the output, information from previous time stages is passed to future time stages. As the inputs and parameters are analyzed from each layer, the output from previous layers is taken into account, providing the network a type of memory

<sup>[15]</sup>. The gradient can become fairly complicated when a time-dimension needs to be drawn into account for RNNs, culminating in the knowledge from previous phases in time either begin to disappear or become greatly enhanced. Such phenomenon are also known as a vanishing or bursting gradient.

**LSTM:** The LSTM networks, comparable to computer memory, enforce a gated cell for storing information. Such as the previously mentioned networks, the LSTM cells also recognize when to make preceding time-steps reads and writes information<sup>[15, 17]</sup>. Therefore the LSTM model addresses the issue of a disappearing or exploding gradient and helps the network to accurately recognize data far back in the sequence<sup>[18, 16]</sup>.

LSTM model architecture is quite complicated to present. The Long Short-Term Memory (LSTM) network is basically the most commonly implemented framework to resolve the gradient disappearance crisis. SeppHochreiter and JrgenSchmidhuber suggested this same network structure in 1997<sup>[19]</sup>. With the emergence of deep architectures, LSTM has been extensively seen in sequence learning currently, and has shown considerable experimental strength<sup>[20]</sup>. Each LSTM is a collection of cells, or device modules, which collect and preserve the data streams. The cells are like a transport line (the upper line in each cell) that links data from the past and gathers them for the current from one module to another. Owing to the utilization of certain gates in each cell, data may be discarded of, filtrated, or incorporated for the following cells in each cell. Thus, the gates that are focused on a sigmoidal neural network layer allow the cells to allow data to transit through or be disposed of optionally. Each sigmoid layer generates numbers in the vicinity of zero and one, reflecting - cell's quantity of each data segment should be allowed through. Most specifically, an estimate of zero value means that "let nothing pass through;" and whilst; an estimate of one suggests that "let anything pass through." Each LSTM includes three styles of gates mostly with purpose of regulating the condition of each cell.<sup>[21]</sup>

- Forget Gate outputs a number between 0 and 1, where 1 shows "completely keep this"; whereas, 0 implies "completely ignore this."
- Memory Gate determines which new data to be retained in the cell. Initially, a sigmoid layer, called the "door input layer," decides which values are to be altered. A tanh layer then allows a vector of fresh agent values which can be attached to the system.
- Output Gate determines what yields every cell can yield. The rendered value, together with the processed and freshly inserted data, will be based on cell state.

This architecture is prone to numerous modifications, however, it has nearly as many changes as the articles which use it. A condensed version of LSTM termed GRU (Gated Recurrent Unit) was implemented in 2014<sup>[22]</sup>. This has the benefit of having fewer computationally complex, since it has smaller parameters and equations. It remains however as powerful as its predecessor. Here are the equations which govern the model's dynamism:

- $Z_t = \sigma(WZx_t + UZht_{-1} + bZ)$  (update gate)
- $R_t = \sigma(WRx_t + URht_{-1} + bR)$  (reset gate)
- $ht = Z_t \circ ht_{-1} + (1 - Z_t) \circ \tanh(Whx_t + Uh(R_t \circ ht_{-1}) + bh)$ .

Last year's curiosity in utilizing LSTM has unexpectedly increased and its predictive accuracy can illustrate its strength in nearly every area of science. It took about two



decades for the strategy to become mainstream. To construct an LSTM model, initially we have to convert the Dataset training and test into a three-dimensional collection of "samples, features, and time measures". We utilize a layer of inputs, a hidden layer of LSTM blocks as well as a single layer of output. The accompanying principles are employed to evaluate the amount of layers and the number of neurons in each layer:

- Input layer: Logically, we possess one entry; the number of neurons in that layer is calculated by the number of columns that can be quantified.
- Output layer: each neural network has a single output.
- Hidden layer: the size of this layer, that is to say the number of neurons, is to determine.

**GAN:** Generative adversarial networks, or GANs for shorter, are a strategy to generative modeling incorporating techniques of deep learning, like neural convolution networks. GANs are an innovative means to teach a generative model by presenting the issue using two sub-models as a supervised learning problem: the generator model, and the discriminator model. There is teaching of the two models together.

Naive Bayes is an illustration of a more commonly used generative model as a discriminative model. Naive Bayes functions by summing up the distribution of probability for each input variable and the output class. When a prediction is made, the likelihood for each potential outcome for each variable is determined, the individual probabilities are merged, and the highly realistic consequence is estimated.

Other examples of generative models include Latent Dirichlet Allocation, or LDA, and the Gaussian Mixture Model, or GMM.

**RBM:** A Restricted Boltzmann Machine is a generative stochastic artificial neural network which is able to learn a distribution of probability over its set of inputs. RBMs were first discovered in 1986 by Paul Smolensky under the name Harmonium. RBMs are a variety of Boltzmann devices, as their name suggests, with the constraint that its neurons should build a bipartisan graph. This limitation enables for increasingly effective training algorithms than those accessible for the general group of Boltzmann machines, in specific the contrastive divergence algorithm based on gradients.

**DBN:** Deep Belief Network is an unsupervised probabilistic deep learning algorithm, made up of multilayered latent stochastic variables. DBN is a graphical model of the generative hybrid. Top two stratification are undirected. Two layers lower have guided interactions from above layers. DBN are pre trained using Greedy Algorithm. By adequately instantiating the weights of all the layers, pre-training improves optimization. DBNs are used in image recognition, video sequences, motion capture data and speech recognition.

#### **FORECAST PERFORMANCE METRICS (Common Accuracy Statistics)**

**RMSE:** The RMSE is the square root of residual variance. This shows the model's absolute fit for the data – how identical the measured data points are to the expected values of the model. RMSE is an absolute fit-measure. It can be assumed, focusing on a thumb rule, that the RMSE values between 0.2 and 0.5 indicate that the model was able to predict the relevant data reasonably. Adjusted R-squared upwards of 0.75 is, however, a very better value for the accuracy display.

For certain cases it is also appropriate to have Modified R-squared of 0.4 or more.

**MSE:** The mean squared error, or MSE, is measured as the square predicted error values average. The squaring of forecast error values causes them to be positive; it also puts greater weight on major errors. A greater MSE implies that the data values are massively scattered around its central moment (mean), and a lesser MSE indicates otherwise and is certainly the preferable and/or desired option because it indicates that the data values are distributed near to its central moment, which is normally large.

**MAPE:** The mean absolute percentage error (MAPE) is the mean or average of forecast errors with an absolute percentage. Error is characterized as real or detected, minus the expected value. Percentage errors are summed up lacking regard to MAPE calculation signing. This calculation is easy to grasp, because it renders the mistake in percentage terms. Furthermore, since absolute percentage errors are considered, the issue of equally dismissing positive and negative errors is prevented. MAPE therefore has a strategic reach and is a metric which is widely included in forecasting. The smaller the MAPE the better the forecast.

#### **RELATED WORKS:**

A comparison analysis was carried out between LSTM and ARIMA in AjilaElmasdotter's forecasting sale retail<sup>[2]</sup>. Furthermore, ARIMA time series is a commonly used methodology for the budgetary time series<sup>23</sup> in econometrics, numerous ARIMA models have been formulated to evaluate and project stock markets<sup>[24-26]</sup>. In addition, ARIMA is used for water expenditure / forecasting of consumption<sup>[27,28]</sup> and the requirement for electricity<sup>[29]</sup>. Most notably, there has been a great deal of concern in the usage of deep learning models<sup>[30,31]</sup>, particularly recursive models like LSTM Neural Network for the estimation of budgetary time series, particularly in the stock market<sup>[32-34]</sup>. In<sup>[32]</sup> presented a simulation and simulation of China stock returns employing LSTM architecture via an accepted 27.2 percent accuracy, in<sup>[33]</sup> studied the usefulness of recurrent neural networks to the assessment of stock market price fluctuations. Shanghai Composite Index and Dow Jones Index were eventually proposed<sup>[34]</sup>. With regard to budget analysis and forecasting, hardly any research has been reported trying to compare or even implementing the two techniques, the majority of which have been adapted to the stock price as discussed above. It is due largely to the complexity of accessing specific data sets and the unreliable nature of such data. None of its authors correlated LSTM and ARIMA models to efficiency.

#### **MAJOR DIFFERENCES BETWEEN ARIMA AND LSTM:**

Even though the work is new, it is obvious that LSTM architectures provide tremendous prospects as contenders for modeling and forecasting time series. We review the major variations between an LSTM and ARIMA in the following Table. The usage of RNNs, especially the LSTM architecture, allows us to set many parameters that we need to change in order to achieve optimum efficiency in the forecasting tasks. The problem lies in selecting the appropriate parameters to identify the correct architecture for the platform. An ARIMA model is easy to customize as it provides a reasonable efficiency, such model also includes the determination of the parameters p, d and q such that p is the order of the autoregressive part (AR), the order of differentiation and q the order of the average moving portion (MA).

ARIMA	LSTM
<ul style="list-style-type: none"> <li>➤ ARIMA is a linear model</li> <li>➤ Small amount of data is also sufficient for ARIMA (i.e., minimum number of observations is 50-100).</li> <li>➤ ARIMA is a parametric model, which is to say for each series we have to define the parameters p, d and q.</li> <li>➤ ARIMA is dedicated only for time series analysis</li> </ul>	<ul style="list-style-type: none"> <li>➤ LSTM is a nonlinear model.</li> <li>➤ LSTM can store and process large amount of data.</li> <li>➤ As LSTM is a non-parametric model, requires adjustment of some hyperparameters.</li> <li>➤ LSTM models can process sequential data as well.</li> </ul>

**Evaluation measures of real world time series data:**

		MAE	RMSE	MAPE	SMAPE
1	Formula	$MAE = \frac{1}{n} \sum_{i=1}^n  y_i - \hat{y}_i $	$RMSE = \sqrt{\frac{\sum_{i=1}^n (f_i - o_i)^2}{n}}$	$MAPE = \frac{1}{n} \sum_{i=1}^n \left  \frac{A_t - F_t}{A_t} \right $	$\frac{100\%}{n} \sum_{i=1}^n \frac{ F_t - A_t }{( A_t  +  F_t ) / 2}$
2	Acceptable range	The lower the MSE, the higher the accuracy of the prediction.	Lower values of RMSE indicate better fit.	Since, MAPE is a measure of error, high number are bad and low numbers are good	Lower the value, higher the accuracy.

**Comparison of Software:**

ARIMA	LSTM
<ul style="list-style-type: none"> <li>➤ EViews</li> <li>➤ Julia.</li> <li>➤ Mathematica</li> <li>➤ MATLAB</li> <li>➤ NCSS</li> <li>➤ Python</li> <li>➤ R</li> <li>➤ Ruby</li> <li>➤ SAS</li> <li>➤ IBM SPSS</li> <li>➤ Stata</li> </ul>	<ul style="list-style-type: none"> <li>➤ Python</li> <li>➤ Rnn</li> <li>➤ Tensor Flow</li> <li>➤ IBM SPSS</li> <li>➤ MATLAB</li> <li>➤ SAS</li> <li>➤ Julia</li> <li>➤ Ruby</li> </ul>

**CONCLUSIONS:**

These methodologies are acquiring prominence amongst investigators across diverse disciplines with the latest enhancement in the development of advanced machine learning-based approaches and especially deep learning algorithms. The major concern then is how specifically and effectively these newly developed techniques compete with conventional methods. The outcomes of the overall discrepancy in error indicate that in regards of both RMSE and MAE, the LSTM-model tended to have greater predictive accuracy than the ARIMA-model. All in all, we have shown that on lengthy-term predictions, neural networks provide more precision. Since the error for the ARIMA model presented by Box Jenkins is not significantly higher, we note that its use remains an advantage in the prediction of time series. We also saw the two models enjoy complimentary differences: ARIMA is far pickier on seasonal peaks while LSTM suits the balanced portion of the series better. For some developed countries dysfunctional economic system makes reliable prediction models complicated. The probability of having a better estimate (without sacrificing the complexity of the data through diff and linearization) emerges with the aid of new approaches in RNN.

**References:**

[1] Chatfield C. Time-series forecasting. CRC press; 2000 Oct 25.

[2] Elmasdotter A, Nyströmer C. A comparative study between LSTM and ARIMA for sales forecasting in retail.

[3] Fathi O. Time series forecasting using a hybrid ARIMA and LSTM model. In Velvet Consulting 2019.

[4] Zhang GP. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 2003 Jan 1; 50: 159-75.

[5] Zhang GP. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing. 2003 Jan 1; 50: 159-75.

[6] Chen L, Wang S, Fan W, Sun J, Naoi S. Beyond human recognition: A CNN-based framework for handwritten character recognition. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) 2015 Nov 3 (pp. 695-699). IEEE.

[7] Siami-Namini S, Tavakoli N, Namin AS. A comparison of ARIMA and LSTM in forecasting time series. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) 2018 Dec 17 (pp. 1394-1401). IEEE.

[8] A. M. Alonso, C. Garcia-Martos. Time Series Analysis - Forecasting with ARIMA models. In 2012 Universidad Carlos III de Madrid, Universidad Politecnica de Madrid.

[9] Earnest A, Chen MI, Ng D, Sin LY. Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore. BMC Health Services Research. 2005 Dec 1; 5(1): 36.

[10] Tang R, Zeng F, Chen Z, Wang JS, Huang CM, Wu Z. The Comparison of Predicting Storm-time Ionospheric TEC by Three Methods: ARIMA, LSTM, and Seq2Seq. Atmosphere. 2020 Apr; 11(4): 316.

[11] Khashei M, Bijari M. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. Applied Soft Computing. 2011 Mar 1; 11(2): 2664-75.

[12] Liu LM, Hudak GB, Box GE, Muller ME, Tiao GC. Forecasting and time series analysis using the SCA statistical system. DeKalb, IL: Scientific Computing Associates; 1992.

- [13] K. Lawrence and M. Geurts. Advances in Business and Management Forecasting. In 2004 vol. 19. Emerald Group Publishing.
- [14] Rhanoui M, Yousfi S, Mikram M, Merizak H. Forecasting financial budget time series: ARIMA random walk vs LSTM neural network. IAES International Journal of Artificial Intelligence. 2019 Dec 1; 8(4).
- [15] Rojas R. Neural networks: a systematic introduction. Springer Science & Business Media; 2013 Jun 29.
- [16] Li X, Peng L, Yao X, Cui S, Hu Y, You C, Chi T. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. Environmental pollution. 2017 Dec 1; 231: 997-1004.
- [17] Cortez B, Carrera B, Kim YJ, Jung JY. An architecture for emergency event prediction using LSTM recurrent neural networks. Expert Systems with Applications. 2018 May 1; 97: 315-24.
- [18] Krstanovic S, Paulheim H. Ensembles of recurrent neural networks for robust time series forecasting. In International Conference on Innovative Techniques and Applications of Artificial Intelligence 2017 Dec 12 (pp. 34-46). Springer, Cham.
- [19] Jia Y, Wu Z, Xu Y, Ke D, Su K. Long Short-Term Memory Projection Recurrent Neural Network Architectures for Piano's Continuous Note Recognition. Journal of Robotics. 2017 Jan 1; 2017.
- [20] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv: 1412. 3555. 2014 Dec 11.
- [21] Schmidhuber J. Deep learning in neural networks: An overview. Neural networks. 2015 Jan 1; 61: 85-117.
- [22] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15; 9(8): 1735-80.
- [23] Mills TC, Markellos RN. The econometric modelling of financial time series. Cambridge University Press; 2008 Mar 20.
- [24] Pai PF, Lin CS. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega. 2005 Dec 1; 33(6): 497-505.
- [25] Jarrett JE, Kyper E. ARIMA modeling with intervention to forecast and analyze Chinese stock prices. International Journal of Engineering Business Management. 2011 Aug 15; 3(3): 53-8.
- [26] Ariyo AA, Adewumi AO, Ayo CK. Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation 2014 Mar 26 (pp. 106-112). IEEE.
- [27] Birylo M, Rzepecka Z, Kuczynska-Siehien J, Nastula J. Analysis of water budget prediction accuracy using ARIMA models. Water Science and Technology: Water Supply. 2018 Jun; 18(3): 819-30.
- [28] Razali SN, Rusiman MS, Zawawi NI, Arbin N. Forecasting of water consumptions expenditure using Holt-Winter's and ARIMA.
- [29] Kandananond K. Electricity demand forecasting in buildings based on ARIMA and ARX models. In Proceedings of the 8th International Conference on Informatics, Environment, Energy and Applications 2019 Mar 16 (pp. 268-271).
- [30] Chong E, Han C, Park FC. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications. 2017 Oct 15; 83: 187-205.
- [31] Singh R, Srivastava S. Stock prediction using deep learning. Multimedia Tools and Applications. 2017 Sep 1; 76(18): 18569-84.
- [32] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: A case study of China stock market. In 2015 IEEE international conference on big data (big data) 2015 Oct 29 (pp. 2823-2824). IEEE.
- [33] Nelson DM, Pereira AC, de Oliveira RA. Stock market's price movement prediction with LSTM neural networks. In 2017 International joint conference on neural networks (IJCNN) 2017 May 14 (pp. 1419-1426). IEEE.
- [34] Jiang Q, Tang C, Chen C, Wang X, Huang Q. Stock price forecast based on LSTM neural network. In International Conference on Management Science and Engineering Management 2018 Aug 1 (pp. 393-408). Springer, Cham.