Comparison of Time Series Forecasting Based on Statistical ARIMA Model and LSTM with Attention Mechanism

Comparison of Time Series Forecasting Based on Statistical ARIMA Model and LSTM with Attention Mechanism

Kun Zhou[1,2*], Wen Yong Wang[1], Teng Hu[1,2] and Chen Huang Wu[1,3]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, Sichuan Province, P.R. China
[2] Institute of Computer Application, China Academy of Engineering Physics, Mianyang 621000, Sichuan, P.R. China
[3] School of Mathematics and Finance, Putian University, 351100, Fujian Province, P.R. China
Email: zhoukun@std.uestc.edu.cn

Abstract. Time series Forecasting (TSF) has been a research hotspot and widely applied in many areas such as financial, bioinformatics, social sciences and engineering. This article aimed at comparing the forecasting performances using the traditional Auto-Regressive Integrated Moving Average (ARIMA) model with the deep neural network model of Long Short Term Memory (LSTM) with attention mechanism which achieved great success in sequence modelling. We first briefly introduced the basics of ARIMA and LSTM with attention models, summarized the general steps of constructing the ARIMA model for the TSF task. We obtained the dataset from Kaggle competition web traffic and modelled them as TSF problem. Then the LSTM with attention mechanism model was proposed to the TSF. Finally forecasting performance comparisons were conducted using the same dataset under different evaluation metrics. Both models achieved comparable results with the up-to-date methods and LSTM slightly outperformed the classical counterpart in TSF task.

1. Introduction

Forecasting facilitates effective and efficient resources planning for future demands. TSF is an important area of machine learning for many prediction problems involving the time component. Time series is defined as a sequence of observations taken sequentially [1]. TSF adds time dimension which is different from other prediction problems in that the timely order must be preserved. One of the time series research branch is Time Series Analysis (TSA, also known as descriptive models) aims at understanding the pattern which contributes to good predictions through decomposition of time series into level (L) and three optional parts, namely, trend (T), seasonality (S), and noise (N). The naïve model may add these components up, which is y = L + S + T + N. Assumptions about these components avail the modelling using traditional statistical methods. Although these four decompositions may not be universal for forecasting problems, they have values in concepts and could be used as input for more complex models. For TSF tasks issues of the available dataset, time horizons of forecasting, the updatability of the forecasting model, and the forecasting frequency should be considered. Time series data requires pre-processing such as cleaning, normalization, scaling, and transformation to deal with outlier or missing data.

The ARIMA models comprehensively consider Auto-Regressive (AR), Moving Average (MA) process, and difference processing, and they are the most general class of models for forecasting a time

series [2]. Generally, the sequence has non-stationary state (the residual of the model should be as stochastic as possible for the stationary state, for instance Gaussian white noise), and the non-stationary state feature could be potentially removed after differential processing. The AR part indicates that the variables are autoregressive at time T_n and the previous i time T_{n-i} (left of the "=" in equation 1), and the MA part indicates that the regression error is a linear combination of the previous errors (right of the "=" in equation 2). ARMA model differential processing was simplified as X_{t-i} substitutes the first order differential with $X_{t-i} - X_{t-i-1}$, and the second order differential $(X_{t-i} - X_{t-i-1}) -(X_{t-i-1} - X_{t-i-2})= X_{t-i} - 2X_{t-i-1} - X_{t-i-2}$

$$X_t - \alpha_1 X_{t-1} - \cdots \alpha_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} \ldots + \theta_q \varepsilon_{t-q} \quad (1)$$

$$(1 - \sum_{i=1}^{p} \varphi_i L^i)(1 - L)^d X_t = (1 + \sum_{i=1}^{q} \theta_i L^i)\varepsilon_t \quad (2)$$

where Time difference lag $L^i X_t = X_{t-i}$ in equation 2, ε_t follows the normal distribution $\varepsilon_t \sim N(0,\sigma^2)$. The model could be expressed as ARIMA (p, d, q) with the aim of "calculating" the training data and select the minimum error loss with (p, d, q) under a certain metrics (AIC, BIC, etc.). Whether the data has seasonality should be determined by analysing the specific data. Sales curve may have the seasonality property. Although the sales revenue is different from quarter to quarter, there are similarities between the four quarters of the previous and this year (the first quarter of the previous year and that of this year). Seasonality does not necessarily refer to the four seasons of the year. The properties of trend, seasonality, cycle in the time series should be investigated on a case by case basis.

We briefly introduced the application of LSTM-based deep learning model in TSF. LSTM (RNN's representative) has achieved good results in TSF performances in the past few years, it could address the challenges of "gradient disappearance" and "explosion" in the training process of long sequences. The basic LSTM network structure [3] was shown in figure 1. The relationship between input and output was given in equation (3).

$$\begin{aligned}
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(output gate)} \\
h_t &= o_t * \tanh(C_t) \\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C_t} \quad \text{(new state } C_t\text{)} \\
\widetilde{C_t} &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \text{ (candidate state } \widetilde{C_t}\text{)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \text{ (input gate)} \\
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \text{ (forget gate)}
\end{aligned} \quad (3)$$

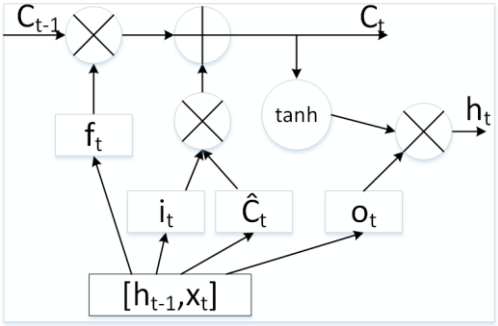

Figure 1. basic structure of LSTM.

The general steps of designing and constructing the basic LSTM models were as follows. Firstly pre-process the dataset, such as feature normalization, scaling, impute for missing values, etc. Split data for training and test after pre-processing. Secondly designing the appropriate LSTM network, such as input

layer, the number of hidden layers, output layer, epoch, batch size, appropriate optimizer and loss function etc. The hyperparameters tuning, more like the art of alchemy than science, plays a critical role in the performance of the models. Thirdly training the model until it converges, evaluate the performance on the training and test set. Lastly model optimization and adjustment: optimize the LSTM model according to data characteristics, such as setting the window parameter w(forecasting horizon). For example, w=3 means using the input variables t-2, t-1, t to forecast the output t+1. The architecture of LSTM model could be improved by designing memory between batch samples and stacking to form a deep network. Then re-train and compare performances and other indicators after model reaches convergence. General steps applied for the LSTM with attention and careful designing of attention layer was the key.

2. Related Papers

Armstrong, etc. proposed [4] the Golden Rule of forecasting, and deduced twenty-eight guidelines. The authors pessimistically concluded TSF failed to improve despite major advances in forecasting methods. To improve TSF accuracy, Khashei, etc. proposed [5, 6] a hybrid of Artificial Neural Networks (ANN) and ARIMA. Results showed around 3 percentage points improve under the metrics of MAE and MSE. Kumar [7] applied ARIMA to forecast air pollutants and evaluated the results with metrics of MAPE, MSE, RMSE. Dastorani, etc. [8] compared AR, MA, ARMA, ARIMA, and identified best one using trial and error method. Liu, etc. [9] proposed using ARIMA to decide the structure of the ANN. Cortez, etc. [10] presented forecasting TCP/IP traffic with NN ensemble approach and ARIMA and Holt-Winters. Results showed the NN and ARIMA achieved the best for different time scale forecasts. The work reported in Refs. [11-14] compared ARIMA with ANN models in predicting Stock Exchange. Results revealed ANN achieved higher accuracy than ARIMA. However, there were contradictory reports for the forecasting performances. The work [15] concluded ARIMA performed better than the back-propagation NN model in forecasting Korean Stock.

Sepp Hochreiter, etc. [3] invented the epoch-making LSTM model. Since its inception great development has been made. We illustrated the basics of the LSTM in figure 1 and equation (3).

Jiang, etc. [16] reviewed recent works on deep learning models for stock market prediction. They compared the different data sources, various NN structures, evaluation metrics, and the implementations. The paper [17] presents a Recurrent Neural Network (RNN) based approach to forecast the price range of the Standard & Poors 500 stock index. The authors [18] proposed using LSTM to predict the stock market. Radityo, etc. [19] compared four ANN methods of backpropagation, genetic algorithm based, genetic algorithm with backpropagation, and neuro-evolution of augmenting topologies NN.

Vaswani, etc. [20] analysed the sequence transduction models based on RNN or CNN in an encoder and decoder configuration. They pointed out models connect the encoder and decoder through attention mechanism was superior to others and that was confirmed by experiments. Their work had a huge impact on the development of attention mechanisms. This paper [21] applied an attention-based LSTM model to predict stock price which achieved impressing results. The authors [22] compared multilayer perceptron (MLP), one-dimensional CNN, stacked LSTM, attention networks for financial TSF. Qiu, etc. [23] proposed LSTM with attention to forecasting stock price. They compared the results among the LSTM, the LSTM with wavelet denoising and the gated recurrent unit (GRU) NN with financial benchmark datasets. Kaji, etc. [24] trained LSTM with attention to predict over two-week patient ICU courses using the MIMIC-III dataset. Ran, etc. [25] proposed LSTM with Attention for Travel Time Prediction. The authors [26] proposed LSTM to model the train dynamic in a nonparametric way. All the papers claimed the superiority of the LSTM with attention over existing methods in TSF tasks, but an in dependent re-evaluation was needed to confirm the conclusions.

3. Methodologies

3.1. Datasets

We collected the benchmark dataset from the Kaggle web traffic competition (https://www.kaggle.com/c/web-traffic-time-series-forecasting) for TSF task and randomly selected one of the time series data to test the performance of our proposed ARIMA and LSTM with attention models. We evaluated the forecasting performances using the same traffic dataset. R package fpp3 [27] was used to construct the ARIMA model and evaluate the performances. Python library GPU version of Tensorflow (backbone) + Keras (deep learning framework) was used for LSTM with attention mechanism. The proposed model was run and evaluated on windows 10 PC with Intel Core i7-2600, 16G RAM, GTX 1080 GPU.

3.2. ARIMA model

3.2.1. Basic Steps of ARIMA Modelling. Generally the forecasting steps involve processes of problem definition, collecting and analysis of the dataset, designing and optimizing of the model, and forecasting and evaluating. The web traffic could be modelled as time series data in that they followed the time order. We selected one randomly from the whole dataset and checked the need for pre-processing, after that the data was graphed to help finding the patterns, trend, seasonality of the data, and even locating the outliers. After analysing the characteristics of the stochastic web traffic which presented little trend or seasonality properties, we proposed using the ARIMA family or ETS (exponential smoothing methods) models. Summarization of the steps was illustrated in figure 2.

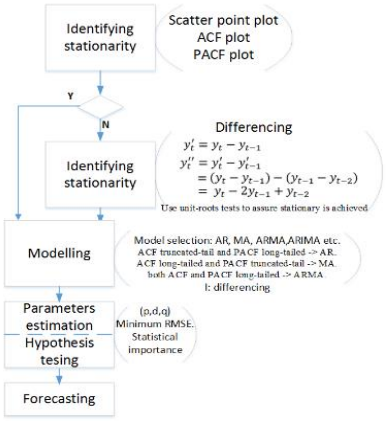

Figure 2. General steps of ARIMA modelling.

3.2.1. Model Designing and Evaluation

Graphs enable many features of the data to be visualised, including patterns, unusual observations, changes over time, and relationships between variables. The features observed from the plots of the data must then be incorporated into the forecasting methods. If the data presents non-stationarity, take orders of differences of the data until the data reach stationarity. Autocorrelation Coefficient (ACF) and Partial Autocorrelation Coefficient (PACF) of stationary time series was examined. By drawing the ACF and PACF diagrams of the first-order difference, the specific web traffic was analysed and judged to be suitable for AR, MA or ARMA models.

ACF and PACF of the time series instruct how to choose from AR, MA, ARMA models. The PACF of stationary series is truncated with ACF tailed, the AR model is applied; if the PACF of stationary series is tailed with ACF truncated, the MA model is applied; if the PACF and ACF of the stationary series are both trailed, it is suitable for ARMA model. The function value is all 0 after the lag period k>q, which is called truncated. Tailing is with k increasing, the function value exhibits exponential or

oscillating attenuation and tends to 0. When the original data exhibit non-stationarity, differencing should be employed. For example the difference d=1, that is: ARIMA (p,1,q), where p means the order of the autoregressive, d degree of first differencing involved, and q order of the moving average.

There are several autocorrelation coefficients, corresponding to each panel in the lag plot. For example, r_1 measures the relationship between y_t and y_{t-1}, r_2 measures the relationship between y_t and y_{t-2}. Metrics of AICc, RMSE, and MAE (equation 4) were used to evaluate and compare the model forecasting performances for a better model.

$$\mathrm{AIC} = -2 \times \mathrm{Log(L)} + 2 \times (\mathrm{p} + \mathrm{q} + \mathrm{k} + 1)$$

$$\mathrm{AICc} = \mathrm{AIC} + \frac{\big(2 \times (\mathrm{p} + \mathrm{q} + \mathrm{k} + 1) \times (\mathrm{p} + \mathrm{q} + \mathrm{k} + 2)\big)}{(\mathrm{T} - \mathrm{p} - \mathrm{q} - \mathrm{k} - 2)}$$

$$SSE = \sum_{n=1}^{i} e_i^2 \quad RMSE = \sqrt{\frac{1}{n} SSE} \quad \mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{x_i} - \widehat{\mathrm{x_i}}) \tag{4}$$

where L stands for the likelihood of the data, p, q are parameters from ARIMA (p,d,q), if c=0 then k=0 else k=1 where c means the average of the changes between the data observations. T stands for the number of values in the time series. Forecasting error for data i: $e_i = x_i - \widehat{x_i}$ where $\widehat{x_i}$ represents the forecast data i. SSE stands for sum of squared error and RMSE root mean squared error.

The residual tests were used to measure the fitness of the model. Check the residuals by plotting the ACF of the residuals, and doing a portmanteau test. If the test of residuals presents random noise like characteristic, the model could be used otherwise the model should be modified. Portmanteau tests whether a group of autocorrelations of the residual time series are different from zero which is often used for autocorrelation in the residuals of a model. Uniroot test the stationarity conditions for the model. If the p complex roots of $\phi(B)$ lie outside the unit circle, and the invertibility conditions are that the q complex roots of $\theta(B)$ lie outside the unit circle. Inverse root means they should all lie inside the unit root to reach the stationarity and invertibility.

ETS could be roughly divided as additive and multiplicative models. Theoretically the two classes of models of ARIMA and ETS have some equivalence relationships for example, ETS (A,N,N) might construct the model comparable to ARIMA(0,1,1). The ARIMA models are presumed to be more general than ETS and we also simply compared the performances of ARIMA model with the ETS in our experiments.

3.3. LSTM with Attention

The attention mechanisms [20] map vectors input (query, key, values) to a vector output which is computed as a weighted sum. The Scaled Dot-Product ($\odot$) Attention computed attention function where queries were packed into a matrix Q, keys and values into matrices K an V. The attention computation is that $\mathrm{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_k}}}\right)$, where $Q=[q^1, q^2, \dots q^n]$, $K=[k^1, k^2, \dots k^n]$, and $V=[v^1, v^2, \dots v^n]$, d means the dimension of the q and k, $a_{1,i} = q^1 \odot k^i / \sqrt{\mathrm{d_k}}$.

We illustrated in figure 3 the basics of the attention mechanism which was used as the self-attention layer in the proposed model.

The architecture of LSTM with attention mechanism model was illustrated in figure 4. We designed one self-attention layer between two LSTM layers with 50 neurons and RELU activation. We reported the performances of losses, mean squared error rate, and the training time with 5 different optimizers (adadelta, adagrad, adam, rmsprop, and sgd).

4. Experiments

4.1. ARIMA Model

We followed the steps of ARIMA modelling. Firstly visualizing the original web traffic and plot of the ACF and PACF of the time series to obtain some information for p,d,q.

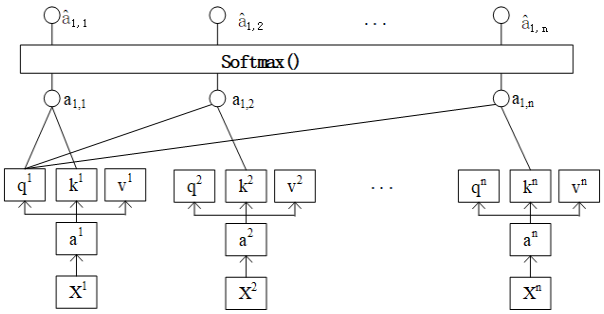

Figure 3. Basics of attention principles.

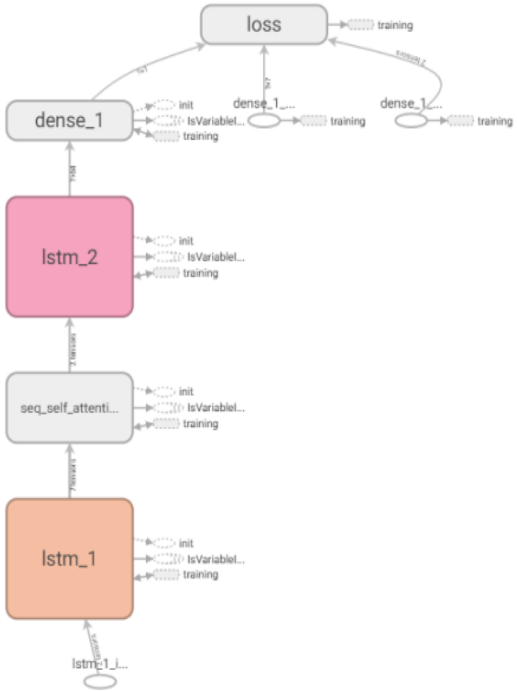

Figure 4. Architecture of LSTM with attention.

Two candidates of (1, 0, 0) and (1, 1, 2) were chosen (see left part in figure 5) for subsequent evaluation. Secondly we checked residuals of these two set of parameters, then we chose the better ARIMA model under the evaluation metrics. Finally, the fitted model (right part of figure 5) and the forecasting comparisons among common statistical models.

```
ARIMA(1,1,2)

Coefficients:
         ar1      ma1      ma2
      0.6131  -0.9667  -0.0333
s.e.  0.0529   0.0667   0.0664

sigma^2 estimated as 9053:  log likelihood=-3280.84
AIC=6569.68   AICc=6569.76   BIC=6586.91
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1     mean
      0.6305  96.7161
s.e.  0.0330  10.9084

sigma^2 estimated as 9024:  log likelihood=-3284.27
AIC=6574.55   AICc=6574.59   BIC=6587.48
```

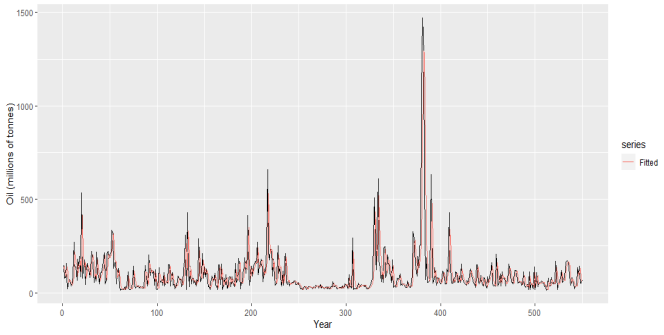

Figure 5. Metrics of ARIMA and the fitted model.

The black curve in the right part of figure 5 was the original web traffic, and the red curve represented the fitted values of the ARIMA model. The ARIMA (1, 1, 2) model was compared with the ETS (A, N, N) model using the RMSE, MAE metrics.

Performances of ARIMA models with different (p, k, q) were made and the more suitable model of ARIMA (1, 1, 2) was chosen as the ARIMA forecasting model. We compared the performances of ARIMA and ETS models (see figure 6). The AIC, AICc and BIC metrics were used as the criteria for model selection and they were displayed in table 1. The less values indicate the better model.

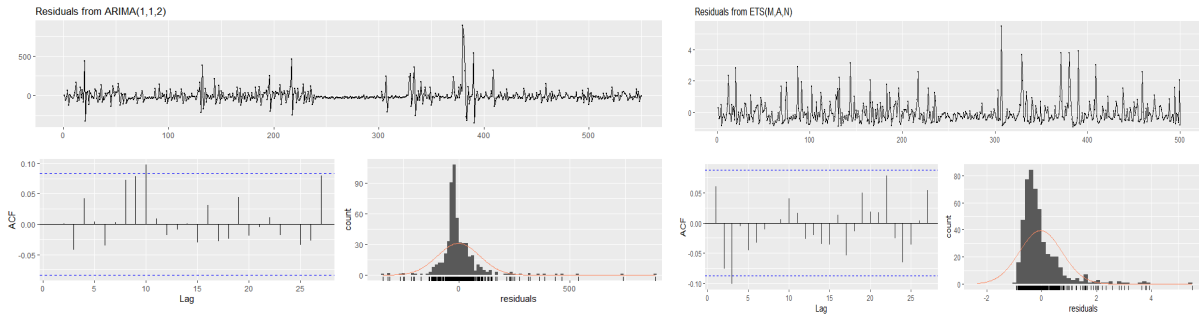

Figure 6. Comparisons between ARIMA and ETS models.

Table 1. Performances comparison between ETS and ARIMA model.

ETS(A,N,N)	RMSE	MAE
Training set	47.71	29.80
Test set	46.04	10.04
ARIMA(1,1,2)	RMSE	MAE
Training set	48.53	30.29
Test set	47.89	23.89

4.2. LSTM with Attention

We trained 100 epochs for each optimizer for 10 times and found that training and testing time ranges from 6 to 10 minutes per optimizer and the best performed optimizer was Adam. This model achieved the comparable performances with the ARIMA models (See figure 7 and table 2 for the results).

From RMSE scores of LSTM with attention (table 1) and ARIMA model (table 2) we found that the LSTM with attention performed slightly better than the traditional statistical ARIMA model. The neural network achieved comparable performances to the traditional statistical models.

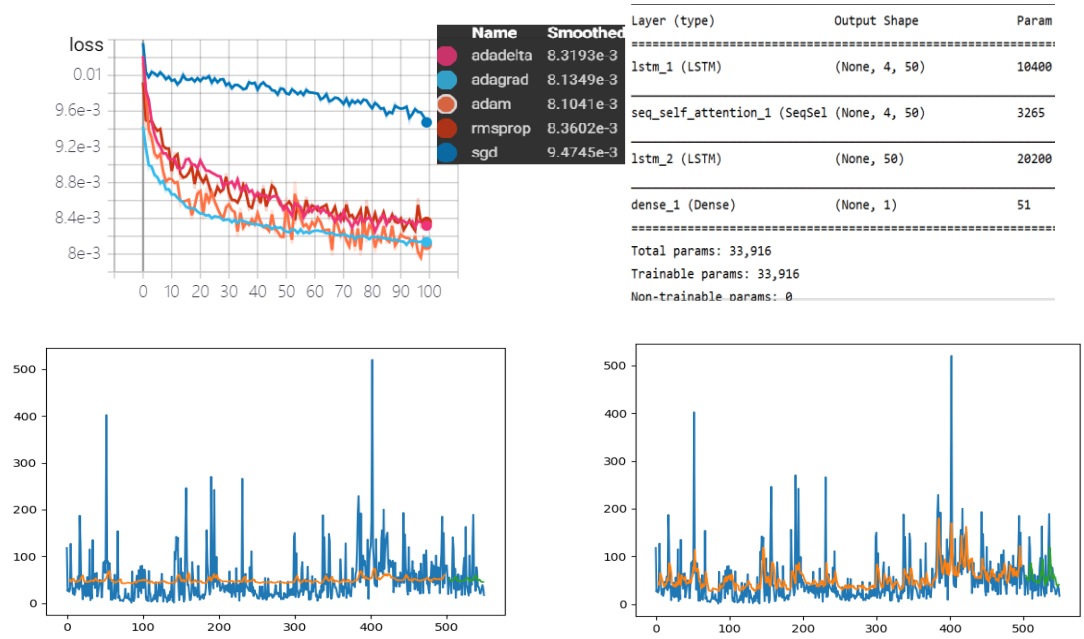

Figure 7. Results for LSTM with attention mechanism.

Table 2. Performances of LSTM with attention mechanism.

Optimizer	Train/Test RMSE	Optimizers Parameters
adadelta	46.77/40.12	learningrate=0.01, rho=0.9
adagrad	46.77/41.29	learningrate=0.01
adam	45.12/40.40	Learningrate=0.01
rmsprop	46.34/40.52	lr=0.01, rho=0.9
sgd	50.32/40.54	lr=0.01, decay=1e-6, momentum=0.9, nesterov=True

5. Conclusions

Time series modelling and prediction can be roughly divided into traditional and deep learning-based methods. Traditional modelling such as stochastic process, ARIMA family, ETS and others are relatively mature. Although deep learning such as LSTM technology has achieved better accuracy than traditional models in the field of sequence prediction, its complexity is generally higher. Deep learning NN need to be designed, while some traditional models can be easily constructed which requires less computing resources. Deep learning models lack interpretability, such as the design of the number of neurons in the input layer and hidden layer, and the tuning of hyperparameters was like alchemy. Although LSTM with attention mechanism sometimes achieved better than traditional models in TSF, ARIMA could be used for small-scale of TSF work considering the network structure, traffic characteristics, accuracy, complexity and other factors.

Acknowledgments

This work was supported in part by the Institute of Computer Application, China Academy of Engineering Physics under Grant “SJ2019A05”. Comments from reviewers are appreciated in advance.

References

[1] Box G E, Jenkins G M, Reinsel G C and Ljung G M 2015 *Time Series Analysis: Forecasting and Control* (John Wiley & Sons).
[2] *Lecture Notes about ARIMA Model* http://people.duke.edu/~rnau/411arim.htm
[3] Hochreiter S and Jürgen S1997 Long short-term memory *Neural Computation 9.8* pp 1735-1780.
[4] Armstrong J S, Green K C and Graefe A 2015 Golden rule of forecasting: Be conservative *Journal of Business Research* **68** (8) 1717-1731.
[5] Khashei M and Khashei B 2011 A novel hybridization of artificial neural networks and ARIMA models for time series forecasting *Applied Soft Computing Journal* **11** (2) 2664-2675.
[6] Khashei M and Bijari M 2010 An artificial neural network (p, d, q) model for timeseries forecasting *Expert Systems with Applications* **37** (1) 479-489.
[7] Kumar U and Jain V K 2010 ARIMA forecasting of ambient air pollutants (O_3, NO, NO_2 and CO) *Stochastic Environmental Research and Risk Assessment* **24** (5) 751-760.
[8] Dastorani M, Mirzavand M, Dastorani M T and Sadatinejad S J 2016 Comparative study among different time series models applied to monthly rainfall forecasting in semi-arid climate condition *Natural Hazards* **81** (3) 1811-1827.
[9] Liu H, Tian H Q and Li Y F 2012 Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction *Applied Energy* **98** 415-424.
[10] Cortez P, Rio M and Rocha M 2012 Multiscale Internet traffic forecasting using neural networks and time series methods *Expert Systems* **29** (2) 143-155.
[11] Babu C N and Reddy B E 2012 Predictive data mining on Average Global Temperature using variants of ARIMA models *International Conference on Advances in Engineering Science and Management (ICAESM)* pp 256-260.
[12] Wadi S A L, Almasarweh M and Alsaraireh A A 2018 Predicting closed price time series data using ARIMA Model *Modern Applied Science* **12** 11.

[13] Ariyo A A, Adewumi A O and Ayo C K 2014 Stock price prediction using the Arima model *Proc. IEEE International Conference on Computer Modelling and Simulation (ICCMS)*.
[14] Wijaya Y B, Kom S and Napitupulu T A 2010 Stock price prediction: Comparison of Arima and artificial neural network methods-An Indonesia stock's case *International Conference on Advances in Computing, Control, and Telecommunication Technologies* pp 176-179.
[15] Lee C K, Sehwan Y and Jongdae J 2007 Neural network model versus SARIMA model in forecasting Korean stock price index (KOSPI) *Issues in Information System* **8** (2) 372-378.
[16] Jiang W 2020 Applications of deep learning in stock market prediction: recent progress *ArXiv abs/2003.01859*.
[17] Lin Y, Ueng Y, Chung W and Huang T 2019 Stock price range forecast via a recurrent neural network based on the zero-crossing rate approach *IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)* pp 1-9.
[18] Gao T W, Chai Y T and Liu Y 2017 Applying long short term memory neural networks for predicting stock closing price *8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*.
[19] Arief R, Munajat Q and Budi I 2017 Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*.
[20] Vaswani, A, Shazeer, N Parmar N, Uszkoreit J, Jones L, Gomez A N and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* 5998-6008.
[21] Chen L C, Huang Y H and Wu M E 2018 Applied attention-based LSTM neural networks in stock prediction *IEEE International Conference on Big Data*.
[22] Kim S and Kang M 2019 Financial series prediction using attention LSTM *arXiv preprint arXiv:1902.10877*.
[23] Qiu J, Wang B and Zhou C 2020 Forecasting stock prices with long-short term memory neural network based on attention mechanism *PLoS ONE* **15** (1).
[24] Kaji D A, Zech J R, Kim J S, Cho S K, Dangayach N S, Costa A B and Oermann E K 2019 An attention based deep learning model of clinical events in the intensive care unit *PloS ONE* **14** (2).
[25] Ran X, Shan Z, Fang Y and Lin C 2019 An LSTM-based method with attention mechanism for travel time prediction *Sensors* **19** 861.
[26] Li Z, Tang T and Gao C 2019 Long short-term memory neural network applied to train dynamic model and speed prediction *Algorithms* **12** 173.
[27] Hyndman R J and Athanasopoulos G 2019 *Forecasting: Principles and Practice* 3rd ed (Melbourne, Australia: OTexts).