14th Annual International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), February 12-14, 2024
Publisher: IEOM Society International, USA DOI: 10.46254/AN14.20240118
Published: February 12, 2024

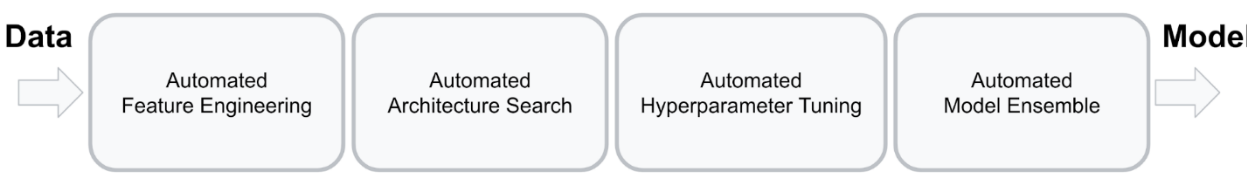
Automated Machine Learning Algorithms for Long-Term Time Series Forecasting

Ying Su and Morgan C. Wang
Department of Statistics and Data Science
University of Central Florida
Orlando, Florida 32816, USA
ying.su@ucf.edu, chung-ching.wang@ucf.edu

Abstract

Long-term time series forecasting is an important research area for automated machine learning (AutoML). Currently forecasting based on either machine learning or traditional statistical model is usually built by experts and it requires significant manual effort: from model construction, feature engineering, and hyper-parameter tuning to construction of the time series model. Automation is not possible since there are too many human interventions. To overcome these limitations, this article proposed to use a Recurrent Neural Network (RNN) variant, Long Short-Term Memory (LSTM), through the memory cell and gates to perform long-term time series prediction. We have shown that this proposed approach is better than traditional Autoregressive Integrated Moving Average (ARIMA). In addition, we also found it is better than other neural network systems.

Keywords
Automated Machines Learning, Autoregressive Integrated Moving Average, Neural Networks, Time Series Analysis, Time Series Forecasting



1. Introduction

Time series forecasting still remains a dynamic and active research area for statistical analysis and machine learning (ML), such as retail and financial industries. Currently, machine learning based tasks need professionals to understand and construct traditional time series forecasting models, and also require significant manual efforts. Automated Machine Learning (AutoML) plays an important role in tackling limitations. AutoML has been studied by Google

Figure 1. AutoML Pipeline (Liang et al. 2020)

Brain Team (2020) and also provides the pipeline of procedures (Figure 1).
The AutoML approach aims to use raw data as input to produce a high-quality model output without human intervention. However, the forecasting accuracy for both traditional analysis and AutoML approaches degrades significantly in the long-term forecasting.

1.1 Objectives

Therefore, the objective of this article is to provide a comprehensive report between the typical and traditional time series model and multiple Neural Network (NN) systems, and empirically shows that LSTM (Hochreiter and Schmidhuber 1997) is suggested to be the best option when forecasting in long term. It also presents the short-term and long-term forecasting accuracy between traditional Autoregressive Integrated Moving Average (ARIMA) (Box et al. 2013) and NN models, which are (1) Fully Connected Network (FNN); (2) Convolutional Neural Network (CNN); (3) Nonpooling Convolutional Neural Network (NPCNN) (Liu et al. 2019); (4) Vanilla RNN (LeCun et al. 2015); (5) LSTM.

2. Literature Review

Time-series data comprises three significant components: trends, seasonality, and time lag correlations. To apply traditional statistical models, such as Seasonal ARIMA, time series stationarity needs to be satisfied in advance.

In the meanwhile, NN has become one of the most commonly used methods to solve time-series tasks. The neurons in NN will recognize and memorize the patterns in the big data, so it produces the high-quality predictive model (Hinton 1992). Based on certain tasks, a number of different NN architectures can be constructed. The baseline model in the experiments is FNN. FNN is a basic NN structure that consists of a series of fully connected layers and each neuron in a certain layer can connect to each neuron in the other layers. CNN is the regularized version of FNN that additionally conducts convolutional and pooling operations. Other than time series forecasting, CNN is widely applied in image, video, and speech recognition (Abdel-Hamid et al. 2014). NPCNN is the same as the structure of a CNN without the pooling layer, and the findings show it is the best option when forecasting seasonal and trended time series. An RNN is one class of neural network and particularly designed for sequence model (Rumelhart et al. 1986), so it has been widely used to solve time series and ordered data tasks, such as sentiment classification and stock price forecasting. And yet, the RNN variant, LSTM, can overcome the exploding and vanishing gradient problem by its memory cells and gating mechanisms.

3. Time Series Modeling

3.1 Traditional Statistical Approaches

Time series data can include two important and common patterns: seasonality and trends. Basic decomposition models in time series analysis are additive and multiplicative. The experiment of our article adopts the multiplicative decomposition of seasonal and trended time series can be described as follows

$$X_t = T_t \times S_t + e_t \tag{1}$$

where at time t, X_t is the value of time series, T_t is the trend component, S_t is the seasonality component and e_t is the noise.

ARIMA and seasonal ARIMA (SARIMA) belong to traditional statistical approaches to forecast time series data. The decomposition of time series should be processed in order to remove the effects of seasonality and trend. ARIMA(p, d, q) is a classical statistical method that generalized by the integration of Autoregressive (AR) and Moving Average (MA) and make the non-stationary data stationary by taking the difference between current data values and previous data values. p is the order of AR, d is the degree of differencing and q is the order of MA. To indicate seasonality in time series, p, d, q represent nonseasonal factors and P, D, Q represent seasonal factors in the ARIMA$(p, d, q) \times (P, D, Q)_s$ model.

3.2 Neural Network Approaches

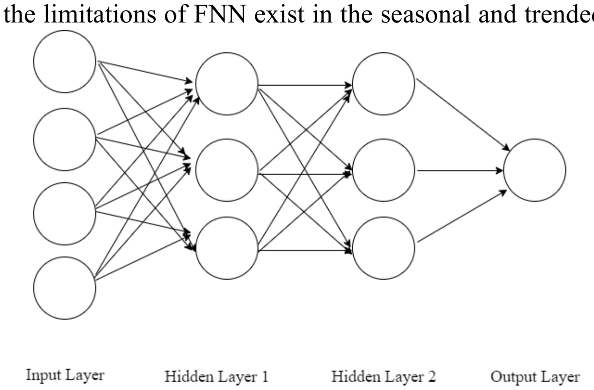
From this approach, no specific assumptions need to be made about the model and the underlying relationship is determined solely through data mining (Zhang and Qi 2005). Given the characteristics of AutoML, NN can automatically recognize the seasonal and trend patterns, and select the optimal NN model.

Given any time series, the equation is

$$X_t = f(X_{t-1}, X_{t-2}, \cdots, X_{t-k}) + \varepsilon_t \tag{2}$$

where $X_{t-1}, X_{t-2}, \cdots, X_{t-k}$ are the previous lagged observations, f are some functions and ε_t is the error.

A FNN is a basic neural network model that contains only fully connected layers. Figure 2 presents the basic FNN architecture. However, the limitations of FNN exist in the seasonal and trended time series because it is difficult for



FNN to access multiple past data points and capture the seasonality (Zhang and Qi 2005)

The purpose of convolutional layer and pooling layer from CNN is to extract the significant features, as well as the patterns that can reflect the important components in time series data. Although NPCNN removes the pooling layers

in CNN, it helps avoid too much loss of information in the univariate time-series data.

RNN architecture that also contains an input layer, hidden layers, and an output layer. Different from an FNN, RNN includes a recurrent network in the hidden layers which can feedback as input data to current step. The design of recurrent network in an RNN enables to capture the sequential information and store it in the memory state by Backpropagation Through Time (BPTT) algorithm.

However, BPTT in RNNs usually suffers from exploding and vanishing gradient problems, since the backpropagated error in temporal evolution exponentially increases or decreases based on the size of the weights. Hochreiter and Schmidhuber (1997) propose an RNN variant, LSTM, to address the problems, that is enforcing constant error flow through internal states of special units. LSTM is more proper to overcome the long-term dependency problem by its memory cells and gating mechanisms. Figure 3 illustrates the operational principles and roles of memory cells within

Figure 1. Basic FNN Architecture

an LSTM architecture.

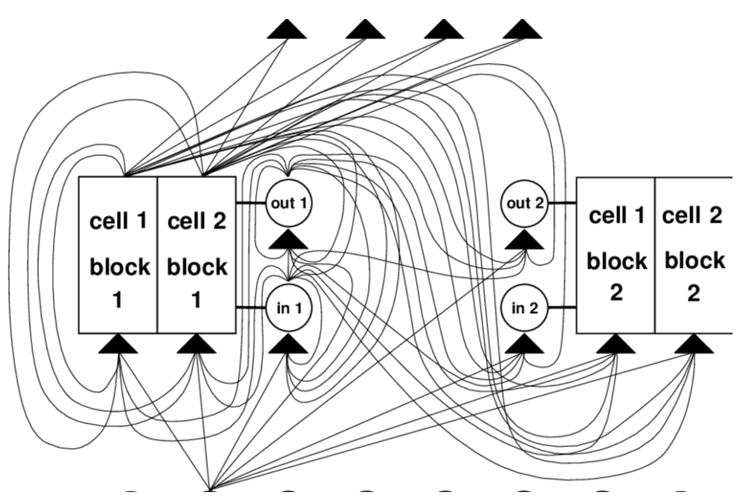


Figure 2. Memory Cell in LSTM Architecture (Hochreiter and Schmidhuber 1997)

4. Experiments

This section presents the experiment design of ARIMA and NN models and demonstrates the potential challenges of long-term time series forecast with using simulated data and real data.

4.1 The Simulated Data

The idea of simulated time series data is originally adopted from (Zhang and Qi 2005), but the simulated data needs to be generated and adjusted in accordance with our research goals. Simulated data is monthly time-series data because it is more challenging to forecast.

To generate simulated time series, the mathematical expression of multiplicative model follows

$$y_t = T_t \times SI_t + e_t \tag{3}$$

where at time t, $T_t = 120 + 0.5t$ represents the linear trend components, SI_t represents seasonal components referred in Table 1, and e_t represents the error term using the Gaussian noise $N(0, 15^2)$.

Table 1. Seasonal Indexes for the Simulated Data

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec
Seasonal Indexes (SI)	0.75	0.80	0.82	0.90	0.94	0.92	0.91	0.99	0.95	1.02	1.2	1.8

The long-term time series forecasting task is difficult and unexpected. Instead of simulated data having a stable linear growth, we decide to modify the data by increasing 40% values from $t = 156$ to $t = 180$. The purpose of modification is to make the simulated data more complex, so it can mimic some real data. Figure 4 is shown as the plot of simulated time series after the adjustment.

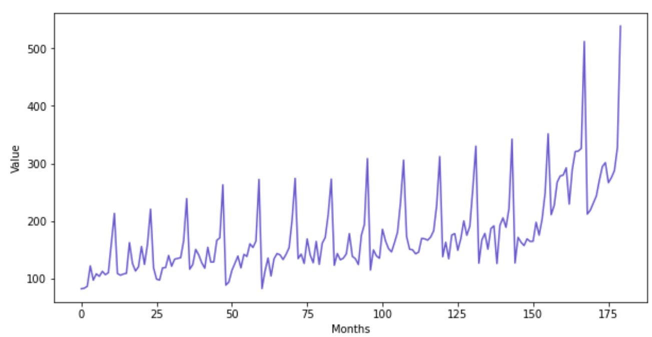


Figure 3. The Simulated Time Series after Adjustments

4.2 The Real Data

The real data is the monthly wine sales collected by the database of analytical results of the Australian Wine Research Institute's Commercial Services Group (Godden et al. 2015). Both real data range from January 1980 to December 1994, and the dataset includes white wine sales and red wine sales in such time order. Figure 5 is shown the plots for white wine sales and red wine sales respectively. In this study, white wine sales and red wine sales are both treated as independent univariate time-series data.

4.3 Modeling Strategy

The number for simulated data points and real data points is 180 respectively. For data partition, the training set includes the first 156 data points, and the test set includes the remaining 24 data points. As results, data splits in such form of 156-month training set and 24-month test set (unknown forecast lengths).

Figure 4. The Real Data

Firstly, to determine if the time series data is stationary, Augmented Dickey Fuller (ADF) test is commonly applied before SARIMA modeling. The results of ADF test show that simulated data and both real data are non-stationary time series, so they need to be deseasonalized and detrended (DSDT). Given a range of parameters in $\mathrm{ARIMA}(p,d,q)\times(P,D,Q)_s$, it requires to use grid search algorithm to target the best combination of parameters' values among all the possible combinations.

Secondly, let's discuss the experimental setting for NNs. Please note all of the neural network models are applied by the open-source software Keras and Tensorflow.

A FNN is considered as a baseline model among all the NNs. The construction of CNN, NPCNN, Vanilla RNN and LSTM will be modified based on the construction of FNN. Due to the state-of-art of AutoML, NN systems do not require time series deseasonalization and detrend, so this data preprocessing step can be omitted. Moreover, data normalization can be applied based on the nature of the task. The scaled training set needs to be proceeded to TimeseriesGenerator, so a univariate time series can generate batches of input-output pairs.

In the article (Box et al. 2013), the empirical findings have shown and suggested that the combination of ReLU activation function and Adam optimizer is the best and most efficient, when modeling in the baseline FNN. Hence, this combination is used by all the NN models.

The FNN in the experiment includes 4 layers: input layer based on the shape of generated_batches, 2 hidden layers (Dense(64) and Dense(8)) both using ReLU activation functions and output layer using Linear activation function.

A CNN is constructed based on the construction of FNN. Other than the layers in the FNN, one convolutional layer with 8 convolutional kernels (Conv1D) and ReLU activation function, and one pooling layer with ReLU activation function (MaxPooling1D) are added in the hidden layers. The construction of NPCNN is the same as CNN, except removing the pooling layer (MaxPooling1D).

Given the FNN as baseline, a fully-connected RNN layer with 128 neurons and ReLU activation function (SimpleRNN) is added and the remaining layers keep to be the same as in FNN. However, a Vanilla RNN can easily suffer from learning dependencies. To improve the performance, we can highly recommend a LSTM. When constructing LSTM networks, 128 neurons and ReLU activation function in LSTM layer (LSTM) are selected addition to FNN baseline model.

5. Empirical Findings

This section aims to show and interpret the empirical results from the experiments. The empirical results include the performances and visualizations of long-term forecasts of the test set. In the experiment, the evaluations of models are not only on the whole test set, but evaluations of first 6-month, 12-month, 18-month and 24-month data of the test set are also shown.

Three accuracy metrics are used for model evaluation: (1) Mean Absolute Percentage Error (MAPE), (2) Mean Absolute Error (MAE), and (3) Root Mean Square Error (RMSE). MAPE and MAE are more important measures in time-series forecasting errors, because MAPE quantifies the absolute percentage errors between predicted and actual values while MAE measures the absolute magnitude of errors between predicted and actual values.

5.1 The Results of Modeling Simulated Data

For the 6-month forecasts, the forecasting errors from CNN and Vanilla RNN are not far different, but both of them have the highest errors. In Table 2, the forecasting errors from ARIMA, FNN and LSTM are very close to each other's, but the errors of LSTM show to be the lowest. 6-month forecasting MAPE of LSTM is about 7.0% lower than that of ARIMA, and 24-month forecasting MAE of LSTM is 8.95% lower than that of ARIMA. To conclude, for 12-month, 18-month and 24-month forecasting, the forecasting errors of LSTM outperform all the models by resulting in the lowest errors.

Table 2. Forecasting Results of the Test Set in the Simulated Data

Forecast Lengths	NNs	*MAPE(%)*	*MAE*	*RMSE*
6-Month	ARIMA-DSDT	30.5301	80.2989	83.0574
	FNN	29.8322	78.9007	82.1851
	CNN	33.0376	86.4791	90.1046
	NPCNN	30.7995	80.6975	84.169
	Vanilla RNN	33.6584	87.6101	90.2922
	LSTM	**28.3919**	**75.1456**	**78.0703**
12-Month	ARIMA-DSDT	29.7999	89.3639	94.4594
	FNN	35.7379	112.9054	130.8676
	CNN	31.0929	91.8531	96.2516
	NPCNN	30.8727	91.0729	94.4444
	Vanilla RNN	33.7313	98.7341	101.6353
	LSTM	**28.0249**	**83.6608**	**88.3844**
18-Month	ARIMA-DSDT	27.4975	78.7479	84.8887
	FNN	32.9382	98.2652	115.0654
	CNN	29.0096	81.8401	87.7812
	NPCNN	28.3856	80.3477	85.6998
	Vanilla RNN	33.6886	93.6318	97.0369
	LSTM	**26.7399**	**75.8134**	**81.0608**
24-Month	ARIMA-DSDT	27.1813	81.6081	88.9764
	FNN	34.8293	109.908	130.8493
	CNN	28.1684	83.3046	90.6289
	NPCNN	28.1708	83.6431	89.9661
	Vanilla RNN	33.641	98.2131	102.942
	LSTM	**25.3491**	**74.3007**	**79.3263**

ARIMA - DSDT = Deseasonalization and Detrend have processed in the data before ARIMA modeling

According to the Figure 6, the ARIMA$(3,1,0) \times (0,1,1)_s$ and LSTM predictions do show capturing and well predict the abrupt increase in the simulated data, but prediction by LSTM eventually and approximately reaches to actual value as time passes. LSTM shows itself to be a more reliable and flexible tool when forecasting long-term time series.

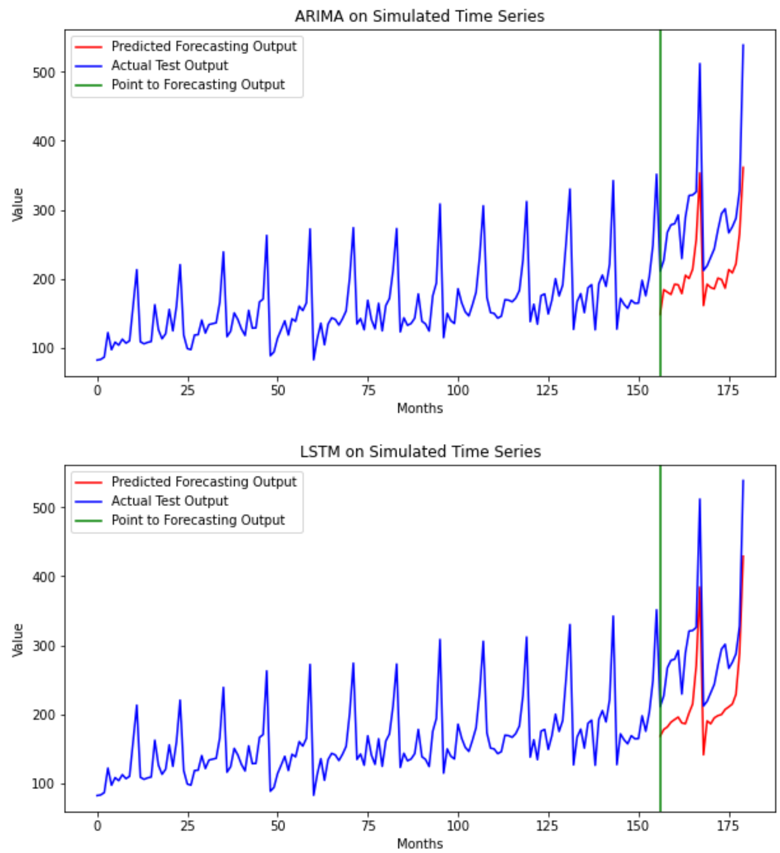


Figure 5. The Simulated Data Forecasting Compared by ARIMA$(\mathbf{3},\mathbf{1},\mathbf{0}) \times (\mathbf{0},\mathbf{1},\mathbf{1})_{\boldsymbol{s}}$ and LSTM

5.2 The Results of Modeling Real Data

The evaluation design of real data is similar to simulated data. Firstly, for the white wine sales data, Table 3 shows that forecasting errors of ARIMA and CNN are the closest to the errors of LSTM, ARIMA and CNN slightly better predict the short-term forecasted values. However, the forecasting errors of FNN are shown to be one time higher than them. The white wine sales data does not reveal enough linear trend but obvious seasonality. The long-term forecasting results of LSTM outperform in terms of all three metrics. Moreover, Figure 7 shows that RNN forecasts are well able to capture small fluctuations of the sequence which ARIMA is not.

The red wine sales data shows more obvious trends and seasonality, comparing to white wine sales data. Table 3 also shows that the forecasting errors of ARIMA are closer to the forecasting errors of Vanilla RNN and LSTM in the red wine data. The LSTM also outperforms in all lengths forecasting. After time series deseasonalizing and detrending, ARIMA can be competitive among the model selections. In Figure 7, the visualization of ARIMA and RNN forecasts can be attested.

In both long-term white and red wine sales forecasting, LSTM model exhibits the most accurate results. 24-month forecasting MAPE of white wine sales by LSTM is 8.36% decrease that of ARIMA, while 18-month forecasting MAE of red wine sales by LSTM is 6.61% decrease that of ARIMA.

Table 3. Forecasting Results of the Test Set of Wine Sales Data

		White Wine Sales			Red Wine Sales		
Forecast Lengths	NNs	*MAPE(%)*	*MAE*	*RMSE*	*MAPE(%)*	*MAE*	*RMSE*
6-Month	ARIMA-DSDT	8.7092	267.3638	367.6425	16.0119	230.6409	270.2548
	FNN	14.5056	366.8641	526.7203	28.0079	469.9322	483.649
	CNN	**8.2626**	266.0152	391.1015	16.4397	267.2242	303.2052
	NPCNN	9.8683	318.8118	418.1253	22.4071	279.4302	344.6334
	Vanilla RNN	10.2952	317.0787	425.2559	14.1835	254.0651	304.787
	LSTM	8.8306	**258.3949**	**306.662**	**12.4402**	**174.1644**	**194.6974**
12-Month	ARIMA-DSDT	**9.8989**	369.4735	541.148	13.2377	251.8652	347.5172
	FNN	17.9799	673.7442	925.8129	27.0217	599.9702	705.9766
	CNN	10.5611	408.5439	597.8329	15.9748	335.4088	389.835
	NPCNN	10.8804	399.9398	542.6419	18.1866	314.2824	410.1849
	Vanilla RNN	10.8321	398.01	567.9462	14.4389	325.539	406.0045
	LSTM	9.9255	**355.3635**	**488.6**	**11.7875**	**233.302**	**311.8323**
18-Month	ARIMA-DSDT	10.7716	393.6367	529.923	12.9352	246.4817	328.2575
	FNN	16.9081	598.4941	813.4253	26.0258	563.6044	652.594
	CNN	12.694	475.0189	629.5812	15.8407	340.4696	403.8087
	NPCNN	12.7756	460.7533	584.4867	18.136	293.6673	393.2292
	Vanilla RNN	11.6856	423.8925	564.119	15.9224	356.7617	434.0644
	LSTM	**9.6394**	**342.0841**	**450.8697**	**11.5444**	**230.1887**	**293.2363**
24-Month	ARIMA-DSDT	11.4001	447.872	563.1053	11.5167	239.589	321.9751
	FNN	19.0744	754.0756	988.2545	27.4151	649.0517	754.1498
	CNN	13.5519	540.6997	667.8237	17.2302	405.4927	479.3757
	NPCNN	12.8508	495.1397	601.7356	15.3284	271.7573	374.4326
	Vanilla RNN	12.4471	483.3543	592.93	16.2465	392.5483	475.9419
	LSTM	**10.4468**	**402.4446**	**493.8023**	**10.6039**	**233.1067**	**315.2351**

ARIMA - DSDT = Deseasonalization and Detrend have processed in the data before ARIMA modeling

6. Conclusion

Nowadays, time series analysis and forecasting have been studied by more techniques such as traditional statistical methods, machine learning and deep learning. The automated search is able to adjust the architecture and hyperparameter choices for different datasets, which makes the AutoML solution generic and automates the modeling efforts (Liang et al. 2020). Meanwhile, the purpose of this study is to provide a solution in term of AutoML novelty when solving the long-term time series tasks. This article has provided analysis between classical statistical model ARIMA and multiple NNs, and empirically shown that ability and significance of neural network models on the seasonal and trended time series. Under the characteristics of AutoML, LSTM is suggested when we solve long-term time-series forecasting task. The finding of Emsafi et al. (2022) reveals how competitive are neural networks time-series forecasting compared with traditional univariate methods.

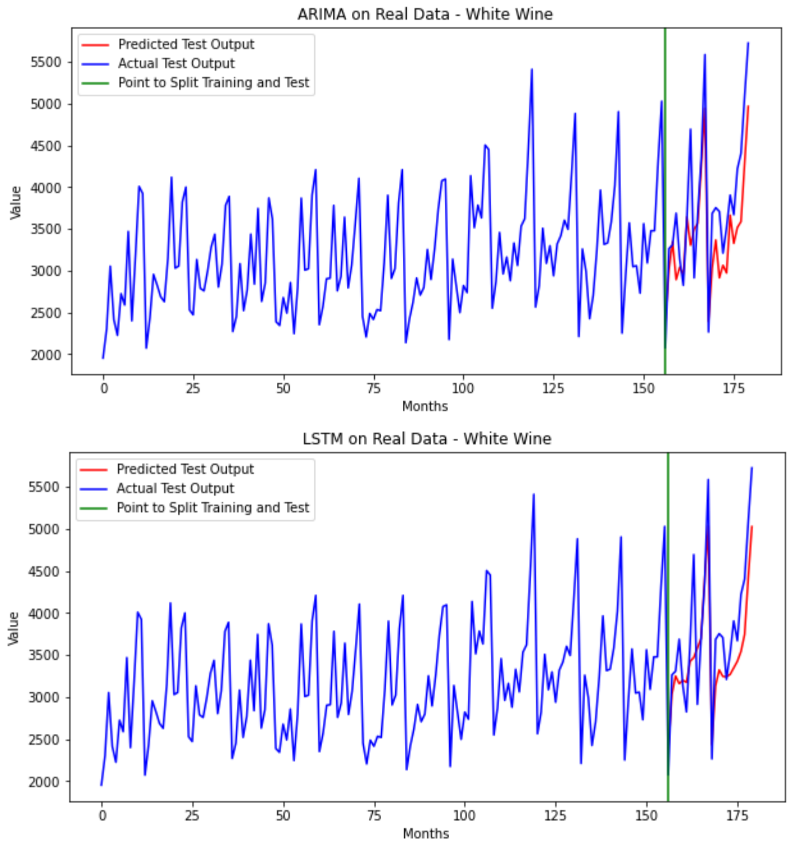


Figure 7. The White Wine Sales Data Forecasting Compared by ARIMA(0,1,1)×(0,1,1)_s and LSTM

References

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G. and Yu, D., Convolutional neural networks for speech recognition, *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533-1545, 2014.

Box, G. E., Jenkins, G. M., Reinsel, G. C. and Ljung, G. M, Time series analysis: forecasting and control, *John Wiley & Sons,* 2013.

Ensafi, Y., Amin, S. H., Zhang, G. and Shah, B., Time-series forecasting of seasonal items sales using machine learning–A comparative analysis, *International Journal of Information Management Data Insights*, vol. 2*,* no.1, p.100058, 2022.

Godden, P., Wilkes, E. and Johnson, D., Trends in the composition of Australian wine 1984–2014, *Australian Journal of Grape and Wine Research*, vol. 21, pp. 741-753, 2015.

Hinton, G. E., How neural networks learn from experience, *Scientific American*, vol. 267, no. 3, pp. 144-151, 1992.

Hochreiter, S. and Schmidhuber, J., Long short-term memory*, Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

LeCun, Y., Bengio, Y. and Hinton, G., Deep learning, *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

Liang, C., Lu Y., Google Research and Brain Team (2020), Using automl for time series forecasting, Available: https://ai.googleblog.com/2020/12/ using-automl-for-time-series-forecasting.html, 2020.

Liu, S., Ji, H. and Wang, M. C., Nonpooling convolutional neural network forecasting for seasonal time series with trends, *IEEE transactions on neural networks and learning systems*, vol. 31, no. 8, pp. 2879-2888, 2019.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J., Learning representations by back-propagating errors, *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.

Zhang, G. P. and Qi, M., Neural network forecasting for seasonal and trend time series, *European journal of operational research*, vol. 160, no. 2, pp. 501-514, 2005.

Biographies

Ying Su received the M.S. degree in Statistics and Data Science from the University of Central Florida, Orlando, Florida, USA in 2021. She is currently a Ph.D. Candidate of Big Data Analytics Program at University of Central Florida, Orlando, Florida, USA. Her current research interests include time series, neural networks, machine learning and data mining.

Morgan C. Wang is a Professor at Department of Statistics with the college of Science, University of Central Florida, Orlando, Florida, USA. His current research interests include automated machine learning, deep learning and enhance AI using Big Data Analytics.