Hybrid ARIMA-LSTM with Attention Mechanism For Time Series Forecasting: Report

G056 (s2015708, s2045321, s2025012)

Abstract

Forecasting task based on time-sequential data is a hot topic in various domains such as, finance, urban management, astronomical forecasting. Very recently, the superiority of deep learning has gained increasing attention in forecasting task. Long Short-Term Memory(LSTM), as a recurrent model naturally has the ability to handle time series forecasting task and present a state of art performance. In this report, we treat LSTM as the baseline model and improved it through two ways: data enhancement and model structure improvement. The idea of data enhancement is inspired from (Zhang, 2003), which establish a superior hybrid model composed of Autoregressive integrated moving average(ARIMA) and LSTM. The idea of model structure improvement is inspired from (Qin et al., 2017b) which import the attention mechanism. We will apply them to SP 500 dataset an the performance will be evaluated by comparing with baseline.

1. Introduction

SP500 (Standard Poor's 500) is a collection of 505 stocks that are selected by the SP 500 Index Committee.Generally, the SP500 index basically reflects the overall performance of large-cap stocks, and is considered by analysts to reflect the overall global stock market and the indicator of US economy. In history, scientists and engineers have never stopped exploring the task of forecasting the prices of S&P 500 stocks. Autoregressive integrated moving average(ARIMA) introduced by (Gwilym, 1970) made a great success in both academia and industry fields. In 1980S, this simple-architectural model is widely used in time series stocks and rainfall forecasts due to its high quality on prediction results and Box-Jenkins Modelling process(a pipeline used to determine the internal paramaters of ARIMA). However, with the coming of the information age and data explosion, the need for high accuracy of time series far exceed ARIMA's power. Autoregressive integrated moving average(ARIMA) as a linear statistic model, is limited by the features of input series which means when ARIMA is appiled to predict non-stationary data, the results will become noisy and meaningless. Moreover, (Gwilym, 1970) also illustrated that ARIMA is demanding on data selection due to its limitation on non-linear input series data.

In recent years,with the great improvement of both hashing rate and graphics processor hardware, deep learning has become a new trend in many scenarios, especially, since (Hochreiter & Schmidhuber, 1997b) proposed Long Short-Term Memory(LSTM), deep learning algorithm has been beginning to show its dominance in time series forecasting task. Many researchers have explored its superior performance in forecasting for sequential series and trying to modifying its architecture to reach a higher quality prediction results. However, inspired from an innovative idea proposed by (Zhang, 2003). We consider that ARIMA and LSTM are complementary to each other in the extent of data augmentation(fully discussed in section 2). In our project, we focus on establishing a state-of-the-art hybrid model based on the baseline LSTM and applying this hybrid method to S&P stocks prices forecasting task. We are expecting that this hybrid ARIMA-LSTM model combines the advantages: feasible for universal data, and complements the limitations of both single model. Moreover, attention algorithm as a popular mechanism for encoder-decoder will be considered as auxiliary method added into this hybrid ARIMA-LSTM model for further improvement.

In order to evaluate its performance, mean square error(MSE), mean absolute deviation(MAD), Root Mean Squared Logarithmic Error(RMSLE) and Mean Absolute Percentage Error(MAPE) are selected be the forecasting accuracy measures.

The present paper is structured as follows: in section 2, we will introduce the work related to our project. Section 3 explains our data set and task description. Some methodology will be introduced in section 4. Finally, results and conclusions will be introduced in section 5 and 6.

2. Related Work

In the era when deep learning was not widely used, people mainly used statistical models to predict time series, such as Moving Average, which includes a series of derivative models aiming at different data features. For example, EWMA(exponential weighted moving average)(Enders, 2004) can be used to simulate data with Exponential features and SARIMAX(seasonal auto regressive integrated moving average with exogeno us factors)(Gwilym, 1970) can be used to simulate data with seasonal characteristics. Since stock data have no obvious seasonal characteristics or other characteristics, people usually use the most basic ARIMA(auto regressive integrated moving average) model

to forecast stock data. Based on this idea,

However, models based on statistics have poor robustness for data. If the data do not conform to the prior assumptions of the model, the data cannot be well fitted. Therefore, when the neural network, a more general model, is put forward, people rapidly apply it on time series prediction. (Hochreiter & Schmidhuber, 1997b) proposed LSTM architecture, a advanced type of RNN. Compared to standard RNN structure, (Hochreiter & Schmidhuber, 1997b) applied three different "gate" structure to RNN to obatin better performance on long term memory. And many scholars has tested its performance on different dataset(Alaa & Mostafa, 2019)(Akbar, 2018). However, a potential problem with LSTM or RNN architectures is that the performance of LSTM or RNN architectures is limited by sequence length, especially when the input sequence is long. Attention mechanism proposed by (Bahdanau & Bengio, 2015) which firstly applied in Machine translation can effectively solve this problem. during the process of training, attention mechanisms can assist in selective learning of input sequences which breaks the limitation on a fixed length vector for traditional sequence-to-sequence structure. Based on this idea, (Qin et al., 2017b) apply it in time series prediction which lead to better result and we will try to implement this attention mechanism in this project.

In addition to improving the structure of the model, another direction of work is that people are trying to preprocess the data. By extracting more features of the data, the neural network model can be better trained. Usually, there are two ways to preprocess data. One is to use mathematical transformation on data, such as Fourier transform, which can bravely extract the frequency domain features of data. The other way is that people view the result of classic statistic model as the feature. For example, (Zhang, 2003) proposed a innovative idea of hybrid deep learning model in 2003. In his paper, Zhang considered time series data as two components, linear component and non-linear components. He has illustrated that traditional forecasting algorithm such as ARIMA is linear model and these linear method can effectively capture the linear pattern from time series, while neural network as a non-linear method is more flexible to obtain the non-linear relationship of series. This hybrid method has a wide range of applications in time series prediction, for example, (Avadhnam, 2007) applied ANN-autoregressive hybrid method to hydrologic time series forecasting using the monthly streamflow data at Colorado River at Lees Ferry, USA. and (Xu et al., 2019) introduced a ARIMA-RNN hybrid model to predict the next 30 days's water level in Taihu Lake. Also, in (Choi, 2018), LSTM is combined with ARIMA to predict stock price correlation coefficient. However, there are very limited number of paper that analyze the combination attention-based LSTM and statistic method. (Zheng et al., 2020) proposed a hybrid attention-based model that conbines two different deep neural network, LSTM and CNN. But in (Zeng & Khushi, 2020), a attention-based RNN-ARIMA is appiled to predict time series, the forex price, which can be a good guide

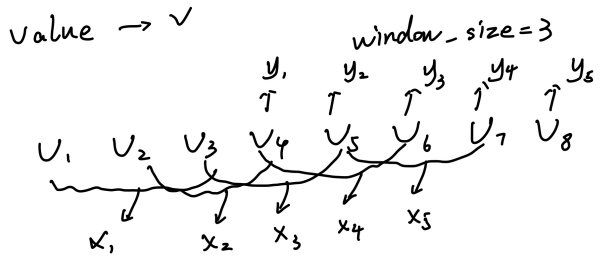

Figure 1. Sliding Window to generate the training set and output label

literature for our project.

3. Data set

In this project, our experimental data is stemed from the S&P500 dataset[1][2] [3]. The S&P data set contains 505 common stocks issued by 500 large cap companies, accounting for 80% of the U.S. stock market in terms of total market value. These stock data are representative and can reflect the running process of general stocks. In our project, we select three company AAL(American Airlines Group),ABC(AmerisourceBergen) and SP500 index from S&P500 dataset as our experimental dataset. We split these data into three groups(table 1) for baseline model. For residual tasks, we will have different size for each group which will be detailedly discussed in Section 5.1.

Abbrevation	size	Train Size	Val Size	Test Size
AAL	1260	900	100	260
ABC	1260	900	100	260
SP500Index	6755	5800	200	755

Table 1. Dataset List

We use window sliding to generate the suitable data form which will be feeded into our model. For example, We use window size = n which means that we input the stock value of the previous n time periods, the model need to predict the stock value of the n+1 time period.

4. Methodology

4.1. Autoregressive integrated moving average

Auto-regressive integrated moving average as a famous traditional statistic forecasting algorithm proposed by (Gwilym, 1970) is widely used in both industry and rapid prediction due to its flexibility and simplicity. ARIMA can be divided in to three steps:

- Integrated(I) - "Integrated" step is aiming at reducing irregular components in time series data and transform-

[1]https://en.wikipedia.org/wiki/List_of_S%26P_500_companies
[2]https://www.kaggle.com/camnugent/sandp5000
[3]https://github.com/andy971022/SP500-ARIMA

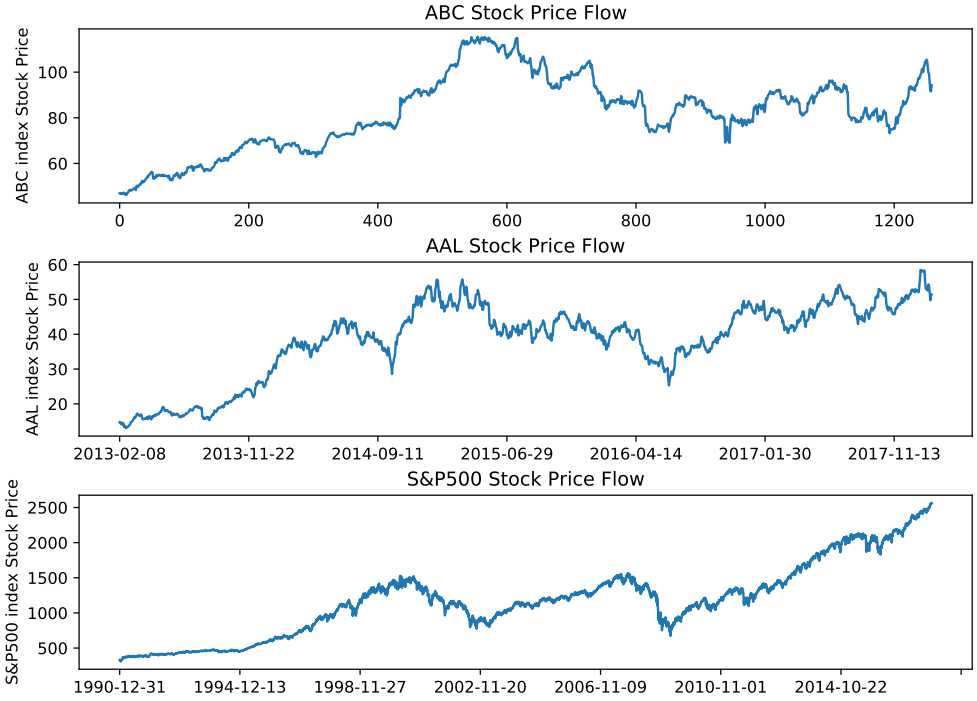

Figure 2. Raw Data,ABC,AAL,S&P500 Index

ing raw data into stationary data. The most commonly used integrated techniques is differentiating the observed series.

- auto-regression(AR) - auto regression step can be visualized as a regression process based on a number of lagged observations p, where p is called dependency length. Also, a random error generated in auto-regression process will be considered.
- Moving Average(MA) - "Moving Average" step can be visualized as a regression process based on the dependency of random error generated in AR process. Same as AR, the length of random error dependency can be defined by notation q, where q can be defined by MA order.

The ARIMA can be defined by following formula

$$X_t = c + \sum_{i=1}^{p} \theta_i X_{t-i} + \sum_{i=1}^{q} \phi_i \epsilon_{t-i} \tag{1}$$

And in the following several subsections, we will introduce the methods for determining the order of Integrated(I), auto-regressive (AR) and moving average (MA) and the method for evaluating the performance of whole ARIMA model.

4.2. Augmented Dickey–Fuller(ADF) test

Augmented Dickey–Fuller test is aiming at testing the degree of instability of a time series by measuring how much for the null hypothesis of a unit root will be rejected in a time series dataset.(Fuller, 1976). Therefore, Augmented Dickey–Fuller(ADF) test is a kind of statistical hypothesis testing.

The definition of unit root is illustrated by (Bierens, 2001) that in an autoregressive process, if the lag coeffi cient is 1 it can be called unit root, and when the unit root is present, the relationship between independent and dependent variables is deceptive.

4.3. ACF & PACF

Autocorrelation function(ACF) is a function aiming at measuring the degree of correlation between the current value of the sequence and its past value including both direct and indirect correlation information in time series(Gubner, 2006), while partical Autocorrelation function(PACF) aims at quantifying the correlation between the current residual and the next lag value.(Gubner, 2006)

In our experiment, ACF and PACF can be used to determine the order of auto-regression(p) and moving-average(q) by evaluating the ACF and PACF graph of time series. And the evaluation criteria are as follows:

	AR(p)	MA(q)	ARMA(p,q)
ACF	continuous	cut in order q	coutinuous
PACF	cut in order p	continuous	continuous

Table 2. ACF and PACF criteria

Therefore, when the graph of ACF is continuous which means there is always a non-zero value, it's not going to be constant to zero (or fluctuating randomly around zero) after k is greater than some constant, also the graph PACF is cut in order p which means the truncation is the rapid approach to 0 after constant p, this time series can be modelled by AR

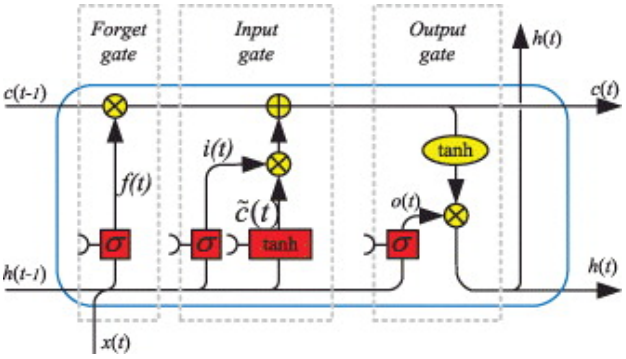

Figure 3. LSTM architecture(Yu et al., 2019)

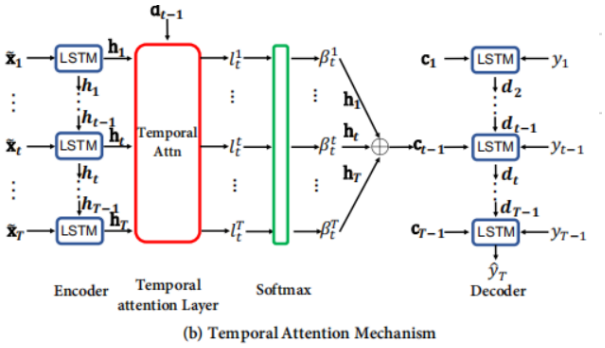

Figure 4. DARNN Decoder (Qin et al., 2017a)

with order p. The same procedure may be easily adapted to obtain behaviour models for any other time series.

4.4. Long Short-Term Memory

LSTM (long short term memory)(Hochreiter & Schmidhuber, 1997a) is a special RNN, which is mainly used to solve the problem of gradient vanishing in the process of long sequence training. LSTM use special "gates" to keep and control the state of cells, so as to avoid the problem of gradient vanishing. There are three kinds of gates in LSTM model: forget gate, input gate and output gate. It also includes cells and hidden units. It should be noted that the cell represents the update value of the cell state. The LSTM architecture can be shown in Figure 1.

Forgetting gate is mainly used to determine how much information the current model should remember about the previous cell state.

$$f(t) = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f)$$

The input gate updates the current cell state by adding the previous cell state and the current cell information.

$$i(t) = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f)$$

$$\hat{C}_t = tanh(W_{\hat{C}h}h_{t-1} + W_{\hat{C}x}x_t + b_{\hat{C}}$$

The $\hat{C}(t)$ are the updated cell state output of this cell.

$$c(t) = c(t-1)f(t) + i(t)\hat{C}(t)$$

The output gate is to output the LSTM model results of this layer to the LSTM in the next time step, so as to pass the historical information to next block. Output gate determines which parts of the cell state we want to output.

$$o(t) = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o)$$

$$h(t) = o(t)tanh(c(t))$$

4.5. Attention based LSTM

Here we use the DARNN model to apply the attention mechanism into sequence to sequence model for time series data. This DARNN was introduced in the (Qin et al., 2017a). There are total 2 stages for this model, encoder, decoder. Data: The input of this model is a driving sequence of $X = (x^1, x^2, ..., x^n)^T = (x_1, x_2, ..., x_T), X \in R^{n*T}$. There are total n driving time series data, and each time series data contain T time step data, so the length of x_n is T. x^k represent the k-th time series, x_t represent the data of n time series in the t-th time step. So the task of the model is to learn a mapping function F which satisfy the model can predict the target value y_t by given X and target outputs before time step t. The Function is shown below.

$$y_T = F(y_1, y_2, ...y_{T-1}, x_1, x_2, ..., x_T)$$

In our experiment, we do not use the encoder section, so we only introduce the decoder part as below.

4.6. ARIMA & LSTM hybrid model

Dataset	p	d	q
ABC	7	1	2
AAL	6	1	0
SP500Index	7	1	1

Table 3. Best parameters for ARIMA in different datasets

Decoder: To avoid the problem of long input sequence decreasing the performance of traditional seq2seq model, the decoder use the attention mechanism. The decoder select the hidden states from LSTM system by using attention. The decoder structure is shown in Figure 4. And the query, key, value is shown below. Query: Concrete the previous time step hidden states d_{t-1} and the cell state s'_{t-1} as the query of attention. $[d_{t-1}; s'_{t-1}]$ Key: key is the hidden state h_t Value: same as key

So we can use query and key to calculate the attention and normalize to get the attention score by softmax.

$$l_t^i = V_d^T tanh(W_d[d_{t-1}; s_{t-1}] + U_d h_i)$$

$$\beta_t^i = \frac{exp(l_t^i)}{\sum_{k=1}^{T} exp(l_t^k)}$$

Then the hidden states with attention c_t can be calculated by

$$c_t = \sum_{k=1}^{T} \beta_t^k h_k$$

Put the concreted data of c_{t-1} and y_{t-1} $[c_{t-1}; y_{t-1}]$ as the input of a linear layer. And then transfer the linear layer output into LSTM to get the d_t and c_t. At the final prediction part, we use both d_T and c_T to predict y_T.

$$\tilde{y}_{t-1} = W_d[d_{t-1}; s_{t-1}] + \tilde{b}$$

$$d_t = LSTM(d_{t-1}, \tilde{y}_{t-1})$$

$$\hat{y}_T = V_y^T(W_y[d_T; c_T] + b_w) + b_v$$

By now we have finish the prediction task as we mentioned at beginning.

$$y_T = F(y_1, y_2, ...y_{T-1}, x_1, x_2, ..., x_T)$$

The hybrid model is based on (Choi, 2018)(Zhang, 2003), all we need to do is set the output of ARIMA as the input of attention-based LSTM. We use the residual values, derived from ARIMA model of the 150 randomly selected SP500 stocks as input for LSTM model.

From (Zhang, 2003) paper, Zhang illustrated that in real life problem, a time series can be considered as being made up of linear and non-linear patterns. As the following equations shows:

$$y_t = L_t + N_t \tag{2}$$

Where y_t is the raw time series information, L_t is the linear pattern and N_t is the non-linear pattern of this time series. According to (Zhang, 2003) statements that traditional statistic model such as ARIMA, AR, MA are linear model which can much easily obtain linear pattern from a time series, while deep learning method such as RNN, LSTM are non-linear model which is more suitable for non-linear series.

$$e_t = y_t - \hat{L}_t \tag{3}$$

The raw dataset will be fitted into linear model first, and the $\hat{L}_t$ denotes the predicted result from linear model, then the residual between raw dataset and $\hat{L}_t$ will be calculated, marking as e_t.

Then, e_t from equation(3) will be considered as the input sequence of deep learning model, and the predicted result from neural network, denoting as $\hat{N}_t$ will be combined with $\hat{L}_t$.Finally we evaluate its hybrid model by comparing raw information y_t and predicted results $\hat{L}_t$+$\hat{N}_t$. And in the following section, we will introduce four evaluation methods used in our experiment.

4.7. Evaluation Methods

We compare the predicted results with the actual results by Root Mean Square Error(RMSE) and regard MSE as the cost function of the model. The formula of RMSE is as follows.Where y_i is the predicted stock value at time i,and the $\hat{y}_i$ is the practical value at time i. We use the MSE as cost function of the model and regression.

$$RMSE = \sqrt{\frac{1}{n^2}\sum_{t=1}^{n}(y_t - \hat{y}_t)} \tag{4}$$

Root Mean Squared Logarithmic Error(RMSLE) as an auxiliary evaluation method of MSE is much suitable for some scenarios where the loss of the forecast is greater which means RMSLE penalizes overvaluation more than undervaluation.

$$RMSLE = \sqrt{\frac{\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}{n}} \tag{5}$$

And another evaluation method is Mean Absolute Percentage Error(MAPE) which is used to measure the performance of model, 0% means perfect model while greater than 100% indicates a poor model. And also Mean Absolute Error(MAE) is used to measure the mean value of the absolute error between the observed value and the true value.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{\hat{y}_i - y_i}{y_i}| \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{7}$$

5. Experiments

5.1. Raw data description

As we discussed in section 3, we will use three dataset for stock series forecasting in our project gathered from company SP, as shown in 2.With respect to S&P500 Index, 6755 observations are recorded from 31th December,1990 to 19th October, 2017.The first 3000 samples are used for training ARIMA model, then use the trained ARIMA model to predict rest 3755 value. So we can get the residual value through using later 3755 raw data subtract the 3755 predicted value. So the 3755 residual values are the data for LSTM model, again, LSTM will set the previous 3000 residual values as the training data, and the rest 755 samples are used for testing. So the final 755 samples are the test set for whole hybird ARIMA and LSTM model. Similar to ABC and AAL dataset, there are only 1260 samples used in experiment and their first 500 samples are used for ARIMA training. First 500 residual values are used for LSTM training. The last 260 samples are used for hybird model predicting. The statistical information of the three datasets is calculated and shown in table 4.

5.2. Data Augmentation

According to section 4.6, the methodology of hybrid can be considered as data augmentation, which decomposes raw data into linear and non-linear components. The traditional statistic model can successfully obtain the linear pattern of time series, while neural network is better for non-linear components.

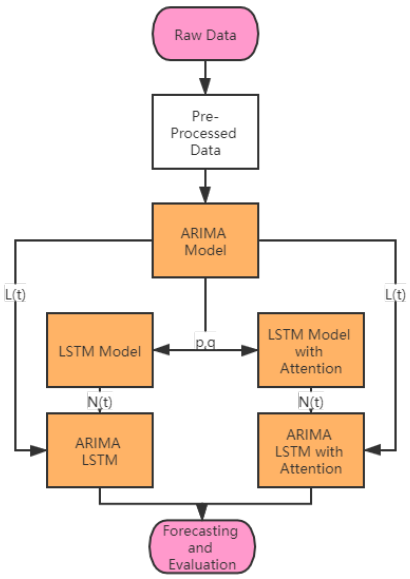

Figure 5. Hybrid ARIMA LSTM flow Chart

Time series data	Count	Mean	Min	Max	Standard derivation
ABC	1259	49.202	33.370	74.820	9.229
AAL	1259	38.393	13.020	58.470	10.957
S&P 500 Index	6755	1175.324	311.489	2562.100	517.822

Table 4. statistical information of three datasets

5.2.1. Linear prediction of ARIMA model

Our experiment is performed with the statistic.model[4]. In order to obtain the linear components and their residuals from raw data, two steps should be followed: firstly, time series data must be stationary when they are fitted into ARIMA model. We apply the differencing process to our raw data to ensure the order of integration d by doing Augmented Dickey–Fuller(ADF) test we discussed in section 4.2. Then, in order to determine the order of p and q, the minimum value of the AIC will be explored by trying different values of p and q through iteration. We determine the range of p and q by evaluaing the ACF and PACF curves as we discussed in section 4.3. And the p,d,q of ARIMA for three different dataset are in table 3.

5.2.2. Non-Linear prediction of LSTM

After the building and training of ARIMA model, we can predict the linear part of the financial data, so in this part, we build LSTM model to predict the nonlinear part of the data. LSTM model is built based on framework of python and Pytorch framework[5] We build a seq2seq model. In order to add attention mechanism more easily in the next part, this seq2seq model is built by two LSTM networks. The former LSTM is used as the encoder of the whole model, and the latter is used as the decoder. We also set

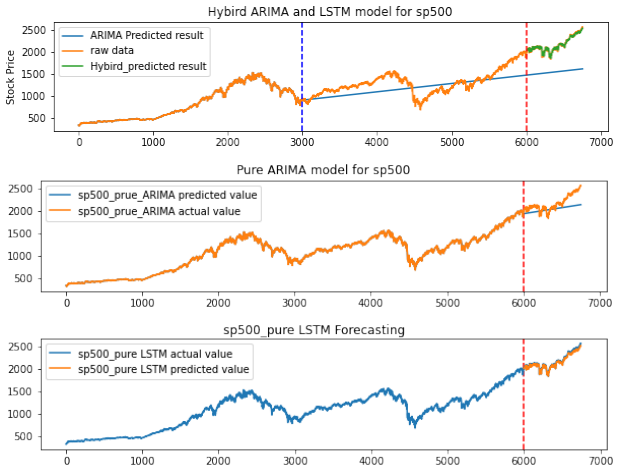

Figure 6. Results of S&P 500

the training epoch number as 50. In addition, the structures of these two LSTM models are (input_dim, hidden_size, num_ layeroutput_ dim). Input_dim is the dimension of the input data we use, because we only use the “close” price as the input and forecast future ”close” price, so both input_dim and output_dim are 1. The seq2seq model is complex enough, so we use num_layer = 1 for each LSTM. In addition, we also use Adam as our training optimizer, so that we can dynamically adjust the learning rate in the process of training. At the same time, we use MSE as the criterion during training. So in general, for SP500 stock

[4]https://www.statsmodels.org
[5]https://pytorch.org/

Dataset	Window Size	Model	MSE	RMSE	MAE	RMSLE	MAPE(%)	Epoch
sp500	**8**	**ARIMA**	32377.8	179.9	145.5	0.081	6.41	50
sp500	**8**	**LSTM**	775.4	27.8	23.7	0.0126	1.07	50
sp500	**8**	**ARIMA-LSTM**	437.4	20.9	16.3	0.0379	2.75	50
sp500	**8**	**ARIMA-LSTM Attention**	552.8	23.5	20.8	0.0417	2.74	50
sp500	**16**	**ARIMA-LSTM Attention**	415.6	20.4	16.5	0.0364	2.69	50
sp500	**16**	**ARIMA-LSTM Attention**	558.5	23.6	19.8	0.0345	3.19	100
sp500	**32**	**ARIMA-LSTM Attention**	1018.5	31.9	28.0	0.0529	4.46	50
sp500	**8**	**Multi-Feature Fusion**	35726.9	175.9	147.6	0.0815	8.4	100
ABC	**8**	**ARIMA**	53.5	7.3	6.61	0.0833	6.61	50
ABC	**8**	**LSTM**	3.2	1.8	1.2	0.0206	1.42	50
ABC	**8**	**ARIMA-LSTM**	56.9	7.5	6.2	XXX	8.26	50
ABC	**8**	**ARIMA-LSTM Attention**	21.9	4.6	3.6	XXX	4.87	50
ABC	**16**	**ARIMA-LSTM Attention**	15.5	3.9	2.9	XXX	3.99	50
ABC	**16**	**ARIMA-LSTM Attention**	6.3	2.5	1.9	XXX	2.52	100
ABC	**32**	**ARIMA-LSTM Attention**	12.5	3.5419	2.7	XXX	3.64	50
AAL	**8**	**ARIMA**	17.1	4.1	3.4	0.084	7.40	50
AAL	**8**	**LSTM**	1.4	1.1	0.9	0.0236	1.85	50
AAL	**8**	**ARIMA-LSTM**	2.8	1.6	1.3	XXX	2.94	50
AAL	**8**	**ARIMA-LSTM Attention**	2.5	1.5	1.2	XXX	2.78	50
AAL	**16**	**ARIMA-LSTM Attention**	1.7	1.3	1.01	XXX	2.25	50
AAL	**16**	**ARIMA-LSTM Attention**	1.19	1.1	0.85	XXX	1.9	100
AAL	**32**	**ARIMA-LSTM Attention**	1.7	1.3	1.012	XXX	2.21	50

Table 5. Performance of different Model(ARIMA,LSTM,ARIMA-LSTM,ARIMA-LSTM with Attention and Multi-Feature Fusi)

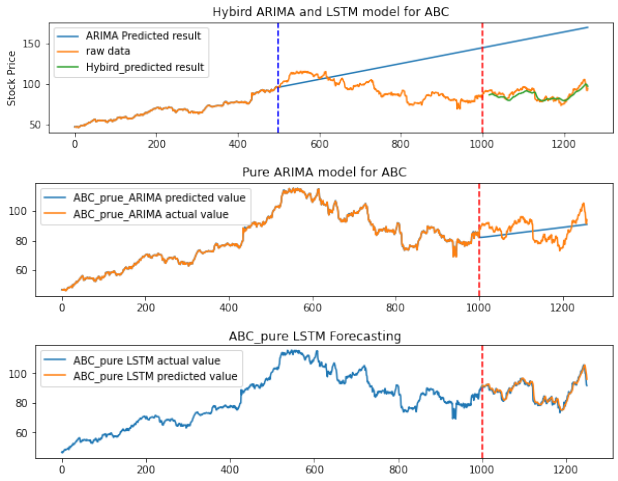

Figure 7. Results of ABC

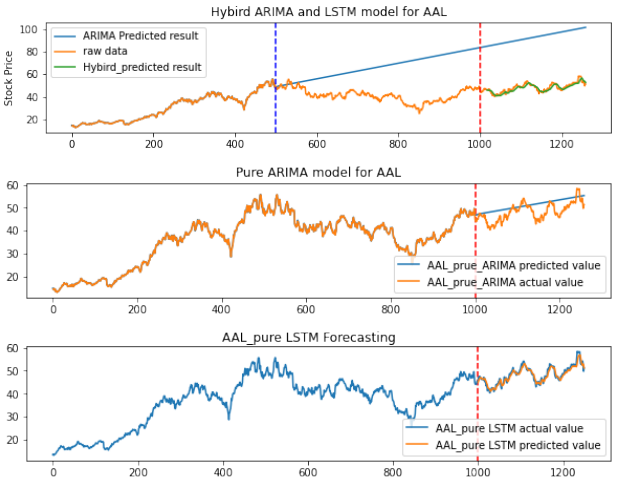

Figure 8. Results of AAL

data set, the whole model is to predict the next 745 data by training previous 6000 data.

Up to now, we are still not sure about the value of window_size and hidden_size. Window_size is that we combine the input values of window size into one input, and use this input to predict the next output value. For example, if window size = 8, we will take the first eight data as the input [0:8], and the eighth data as the label of the input. Hidden_size determines the complexity of the LSTM system. The larger the hidden size is, the stronger the learning ability of the model is. However, when the hidden size is too large, the model may also learn the noise in the data, resulting in over-fitting. In order to find the best combination of model hyper-parameters, we test window size = [4,8,16], hidden size = [2,4,8,16,32,64]. We use the results of MSE and MAE to indicate the performance of these hyper-parameters combinations, as shown in the Table 6.

From the Table 6, when window size = 8, hidden size = 32, the performance is the best, and MSE is only 1446. Therefore, we use these two values as the basic values of the later training model.

	window size	hidden size	MSE	MAE
Hidden	8	2	9816	83.27
Size	8	4	5526	78
Adjusting	8	16	2284	43
	8	32	1446	32
	8	64	3459	54
Window	16	32	3288	52
Size	10	32	1546	34
Adjusting	8	32	1446	32
	6	32	4862	64
	4	32	2381	43

Table 6. Model performance of various hyper parameters combinations

5.2.3. Improving model performance by adding attention

To improve the performance and try more methods, we applied the attention to the seq2seq model in this section dependent on the section 4.5.

Comparing with the seq2seq model mentioned in previous section, the attention seq2seq model has the same encoder structure. The newest attn_decoder is much complex than the LSTM decoder.

6. Results and Analysis

Besides the experiments mentioned before, we also add the pure-ARIMA predicting result and pure-LSTM predicting results for SP500, AAL and ABC data. The final results are listed in table 5. There are some empty evaluation results in RMSLE shown as "XXX". This is because RMSLE is calculate the log value of evaluated data, part of our residual values are negative is exmpty as "XXX" Besides, it is worth to mention that, as for SP500, the pure models will set the previous 6000 data as the training set, and the rest 755 as the test set which is different from that of hybird model. Similar for AAL and ABC data, pure models use previous 1000 data as training set and the rest 260 data as test set.

For SP500 data, from the Table 5 and Figure 6, we can clearly find that ARIMA has no particularly good performance on SP500, and the MSE even reaches to 32377. However, as we mentioned before, ARIMA can only predict the linear feature of data, so for pure-ARIMA model, we get a straight line, which is generally extended along the trend of current data. When we use the pure LSTM model to train and predict the data, we find that MSE decreases a lot, from 32377 of ARIMA to 775. This is because LSTM can learn the nonlinear part of the data, and for the whole data, the nonlinear part is the main factor that determines the amount of MSE. Therefore, in order to learn the linear and nonlinear features of the data well, we build a hybrid model of ARIMA and LSTM, and the MSE is reduced to 437, which is better than the prediction results of pure-ARIMA and pure-LSTM as expected. By combining the linear feature learned by ARIMA and the nonlinear feature of LSTM, the hybrid model can be used for better prediction. Later, when we added the attention mechanism, we found that the MSE increased. This is because our window size is too small for the attention mechanism. Although window size = 8 is a good parameter for the pure-LSTM model, the attention mechanism ignores some factors by adding the attention score to the data. When we increase the window size appropriately, window size = 16, the model becomes better as expected. MSE is reduced from 552 of windows size = 8 to 415 with even nearly 25% improvement which also proves that attention mechanism has good predicting performance for time series data. However, when the window size is increased to 32 again, the effect of the model becomes worse. We guess that the noise part of the data is added when the window size increases, so the model does not learn very well. Overall, for SP500 data, the performance of our models are generally consistent with the hypothesis. The hybrid model learns the linear and nonlinear characteristics of the data, and the attention mechanism optimizes the attention of the data with appropriate length, so as to get better performance.

For ABC and AAL models the result is shown in the Table 5, Figure 7 and Figure 8. Because of the small amount of data which is only 1260. When using the hybrid model to train and predict, only 500 training data can be obtained for ARIMA and LSTM respectively. So the data of the hybrid model is not particularly good. When we input 1000 training data to the pure LSTM model, LSTM model shows strong learning ability. The MSE of ABC is only 3.2 and that of AAL is only 1.4. But this doesn't mean that our hybrid model is not powerful, just because the amount of data is too small, and hybird model don't learn the features of the data. Because the hybird model is powerful for large data set such as the good performance in sp500. When we increase the number of training epochs of the hybrid model with the attention mechanism to 100, the data is also greatly improved which can be seen when the window size of ABC is 16, the MSE of hybird model is reduced from 15.5 to 6.28. Although it is still not as good as pure LSTM, it also proves that the hybrid model does not learn the features of the data well with a small amount of training data and a small number of training epoch.

In addition, we also apply multiple features to predict the target value. As shown in the Table 5, we introduce the values of "value" and "close" as the data features to predict the future "close" value. However, because the feature values are not very good, the model does not get very good results.

7. Conclusion and Future work

In a word, the hybrid model has good performance for large amount of data and can effectively improve the prediction performance. For example, the prediction result of SP500 data by the hybrid model under the attention mechanism is the best among various models. However, LSTM is more suitable to predict a small amount of data for AAL and ABC. However, this does not mean that the learning ability of mixed data is poor, a small number of training set limits

the model to learn good features.

In the future, we will continue to import multiple features of data into our hybird attention model, such as "value", "high", "low" and "close" of stocks to predict the future "close" values. Besides, find larger size of training data to train our model.

References

Akbar, Siami-Namini Sima Siami Namin. Forecasting economics and financial time series: Arima vs. lstm. *CoRR*, 2018.

Alaa, Sagheer and Mostafa, Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing (Amsterdam)*, 323:203–213, 2019.

Avadhnam, Jain Ashu Kumar Madhav. Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, 7, 2007.

Bahdanau, Dzmitry and Bengio, KyungHyun Cho Yoshua. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Bierens, H. J. A companion to econometric theorys. *Oxford: Blackwell Publishers. pp. 610–633. "2007 revision"*, 2001.

Choi, Hyeong Kyu. Stock price correlation coefficient prediction with arima-lstm hybrid model. *arXiv preprint arXiv:1808.01560*, 2018.

Enders, Walter. Stationary time-series models. *Applied Econometric Time Series (Second ed.). New York: Wiley. pp. 48–107.*, 2004.

Fuller, W. A. Introduction to statistical time series. *New York: John Wiley and Sons*, 1976.

Gubner, John A. Probability and random processes for electrical and computer engineers. *Cambridge University Press*, 2006.

Gwilym, Box George Jenkins. Time series analysis:forecasting and control. *San Francisco: Holden-Day*, 1970.

Hochreiter, Sepp and Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997a.

Hochreiter, Sepp and Schmidhuber, Jurgen. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997b.

Qin, Yao, Song, Dongjin, Chen, Haifeng, Cheng, Wei, Jiang, Guofei, and Cottrell, Garrison. A dual-stage attention-based recurrent neural network for time series prediction. 2017a.

Qin, Yao, Song, Dongjin, Chen, Haifeng, Cheng, Wei, Jiang, Guofei, and Cottrell, Garrison. A dual-stage attention-based recurrent neural network for time series prediction. pp. 2627–2633, 08 2017b. doi: 10.24963/ijcai.2017/366.

Xu, Guoyan, Cheng, Yi, Liu, Fan, Ping, Ping, and Sun, Jie. A water level prediction model based on arima-rnn. *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2019.

Yu, Yong, Si, Xiaosheng, Hu, Changhua, and Zhang, Jianxun. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31 (7):1235–1270, 2019.

Zeng, Z. and Khushi, M. Wavelet denoising and attention-based rnn- arima model to predict forex price. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020. doi: 10.1109/IJCNN48605.2020.9206832.

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

Zheng, H., Lin, F., Feng, X., and Chen, Y. A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020. doi: 10.1109/TITS.2020.2997352.