Financial Time Series Forecasting

Yiqing Liu, *yq.liu@rutgers.edu*
Wenzhe Zhang, *wenzhe.zhang@rutgers.edu*

***Abstract*—Stock price forecasting is a hot and important topic. This project focuses on comparing different financial time series models. Our dataset is monthly Apple stock adjusted price and models covered SARIMA, LSTM, and SVM. Results show that SVM method achieves better forecasting accuracy than the other two methods.**

Keywords*—*Time Series, Forecasting, SARIMA, LSTM, SVM.

I. Introduction

Financial time series are related to various economic and business data, such as stock price, exchange rate, housing price, etc. Forecasting is tracking of a particular variable over time. The prime goal of financial time series forecasting is to provide reliable information about future which is crucial for planning or dealing with uncertainty. Financial time series forecasting is considered as one of the most challenging problems because financial time series are naturally noisy and non-stationary [1].

The seasonal autoregressive integrated moving average (SARIMA) model is a useful method to account for seasonal and nonstationary time series[2]. Despite the simplicity and notable forecasting accuracies of SARIMA models, the main drawback is the inherent linear form. It fails to capture any nonlinearity which may be present in a financial time series.

The artificial neural networks (ANN) have been widely used for predicting financial markets because they are capable of detecting and reproducing linear and nonlinear relationships[3]. Traditional neural networks cannot use its previous information to inform later ones. Lack of memory has poor performance when working with anything related to time series. Recurrent neural networks (RNN) address this issue. They are networks with loops in them, allowing information to persist. The Long Short-Term Memory Network (LSTM) is a type of recurrent neural networks which are powerful and popular for time series forecasting.

Support vector machines (SVM) are promising methods for prediction of financial time series because they use a risk function consisting of the empirical error and a regularized term which is derived from the structural risk minimization principle. The solution of SVM may be global optimum, so overfitting is unlikely to occur with SVM[4].

II. Structure

Our paper has six sections. The first section includes the introduction and background of our project. The second section holds a detailed description of the dataset. The third section involves selection of models and detailed description of each the model we generated to derive the final model. The fourth section compares results. The next section summarizes the model. Finally, the last section includes conclusions.

III. Data

In this paper, we used Apple stock monthly price time series from Jan 1981 to March 2017. From the figure 1,

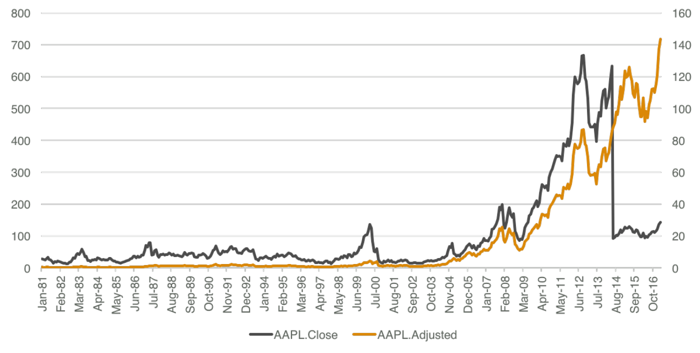


Fig. 1. Apple Stock Price

the closing price has several sudden drops because of the stock split. The adjusted price is more continuous than closing price. Thus we used this price in further study. The variance of time series grew with time, so we did log transformation for data.

We divided the dataset into the training set (Jan 1981 to Dec 2015) and testing set (Jan 2016 to Mar 2017).

IV. Models

A. *SARIMA*

We fitted SARIMA model to time series data by plotting the data, transforming the data, identifying the dependence orders of the model, parameter estimation, diagnostics, and model choice.

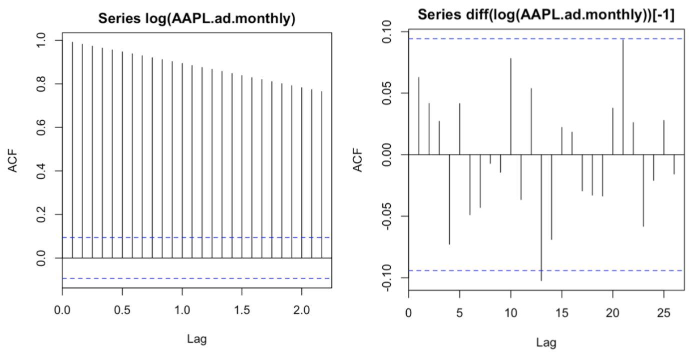
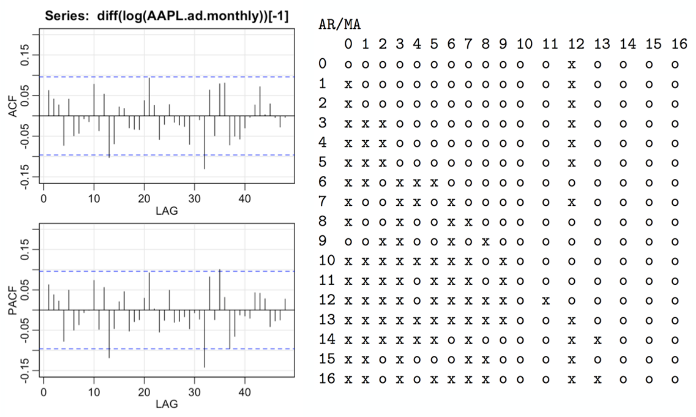


Fig. 2. Sample ACF

By log transformation, the variance of data is stable. Then the sample ACF which decaying very slowly indicates that differencing is needed. Therefore, d should be 1.



```
AR/MA
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
0  o o o o o o o o o o o  o  x  o  o  o  o
1  x o o o o o o o o o o  o  x  o  o  o  o
2  x o o o o o o o o o o  o  x  o  o  o  o
3  x x x o o o o o o o o  o  x  o  o  o  o
4  x x x o o o o o o o o  o  x  o  o  o  o
5  x x x o o o o o o o o  o  x  o  o  o  o
6  x x o x x x o o o o o  o  o  o  o  o  o
7  x o o x x o x o o o o  o  x  o  o  o  o
8  x o o x o o x x o o o  o  o  o  o  o  o
9  o o x x o o x o x o o  o  o  o  o  o  o
10 x x x x x x x x o x o  o  o  o  o  o  o
11 x x x x o x x x o x o  o  o  o  o  o  o
12 x x x x o x x x x x o  x  o  o  o  o  o
13 x x x x x x x x x x o  o  x  o  o  o  o
14 x x x x x x o x o o o  o  x  x  o  o  o
15 x x o x o o o x x o o  o  x  o  o  o  o
16 x x o x o x x x x o o  o  x  x  o  o  o
```

Fig. 3. Sample ACF, PACF, and EACF

The next step is to look at the sample ACF, PACF, and EACF of $\nabla^1 x_t$. There are no clear indicates what order should be used, but we could know that we can try low orders. Also, time series have seasonality, so we should try Seasonal ARIMA model.

TABLE I. SARIMA Model

Model	AIC	RMSE	MAE
ARIMA(1,1,0)(1,1,0)[12]	-308.72	0.1636	0.1205
ARIMA(1,1,0)(2,0,0)[12]	-490.59	0.1358	0.1010
ARIMA(2,1,0)(1,0,0)[12]	-490.84	0.1357	0.1008
ARIMA(1,1,1)(0,0,1)[12]	-490.88	0.1357	0.1007
ARIMA(1,1,0)(2,0,1)[12]	-491.69	0.1349	0.1002
ARIMA(1,1,0)(1,0,0)[12]	-492.31	0.1358	0.1010
ARIMA(1,1,0)(1,0,1)[12]	-493.26	0.1350	0.1004
Mean	-465.47	0.1395	0.1035

We tried seven different models with low orders, Table I. Selecting model with smallest AIC, ARIMA$(1,1,0) \times (1,0,1)_{12}$ is the best model. Then, we plot the residuals and perform Ljung-Box test to residuals, Figure 4. Most residuals are random, ACF are clean, and p values are higher than 0.05, so residuals could be regarded as white noise. Therefore, ARIMA$(1,1,0) \times (1,0,1)_{12}$ is final SARIMA model.

In the last, we used ARIMA$(1,1,0) \times (1,0,1)_{12}$ to get rolling forecast values on both training and testing set.

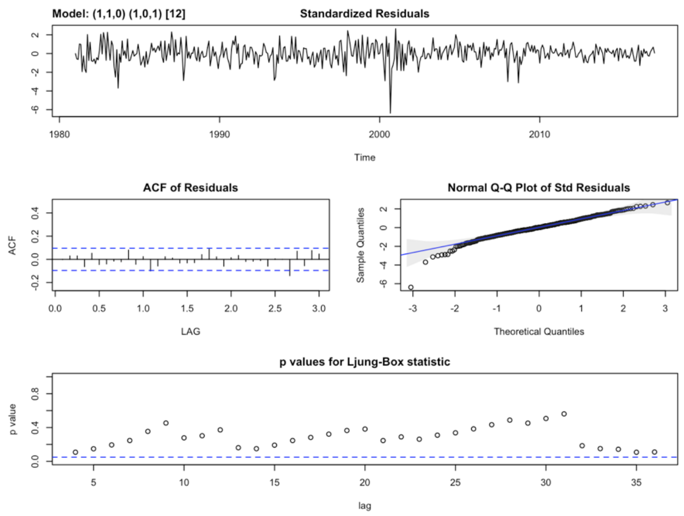


Fig. 4. Diagnostics

B. *ANN*

Traditional artificial neural networks (ANN) cannot use its previous information when dealing with time series. Figure 5. It has no notion of order in time, and the only input it considers is the current input.

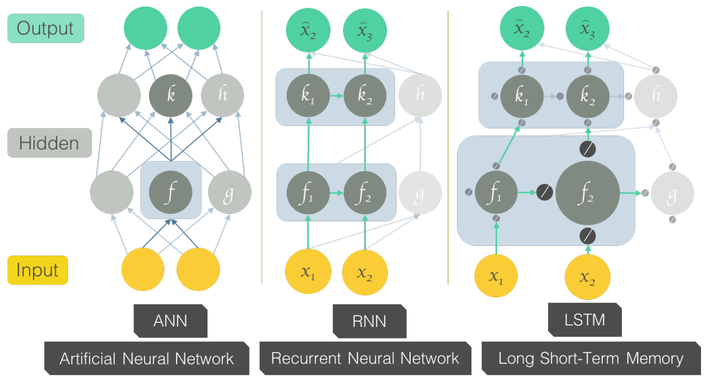


Fig. 5. ANN, RNN, LSTM

Recurrent neural networks (RNN) are designed to handle sequence dependence. It takes as their inputs, not just the current input data, but also what model perceived one step back in time. That sequential information is saved in the recurrent networks hidden state, which manages to span many time steps as it cascades forward to affect the processing of each new data. But the RNN only can use information from the last step, short memory, and cannot control whether use it or not.

The Long Short-Term Memory Network (LSTM) is a particular kind of RNN which has achieved impressive success in a wide range of sequence modeling task. LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell, much like data in a computers memory. The cell makes decisions about what to store, and when to allow reads, writes, and erasures, via gates that open and close[5].

We choose lag 1 as input label and one LSTM layer to training model.

C. SVM

1) Introduction: Support Vector Machine (SVM) constructs a linear model to implement nonlinear class boundaries through the transforming the inputs into the high-dimensional feature space. [4]

$$y = b + \sum \alpha_i y_i K(\mathbf{x}(i) \cdot \mathbf{x}) \tag{1}$$

where y_i is the class value of training example x(i), $\cdot$ represents the dot product. b and α_i are parameters that determine the hyperplane. $K(\mathbf{x}(i) \cdot \mathbf{x})$ is defined as the kernel function. SVM is known as the algorithm that finds a special kind of linear model, the maximum margin hyperplane.

2) Applications of SVM in financial time-series forecasting: As we know, neural networks have been broadly used in the area of financial time series forecasting, so we can apply SVM to our data based on its model to get the fitted values of the non-linear boundary. There are mainly three steps to implement our SVM model:

Kernel selection: Kernel method is used to map the data into higher dimensional spaces, hoping that the data could be more easily separated. Generally, before fitting the model, we don't know what exact kernel method to use. However, prior knowledge shows that some of the classical kernels may have good performance. Therefore, we try three most commonly used ones[6]:

Gaussian Radial Basis Function:

$$K(x, y) = exp(-\frac{1}{\delta^2 (x-y)^2})$$

where δ^2 is the bandwidth of the Gaussian radial basis function kernel.

Linear Kernel:

$$K(x, y) = x^T y + c$$

where c is a regularisation parameter, which controls the trade-off between achieving a low error on the training data and minimizing the norm of the weights

Polynomial Kernel:

$$K(x, y) = (1 + xy)^d$$

where d is the degree.

Lag decision: When we use SVM to fit time series model, we set y_t as our dependent variable and some lag terms as our independent variables. After trying lags from 1 to 12, we find the lag=1 term builds the best model:

$$y_t = b + \alpha y_t K(y_{t-1}, y_t)$$

Parameters: In our model, we use default parameters at first. For Gaussian Kernel, $\delta^2 = 10$ which also means $\gamma = 1\frac{1}{\delta^2} = 0.1$. For Linear Kernel, we set C=1000. For

Polynomial Kernel, we set degree d=2.
Then we make some adjustments to these parameter values to tune the model. However, we find the results are very close and all good enough, so we finally decide to use the default values.

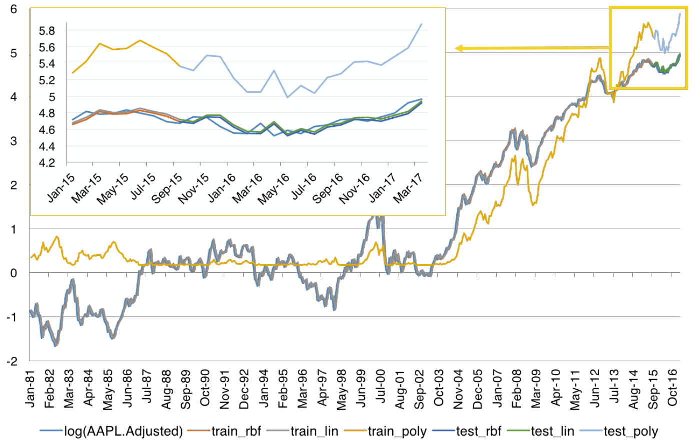


Fig. 6. SVM comparison

3) Comparisons: We can see in the plot; the polynomial model is far from the true value on both datasets whereas the other two kernels are very close. We can see linear kernel and Gaussian kernel have very similar curves on the testing dataset and both perform well. However, we have to decide the best one to compare with the other two models in the next step. After checking RMSE and MAE, we determine the Linear Kernel model to be the best SVM model.

V. Results

Here comes our final comparison as well as our project target, which aims to compare different models and determine the best one:

TABLE II. Error

	Method	SARIMA	LSTM	SVM: Linear
RMSE	Train	0.1881	0.1520	0.1387
	Test	0.1558	0.0837	0.0733
MAE	Train	0.1505	0.1153	0.1037
	Test	0.1307	0.0654	0.0592

- From the plot, we can see that SARIMA's fitted values are higher than the actual values, which may indicate it's overestimated.

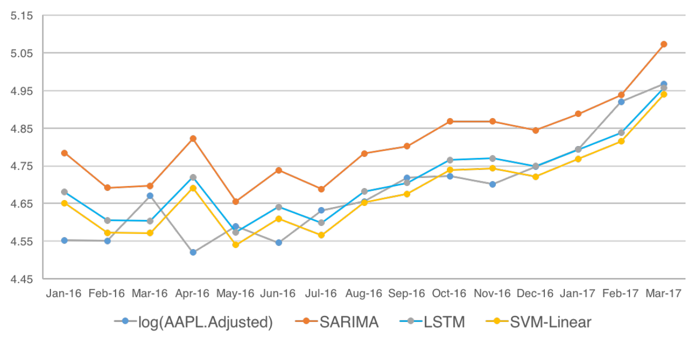


Fig. 7. SARIMA, LSTM, and SVM comparison

- The two neural network models are very close and very alike in shape. They both perform very well on the test dataset.
- For the reason that we can't decide whether LSTM has better performance than SVM or not, we have to check their errors. According to the table, we determine that SVM with linear Kernel is the best model among the three.

VI. Summary

A. Data analysis

After comparing some datasets, we finally decide to select AAPL stock price. In order to test the model as well as to detect the overfitting problem, we separate the dataset into training and testing parts. Then we take a log transformation of original data because it shows an exponential trend.

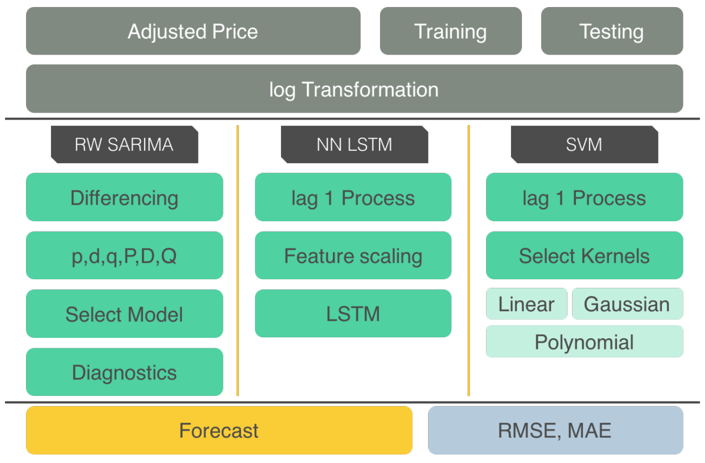


Fig. 8. Summary

B. Models

1) RW(SARIMA): According to the slow decay speed of acf, we take first order difference. By acf, pacf, eacf, we want to decide model orders (p,d,q,P,D,Q). We tried a variety of possible models and calculated their AICs, from which we select the best model. After deciding our final SARIMA model, we make residual analysis through time.

2) LSTM: When applying neutral network to time series models, we tried lags from 1 to 20 and decide lag=1 variable to be our independent variable for its model's best performance. After adjusting the data by normalizing it, which is also known as feature scaling, it's proper and convenient to fit the model using python package sklearn with default parameters.

3) SVM: SVM is another try we take for the neural network model. First, we choose Kernel functions according to prior knowledge. Second, we decide lag=1 be our independent variable just the same way we dealt with LSTM. Finally, we fit the model using python package sklearn with default parameters.(Here we tried to adjust the parameters, however, we find there's no significant differences, so we just use the default values.)

C. Forcast and comparison

We use models we have built on training dataset to make a forecast based on both training and testing datasets to get our fitted values.

By comparing three models' RMSE and MAE on the testing dataset, we decide the best model.

VII. Conclusions

A. Performance(Accuracy)

SVM(linear kernel) has the best performance while SARIMA is overfitted, which is the worst among the three. Overall, neural network models have better accuracy.

B. Implementation

Neural networks are easier to perform than SARIMA models because when building SARIMA, we have to make some judgments and diagnostics while machine learning needn't.

C. Interpretation

ARIMA has its advantage in interpretation, its parameters compared to Neural Network models, which are always black boxes. The weighting coefficients of the Network can hardly be explained.

D. Overview

In all, our project proves that when the dataset is not very large, and dimension is not very high, Neural Network models have excellent performance both in accuracy and efficiency. However, when it comes to big data, machine learning may require good hardware to implement, and the training process may cause a much longer time, which cannot be so efficient. Therefore, if we want to look further into this topic, we have to deepen our network and enlarge our dataset both in quantity and dimension. Also considering some other classical time series model like ARMA+GARCH, TAR, etc., in this way, we might be able to reach our conclusion.

Acknowledgment

We would like to thank Professor Han Xiao. His guidance along with his unconditional support towards making this paper original proved to be an asset.

References

[1] R. Adhikari and R. Agrawal, "A combination of artificial neural network and random walk models for financial time series forecasting," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1441–1449, 2014.

[2] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2010.

[3] L. Bodis, "Financial time series forecasting using artificial neural networks," *Mestrado–Babeş-Bolyai University*, 2004.

[4] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[5] C. N. AdamGibson, "A beginner's guide to recurrent networks and lstms," 2017, [Online; accessed 30-April-2017]. [Online]. Available: https://deeplearning4j.org/lstm.html

[6] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, no. 4, pp. 309–317, 2001.