US Stock Market Data Analysis, Visualization and Prediction

Kuntel Patel
University of North Texas
kuntalpatel22@gmail.com

Zhaochen Gu
University of North Texas
ZhaochenGu@my.unt.edu

Srikala Murugan
University of North Texas
msrikala87@gmail.com

***Abstract*— In this data mining project, our group analyze the stock market data and predict the stock price using a time series based algorithm called LSTM. Finance market is very influential in business, economy and affects our life significantly. However, the data varies drastically and it requires fast trading decision. Without computer, we cannot do a better work. As the machine learning algorithm improved life quality from all kinds of side. We wish can use this new methods to solve some problems that are traditional but closely related to our daily lives. So, we decide to use machine learning algorithm to predict us stock market data. We hope our method can help more people to make a better decision on stock selection and investment.**

I. INTRODUCTION

All publicly known information about a company, which obviously includes its price history, would reflected in the current price of the stock. The successful prediction of a stock's future price could yield significant profit. The main goal of our work is to predict future price of the stock by using historical information and informative Data visualization. With the help of Recurrent neural network (RNN), specifically Long short-term memory (LSTM) we have implemented predictive model which uses Adam optimizer with default parameters.

Our work is divided into four parts:

1) Data collection: To check performance of Algorithm on data from different time frames we have tried to scarp data directly from Google finance. But because of API limitations we could not able to scrap data using web scraping method. Instead we have manually downloaded data from direct website from 2010 to 2017 date range. These data was in raw format which has some extra rows with Dividend, Split, and Merger information. For this project we have discarded those rows and made CSV file which contains only price and volume information about total 31 stocks.
2) Data prepossessing: There are some missing values which we have filled using back-fill method. Also dropped some values which was not possible to back-fill.
3) Data visualization: For initial data exploration we have used WEKA tool. With the help of Seaborn library and matplotlib library we have visualized top 5 best vs top 5 worst stocks and correlation between all 31 stocks.
4) Future price prediction: We have implemented prediction method for Apple stock based on data from 2010 to 2016 to predict 2017 stock price. We have studied related scholarly articles and referred some technical blogs to decide which method will be preferred method for prediction. Stock price prediction is temporal series kind of problem which can be solved by the method which can infer future stock price after learning from historical data and we have found LSTM network method would be interesting to experiment.

II. RELATED WORK

A. Stock Market's Price Movement Prediction with LSTM Neural Networks [1]

This article studies the usage of LSTM networks to predict future trends of stock prices based on the price history, alongside with technical analysis indicators. For that goal, a prediction model was built, and a series of experiments were executed and theirs results analyzed against a number of metrics to assess if this type of algorithm presents and improvements when compared to other Machine Learning methods and investment strategies. The results that were obtained are promising, getting up to an average of 55.9% of accuracy when predicting if the price of a particular stock is going to go up or not in the near future.

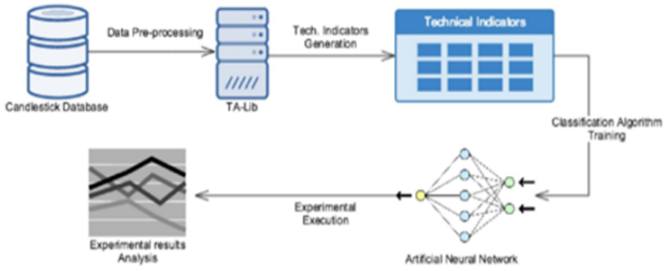


Fig. 1. Perform predictions of stock price based on LSTM networks

B. Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes [2]

In this research paper a novel model was implemented to combine real-time wavelet denoising functions with the LSTM to predict the East Asian stock indexes in which the wavelet denoising adopts a sliding window mechanism to exclude the future data while its system configuration is flexibly optimized based on some predefined criteria. The empirical results reveal that the performance of our proposed prediction model shows significant improvements when compared to those of the original LSTM model without utilizing the wavelet denoising function.

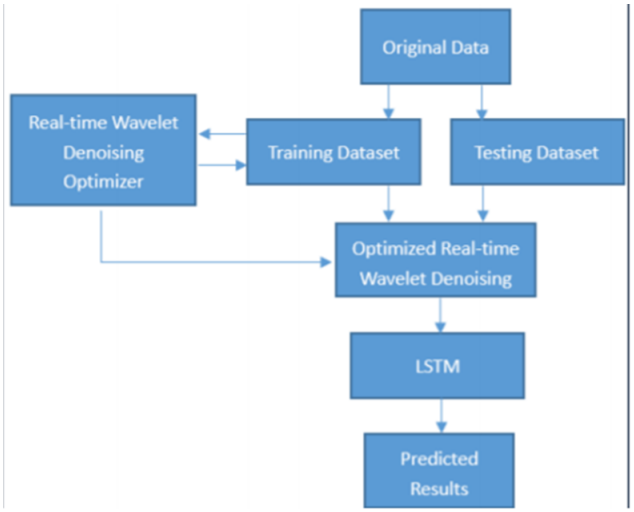


Fig. 2. Framework of Predicting Indexes

C. Sentiment-aware stock market prediction: A deep learning method [3]

This research paper proposes a new method for stock market prediction, which adopts the Long Short-Term Memory (LSTM) neural network and incorporates investor sentiment and market factors to improve forecasting performance. By extracting investor sentiment from forum posts using Naive Bayes, this paper makes it possible to analyze the irrational component of stock price. Their empirical study on CSI300 index proves that our prediction method provides better prediction performance. It gives a prediction accuracy of 87.86%, outperforming other benchmark models by at least 6%. Furthermore, their empirical study reveals evidence that helps to better understand investor sentiment and stock behaviors. Finally, this work also shows the potential of deep learning financial time series in the presence of strong noises.

III. METHODOLOGY

A. Dataset Preprocessing

After studying above related work and few blogs and articles we have decided to experiment LSTM method on real stock values. The data can be collected using the pandas_datareader package which fetches data from

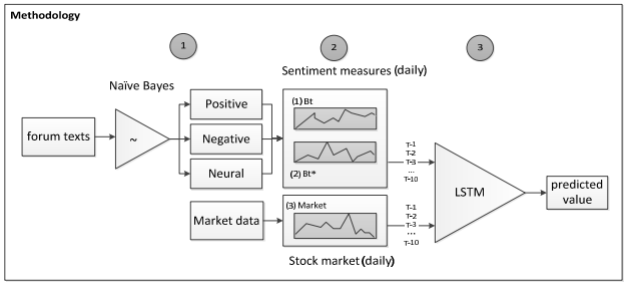


Fig. 3. Three Phases of Building Stock Market Price Prediction Model

Google Finance API.Since large amount of financial data is not possible to scrap any more we have to run all my experiments on existing dataset only. Dataset which we are using can be downloaded from Google finance website. Our final dataset is in CSV format. Python libraries we have used are pandas, numpy, matplotlib. APIs we have used are seaborn, Keras and Tensorflow. Dataset has 7 columns with information about any particular day opening price, closing price, Peak price, Lowest traded price, Number of shares traded and, trading date.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62396 entries, 0 to 62395
Data columns (total 7 columns):
Date      62396 non-null datetime64[ns]
Open      62371 non-null float64
High      62386 non-null float64
Low       62376 non-null float64
Close     62396 non-null float64
Volume    62396 non-null int64
Name      62396 non-null object
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 3.3+ MB
```

```
df.describe()
```

	Open	High	Low	Close	Volume
count	62371.000000	62386.000000	62376.000000	62396.000000	62396.000000
mean	100.730692	101.522927	99.909877	100.751459	16484316.887188
std	125.632592	126.554663	124.558485	125.588264	24400156.581981
min	11.300000	11.800000	11.090000	11.090000	0.000000
25%	39.640000	40.000000	39.300000	39.650000	4378555.000000
50%	72.300000	72.880000	71.735000	72.350000	8365206.000000
75%	108.540000	109.300000	107.690000	108.622500	18139611.750000
max	1204.880000	1213.410000	1191.150000	1195.830000	618237630.000000

There are some days for which we have missing values because for some reasons Google Finance API was down and did not show trading prices for one or more attributes. Below screenshot will show list of the days where we had some missing values. Since stock market trading only runs from Monday to Friday we are only calculating working Business days.

```
#The number of  days for which the records are missing.(freq = 'B')

missing_Days = pd.date_range(start='2010-01-01', end='2018-01-01', freq='B')

#Reference  https://wiki.python.org/moin/BitwiseOperators
missing_Days[~missing_Days.isin(df.Date.unique())]


#print(len(missing_Days[~missing_Days.isin(df.Date.unique())]))
```

```
DatetimeIndex(['2010-01-01', '2010-01-18', '2010-02-15', '2010-04-02',
               '2010-05-31', '2010-07-05', '2010-09-06', '2010-11-25',
               '2010-12-24', '2011-01-17', '2011-02-21', '2011-04-22',
               '2011-05-30', '2011-07-04', '2011-09-05', '2011-11-24',
               '2011-12-26', '2012-01-02', '2012-01-16', '2012-02-20',
               '2012-04-06', '2012-05-28', '2012-07-04', '2012-09-03',
               '2012-10-29', '2012-10-30', '2012-11-22', '2012-12-25',
               '2013-01-01', '2013-01-21', '2013-02-18', '2013-03-29',
               '2013-05-27', '2013-07-04', '2013-09-02', '2013-11-28',
               '2013-12-25', '2014-01-01', '2014-01-20', '2014-02-17',
               '2014-04-18', '2014-05-26', '2014-07-04', '2014-09-01',
               '2014-11-27', '2014-12-25', '2015-01-01', '2015-01-19',
               '2015-02-16', '2015-04-03', '2015-05-25', '2015-07-03',
               '2015-09-07', '2015-11-26', '2015-12-25', '2016-01-01',
               '2016-01-18', '2016-02-15', '2016-03-25', '2016-05-30',
               '2016-07-04', '2016-09-05', '2016-11-24', '2016-12-26',
               '2017-01-02', '2017-01-16', '2017-02-20', '2017-04-14',
               '2017-05-29', '2017-07-04', '2017-09-04', '2017-11-23',
               '2017-12-25', '2018-01-01'],
              dtype='datetime64[ns]', freq=None)
```

We have used Backfill method to fill these missing values. If any value is missing then we propagated previous day value and copy into missing places.

```
values = np.where(df['2017-07-31']['Open'].isnull(), df['2017-07-28']['Open'], df['2017-07-31']['Open'])
print(values)
df['2017-07-31']= df['2017-07-31'].assign(Open=values.tolist())
```

After handling all missing values checked there are no missing values. Please see the screen shot below:

```
#no missing values any more.
df.isnull().sum()
```

```
Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
Name      0
dtype: int64
```

B. Data Exploratory analysis

In order to do an exploratory analysis for our dataset, we need extra features to do analysis. We have created extra column in data frame which is average of all 'High', 'Low', 'Open' and 'Close'. In Dataset main information is within Price columns, but to consider cumulative effect of all, average price is the proper new measure on which we can do data analysis. We have studied how long term growth relationship can be plotted and found Cumulative Compound Growth will be other extra column ,based on which we can plot five worst vs best performed stocks from 2010 to 2017 among 31 stocks. New data frame have two extra columns named AvgPrice and Growth.

	Date	Open	High	Low	Close	Volume	Name	AvgPrice	Growth
0	2010-01-04	83.09	83.45	82.67	83.02	3043663	MMM	83.06	0.00
1	2010-01-05	82.80	83.23	81.70	82.50	2849586	MMM	82.56	-0.01

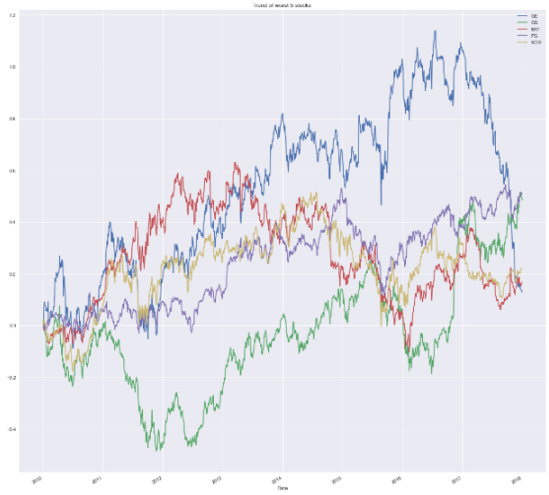
Next, we will rank these records based on the growth rate for each company. By sorting Growth values we could easily visualize which top five stocks are giving High returns compared to top five worst performers. According to results GE,IBM, XOM, GS and PG are worst performers and AMZN, UNH, HD , AAPL and BA are best performers. Here are our implantation

```
Worst_Stocks = df[df.Date == df.Date.max()].sort_values('Growth').head()

Best_Stocks = df[df.Date == df.Date.max()].sort_values('Growth', ascending=False).head()
```

Here are the performance of selected five worst stocks market:

```
Name
GE      AxesSubplot(0.125,0.2;0.775x0.68)
GS      AxesSubplot(0.125,0.2;0.775x0.68)
IBM     AxesSubplot(0.125,0.2;0.775x0.68)
PG      AxesSubplot(0.125,0.2;0.775x0.68)
XOM     AxesSubplot(0.125,0.2;0.775x0.68)
Name: Growth, dtype: object
```



The following are the graph for our five best stocks' performance:

```
Name
AAPL    AxesSubplot(0.125,0.2;0.775x0.68)
AMZN    AxesSubplot(0.125,0.2;0.775x0.68)
BA      AxesSubplot(0.125,0.2;0.775x0.68)
HD      AxesSubplot(0.125,0.2;0.775x0.68)
UNH     AxesSubplot(0.125,0.2;0.775x0.68)
Name: Growth, dtype: object
```

C. Stock Correlation and Data Visualization

1) LSTM Simple Model Predict Stock Price: To make balanced portfolio it is important to have good mixture of stocks. Price variation of stocks is unpredictable but there exists always some kind of positive or negative price relationship among same sectored stocks. To visualize this we have plotted relationship among all 31 stocks in heat map and found IBM is negatively correlated and hence interesting to study its company profile and Business model. In this heat map lighter color indicates positive correlation.

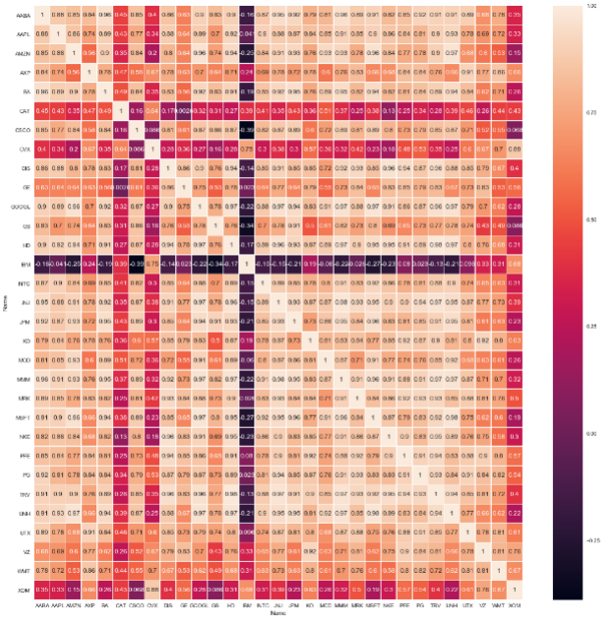


Fig. 4. Heat map of 31 stocks' correlations

D. Stock Market Price Prediction

First we have implemented simple predictive model to check can it predict general trend! As a input of LSTM neural network we are giving AvgPrice column data and predicting for Apple stock. As a training data we have considered 2010 to 2016 date range and for testing data 2017 year. After data scaling data can be fed into LSTM network.

1) LSTM Simple Model Predict Stock Price:

```
Training data size:  1761
Data from 2010 to 2016:[ 30.51    30.6375  30.4075 ... 117.125  116.6725
116.275 ]
Testing data size:  251
Data of 2017:[115.76   116.0325 116.3    117.33   118.5775 118.89   119.2
55  118.915
```

Simple model has a defined framework which consists of Defining network, Compiling network, Fitting data to network, Evaluating network and make predictions.

```
simplemodel.summary()
```

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 10)	480
dense_3 (Dense)	(None, 1)	11

```
Total params: 491
Trainable params: 491
Non-trainable params: 0
```

	loss
0	0.360730
1	0.069860
2	0.022819
3	0.011801
4	0.005271
5	0.002299
6	0.001516
7	0.001417
8	0.001387
9	0.001354
10	0.001351
11	0.001314
12	0.001299
13	0.001265
14	0.001274
15	0.001223
16	0.001197
17	0.001185
18	0.001150
19	0.001159

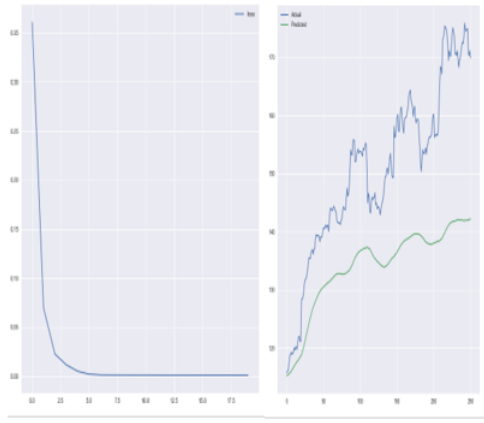


Fig. 5. LSTM Simple Model applied on Apple Stock

Simple model was predicting general upward trend for Apple in year 2017, but it is still far away from actual prices. For these reason we have implemented Complex model with 4 interactive RNN layers. We have found quite satisfactory results after these changes. Complex model can predict actual price on few occasions which means in year of 2017 if we trade only few days based on intersection points of actual price and predicted price we can get good returns. To predict stock prices for stock other than Apple need to modify code accordingly.

2) LSTM Complex Model Predict Stock Price:

```
complexmodel.summary()
```

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 30, 50)	10400
lstm_3 (LSTM)	(None, 30, 50)	20200
lstm_4 (LSTM)	(None, 30, 50)	20200
lstm_5 (LSTM)	(None, 50)	20200
dense_2 (Dense)	(None, 1)	51

```
Total params: 71,051
Trainable params: 71,051
Non-trainable params: 0
```

IV. Conclusion

In this project, we developed a model to predict the stock market price among the U.S. We selected thirty one stocks from various type of companies. We collected and processed data set from online stock historical data site. We handling

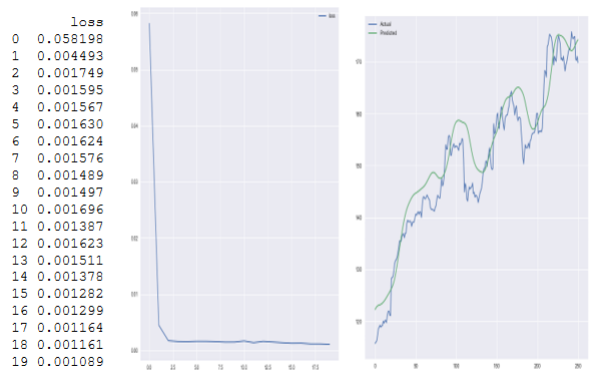


Fig. 6. LSTM Complex Model applied on Apple Stock

the missing values and add two new features, average price and growth rate, to our formatted data set for predicting easily. Then we explored the data set and discovered five best and five worst rank stock sectors which can gave us the side feedback about which stock is suggested to buy and which is not. Then, we visualize the result for us to easily see the tendency of each of the stocks among the last 6 years. In addition, we calculate the correlation among each stocks. Finally, we predict the stock price using LSTM algorithm. In our experiment, we chose the stock of Apple Inc. as our test case and we can see the the LSTM complex perform better on prediction than the LSTM simple mode.

V. Future Plan

Our future plan involves three aspects:

- Test using different optimizer such as stochastic gradient descent, RMSprop, Adagrad, Adadelta, Adamax, Nadam, TFOptimizer
- Change the parameter while apply the test model. We have been used the default parameter for applying the LSTM to predict the stock price. However, if we can test using various parameters, we would be able to compare the different result and see which one has the best performance for our data set, or in general. That would be a model selection and optimizing issue.
- Test different model. We currently using the LSTM model to predict the result. Even though most of the related paper has showed the better result of predicting time series based data. However, there are more new methods that was not used for predicting finance data before, however, it performed well in other field of prediction like ARIMA. We expect to gain more comparison and pros and cons about different algorithm on stock price prediction

References

[1] D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 1419-1426.

[2] Z. Li and V. Tam, "Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1-8.

[3] Jiahong Li, Hui Bu and Junjie Wu, "Sentiment-aware stock market prediction: A deep learning method," 2017 International Conference on Service Systems and Service Management, Dalian, 2017, pp. 1-6.