Fall 2019

Stock Prediction with Random Forests and Long Short-term Memory

Shangxuan Han

Stock Prediction with Random Forests and Long Short-term Memory

by

Shangxuan Han

A report submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Joseph Zambreno, Major Professor

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to my major professor Dr. Joseph Zambreno for his professional guidance and comprehensive support throughout the whole course of this report.

In addition, I would also like to thank my friends, colleagues, the department faculty and staffs for making my time at Iowa State University a wonderful experience. I want to also offer my gratitude to my parents for their mental and material support on my decade's academic path.

ABSTRACT

Machine learning as a popular computer science area has been promoted and developed for more than two decades. It has been applied in many fields in our life, like domestic products such as Alexa from Amazon, photographic products such as Mavic from Dji and so many other areas. This report represents an interesting way to apply machine learning and deep learning technologies on the stock market. We explore multiple approaches, including Long Short-Term Memory (LSTM), a type of Artificial Recurrent Neural Networks (RNN) architectures, and Random Forests (RF), a type of ensemble learning methods. The goal of this report is to use real historical data from the stock market to train our models, and to show reports about the prediction of future returns for picked stocks.

Keywords: machine learning, deep learning, artificial neural network, long short-term memory, random forests, ensemble learning.

CHAPTER 1. INTRODUCTION

1.1 Background

Since the development of Artificial Intelligence (AI) technology during recent decades, Artificial Intelligence has been deployed on many areas, such as computer vision algorithms, natural language understanding, and so on. Stock forecasting is also one hot topic. Making stock transaction decisions is a generally an experiential process. This makes the prediction on future stock prices difficult when using traditional statistics technology, because of the non-linearity and non-stationary characteristics of the stock market [32]. As a result, many researchers began to use machine learning technologies on the prediction of the financial market to improve the accuracy of results. Using machine learning to mimic the process of making investment decisions is novel and advanced. This usage is designed to simulate the thought process of the human mind [31]. Machine learning is an application of AI, which can grant machines the ability to access data and learn from it. Nowadays, there are many algorithms in machine learning, including Bayesian algorithm, Decision Trees algorithm, Artificial Neural Networks algorithm and so on.

These algorithms are designed for many different leaning directions. For example, Artificial Neural Networks are especially efficient in solving complex problems that cannot be reduced to straightforward decision rules [31]. Artificial Neural Networks belong to a sub-field of machine learning, known as deep learning. Artificial Neural Networks have had a massive growth in recent years. Artificial Neural Networks are built similar to the human brain because it is constructed by multiple layers of neurons and data are transferred from layer to layer, which works like a biological neural network. This is the reason why we call

this algorithm Artificial Neural Networks.

1.2 Related Work

There have been many researchers who have tried to apply machine learning techniques to problems in the financial domain. The use of soft computing techniques such as Neural Networks without getting into complex statistical issues of technical analysis enables us to achieve daily buy and sell signals with a higher speed and an acceptable accuracy [22]. Financial portfolio management is one hot topic. Portfolio management is the decision-making process of continuously reallocating an amount of funds into a number of different financial investment products, aiming to maximize the return while restraining the risk [13]. Portfolio management influences multiple markets. In general, people separate portfolio management strategies into four methods: "Follow-the-Winner", "Follow-the-Loser", "Pattern-Matching" and "Meta-Learning" [20]. It is easy to understand the first two methods. For the third method, using some pattern matching algorithms, like Artificial Neural Networks, to predict the market distribution in the next period depending on the historical data is an efficient way to solve the portfolio management problem. The last method tries to combine all other methods together to achieve better performance. Zhengyao Jiang and Jinjun Liang [12] have published their research on cryptocurrency portfolio management with deep reinforcement learning in 2017. They used Convolutional Neural Networks (CNN) as a model and fed it by using historical cryptocurrency price data.

Compared with the portfolio management direction, financial time series forecasting is another popular field. The financial market is a complex, evolutionary, and non-linear dynamical system [11][19]. Stock market prediction is usually considered as one of the most challenging issues among time series predictions [1]. When we are forecasting the time series trend of the financial market, the data used is full of noise, which makes the prediction less accurate. Some researchers believe that it is impossible to predict the value of financial assets. The efficient market hypothesis (EMH) suggests that the task of predicting future

prices based on financial assets' past behavior cannot achieve abnormal returns [23]. The reason for this is that the distribution function of a financial time series denotes a Brownian motion, which has random, independent, and Gaussian distribution characteristics [23]. However, some researchers disagree with EMH, and they think there exists a long term pattern, which can help us to predict the future value [4]. In the past decades, more and more machine learning models started to be applied on financial time series forecasting, like Artificial Neural Networks (ANN) and Support Vector Machines (SVM) [9][25]. These applications of machine learning models improve the accuracy of financial time series forecasting. There are three major ways to realize the time series forecasting. They are Convolutional Neural Networks [17], deep belief networks [10] and stacked autoencoders [2]. Bao Wei [1] has published his work on forecasting of stock market by using ANN and Wavelet Transforms (WT).

1.3 My Work

In this report, I explored both the portfolio management and the time series forecasting by building different machine learning models depending on the strength and weakness of these two directions. For the portfolio management problem, the base method is using Decision Trees. The reason for this is that portfolio management is an experiential issue. It heavily depends on personal experience and knowledge and how the trader thinks before making the management decision. To mimic the process of these human thoughts, I chose the Decision Trees method. Because it is tree structure, like Figure 1.1, it has some similarities to the process when human beings making decisions [3]. However, this model may cause over-fit problems when training data. To fix the over-fit problems, I used Random Forests algorithm to predict the returns for different stocks and pick corresponding stable stocks with high returns to form the portfolio. Random Forests are formed by multiple Decision Trees. The machine will learn which trees are high influence and set weights on each feature point.

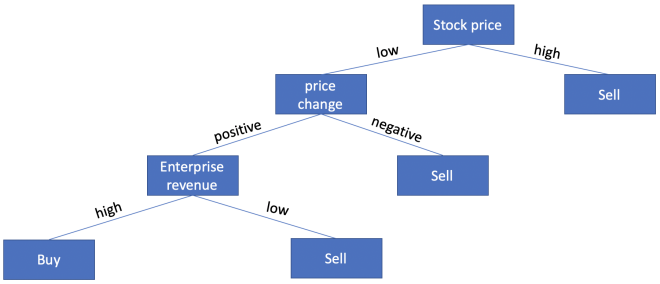

Figure 1.1 Decision Trees

The other model is used for time series forecasting on stock prices. Artificial Neural Networks was our first choice in the beginning. However, we should consider different influences of previous stock prices on the future price prediction. For example, the stock prices may be more influenced by recent news and price fluctuation. There exists one model called Long Short-term Memory, which can adjust the short-term and long-term influences on the future prediction. I used this method to train and predict the time series pattern on many particular stocks and to improve the predictable accuracy.

CHAPTER 2. DATA PREPROCESSING

2.1 Time Series Forecasting

For the time series forecasting model, I used the Neural Networks structure. The most important data for a Neural Network is historical data. It is easy to download historical stock data from the internet [8]. In this report, I used data from the Yahoo Finance website. The stock dataset used by the time series forecasting model included NASDAQ index (one year and ten years), APPL (one year and ten years) and MSFT (Microsoft Inc one year and ten years). For each stock dataset, there are seven columns. They are date, open price, highest price, lowest price, close price, adjusted close price and transaction volume. The dataset is shown as Figure 2.1. Two major parts of information used for the Neural Network model is date and close price. The model should be fed with the close price and date.

2.2 Portfolio Forecasting

For portfolio forecasting, we not only need the time series price data, but also need financial statistics data, like the Income Statement, Balance Sheet, Dividends info and so on. The reason financial statistics data is necessary is that for the Decision Trees method, the model needs some feature data as decision points. To realize the classification, we need these feature data. You can find those fundamentals data from Quandl.com. However, many data on this website are not free to download. I alternatively use data captured from Yahoo Finance website. There are many statistics data like Figure 2.2. we should have corresponding fundamentals info for each stock following the time series.

nasdq

Date	Open	High	Low	Close	Adj Close	Volume
2018/10/16	7501.779785	7658.140137	7493.439941	7645.490234	7645.490234	2623300000
2018/10/17	7669.259766	7670.490234	7563.089844	7642.700195	7642.700195	2364720000
2018/10/18	7616.470215	7616.859863	7452.459961	7485.140137	7485.140137	2545950000
2018/10/19	7530.160156	7582.890137	7428.299805	7449.029785	7449.029785	2532220000
2018/10/22	7486.740234	7520.540039	7424.740234	7468.629883	7468.629883	2282400000
2018/10/23	7328.549805	7472.580078	7260.129883	7437.540039	7437.540039	2735820000
2018/10/24	7423.209961	7435.689941	7099	7108.399902	7108.399902	2935550000
2018/10/25	7197.490234	7364.819824	7178.540039	7318.339844	7318.339844	2741810000
2018/10/26	7125.180176	7283.319824	7057	7167.209961	7167.209961	2964780000
2018/10/29	7272.419922	7295.609863	6922.830078	7050.290039	7050.290039	2689550000
2018/10/30	7017.870117	7166.839844	7001.47998	7161.649902	7161.649902	2681310000
2018/10/31	7276.620117	7368.490234	7270.629883	7305.899902	7305.899902	2899730000
2018/11/1	7327.819824	7435.879883	7286.5	7434.060059	7434.060059	2708880000
2018/11/2	7424.02002	7466.529785	7298.680176	7356.990234	7356.990234	2889910000
2018/11/5	7344.080078	7349.22998	7255.879883	7328.850098	7328.850098	2166470000
2018/11/6	7326.069824	7400.640137	7320.890137	7375.959961	7375.959961	2285680000

Figure 2.1 NASDAQ Index Dataset

Algorithm 1: How to pre-process stock statistics data

Result: Return .csv file like Figure 2.3

Repeat for every stock;

while *In the same stock* **do**

    feature_array = {P/E ratio, Enterprise Value, ...} //Store all features;

    if *current_feature == element in feature_array* **then**

        Write_to_csv(row, col, current_value);

    else

        Continue;

    end

end

VALUATION MEASURES	
Market Cap (intraday):	7.65B
Enterprise Value (7-Dec-03)[3]:	3.38B
Trailing P/E (ttm):	112.70
Forward P/E (fye 27-Sep-05)[1]:	40.67
PEG Ratio (5 yr expected)[1]:	4.41
Price/Sales (ttm):	1.25
Price/Book (mrq):	1.84
Enterprise Value/Revenue (ttm)[3]:	0.55
Enterprise Value/EBITDA (ttm)[3]:	N/A

FINANCIAL HIGHLIGHTS	
Fiscal Year	
Fiscal Year Ends:	27-Sep
Most Recent Quarter (mrq):	30-Sep-03
Profitability	
Profit Margin (ttm):	1.10%
Operating Margin (ttm):	-0.02%
Management Effectiveness	
Return on Assets (ttm):	1.06%

Stock Price History	
Beta:	1.742
52-Week Change:	39.46%
52-Week Change (relative to S&P500):	19.85%
52-Week High (15-Oct-03):	25.01
52-Week Low (17-Apr-03):	12.72
50-Day Moving Average:	22.15
200-Day Moving Average:	19.05
Share Statistics	
Average Volume (3 month):	4,837,318
Average Volume (10 day):	4,171,000
Shares Outstanding:	366.73M
Float:	363.80M
% Held by Insiders:	0.80%
% Held by Institutions:	64.42%
Shares Short (as of 10-Nov-03):	11.99M
Daily Volume (as of 10-Nov-03):	N/A
Short Ratio (as of 10-Nov-03):	2.356
Short % of Float (as of 10-Nov-03):	3.30%
Shares Short (prior month):	13.61M

Figure 2.2 Statistics Data Sample

These fundamentals data are just the first step for the model, because they are not usable for training a model as a .csv file yet. The second step is transferring these key features into a .csv file. The corresponding code is shown as Algorithm 1. The model should memorize every feature related to every stock, and also compute the price changes in this period. the price change values are used for validation during the training process. Besides that, only looking at a single stock price change is not enough. The model also stores the S&P 500 index changes during the same time period. The model used index changes to filtrate those stocks who perform worse than S&P 500 index performance. this produces a .csv file looks like Figure 2.3.

In the dataset of Figure 2.3, We fed our model with features after column 7. Then the model would pick up valuable features and validate the difference between prediction results and real results shown in the price column.

Date	Unix	Ticker	Price	SP500	SP500_p_ch	Market Cap	Enterprise V	Trailing P/E	Forward P/E	PEG Ratio	Price/Sales	Price/Book	Enterprise V
7/11/07 5:59	1184151568	ctas	33.538876	122.084023	-15.9	6350000000	7080000000	19.13	17.79	1.49	1.74	3.09	1.94
2/28/06 20:38	1141180739	ctas	34.026257	100.297188	11.94	6910000000	7160000000	22.55	18.79	1.39	2.16	3.21	2.22
4/20/05 15:38	1114029524	ctas	31.885536	87.81147	17.27	6750000000	7020000000	23.35	19.72	1.46	2.31	3.33	2.34
2/22/05 13:02	1109098963	ctas	35.317966	91.157219	10.92	7550000000	7670000000	26.61	21.78	1.57	2.57	3.68	2.61
4/26/13 17:42	1367016160	ctas	41.474506	143.779022	20.1	5480000000	6550000000	18.46	16.09	1.75	1.3	2.59	1.55
10/23/06 13:27	1161628069	ctas	34.183594	108.906029	12.46	6630000000	7360000000	20.6	16.72	1.24	1.9	3.23	2.11
10/26/04 7:45	1098794757	ctas	34.682678	85.075638	9.17	7140000000	7360000000	25.47	20.28	1.47	2.49	3.65	2.55
9/6/05 22:33	1126063990	ctas	33.462154	95.835831	7.41	6860000000	6940000000	23.41	18.03	1.25	2.21	3.26	2.26
9/5/06 15:34	1157488469	ctas	31.392931	103.854393	14.3	6110000000	6670000000	19.58	15.57	1.09	1.8	2.97	1.96
2/19/06 17:57	1140393429	vz	17.403154	100.750832	15.23	9.617E+10	1.319E+11	13.14	13.12	4.53	1.27	2.41	1.76
11/2/08 21:33	1225683224	vz	17.552139	79.804276	10.53	8.428E+10	1.2738E+11	13.56	10.83	1.7	0.9	1.7	1.32
10/24/12 21:51	1351133481	vz	34.875725	126.666946	26.92	1.2623E+11	1.6827E+11	40.88	15.47	1.59	1.1	3.34	1.47
5/29/07 16:49	1180475354	vz	23.741819	121.760048	-6.72	1.2453E+11	1.5462E+11	20.65	16.31	3	1.38	2.53	1.73
9/2/11 15:03	1314993819	vz	26.353117	103.102608	22.32	1.0114E+11	1.4872E+11	16.03	13.74	1.86	0.95	2.58	1.39
4/22/12 4:43	1335087833	vz	29.869682	122.616142	15.73	1.0982E+11	1.5102E+11	45.56	13.93	1.44	0.98	3.01	1.36
1/1/12 14:15	1325448914	vz	30.146585	111.063309	15.99	1.1358E+11	1.5764E+11	16.13	15.73	1.52	1.04	2.9	1.45
6/20/12 20:46	1340243179	vz	33.394207	121.044655	19.65	1.2302E+11	1.6929E+11	46.46	15.52	1.62	1.11	3.39	1.51
2/25/05 16:05	1109369148	vz	17.254591	93.332367	8.45	9.957E+10	1.3302E+11	13.88	13.47	2.76	1.38	2.62	1.87
1/17/05 1:53	1105948382	vz	17.559643	90.880516	10.45	1.0201E+11	1.4093E+11	17.3	14.01	2.94	1.46	2.92	2.01
4/29/07 10:38	1177861135	vz	20.823559	119.592636	-4.8	1.1026E+11	1.4096E+11	17.87	14.74	3.26	1.25	2.28	1.6
6/28/04 23:14	1088482462	vz	16.637049	86.173737	8.02	9.962E+10	1.4105E+11	35.85	14.1	2.58	1.45	2.96	2.06
11/19/11 5:42	1321702949	vz	27.396427	107.267929	14.19	1.0322E+11	1.4796E+11	14.66	14.3	1.65	0.95	2.66	1.36
10/3/03 1:05	1065161153	vz	15.113293	77.606743	11.7	9.173E+10	1.3811E+11	9.11	13.97	4.7	1.34	2.66	2.04
4/14/04 16:43	1081978998	vz	17.155228	85.814934	4.1	1.0302E+11	1.4536E+11	29.48	14.76	6.83	1.52	3.07	2.15
12/8/03 15:05	1070917501	vz	14.958208	80.744354	11.89	9.201E+10	1.3658E+11	12.74	14.09	6.4	1.36	2.63	2.02
10/24/07 16:35	1193261758	vz	24.885719	122.251602	-38.19	1.2704E+11	1.6023E+11	20.79	16.33	2.59	1.43	2.61	1.76
12/20/08 19:10	1229821818	vz	19.634489	73.273483	27.86	9.428E+10	1.3738E+11	15.17	12.25	1.98	0.98	1.86	1.42
3/11/07 21:08	1173665299	vz	19.818768	112.149643	-7.4	1.0607E+11	1.3677E+11	17.19	14.13	3.1	1.2	2.19	1.55
12/9/04 7:57	1102600650	vz	19.448898	91.195282	7.78	1.145E+11	1.5353E+11	19.42	15.55	3.15	1.63	3.26	2.19
6/29/07 14:18	1183144691	vz	22.626175	120.830994	-13.51	1.1936E+11	1.5047E+11	19.79	15.57	2.57	1.33	2.44	1.68
5/2/07 0:44	1178084655	vz	21.76333	119.600624	-3.79	1.1206E+11	1.4181E+11	18.17	14.98	3.26	1.26	2.29	1.61
6/17/06 9:12	1150553569	vz	16.476055	98.317375	25.06	9.493E+10	1.3405E+11	12.66	12.1	4.5	1.18	2.1	1.68
7/3/07 16:01	1183496489	vz	23.148273	122.365211	-15.5	1.2148E+11	1.5195E+11	20.14	15.85	2.42	1.35	2.47	1.7
9/6/13 1:19	1378448351	vz	37.732414	151.664505	23.5	1.3347E+11	1.8419E+11	85.42	14.48	1.64	1.13	3.93	1.56
2/6/13 9:42	1360165333	vz	35.697113	136.7379	19.78	1.274E+11	1.7582E+11	145.15	14.19	2.01	1.1	3.84	1.52

Figure 2.3 Statistics Data Table

CHAPTER 3. METHODS AND PROCEDURES

3.1 Time Series Forecasting Model

To create the time series forecasting model, I used an Artificial Neural Networks (ANN) algorithm. In the past few years, many researchers have used ANNs to analyze traditional classification and regression prediction problems in accounting and finance [5]. Many researchers have already proved ANN is an efficient algorithm to predict non-linear paths. ANN also has been proved that it performs very well in feature extraction from raw data [6]. In this report, I used a particular method called Long Short-term Memory (LSTM). LSTM is a type of Recurrent Neural Networks (RNN), and RNN is a type of ANN. ANN has one input layer, one or multiple hidden layers and one output layer, as shown in Figure 3.1. We can see that each input neuron is connecting to each hidden neuron and each hidden neuron is also connecting to the output neurons. There is no connection among neurons in the same layer. Every connection has its own weight which can be learned and adjusted from the training process.

Compared with ANN, RNN adds recurring connections shown in Figure 3.2. Recurrent neural networks (RNNs) are able to process input sequences of arbitrary length via the recursive application of a transition function on a hidden state vector. At each time stamp t, the hidden state h_t is resulted from a function of the input vector x_t at time t and its previous hidden state h_{t-1}. For example, the input vector x_t could be a vector representation of the t-th word in the body of text [30] [27]. As a result, recurrent networks can recirculate previous outputs back to the inputs, similar to the concept of using lagged variables in forecasting [5]. Because RNN can learn the result from the previous round, this gives RNN

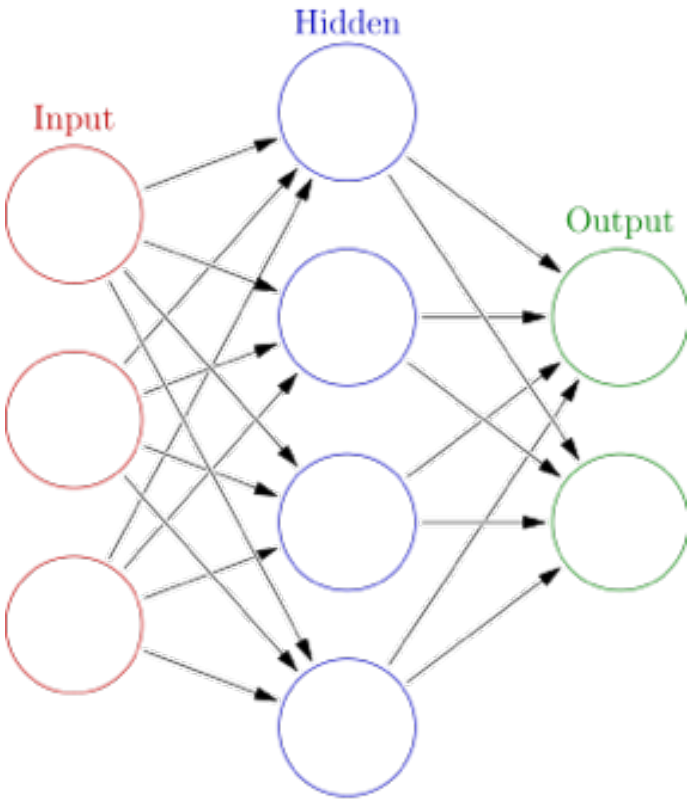

Figure 3.1 Artificial Neural Networks [18]

a time series context. However, RNN can only read the results from the previous round. For stock price data, there are thousands of points of data. Therefor, only remembering the previous round result is not enough, because the stock price might also be influenced by the price weeks ago or months ago. LSTM is introduced into this report because LSTM can address this problem.

The LSTM architecture fixes the lack of ability to learn long-term dependencies by introducing a memory cell that is able to preserve states over long periods of time [30], as shown in Figure 3.3. There are two horizontal chains, the upper chain contains memory cells, which store the long-term results from previous training known as c_{t-1}, c_t, c_{t+1} and so on. The basic architecture to realize long-term memory capability is by using three gates, input gate, forget gate and output gate. The memorization of the earlier trend of the data

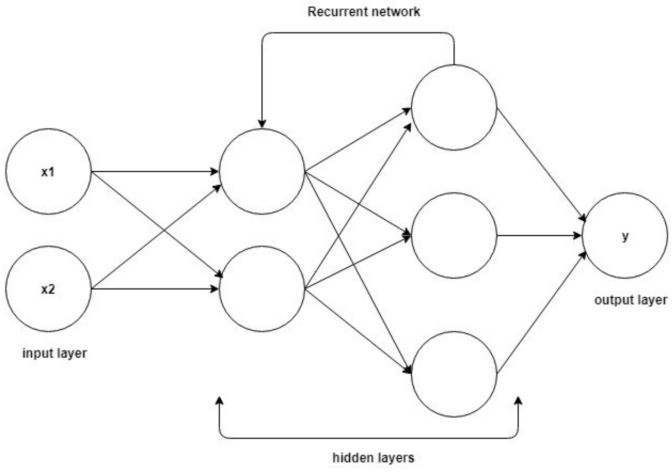

Figure 3.2 Recurrent Neural Networks [24]

is possible through these gates along with a memory line incorporated in a typical LSTM [29][28].

The first step of LSTM is deciding which information should be thrown away by the forget gate. All information kept will be stored into a memory cell c_{t-1} for the next process. The next step is deciding what information included in the input gate. There are two parts:

$$i_t = \sigma(W_i * [X_t, h_{t-1}] + b_i) \tag{3.1}$$

$$\bar{C}_t = \tanh(W_c * [X_t, h_{t-1}] + b_c) \tag{3.2}$$

In Equation 3.1 and 3.2. W_i and W_c means the weight, X_t means the input data at time t, h_{t-1} means the hidden state at time $t-1$. Equation 3.1 uses sigmoid function to let input gate decide which information should be kept, and Equation 3.2 uses tanh function to creates a vector of new value $\bar{C}_t$, which will combined with i_t to store into a memory cell. The current cell state C_t will be calculated as:

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \tag{3.3}$$

$$O_t = \sigma(W_o * [X_t, h_{t-1}] + b_o) \tag{3.4}$$

$$h_t = O_t * \tanh(C_t) \tag{3.5}$$

In Equation 3.3, f_t means the result from forget gate, which is 0 or 1. Then use C_t and O_t to compute the new hidden state h_t, which could be an input of a next time stamp or be the result as shown in Equation 3.4 and 3.5.

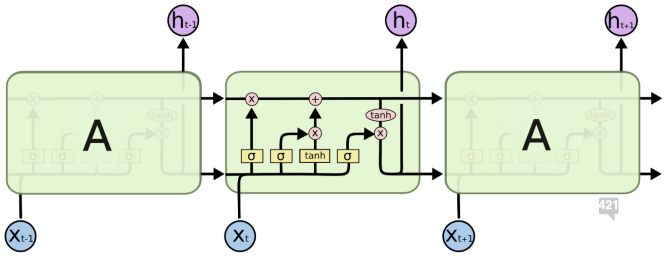

Figure 3.3 LSTM [14]

3.2 Portfolio Forecasting Model

For this prediction model, I used Random Forests, which is a type of ensemble learning method. Random Forests is constructed by multiple Decision Trees during training process and output the mode or mean of individual Decision Trees. Decision Trees are a useful machine learning algorithm. Normally there are four steps. First, begin with the dataset which should have multiple features. Second, find the best features in the dataset. Third, split the data into sub-data, which contains the possible values with the best features. Last, recursively split the sub-data from step 3 by finding possible suitable features [16]. Decision Trees can be used for various machine learning applications, but trees, that are grown really deep to learn highly irregular patterns, tend to over-fit the training sets. A slight noise in the data may cause the tree to grow in a completely different manner [15][21]. Random Forests perform well on removing the over-fit problem caused by Decision Trees.

CHAPTER 4. RESULT AND PERFORMANCE

4.1 Time Series Forecasting Results

4.1.1 Linear Regression Results

As Figure 4.1, 4.2 and 4.3 show, they are the prediction results by using linear regression model from scikit-learn library in python. Figure 4.1 shows the index graph of NASDAQ in 2019. Figure 4.2 shows the index graph of NASDAQ in 2015 and Figure 4.3 shows the last ten year's NASDAQ index. The first 80% data is the real data used for training set. The last 20% data leaves for prediction and comparison. We can tell that none of three predictions shows a strong performance, compared with the LSTM model predictions in Figure 4.7, 4.8 and 4.9.

4.1.2 K-Nearest Neighbours Results

As Figure 4.4, 4.5 and 4.6 show, they are the prediction results by using KNN model. Figure 4.4 shows the Microsoft Inc stock index in 2019. Figure 4.5 shows the Microsoft Inc stock index in recent five years. Figure 4.6 shows the Microsoft Inc stock index in recent ten years. In the same, they are using first 80% data as training set and last 20% as validation set. We can tell that KNN model's prediction provide huge vibrations, which is hard to be used as a persuasive prediction.

4.1.3 LSTM Results

As Figure 4.7, 4.8, 4.9 shows, I predicted NASDAQ index on three different datasets. For the ten year dataset, there are 2516 transaction days in total. The head 2100 days are

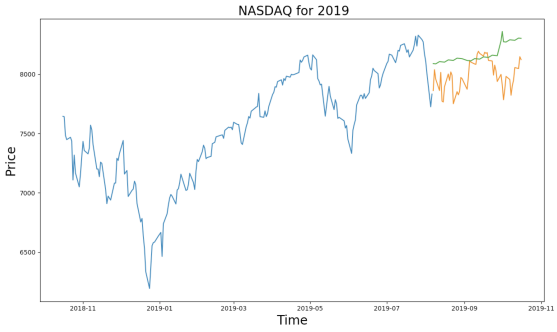

Figure 4.1 NASDAQ for 2019

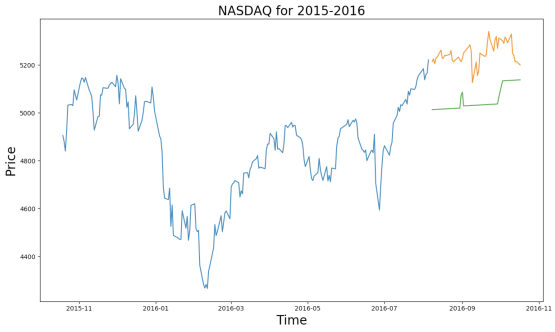

Figure 4.2 NASDAQ for 2015-2016

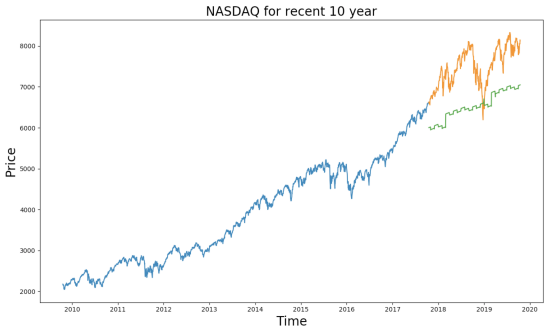

Figure 4.3 NASDAQ for recent 10 year

used as training set and the left days are used for validation. For the other two one year test, both of them have 252 transaction days in dataset. I use head 210 days as training set and left days are used for validation. We can tell from the figure that LSTM is doing much better in the test with ten year dataset than the test with one year dataset.

The reason I did both 2015 to 2016 and 2018 to 2019 tests is because that the trade war exploded in 2018, which may cause abnormal trend in stock market. As a result, I also test the prediction performance for 2015 to 2016 to check if there exists any apparent differences between these two tests.

As Figure 4.10, 4.11, 4.12 shows, compared with the NASDAQ index, this time I picked a single company stock, Microsoft Inc. These figures shows 1 year, 5 year and 10 year price change and prediction. For 1 year test, there are 251 days in total. I use head 210 as training data and the left days for the validation. For the 5 year test, there are 1258 days in total, I use head 1000 as training data and the left days for the validation. For 10 year

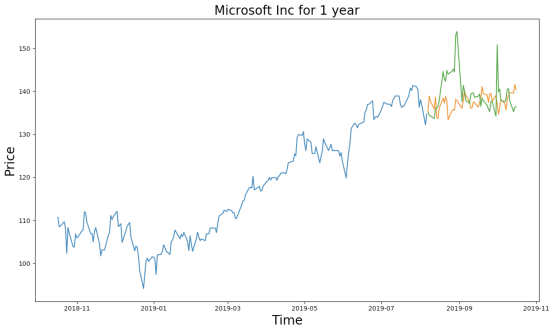

Figure 4.4 Microsoft Inc for 1 year

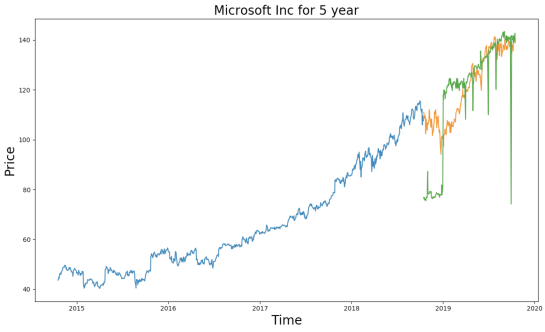

Figure 4.5 Microsoft Inc for 5 year

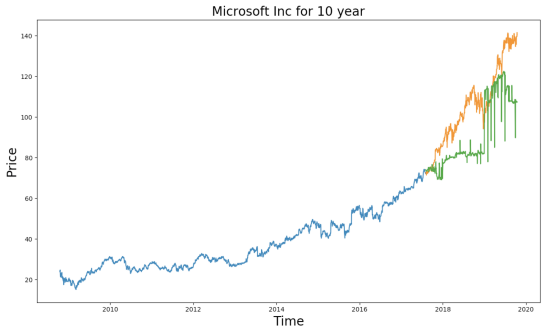

Figure 4.6 Microsoft Inc for 10 year

test, there are 2768 days in total, I use head 2100 as training data and the left days for the validation. From the figures, we can tell that the longer time period makes LSTM model doing better.

4.2 Portfolio Prediction Model Results

To do the Random Forests model, we need to preprocess the dataset. First, we need a benchmark, which can show us the average market performance. Here we use S&P 500 index as the benchmark. Compared with the time series forecasting model, portfolio model need a bunch of stocks data. We also need the price change for each stock and each time period, so we can consider 10 transaction days. So for every 10 transaction days, we consider the key features contained in Figure 2.2. We feed these features into Random Forests model and get the following results.

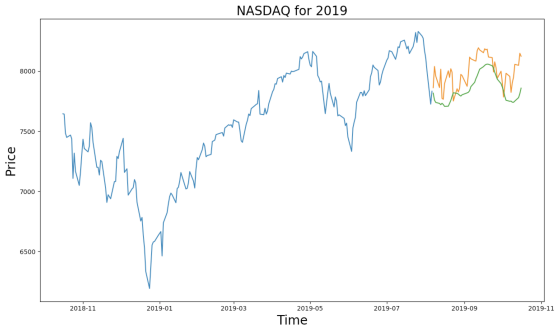

Figure 4.7 NASDAQ for 2019

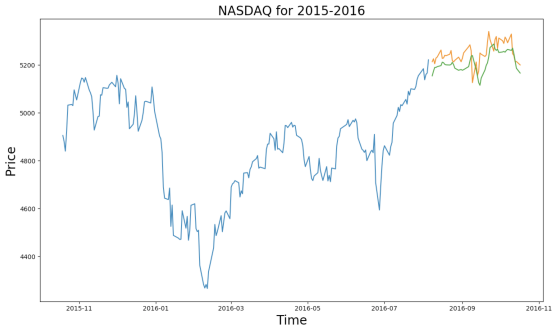

Figure 4.8 NASDAQ for 2015-2016

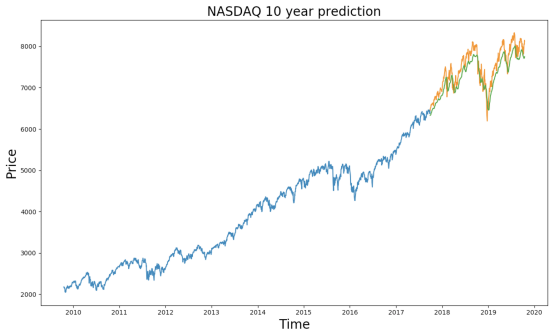

Figure 4.9 NASDAQ for recent 10 year

As Table 4.1 shows, the whole database includes 3384 stocks. Insides, 2707 stocks are used to train our Random Forests model with the stock features. The left 677 stocks are used for validation. Among these 677 stocks, there are 217 stocks have more than 10% more return than the market return (S&P 500 index increase), and 460 stocks do not hit 10% threshold. After the training process, we get 195 stocks, whose return is higher than the threshold (10%), and 482 stocks that do not hit the threshold. The model accuracy reaches 82%. Figure 4.13 shows part of the stocks, which are defined as positive by the model. In other words, they are all considered as they can perform better than the threshold by our models. We can tell that most of stocks have a good performance, but there still exist a few stocks with bad performance like `cce` and `ppg`. Their increases are even slower than market return. Also there exist a few stocks that grow faster than market return, but still below the threshold, like `goog` (Google Inc).

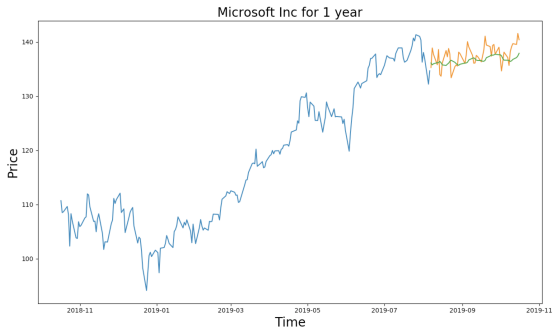

Figure 4.10 Microsoft Inc for 1 year

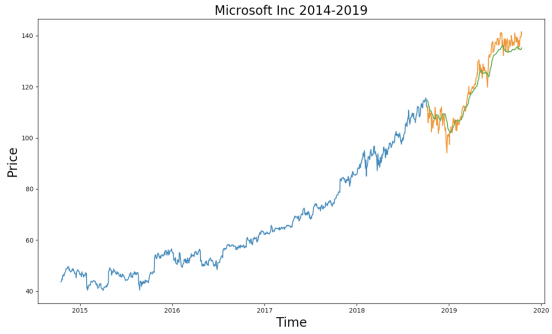

Figure 4.11 Microsoft Inc for 5 year

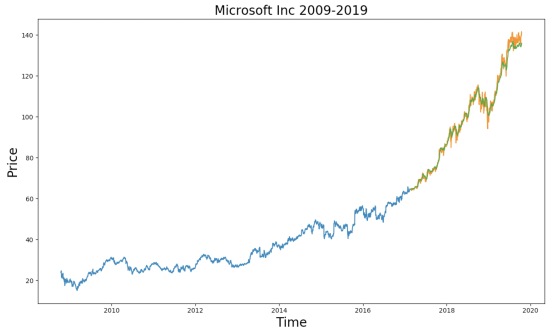

Figure 4.12 Microsoft Inc for 10 year

Stocks Return Result	
Total stocks	3384
Train stocks	2707
Test stocks	677
Number of stocks hit 10% return (actual)	217
Number of stocks missed 10% return (actual)	460
Number of stocks hit 10% return (predict)	195
Number of stocks missed 10% return (predict)	482
Model Accuracy	82%

Table 4.1 Random Forests result

	stock code	actual return	marcket return	performance
0	fosl	1.7198	1.1434	0.5764
1	mur	1.5234	0.94	0.5834
2	ibm	1.4531	1.1437	0.3094
3	jnpr	1.2925	0.9965	0.296
4	acn	1.4507	1.1477	0.303
5	pfe	1.1707	1.0491	0.1216
6	goog	1.1015	1.0295	0.072
7	cf	1.3449	1.1989	0.146
8	mo	1.4036	1.2635	0.1401
9	stx	1.9034	1.2184	0.685
10	cce	0.9867	1.0662	-0.0795
11	cme	1.7087	1.2962	0.4125
12	de	1.583	0.9712	0.6118
13	cbs	1.8446	1.3477	0.4969
14	aa	1.6408	1.2952	0.3456
15	apc	1.4479	0.9481	0.4998
16	ko	1.3496	1.2731	0.0765
17	ppg	0.8502	0.9199	-0.0697
18	xom	1.2249	1.0642	0.1607
19	vlo	1.4611	1.2232	0.2379
20	adbe	1.6851	1.2952	0.3899
21	etr	1.5439	1.232	0.3119
22	pbi	1.7127	1.1989	0.5138
23	nwl	1.3572	1.2259	0.1313
24	msft	1.6056	1.3352	0.2704
25	ko	0.7756	0.6331	0.1425
26	ci	0.8673	0.9576	-0.0903
27	kr	1.0141	0.9506	0.0635

Figure 4.13 Table of stock returns

CHAPTER 5. Conclusion

Some researchers believe that predicting the stock market return is impossible because stock market is random and closely correlated with real time events. No one can forecast what will happen in the future. This report aims to point out and practice the possible way to try to predict and use it as an assistant way to human decision making. From the tests in this report, we can tell that the predictions cannot achieve a very high precision, but see from a long-term angle, the prediction results indicate general correct direction. We can use these results cooperated with human experiential knowledge to make financial decisions.

There are also some potential ways to extend this project. For example, in the Random Forests model, we can add sentimental characteristics in the training set. Rui Ren [26] and Ronen Feldman [7] have published some interesting ways that explore the sentimental analysis. Their model analyzed the most recent related articles to consider if common medias conceive positive or negative opinions. By using sentimental as new feature with relatively high weight in Random Forests model, it may improve the accuracy significantly.

BIBLIOGRAPHY

[1] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[2] Y Bengio, P Lamblin, D Popovici, and H Larochelle. Greedy layer-wise training of deep networks advances in neural information processing systems 19 (nips'06) 1 (2), 2007.

[3] Christian Catalini, Chris Foster, and Ramana Nanda. Machine intelligence vs. human judgement in new venture finance. Technical report, Mimeo, 2018.

[4] Roberto Cervelló-Royo, Francisco Guijarro, and Karolina Michniuk. Stock market trading rule based on pattern recognition and technical analysis: Forecasting the djia index with intraday data. *Expert systems with Applications*, 42(14):5963–5975, 2015.

[5] James R Coakley and Carol E Brown. Artificial neural networks in accounting and finance: Modeling issues. *Intelligent Systems in Accounting, Finance & Management*, 9(2):119–144, 2000.

[6] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

[7] Ronen Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, 2013.

[8] Yahoo Finance. Yahoo finance, 2019.

[9] Zhiqiang Guo, Huaiqing Wang, Quan Liu, and Jie Yang. A feature fusion based forecasting model for financial time series. *PloS one*, 9(6):e101113, 2014.

[10] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[11] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & operations research*,

32(10):2513–2522, 2005.

[12] Zhengyao Jiang and Jinjun Liang. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 905–913. IEEE, 2017.

[13] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017.

[14] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[15] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*, 2016.

[16] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9):1011, 2008.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Ben Kröse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks. 1993.

[19] Manish Kumar and M Thenmozhi. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[20] Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):35, 2014.

[21] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[22] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2):115–125, 2009.

[23] Felipe Dias Paiva, Rodrigo Tomás Nogueira Cardoso, Gustavo Peixoto Hanaoka, and Wendel Moreira Duarte. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115:635–655, 2019.

[24] Lamia Rahman, Nabeel Mohammed, and Abul Kalam Al Azad. A new lstm model by introducing biological cell state. In *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–6. IEEE, 2016.

[25] Apostolos Nicholas Refenes, Achileas Zapranis, and Gavin Francis. Stock performance modeling using neural networks: a comparative study with regression models. *Neural networks*, 7(2):375–388, 1994.

[26] Rui Ren, Desheng Dash Wu, and Tianxiang Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, 13(1):760–770, 2018.

[27] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[28] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[29] Sima Siami-Namini and Akbar Siami Namin. Forecasting economics and financial time series: Arima vs. lstm. *arXiv preprint arXiv:1803.06386*, 2018.

[30] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[31] Robert R Trippi and Efraim Turban. *Neural networks in finance and investing: Using artificial intelligence to improve real world performance*. McGraw-Hill, Inc., 1992.

[32] Xiao-dan Zhang, Ang Li, and Ran Pan. Stock trend prediction based on a new status box method and adaboost probabilistic support vector machine. *Applied Soft Computing*, 49:385–398, 2016.