Media Engineering and Technology Faculty
German University in Cairo

Machine Learning for Stock Trading

Bachelor Thesis

Author:	Ahmed Hamdi Ebied
Supervisor:	Dr. Aysha Alsafty
Submission Date:	30 May, 2018

Media Engineering and Technology Faculty
German University in Cairo

Machine Learning for Stock Trading

Bachelor Thesis

Author:	Ahmed Hamdi Ebied
Supervisor:	Dr. Aysha Alsafty
Submission Date:	30 May, 2018

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgement has been made in the text to all other material used

Ahmed Hamdi Ebied
30 May, 2018

Acknowledgments

To my father.

I would like to express my gratitude to my supervisor, Dr. Aysha Alsafty, for her continuous guidance and assistance.

Abstract

Stock market prediction is an interesting realm to test the capabilities of machine learning on. The nature of the stock market is volatile, sophisticated, and very sensitive to external information, which makes it difficult to predict. Different machine learning models are developed to forecast future stock prices. Using historical stock market data, technical indicators are computed and used along with a stock's price as features associated with a target output, which is the future stock price. This provides a dataset that the machine learning models use to train upon, and thus the models become capable of predicting future prices. The models used are: linear regressor, kNN regressor, Feedforward Neural Network (FFNN), and Long Short Term Memory (LSTM) Recurrent Neural Network (RNN). The prediction models are compared and evaluated using different metrics. Several case studies are performed to evaluate the performance of the machine learning models. From the case studies, few insights have been made. Firstly, the LSTM RNN outperformed all the other models. Secondly, the LSTM RNN model is capable of accurately predicting the next-day price unless a major external event impacts the stock price suddenly. Lastly, the LSTM RNN model naturally lags on picking up on external events that impact the stock price suddenly.

Contents

Chapter 1

Introduction

1.1 Goals and Research Questions

This research aims to exploit the capabilities of machine learning in the field of stock trading. Using concepts and techniques in technical analysis and machine learning, stock price prediction models will be developed. The models will be compared and assessed by using several evaluation metrics, and running several experiments and tests. The research questions for this bachelor thesis are:

1. *Can machine learning be used to predict future stock prices?*

   Is it possible to design machine learning models that are trained on historical prices and technical indicators of a certain stock, and to be able to query the model for future prices? How reliable will the model be? What are the model's constraints, guarantees, and weaknesses?

2. *How does the performance of different machine learning algorithms vary?*

   Which machine learning algorithm does the best job, and how do the different algorithms compare with each other? A grading criteria should be designed to compare and assess the algorithms.

1.2 Outline

This thesis is divided into six chapters: Introduction, (2) Literature Review, (3) Background Knowledge, (4) Implementation, (5) Experiments and Results, and (6) Conclusion. The literature review chapter provides insight into previous work related to the research topic of this thesis. The background knowledge chapter explains the theory used in the implementation. The implementation chapter contains an overview of the data and explains how the algorithms were implemented. The experiments and results chapter presents the different tests and case studies performed and the results and their implications. The conclusion chapter attempts to draw some conclusions from the results.

Chapter 2

Literature Review

In this chapter, several research papers and theses that have investigated the application of machine learning on stock trading are presented. This chapter goes through each paper's or thesis' research question(s), overall process, and conclusion and results briefly. This chapter thus serves as a high-level introduction to this research topic.

Dingli and Fournier (2017) paper covered various regression and classification machine learning approaches to attempt to predict the stock market over short and long terms. The technical indicators used included: momentum, volume, volatility and cycle-based indicators. The outputs considered were next period: direction, price change, and actual price. The periods were: daily, weekly, monthly, quarterly and yearly. Classification machine learning algorithms used included: K-Nearest Neighbors, Logistic Regression, Naive Bayes, Support Vector Classifier (SVC), Decision Trees, Random Forest, Multi Layer Perceptron (MLP), Ada Boost, and QDA. Regression machine learning algorithms included: Linear Regressor, Support Vector Regressor, Decision Tree Regressor, Ada Boost Regressor, Random Forest Regressor, K-Nearest Neighbors Regressor, and Bagging Regressor. Results showed 81% accuracy for future trend direction using classification. Using regression techniques, results showed 0.0117 RMSE for next day price and 0.0613 RMSE for next day change in price [1].

Patel, Shah, Thakkar, & Kotecha (2015) study focused on comparing prediction performance of four machine learning algorithms: Artificial Neural Network (ANN), Support Vector Machine (SVM), random forest and naive-Bayes. The algorithms were given the task of predicting stock and stock price index movement. The research focused on comparing the performance of these prediction models when the inputs are represented in the form of real values and trend deterministic data. Ten technical parameters were used as the inputs to these models. Some of the technical indicators used are: Simple n-day Moving Average, Weighted n-day Moving Average, Momentum, Stochastic K%, Stochastic D%, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Larry Williams R%, A/D (Accumulation/Distribution) Oscillator, and CCI (Commodity Channel Index). The results suggested that when the inputs are represented in the form of real continuous values, random forest outperforms the other three prediction models on overall performance. In addition, the performance of all the prediction models improve when the technical parameters are represented as trend deterministic data [2].

R. Dash and P. K. Dash (2016) paper proposed a novel decision support system using a Computational Efficient Functional Link Artificial Neural Network (CEFLANN) and a set of rules to generate the trading decisions more effectively. The problem of stock trading decision prediction was articulated as a classification problem with three class values representing the buy, hold and sell signals. The CEFLANN network used in the decision support system produced a set of continuous trading signals by analyzing the nonlinear relationship existing between few popular technical indicators. The technical indicators used included: Simple Moving Average, Moving Average Convergence and Divergence, Stochastic KD, Relative Strength Index, and Larry William's R%. Further, the output trading signals were used to track the trend and to produce the trading decision based on that trend using some trading rules. For assessing the potential use of the proposed method, the model performance was also compared with some other machine learning techniques such as Support Vector Machine (SVM), Naive Bayesian model, K nearest neighbor model (KNN) and Decision Tree (DT) model. The CEFLANN showed the highest profit percentage [3].

Larsen (2010) thesis used a knowledge-intensive first layer of reasoning based on technical analysis of the price data. The second layer was a reasoning layer based on machine learning, which was used for further analysis. The first layer allowed the second layer to focus on important aspects of the price data rather than the raw price data itself. The technical analysis features used were: trend, moving average crossover, candlestick chart, stochastic indicator, volume, and ADX. The machine learning technique used is agent decision trees. The results when evaluated on the Oslo Stock Exchange showed that some portfolio simulations increased the initial investment capital by almost 300% from January 2009 to May 2010, thus beating the Oslo Benchmark Index by approximately 250%. Based on this result, Larsen believes he was able to achieve the main purpose and research question of his thesis, which is to create a stock price prediction model that can be used as a decision support tool or as an autonomous artificial trader [4].

Olden (2016) thesis mainly aimed to determine if it is possible to make a profitable stock trading scheme using machine learning on the Oslo Stock Exchange. The data about the stock used included price at different points in a trading day (open, high, low, best ask, and closing), as well as, Norwegian indices from Oslo Benchmark Index (international, price total return, and volume-weighted). Different binary prediction machine learning algorithms were used. The results yielded show that there are binary prediction machine learning algorithms that are profitable in the test period, and can be optimized to a much greater extent, which likely will increase potential profit yielded. However, it is unknown whether the test period is representative for any future time period [5].

Chapter 3

Background Knowledge

3.1 Stock Trading

A stock is a piece of a publicly-traded company. A person who owns one or more shares is called a shareholder. A shareholder is a partial owner of the underlying company. A shareholder is entitled to a set of rights including, but not limited to:

1. Voting power on major issues
2. Ownership in a portion of the company
3. The right to transfer ownership
4. An entitlement to dividends
5. Opportunity to inspect corporate books and records
6. The right to sue for wrongful acts

The influence a shareholder has on a company is directly proportional to the amount of shares owned by the shareholder.

The main reference used in section 3.1 is Stock Market Investing for Beginners Udemy course [6].

3.1.1 Stock Description

Stocks can be described in different ways. Some of the measures that can be used include:

- Market Capitalization

  Market capitalization refers to the total market value of a company's outstanding shares. Commonly referred to as market cap, it is calculated by multiplying a company's outstanding shares by the current market price of one share. The investment community uses this figure to determine a company's size, as opposed to using sales or total asset figures [7]. Usually companies with market cap under 1 billion dollars, between 1-5 billion dollars, over 5 billion dollars are classified as small-cap, mid-cap, large-cap respectively. Large-cap companies tend to be much less vulnerable to the ups and downs of the economy, a large part due to their huge financial reserves. On the other hand, small-cap companies are greatly affected by turmoil, but can provide for a lot more growth versus large-caps.

  $$market_{cap} = price \times n \tag{3.1}$$

  where:

  $market_{cap}$ = market capitalization
  $price$ = market share price
  n = number of shares outstanding

- Sectors and Industries

  Companies are divided into sectors and industries. A sector represents a large part of the economy. Common sectors include electronics, finance, energy, etc. On the other hand, industries are much more specific and are considered to be a subset of some sector. In terms of stock price movement, industries within the same sector tend to exhibit similar reactions to changes in the economy. In order to minimize risks, it is suggested to make sure that a portfolio, which means an investor's collection of owned stocks, includes companies from different sectors and industries.

- Seculars and Cyclicals

  Stocks can be categorized into two different categories: seculars and cyclicals. They mainly differ in the way they make profit. Secular stocks consist of stocks of companies that provide essentials. In other words, it provides people needs they can not live without. For example: consumer staples, and healthcare. Cyclical stocks typically relate to companies that sell inessential items consumers can afford to buy more of in a booming economy and cut back on during a recession. For example: travel, and luxury goods. A cyclical stock price is highly reactive to the economy. It requires strong economy to succeed. Seculars, however, are much less volatile and are stable regardless of the economy.

3.1.2 Making Profit

There are two main ways by which profit can be made with stocks.

1. Capital Gains

   Capital gains occur whenever the price of the underlying stock increases from the original purchase price. Capital gains depend on the investors' demand for the stock. It is important to note that in case the stock price increases, the gain is not realized until the stock is sold. A capital gain, once incurred, subjects the investor to a capital gain tax which varies from one company to another.

2. Dividends

   A dividend is a distribution of a portion of a company's earnings, decided by the board of directors, paid to a class of its shareholders. Dividends can be issued as cash payments, as shares of stock, or other property [8]. Dividends are decided by a company's board of directors. Dividends are used in times of profit to attract and keep investors. The more stable the company, the more dividends it would offer. On the other hand, a higher growth company will rarely distribute dividends in order to reinvest in the company to maintain its high growth. Dividends are based on a company's earnings.

3.1.3 Stock Types

A stock offered by a company can either be a common stock or a preferred stock. Common stock is issued by all publicly-traded companies, while preferred stock is only offered by some. Preferred stock provides a better shield from risk, when compared to common stock. However, it offers less potential for total return. Total return includes the capital gains and the dividends an investor receives from an investment. A common stock is more reactive to a company's share price movement. Companies offering common stocks may pay dividends, but are not required to do so. A company could even cut or eliminate the dividends, though it would be a rare scenario as it would provide a negative indication about the company's financial health. Dividend payments are usually guaranteed to preferred stockholders. Dividends are paid out to preferred stockholders before common stockholders. Compared to common stock, preferred stock is less reactive to a company's share price movement. In case a company goes bankrupt, the priority and obligations go first to the preferred stockholder before the common stockholder. Stockholders with capital gains in mind often go for a common stock, but stockholders looking for dividends prefer preferred stock.

3.1.4 Stock Prices Movement

At the most basic level, stock prices move based on market forces. Market forces is a term used to coin supply and demand. A stock price will move depending on its supply and demand. If the demand of a stock is higher than its supply, the price increases. On the other hand, if the supply of a stock is higher than its demand, the price decreases.

> *It takes big demand to move supply up, and the largest source of demand for stocks is by far the institutional buyer*
>
> — William O'Neill

The major influence on the stock price is caused by institutional buyers, due to the massive volume by which they trade. Institutional buyers include mutual funds, pension funds, and banks. A mutual fund is an investment vehicle made up of a pool of money collected from many investors for the purpose of investing in securities such as stocks, bonds, and other assets. Mutual funds are operated by professional money managers, who allocate the fund's investments and attempt to produce capital gains and/or income for the fund's investors. A mutual fund's portfolio is structured and maintained to match the investment objectives stated in its prospectus [9]. Pension funds usually invest with huge sums of money, and are the major investors in public companies.

Price movement of a stock indicates what investors feel a company is worth. The stock price of a company shows the company's current value, and takes into account future growth. Publicly-traded companies report on earnings once every year quarter. Stock prices are highly correlated to companies earnings. Investors review earnings to evaluate and compare between companies. Investor sentiment and expectations are what ultimately affect a stock price.

3.1.5 Stock Exchanges

The stock exchange is an exchange for brokers and traders that trade stocks. It serves the purpose of ensuring fair and orderly trading and disseminating price information of any stock on the exchange. The exchange gives companies a platform to sell their stock to the investing public. In order to trade a stock, it must be listed in the exchange. There usually used to be a central location for record-keeping but it is currently becoming less linked to a physical place as markets are now electronic networks, which gives the advantage of speed and reduced cost for transactions.

When a company chooses to go public, they go through an Initial Public Offering (IPO). The IPO takes place in primary markets. The primary market is the market that issues new stocks on the exchange. An investment bank sets the beginning price for the stock and sells it to the investors. Once the IPO is done, the actual trading of the stock happens in the secondary market, which is the stock market. In addition, market transactions between brokers and dealers as well as large institutions can be carried out through the Over-The-Counter (OTC) market. The OTC market is a decentralized market. OTC markets are much less transparent and are subject to a few regulations. Tradings in an OTC market does not involve a formal exchange. Exchanges are located all around the globe, some of the famous ones are New York Stock Exchange (NYSE), NASDAQ, and Tokyo Stock Exchange (TYO). The NYSE is primarily auction-based, with specialists being physically present on the exchange trading floor. Each of these specialists specializes in a particular stock, buying and selling it in the auction. The NASDAQ is another large exchange in the US. It is an electronic exchange and sometimes called screen-based because buyers and sellers are connected only by computers.

In addition to buying and selling, a stock can also be shorted. If an investor believes that a certain stock's price will decline, the investor can sell that stock by borrowing it before the price falls. The investor can then buy the stock shorted at a lower price to return it to the lender, thus making profit.

3.1.6 Initial Public Offering

The Initial Public Offering (IPO) takes place in the primary market and basically occurs when a company chooses to list itself on the stock market. Any company that is created and incorporated is initially a private company, meaning that it is completely controlled by a group of owners. The reason why it is considered private and not public is that no other person can come in and buy a stake in the company unless there is a special agreement. At the same time, no shares are traded on the stock exchange. On the other hand, a public company simply means that it trades on the exchange. A public company has some characteristics. The first is that the company is listed on a stock exchange. Another vital characteristic is that a public company can be owned by thousands if not millions of different people, all of whom become shareholders. This means that the original owners lose some control of the company, since anyone can buy shares in the company.

Companies choose to go public for many reasons, the most significant being raising capital to expand a company's operations. To raise capital, a company may choose to raise either debt or equity. Debt basically means taking a loan from a bank, and equity means issuing shares in the company and allowing the people to become shareholders. The advantage of equity over debt is that equity does not have to be paid back. An IPO can represent a great buying opportunity. The biggest issue that an average investor can face when trying to invest in an IPO is actually getting a hold of shares to buy. Since it's become so competitive, priority is given to large institutions. All public companies are required by law to report all financial information, whether it be revenue, expenditures or assets to the public. An investor should take a look at these results to understand how a company operates and determine whether it would make a good investment.

3.1.7 Financial Statements

The three main financial statements are the balance sheet, income statement, and cash flow statement.

- Balance Sheet

  A balance sheet basically consists of assets, liabilities, and stockholders' equity. An asset is a any resource with economic value that a company owns or controls for the company's benefit. A liability is a company's financial debt or obligations that are accumulated as the company carries out its business operations. Stockholders' equity represents the capital received from investors in exchange for stock, capital and retained earnings. The balance sheet equation is:

  $$equity = assets - liabilities \tag{3.2}$$

  Assets are split into three main categories: current assets, other assets, and property. Current assets are assets that can be converted to cash quickly, that is within one year. Current assets include cash, inventory, marketable securities, prepaid expenses, and other liquid assets that can be easily converted to cash. Liabilities are split into three main categories: current liabilities, long-term liabilities, and deferred income taxes. Current liabilities are obligations that have to be paid within one year. Current liabilities include production costs, payroll, taxes, and other short-term obligations. A balance sheet is a measure of a company's overall strength.

- Income Statement

  The income statement contains two key financial metrics: revenues and expenses. An income statement provides the investor with insight on how a company is receiving and spending money. The revenue is what is going in the company, expenses are what is going out of the company. An income statement can help an investor figure out whether a company can grow in the future or not. An income statement also includes earnings, which is one of the most important measures for an investor. For a stock to advance, earnings have to increase constantly. Earnings are listed as net income in the income statement.

  $$income_{net} = income_{operating} - taxes \tag{3.3}$$

- Cash Flow Statement

  The cash flow statement is split into three main parts: operating activities, investing activities, and financing activities. Overall, what is taken in is cash flow from operating activities, and what is taken out is cash flow from investing and financing activities. The cash flow statement shows inflows and outflows.

Features that make a company attractive to investors include high level of assets relative to liabilities, high net income growth, and high degree of cash inflows.

3.1.8 Types of Trading

In trading and investing, there are many different types of strategies that market participants employ. The two main types of investing are fundamental-analysis-based investing, and technical-analysis-based investing.

Fundamental analysis mainly involves investigating a company's financial statements and looking at different aspects of a company, such as revenue, assets, expenses, liabilities, etc. It involves a wide range of calculations to figure out stock prices in the future. Fundamental analysis is based on the concept that the price of a stock does not fully reflect its real value. The real value is called the intrinsic value. Stock price is compared to the intrinsic value to determine whether the stock price is overvalued, or undervalued. The largest assumption in fundamental analysis is that in the long run, the stock will always reflect the fundamentals. But the "long run" is an indefinite period of time. An undervalued stock would represent a good buying signal, and an overvalued stock would represent a good selling signal. This is based on the fundamental analysis assumption the market would catch up to fundamentals over time.

On the other hand, technical analysis does not care about the value of the company or the company as a whole. Technical analysis is mainly concerned with price movements. In order to attempt to determine future direction or trends, technical analysts study the supply and demand of a stock. Technical analysts evaluate stocks by analyzing the statistics generated by market activity, such as price and volume. Technical analysis is based on three assumptions: the market represents everything, price moves in trends, and history tends to repeat itself. Technical analysts only consider price movements and ignore the fundamental factors of a company. This is based on the assumptions that at any given time, the stock price reflects everything that has or could affect the company, stock prices move in the direction of a trend, and stock price movements occur over and over again throughout history.

Usually fundamental analysis is used for long-term investing, while technical analysis is used for short-term investing.

The main reference used in section 3.1 is Stock Market Investing for Beginners Udemy course [6].

3.2 Machine Learning

Arthur Samuel described machine learning as the field of study that gives computers the ability to learn without being explicitly programmed. In general, any machine learning problem can be assigned to one of two broad classifications: supervised learning and unsupervised learning.

- Supervised Learning

  Supervised learning relies on using a dataset with an established relationship between the inputs and the output, which indicates what the correct output should look like for a set of inputs. Supervised learning is categorized into regression and classification.

  A regression learner attempts to predict results within a continuous output, it tries to map input variables to some continuous function.

  A classification learner attempts to predict results in a discrete output. In other words, it tries to map input variables into discrete categories.

- Unsupervised Learning

  Unsupervised learning approaches problems with little or no idea what the results should look like. A structure is driven from data where the effect of the variables is not necessarily known. This structure is driven by clustering the data based on relationships among the variables in the data. With unsupervised learning, there is no feedback based on the prediction results.

  The clustering approach receives a collection of different items, and tries to find a way to automatically group these items into groups that are somehow similar or related by different variables.

  The non-clustering approach tries to find structure in a chaotic environment.

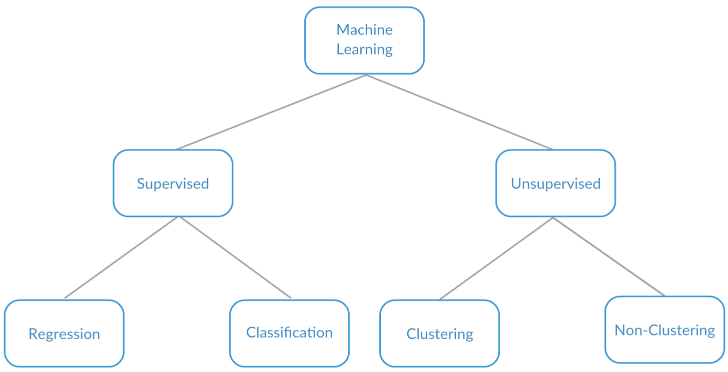


Figure 3.1: Machine Learning

The main reference used in section 3.2 is Machine Learning Coursera course [10].

3.2.1 Supervised Machine Learning

- Model Representation

  A machine learning algorithm requires one or more input variables (denoted as x_i), also called input features, and produces an output variable or prediction target variable (denoted as y_i). A pair (x_i, y_i) is called a training example, and the dataset to be used to learn is a list of n training examples (x_i, y_i); $i = 1, ..., n$, this is called a training set. The subscript (i) in the notation is simply an index into the training set.

  With a supervised learning problem, the goal is given a training set, learn a function $h : X \to Y$ so that $h(x)$ is a good predictor for the corresponding value of y. For historical reasons, this function, h, is called a hypothesis. The process is depicted in figure 3.2 below.

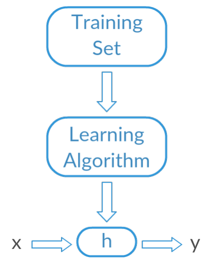


Figure 3.2: Model Representation

  When the target variable to be predicted is continuous, the learning problem is called a regression problem. When the target variable can take on only a small number of discrete values, the learning problem is called a classification problem.

- Cost Function

  The accuracy of a hypothesis function can be measured by using a cost function. A cost function computes an average difference between the results of the hypothesis and the actual output. Mean Squared Error (MSE) is a popular cost function.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(h(x_i) - y_i)^2 \tag{3.4}$$

  where:

  i = index into the training set
  n = number of training pairs
  $\hat{y}_i$ = predicted output of index i
  y_i = actual output of index i
  h = the hypothesis function
  x_i = input features of index i

- Optimization Algorithm

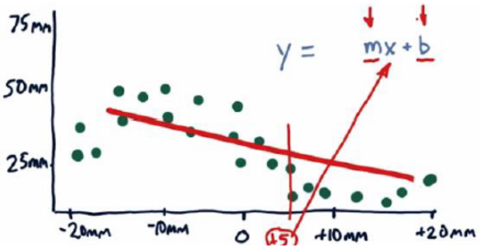
  An optimization algorithm minimizes a cost function by finding the optimal coefficients of a hypothesis function. Any hypothesis function has internal learnable coefficients, which are used in computing the target output values. An optimization algorithm's job is to find the optimal values for these coefficients. The cost is minimized with respect to the coefficients of the hypothesis function to compute the optimal coefficients of the hypothesis function that would provide the best fit to the training examples.

3.2.2 Regression

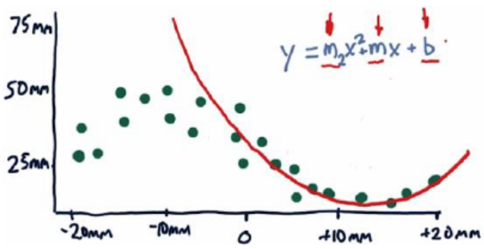
Regression uses the data to build a model that predicts a numerical output based on a set of numerical inputs. Regression can be parametric or non-parametric.

- Parametric Regression

  Parametric regression is where the model is represented with a number of parameters. Given a dataset where you have a dependent variable and some independent variables, fitting a line to this dataset can provide an estimation on the relation between the inputs (independent variables) and output (dependent variable). The fit line can be a linear line or a polynomial with a higher degree to achieve better accuracy. Linear parametric regression is shown in figure 3.3(a), and polynomial parametric regression is shown in figure 3.3(b). The fitted line is the model that can be queried for making predictions on the the dependent variable, when given the independent variables. For linear regression, the parameters of the model are simply the slope and the y-intercept of the line. For polynomial regression, the parameters are the coefficients of the polynomial function of the fitted polynomial line.



(a) Linear



(b) Polynomial

Figure 3.3: Parametric Regression

- Non-Parametric Regression

  The non-parametric approach is a data-centric or instance-based approach where the data is used when querying; unlike the parametric approach, which uses the data to create the model, and directs all querying to the model without consulting the dataset. An example of a non-parametric regression algorithm is kNN regression. kNN regression consults the dataset with each query. The dataset can not contain all possible input values, so whenever a query that involves an unknown input is made, all k nearest inputs' outputs are considered and a mean of the outputs is returned. k is a natural number that denotes the number of nearby elements to consider. Figure 3.4(a) below is an example where $k = 3$.

(a) Query Example

(b) kNN Approximation

Figure 3.4: kNN

If kNN were to be used to make predictions many points along the x-axis, the resulting curve would fit between the data points smoothly as shown in figure 3.4(b) above. When compared with parametric regression, kNN could provide a better fit to the training dataset, since in most causes it would be impossible to deduce a linear or a polynomial function that can fit all the data points as smoothly as kNN does.

When deciding on an approach to solve a regression problem whether by a parametric model or a non-parametric model, a few factors should be considered. The parametric model should be favoured whenever the relationship of the input(s) and output can be defined clearly in terms of a mathematical function. To compare between the two approaches, three things should be considered: storage space, querying speed, and adjusting to new training examples speed.

For a parametric approach, the training set does not have to be stored, so it's very space efficient, but the model can not be updated as more data is gathered. A complete rerun of the learning algorithm has to be done to update the model, thus for parametric approaches, training is slow but querying is fast.

For non-parametric approaches, the entire training set should be stored, so it's hard to apply when the data set is huge, but new training examples can be added easily since no parameters need to be learned, thus training is fast, but querying is potentially slow. Additionally, non-parametric approaches avoid having to assume a certain type of model, whether it's linear or polynomial. Therefore, they're suitable to fit complex patterns where the underlying model is unknown.

Section 3.2.2 reference is Machine Learning for Stock Trading Udacity course [11].
Figures 3.3 and 3.4 are from Machine Learning for Stock Trading Udacity course [11].

3.2.3 Artificial Neural Network (ANN)

An ANN is a computational model. It is artificial in the sense that it tries to process information in the same way biological neural networks in the human brain do. The basic unit of computation in an ANN is the neuron, also known as a node or unit. It receives input either from some other nodes, or from an external source, and a bias to provide every node with a trainable constant value. Each input has an associated weight, which is assigned based on its relative importance to other inputs. A node applies a function called the activation function to the weighted sum of its inputs. The activation function serves the purpose of introducing non-linearity into the output of a neuron. The output of the neuron is the output of its activation function. It is important to introduce non-linearity, because most real world data is non-linear and neurons should be capable of learning these non-linear representations. An ANN consists of three types of nodes: input, hidden, and output.

1. Input Nodes

   The input nodes transmit the dataset into the ANN. Input nodes are referred to as the input layer. No computation is performed in any of the input nodes, they just pass on the dataset to the hidden nodes.

2. Hidden Nodes

   The hidden nodes perform computations and transfer information from the input nodes to the output nodes. A group of hidden nodes forms a hidden layer.

3. Output Nodes

   The output nodes are collectively referred to as the output layer. They are perform computations and transfer the output out of the ANN.

In this research, we are mainly interested in two different types of ANNs: Feedforward Neural Network (FFNN), and Long Short Term Memory (LSTM) Recurrent Neural Network (RNN).

- FFNN

  The FFNN is the first and simplest type of an ANN. It contains multiple neurons arranged in layers. There are connections or edges between nodes from neighboring layers. There are weights associated with all these connections. In a FFNN, the information moves only in the forward direction from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the FFNN.

A Quick Introduction to Neural Networks [12]

- RNN

  RNNs are ANNs with loops in them, allowing information to persist. A loop allows information to be passed from one step of the network to the next. A RNN can be thought of as multiple copies of the same network, each passing a message to a successor. This chain-like nature reveals that a RNN in its core design is related to sequences and lists. RNN's natural architecture makes it optimal for datasets that involve sequences, lists, or time series. One of the applications of RNNs is that they might be able to connect previous information of a previous task to a present task. For example, using previous neurons' inputs might help in the understanding of the present neuron input. However, one important thing to consider is how far are the related tasks from each other. In some cases, only recent information is needed to perform the present task. In such cases, where the gap between the relevant information and the place where it is needed is small, RNNs can learn to use the past information. However, there are cases where more context is needed. It is feasible for the gap between the relevant information and the point where it is needed to become very large. Unfortunately, as that gap grows, RNNs become unable to learn to connect the information.

  - LSTM

    LSTMs are a special kind of RNNs. They are capable of learning long-term dependencies. LSTMs are capable of handling the long-term dependency problem. They specifically remember information for long periods of time. All RNNs have the structure of a chain of repeating modules of neural network. What is special for LSTMs is the cell state, the cell state runs straight through the entire chain, with only some minor linear interactions. It is very easy for information to just flow along it unchanged. An LSTM can remove or add information to the cell state. This is controlled by structures called gates. Gates are a way to optionally let information through. They are constructed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, that controls how much of each component should be let through. A value of zero lets nothing through, while a value of one lets everything through. An LSTM has these gates to protect and control the cell state.

Understanding LSTM Networks [13]
The main reference used in section 3.2 is Machine Learning Coursera course [10].

3.3 Machine Learning for Stock Trading

3.3.1 Computational Investing

Stock investing is the act of committing money or capital to trading stocks with the expectation of obtaining a profit. Computational stock investing is investing with the help of computations to improve investing efficiency and reduce risks. It relies on the use of statistics and market indicators to improve profit opportunities.

One of the most important aspects of stock trading is the company value of a company one chooses to trade its stocks. On the most basic level, an investor should aim to buy a stock when it is undervalued, that is when the market price of a stock is less than the actual value of a stock, and to sell a stock when it is overvalued, that is when the market price of a stock is greater than the actual value of a stock. The are a lot of mechanisms to estimate the true value of a company. One mechanism that can be used is computing the intrinsic value of a company. Intrinsic value is computed by different methods; the method discussed here is based on dividends. To discuss the concept of intrinsic value, the equation of the future value is first discussed.

$$FV = PV \times (1 + IR)^i \tag{3.5}$$

where:

FV = future value
PV = present value
IR = interest rate
i = some duration count

This equation is used to compute the future value of an investment. The interest rate should correspond to how risky the investment is. The interest rate is applied each i, i can be a number of years, months, or any other duration. The interest rate is directly proportional to the the risk involved with the investment. We use this equation in the following form: $PV = \frac{FV}{(1+IR)^i}$ to find out what is the value of the dividend that is going to be paid by the company in the future. If this equation were to be applied to get the sum of dividends paid each duration i from now till the very far future where $i = \infty$, we would get an equation that would allow us to estimate the intrinsic value of a company.

The main reference used in section 3.3 is Machine Learning for Stock Trading Udacity course [11].

Figures used throughout section 3.3 are from Machine Learning for Stock Trading Udacity course [11].

$$value_{intrinsic} = \sum_{i=1}^{\infty} \frac{FV}{(1+IR)^i} = \frac{FV}{IR} = \frac{FV}{DR} \tag{3.6}$$

where:

$value_{intrinsic}$ = intrinsic value
DR = discount rate, which is another term for an interest rate

The true value of company can also be estimated by the book value, which is simply $assets_{total} - liabilities_{total}$. The market value of a company can be estimated by using the market capitalization equation 3.1. Many stock trading strategies look for deviations between intrinsic value and market capitalization. If intrinsic value drops significantly and the stock price is high, it may be worthwhile to short that stock. If dividends are going up and the market capitalization is low, it might be an opportunity to buy the stock. Book value provides a measure of the lowest price. When stock price begins to approach book value, it can be assumed that the price is not going to go below book value. Afterall, book value is the pricing of a company in terms of what it owns.

3.3.2 Technical Analysis

The machine learning algorithms to be developed will mainly depend on technical analysis as input to come up with predictions. As mentioned before in section 3.1.8, technical analysis is entirely concerned with price and volume. Historical prices and volumes of a stock are studied to compute statistics on a time series, and these statistics are called indicators. Indicators are heuristics that may hint at a buy or sell opportunity. There are some factors to note for the effective use of technical indicators. Firstly, individual indicators are weak. Combining multiple indicators adds value. Combinations of three to five different indicators, in a machine learning context, provide a much stronger predictive system than just an individual indicator. Secondly, contrasts should be given further attention when analyzing stocks. In other words, if all stocks are behaving in the same way as the market, there's no reason to pick any one stock over another, but if certain stocks are behaving differently than the market, then they are worth a further look. Finally, technical analysis should be used over short time periods, because generally it works better over shorter time periods than longer time periods.

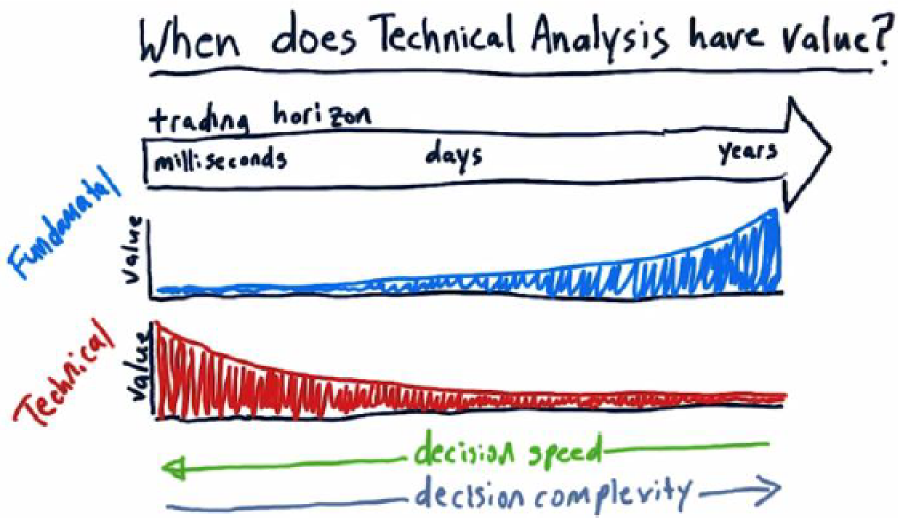


Figure 3.5: Fundamental and Technical Analysis Effectiveness over Time

As shown in figure 3.5, fundamental analysis has low value over short periods of time, and significant value over long periods of time. On the other hand, technical analysis' effect on the price of a stock over long terms is not so valuable. Over very short periods of time, it potentially has high value. Other factors to consider are decision complexity and decision speed. The decision to buy or sell a stock if you're going to hold it for years becomes more complex going from left to right. Decision speed, which is how fast do the decisions have to be made, increases going from right to left.

3.3.3 Technical Indicators

A technical indicator is a mathematical calculation that is computed by using a stock's price or volume. Technical indicators do not measure fundamental factors such as revenue, assets, and expenses. As mentioned before in section 3.3.2, technical indicators which are a part of technical analysis are highly effective when it comes to short-term price movements. Technical indicators alert the trader about a trend, predict the direction of future prices, and confirm technical analysis suggested by other indicators. Technical indicators can be categorized in two categories: leading and lagging. Each technical indicator also belongs to one of four types: trend, momentum, volatility, and volume [14].

- Technical Indicators Categories

  1. Leading

     Leading indicators give trade signals when a trend is about to start. They try to predict price by using a shorter period in their calculation, thereby leading the price movement.

2. Lagging

   Lagging indicators follow the price action. They give a signal after the trend or reversal has started. They are used to determine a trend.

- Technical Indicators Types

  1. Trend

     By using an averaging mechanism to set up a baseline, trend indicators measure the direction and strength of a trend. As price moves above the average, an upwards trend signal is generated, conversely as price falls below the average, a downwards trend signal is generated.

  2. Momentum

     By comparing prices over time, momentum indicators identify the speed of price movement. Momentum indicators are generally calculated by computing a ratio between current closing prices and previous closing prices.

  3. Volatility

     Volatility indicators are mainly concerned with the rate of price movement, regardless of direction. They are mainly based on the difference between the highest and lowest prices. They indicate the range of trading of a certain stock in the market. Volatility indicators often help in determining points where a certain stock price may change direction.

  4. Volume

     Volume indicators measure the strength of a trend by taking a volume-analysis approach. Volume increase is highly correlated with strong trends. Large movements in price are often caused by an increase in trading volume.

3.3.4 Building a Machine Learning Model

Supervised machine learning will be used to create models, as discussed in section 3.2.1, that can predict future prices for stocks. Features that can be used as input variables, (x_i), include the technical indicators and the price. Remember that the data or the training set provided to the machine learning algorithm to build the model is supposed to be the true values that map a certain input or group of inputs to a certain output. When a tuple (x_i, y_i) is provided to the machine learning algorithm during the training phase, it is important to note that the actual output to x_i is y_i. When this is applied to the stock trading problem, the desirable output is a future price prediction. So for example, if x_i encapsulates features at day t, y_i is supposed to be the price at day $t + T$, where T is a number of days. Using this technique while training the machine learning algorithm to build the model allows mapping current inputs to future prices. The predictive factor is integrated into the model. As shown in figure 3.6, the gap chosen between x_i and y_i in a tuple is 5 days, $T = 5$, for this example.

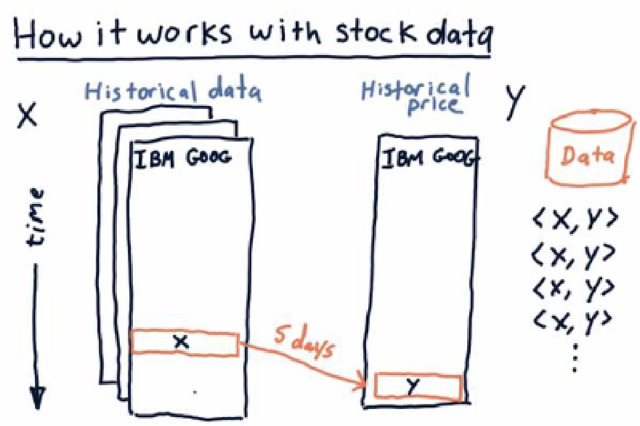


Figure 3.6: Building the Training Set

To build a machine-learning based forecaster. The first step is to select which factors are to be used as input, these are encapsulated as x_i. The next step is to select what should be predicted. Usually, predictions involve change in price, market relative change in price, or future price. An entry of input(s), x_i, is paired with the relative output entry, y_i, to form a tuple, (x_i, y_i), which is called a training example. The training set which is a set of training examples is the data used to train the model. Next, the breadth and depth of the data that is going to be used to train the system should be considered. That includes, for instance, time period. How far back in time does the training phase consider? What is the stock universe? What universe of data, and which symbols are going to be used to train the system as well? After considering the previous steps, the machine learning algorithm is trained with the data to produce the model. The algorithm takes the data and converts it into a model. The model, once produced, is ready to make predictions. The x_i of the present is plugged into the model and the model should provide y_i, the future prediction. The overall process is shown in figure 3.7 below.

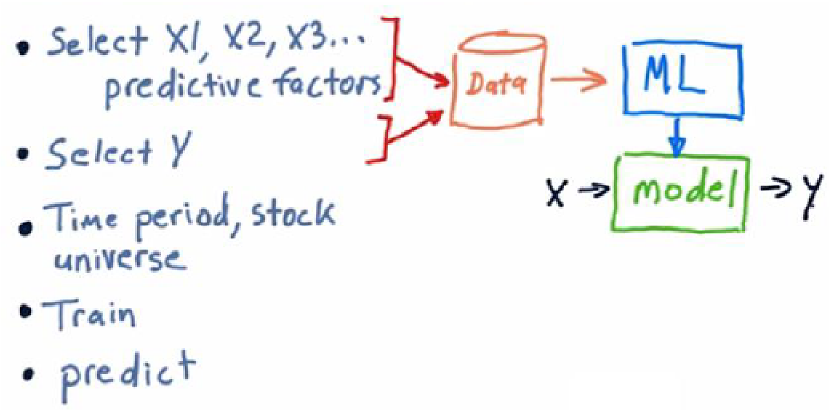


Figure 3.7: Building the Model

3.3.5 Training and Testing

In order to evaluate a learning algorithm in a scientific manner, data should be split into at least two sections: a training section and a testing section. If training is done over the same data over which testing is applied, the results would be suspicious because it should be able to do very well. The procedure of separating testing and training data from one another is called out-of-sample testing. This is a very important and essential technique.

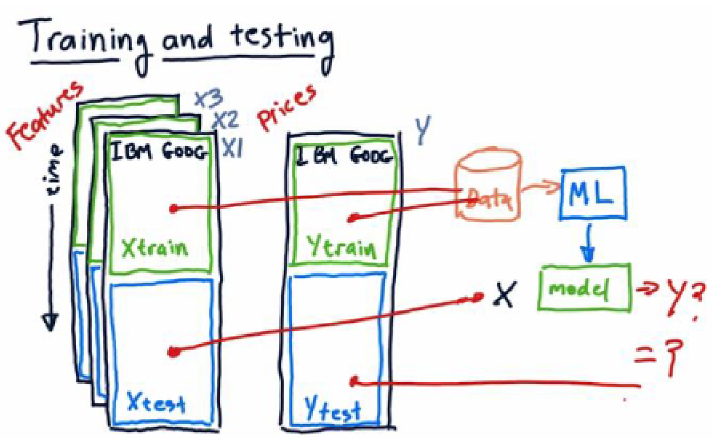


Figure 3.8: Training and Testing

Figure 3.8 above shows an overall schematic of the process of using a machine learning algorithm to produce a predictive model, and training and testing the model. The process of building the model was discussed in section 3.3.4. X_{train} and X_{test} are used to denote the sections of the input features that are used in the training phases, and the testing phases respectively. Same goes for Y_{train} and Y_{test}. After using X_{train} data and Y_{train} data by the machine learning algorithm to generate a model, the accuracy of that model is tested using the X_{test} data and Y_{test} data. During the testing phase, the input to the model is a value from the X_{test} section, and what the model provides is $y_{prediction}$. And the question is, is that $y_{prediction}$ equal to the value from Y_{test}, which is known to be the ground truth and corresponds to the input value given to the model from the X_{test} section. The more closely the model outputs $y_{prediction}$s that reflects data from Y_{test}, the more accurate the model is. Note that the data is time oriented. So, as we move downward in figure 3.8 above, we're going forward in time. The data is typically split up according to time. The model is trained on older data and tested it on newer data.

The main reference used in section 3.3 is Machine Learning for Stock Trading Udacity course [11].

Figures used throughout section 3.3 are from Machine Learning for Stock Trading Udacity course [11].

Chapter 4

Implementation

4.1 Raw Data

The process of building the dataset starts with fetching the raw data. The data in the raw form consist of historical prices of a company in the stock market. They are obtained from Yahoo! Finance. The data can be represented as a table depicting a daily time series where a row entry represents a trading day. The columns of the table are:

- Date : trading day date
- Open : opening price pre-market trading day
- High : highest price of the stock throughout the trading day
- Low : lowest price of the stock throughout the trading day
- Close : price of the stock at market close
- Adj Close : closing price adjusted for splits and dividends
- Volume : the frequency of transactions of the stock throughout the trading day

Table 4.1 below shows a sample of the raw data. The sample shown below belongs to Alphabet Inc.

Date	Open	High	Low	Close	Adj Close	Volume
2004-08-19	49.676899	51.693783	47.669952	49.845802	49.845802	44994500
2004-08-20	50.178635	54.187561	49.925285	53.805050	53.805050	23005800
2004-08-23	55.017166	56.373344	54.172661	54.346527	54.346527	18393200
2004-08-24	55.260582	55.439419	51.450363	52.096165	52.096165	15361800
2004-08-25	52.140873	53.651051	51.604362	52.657513	52.657513	9257400

Table 4.1: Raw Data Sample

4.2 Building the Dataset

The raw data is read and traversed to build the dataset. The dataset is indexed by the date column, and columns of the raw data not needed to compute the technical indicators are discarded; these columns are: Open, High, Low, and Close. We begin by computing the the technical indicators. Section 4.2.1 below discusses the indicators used to build the dataset.

4.2.1 Technical Indicators

The indicators used are : momentum, Simple Moving Average (SMA), Bollinger Bands (BBs), volatility, and Volume Rate of Change (VROC). Each Indicator will be discussed, and its use will be justified. Indicators' categories and types based on what was discussed on section 3.3.3 will also be mentioned.

- Momentum

  Momentum is a measure of how much has the price changed over some number of days. Momentum can be either positive or negative. By looking at the recent momentum, an investor may detect trading signals indicating the direction a price is taking. To compute momentum, the price of a particular day is divided by the price of some days earlier and subtracted by 1. Momentum is a leading momentum indicator.

  $$m_t = \frac{price_t}{price_{t-n}} - 1 \tag{4.1}$$

  where:

  t = any given day
  n = number of days; t > n and n $\geq$ 1
  m_t = momentum ratio of day t

- SMA Ratio

  SMA is computed over n, which is how many days are we looking back. The value of the SMA at a particular day involves looking back over n days, and this is called an n-day window. There are at least two different ways that technicians use SMA as a part of trading strategies. The first is they look for places where the current price crosses through the SMA. Those tend to be important events, especially if the average is over many days. SMA and momentum can be used together to produce trading signals. If for example price has strong momentum, and it's crossing through the SMA, that can be a signal that the price is in a current surge. Another way that technicians use SMA is as a proxy for underlying value. In other words, if you look back over a certain period of time and take that average price, that might represent the true value of the company. If we see a large excursion from that price, we should expect that the price is eventually going to come back down or up to that average.

SMA ratio is calculated by comparing the current price with the current SMA, and subtracting 1. SMA ratio is a lagging trend indicator.

$$\text{SMA}_t = \frac{price_t}{\sum_{i=t-n}^{t} \frac{price_i}{n}} - 1 \tag{4.2}$$

where:

$$\begin{aligned} t &= \text{any given day} \\ n &= \text{number of days; t > n and n} \geq 1 \\ \text{SMA}_t &= \text{SMA ratio of day t} \end{aligned}$$

- BBs Ratio

  BBs are SMA added to two-standard-deviations bands above and below. BBs represent a measure for how strong a deviation should be before responding to it. BBs offer trading signals when the price is outside one of these BBs and starts crossing back to the inside. When the price is outside the upper band, and it crosses back to the inside, this indicates a sell signal. Conversely, when moving from below the lower band back towards the SMA, that would be a buy signal.

  To calculate the BBs ratio on a particular day t, the value of the SMA at day t is subtracted from the value of the price on day t, and then divided by 2 times the standard deviation. BBs ratio is a lagging volatility indicator.

$$\text{BBs}_t = \frac{price_t - \text{SMA}_t}{2 \times std_{rolling}} - 1 \tag{4.3}$$

  where:

$$\begin{aligned} t &= \text{any given day} \\ std_{rolling} &= \text{rolling standard deviation} \\ \text{BBs}_t &= \text{BBs ratio of day t} \end{aligned}$$

- Standard Deviation

  Standard deviation is used to measure risk and determine the significance of certain price movements. The prices are grouped together in a group of n prices, n represents the window for computing the rolling standard deviation. Standard deviation is a lagging volatility indicator.

$$(std_{rolling})_t = \sqrt{\frac{\sum_{i=t-n}^{t} \mid price_i - price_{mean} \mid^2}{n}} \tag{4.4}$$

where:

t = any given day
n = number of days; t > n and n ≥ 1
$(std_{rolling})_t$ = rolling standard deviation of day t
$price_{mean}$ = mean of the prices in the summation

- VROC

  VROC is a measure of how much has the volume changed over some number of days. VROC can be either positive or negative. To compute VROC, the volume of a particular day is divided by the volume of some days earlier and subtracted by 1. VROC is a lagging volume indicator.

$$\text{VROC}_t = \frac{volume_t}{volume_{t-n}} - 1 \tag{4.5}$$

where:

t = any given day
n = number of days; t > n and n ≥ 1
VROC_t = VROC of day t

Different technical indicators have different ranges that they typically operate over. If values of a certain indicator operate in a range higher than the others, that indicator would tend to overwhelm these other indicators and become the most important one. The solution to this problem is something called normalization. Normalization takes each of these indicators and essentially compresses them or stretches them so that they vary on the same range.

4.2.2 Future Gap

The dataset consists of the features and the target output. The features are the price and the technical indicators mentioned above in section 4.2.1. The target output is the future price. An entry in the dataset which is indexed by the date consists of the features from an earlier day, and the price of the day indicated by the date index. As shown in the header of table 4.2, the features are all suffixed with $(t - i)$, the t indicates the day indexed by the dates index, and the i indicates that the features are from i days ago. The i is a variable that can be modified to choose a future gap; the future gap is used to integrate the predictive nature into the dataset. Since the model learns by associating the features of an entry to the output, the dataset is built with a predictive factor in mind. So for example, an entry in the dataset with index dd/mm/yyyy, has the features of day (dd/mm/yyyy - future gap) and the output(price) of (dd/mm/yyyy).

The dataset consists of different values that lay in different ranges, which can cause the model to become biased in the training process. Therefore, the dataset is scaled so that all the values lie in the range between 0 and 1. After training, and making the predictions, the prices are inverse-scaled to get the actual values back and compare between the actual and predicted prices. The dataset sample shown below in table 4.2 is built from the raw data sample shown in table 4.1. Values are rounded for a better display.

price(t-i)	moment(t-i)	SMA(t-i)	BBs(t-i)	std(t-i)	VROC(t-i)	price(t)
6.11e-03	3.99e-01	4.37e-01	9.99e-01	3.27e-01	6.27e-04	0.00316
3.16e-03	2.36e-01	2.65e-01	7.12e-06	2.43e-01	1.03e-03	0.00389
3.88e-03	4.01e-01	4.39e-01	9.99e-01	2.46e-01	7.73e-04	0.00514
5.14e-03	4.24e-01	4.63e-01	9.99e-01	3.42e-02	7.15e-04	0.00399
3.99e-03	3.16e-01	3.50e-01	7.12e-06	1.62e-01	1.04e-03	0.00130

Table 4.2: Dataset Sample

4.3 The Machine Learning Model

Different machine learning models were used to serve the purpose of finding the optimal model for the stock price forecasting problem. The models developed in this research are:

- Linear Regressor
- kNN Regressor
- FFNN
- LSTM RNN

Building the dataset for the models is done as discussed in section 4.2, except for the LSTM RNN model. The LSTM RNN works differently from the other models, and requires some changes in the dataset that will be discussed in section 4.3.4.

4.3.1 Linear Regressor

The linear regressor model is the most simple, primitive machine learning model. To implement the model as mentioned in 3.2.2, two functionalities have to be developed and integrated together. The first is an error function that calculates the error difference between the actual values and the model's predicted values during training. As mentioned in 3.2.2, the linear regressor model is a simple a linear function, where the independent variables are the dataset features, and the output is the price. The error function used is the Root Mean Squared Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_{p_i} - y_{a_i}\right)^2}{n}} \tag{4.6}$$

where:

i = dataset entry of index i
n = number of entries in the dataset
y_{p_i} = predicted value of entry i
y_{a_i} = actual value of entry i

The second thing needed is an optimizer. The optimizer used with the linear regressor can be described as a function, that takes as input an error function, and a function that assigns coefficients to the model. The optimizer then tries out several coefficients values to find out the optimal coefficients that minimize the error calculated by the error function.

The linear regressor model can be described by the function below.

$$f(p, m, s, b, d, v) = C_1p + C_2m + C_3s + C_4b + C_5d + C_6v + C_7 = p_f \tag{4.7}$$

where:

p = price
m = momentum
s = SMA ratio
b = BBs ratio
d = standard deviation
v = VROC
p_f = future price
C_i = constant coefficient

What decides how far into the future is the price is the future gap chosen when training. Below is a sample of the linear regressor model with optimal coefficients.

$$f(p, m, s, b, d, v) = 1.00p - 0.0822m + 0.092412s + 0.000129b + 0.00162d + 0.00205v - 0.0157 = p_f \tag{4.8}$$

4.3.2 kNN Regressor

The kNN regressor is a non-parametric regressor, that means it is instance-based. The model is not as explicit as other machine learning models. The training dataset is stored and used to make the predictions. So, here is the process that happens when making a prediction. First, recall that the training dataset values are all scaled and lie in the range between 0 and 1. This is especially important for kNN, since it involves combining values together, and the presence of positive and negative values would lead to inaccurate results. The features of the training set are all summed into into one value to represent all the features values in one. Secondly, looking at one query/prediction only to provide a clearer explanation, when a certain prediction is to be made, the features of that query are also summed up to one value. That value is then compared with all the values of the features summation column of the training set. Getting on a more technical level, the comparison is done by subtracting the query value from all the features summation column of the training set, and then the predicted price is computed as the mean of the future prices of the entries with the k smallest differences. The same is applied when dealing with the entire testing dataset, each entry in the testing dataset is dealt with as just described to produce the predicted future prices.

4.3.3 FFNN

The FFNN model has one input layer, with 6 nodes, one for each feature in the dataset. There are three hidden layers, the first two layers are 256 nodes each, and the last hidden layer is 64 nodes. The output layer consists of one node only, which outputs the predicted price. The activation function used in the hidden layers is the Rectified Linear Unit (ReLU). The ReLU activation function takes a single number as an input, returning 0 if the input is negative, and the input if the input is positive. The activation function used in the output layer is the linear activation function, which outputs the values as they are.

Overfitting is a common problem to any ANN. Overfitting occurs when an ANN specifically learns the detail and noise in the training dataset too well. That causes the ANN to be specifically tailored to the training dataset, which negatively impacts the performance of an ANN on new data [15]. To reduce overfitting, a dropout of 0.2 is added between the first and second hidden layers, and between the second and the third hidden layers. Dropout is a technique where a set of randomly chosen neurons are ignored during the training phase [16]. Adam optimizer is used with a learning rate decay of 0.1 to avoid overshooting the minimum loss, and to safely converge to the global minimum loss.

4.3.4 LSTM RNN

So far, the dataset used with the previous models is constructed in the same way. The dataset can be considered as a 2-dimensional array. For any machine learning model, the dataset is split into two sections: the features, X, and the target output, Y. For the previous models, the Y was the last column of the dataset, the $price(t)$ column in table 4.1. The X was the entire dataset except for the $price(t)$ column. The Y is similar between the previous models and the LSTM RNN model.

The X for the LSTM RNN is a 3-dimensional array. It can be described as a list of 2-dimensional arrays. The length of the 2-dimensional arrays list is called the *samples*. Each 2-dimensional array is of the same dimensions, the number of rows is called the *time steps* and the number of columns is called the $features$. So basically, you can think of the X as the normal X of the previous models, but for each entry/row in the X of the previous models, for the LSTM RNN there is a group of entries of a size indicated by the time steps. The three dimensions of an LSTM RNN are the samples, time steps, and features.

- **Samples** represent the number of entries in an X. Each entry is a 2-dimensional array.
- **Time steps** represent the number of rows in a single sample.
- **Features** represent the number of columns in a single sample.

To further clear the concept, consider an example where the X of the previous models is of dimensions (20, 6). For the LSTM RNN model the X with 5 time steps would be of dimensions (16, 5, 6)/(samples, time steps, features). The samples number can be computed as $n - t + 1$, where n is the number of rows of the traditional X, and t is the time steps. To create the LSTM RNN's X, the traditional X is traversed, taking the first t rows to make up the first sample. Then moving to the second row of the traditional X and collecting t rows to make up the second sample, and so on until reaching the final t rows of the traditional X, which make up the final sample. The future gap is taken into consideration by associating each sample of the X, with a target price that is g days later, where g denotes the future gap.

The LSTM RNN model has one input layer, where a single input is a 2-dimensional array, (time steps, features). There are three hidden layers, the first two layers are 256 nodes each, and the last hidden layer is 32 nodes. The output layer consists of node only, which outputs the predicted price. The activation function used in the hidden layers is the ReLU. The activation function used in the output layer is the linear activation function. To reduce overfitting, a dropout of 0.2 is added between the first and second hidden layers, and between the second and the third hidden layers. Adam optimizer is used with a learning rate decay of 0.1 to avoid overshooting the minimum loss, and to safely converge to the global minimum loss.

4.4 Training and Testing

In this brief section, the training and testing process of the four models is discussed. What is common among all the models is that the dataset is horizontally split into two parts: training and testing. The model is trained on the training dataset and is tested on the testing dataset, which is completely unknown to the model. Both datasets are also split vertically to two parts the features, X, and the target output, Y. The X is the entire dataset except for the $price(t)$ column, the Y is the $price(t)$ column.

- Linear Regressor
  - Training
    The optimizer is given the training dataset and the RMSE loss function to minimize the loss, and compute the optimal coefficients. Computing the optimal coefficients marks the end of the training phase of the linear regressor.
  - Testing
    Testing of the linear regressor model is done by simply going through the features of the testing dataset, and computing the expected price based on the model's coefficients. So for a single entry in the testing dataset, each feature value is multiplied by the model's coefficient for that feature, then all the weighted features are added together along with the model's constant, the result of that summation is the future predicted price. By doing so to all the testing dataset entries, we have an entire forecast for a specified time period that is dependent on the size of the testing dataset.
- kNN Regressor
  - Training
    The kNN regressor is a non-parametric model, which means that all the model does during the training phase is storing the training dataset to be able to base its forecasts on it during the testing phase. The features of the training set are all summed into into one value to represent all the features values in one.
  - Testing
    Testing of the kNN regressor model is done by going through the features of the testing dataset, and computing the mean of the k target prices of the entries in the training dataset, whose features are closest to the features of the entry in the testing dataset that we are trying to predict. So for a single entry in the testing dataset, all the features values are summed up to one value, then compared with all the values of the features summation column of the training set, the mean of the k target prices of the entries in the training dataset whose features summation value is closest to the features summation value of the testing entry being processed is the future predicted price. By doing so to all the testing dataset entries, we have an entire forecast for a specified time period that is dependent on the size of the testing dataset.

- FFNN
  - Training
    The FFNN was trained on the training dataset with a batch size of 128 for 200 epochs.
  - Testing
    The FFNN was tested using the testing dataset.
- LSTM RNN
  - Training
    The LSTM RNN was trained on the training dataset with a batch size of 2048 for 300 epochs.
  - Testing
    The LSTM RNN was tested using the testing dataset.

4.5 Evaluation

After the models make predictions, the predictions should be assessed and compared with the actual values. The metrics were computed on the normalized/sampled testing dataset. Below are the evaluation metrics used to evaluate the models.

- RMSE

  RMSE is a quadratic scoring rule that measures the average magnitude of the error. It expresses the model prediction error in units of the variable of interest, which for our case is stock price. Compared to other metrics, RMSE has the advantage of penalizing large errors more [17].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_{p_i} - y_{a_i}\right)^2}{n}} \tag{4.9}$$

where:

i = dataset entry of index i
n = number of entries in the dataset
y_{p_i} = predicted value of entry i
y_{a_i} = actual value of entry i

- Normalized Root Mean Squared Error (NRMSE)

  NRMSE is helpful when comparing between datasets with different scales or ranges [18].

$$\text{NRMSE} = \frac{\text{RMSE}}{\overline{Y_a}} \tag{4.10}$$

  where:

  $\overline{Y_a}$ = mean of the actual prices

- Mean Absolute Error (MAE)

  MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It expresses the model prediction error in units of the variable of interest, which for our case is stock price [17].

$$\text{MAE} = \frac{\sum_{i=1}^{n} \mid y_{p_i} - y_{a_i} \mid}{n} \tag{4.11}$$

  where:

  i = dataset entry of index i
  n = number of entries in the dataset
  y_{p_i} = predicted value of entry i
  y_{a_i} = actual value of entry i

- Mean Absolute Percentage Error (MAPE)

  MAPE measures the error size in terms of percentage [18].

$$\text{MAPE} = \frac{\sum_{i=1}^{n} \mid \frac{y_{p_i} - y_{a_i}}{y_{a_i}} \mid}{n} \times 100 \tag{4.12}$$

  where:

  i = dataset entry of index i
  n = number of entries in the dataset
  y_{p_i} = predicted value of entry i
  y_{a_i} = actual value of entry i

- Coefficient of Correlation (R)

  R measures the strength and direction of the linear relationship between the predicted and actual values. R values lie between -1 and 1, where -1 is total negative linear correlation, 0 is no linear correlation, and 1 is total positive linear correlation [19].

$$R = \frac{n\sum_{i=1}^{n}(y_{p_i} \times y_{a_i}) - (\sum_{i=1}^{n} y_{p_i}) \times (\sum_{i=1}^{n} y_{a_i})}{\sqrt{[n\sum_{i=1}^{n} {y_{p_i}}^2 - (\sum_{i=1}^{n} y_{p_i})^2] \times [n\sum_{i=1}^{n} {y_{a_i}}^2 - (\sum_{i=1}^{n} y_{a_i})^2]}} \tag{4.13}$$

  where:

  i = dataset entry of index i
  n = number of entries in the dataset
  y_{p_i} = predicted value of entry i
  y_{a_i} = actual value of entry i

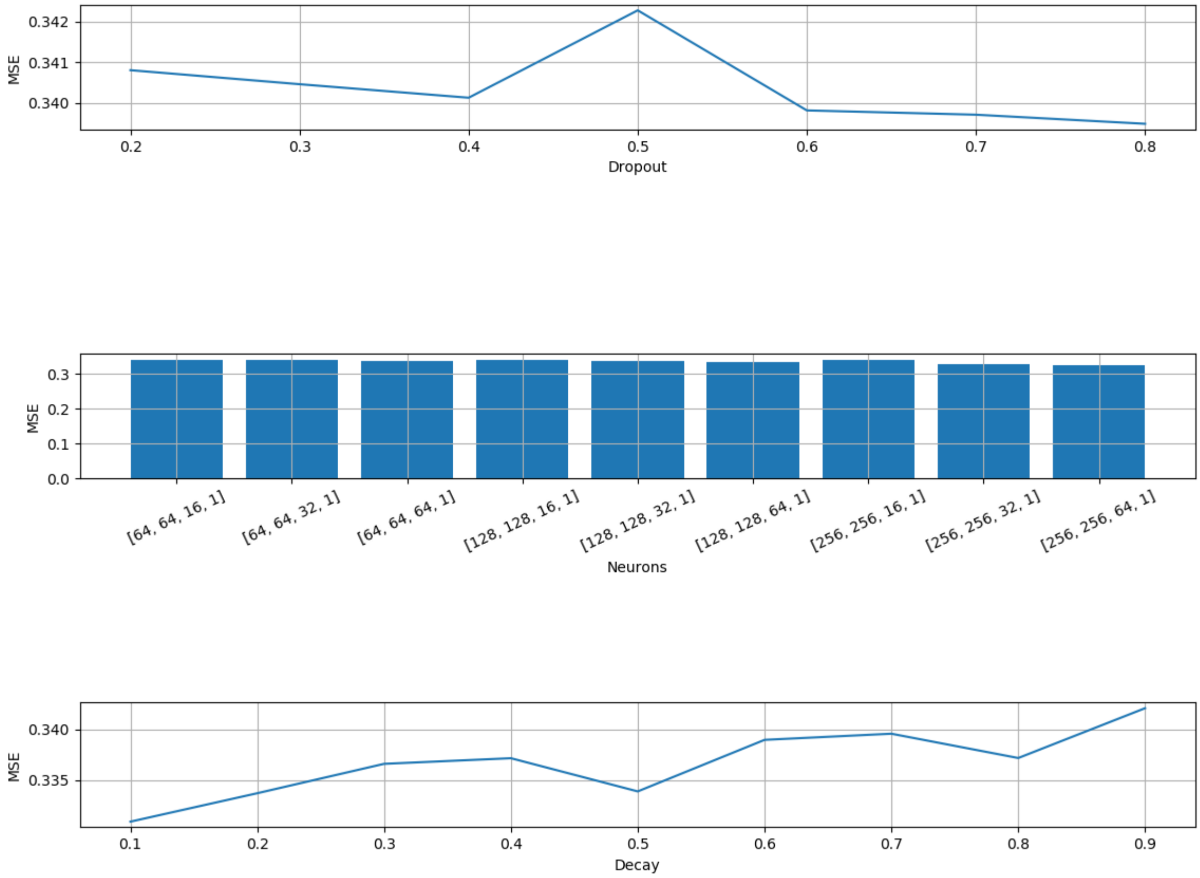
- Coefficient of Determination (R^2)

  R^2 is a statistic used to measure how close the actual prices are to the predicted prices. Its values lie between 0 and 1, where 0 indicates that the model explains none of the variability of the actual prices, and 1 indicates that the model explains all the variability of the of the actual prices. R^2 is the calculated as the squaring of R mentioned in the equation 4.13 [20].

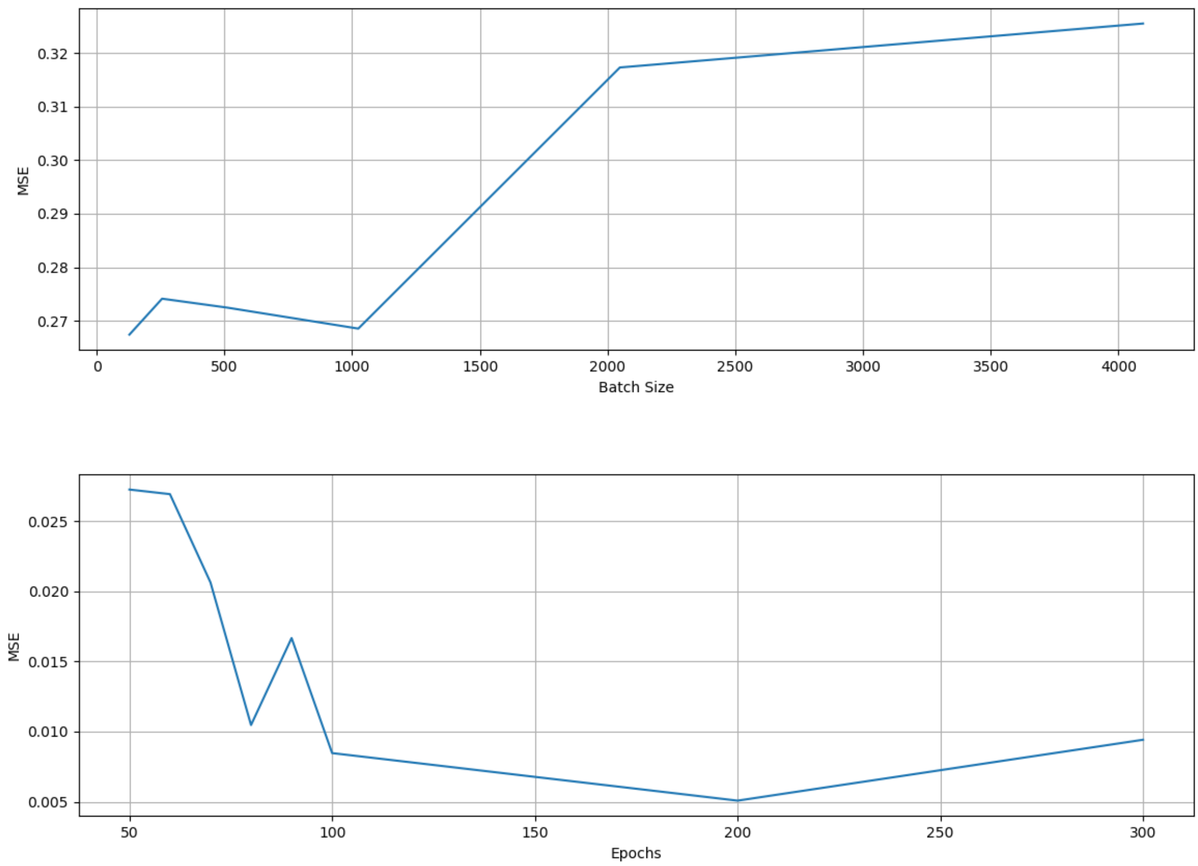
$$R^2 = (R)^2 \tag{4.14}$$

4.6 Artificial Neural Network Hyperparameter Tuning

Hyperparameter tuning is a process where a machine learning model is trained and tested to find a set of optimal hyperparameters. A hyperparameter is a parameter whose value is set before the training process begins. Hyperparameters include dropout, neurons, decay, batch size, time steps, and epochs. The hyperparameter tuning technique used here is called grid search. Grid search is where a set of values are chosen for each hyperparameter, and for each value, the ANN is trained and tested. The loss of the ANN with the different values of a hyperparameter are then compared to find the minimum loss, and thus the optimal hyperparameter. Shown below in figure 4.1 is the hyperparameter tuning of the FFNN model, and in figure 4.2 the hyperparameter tuning of the LSTM RNN model. Table 4.3 shows the optimal hyperparameters for the FFNN model, and table 4.4 shows the optimal hyperparameters for the LSTM RNN model.



(a) Dropout, Neurons, and Decay



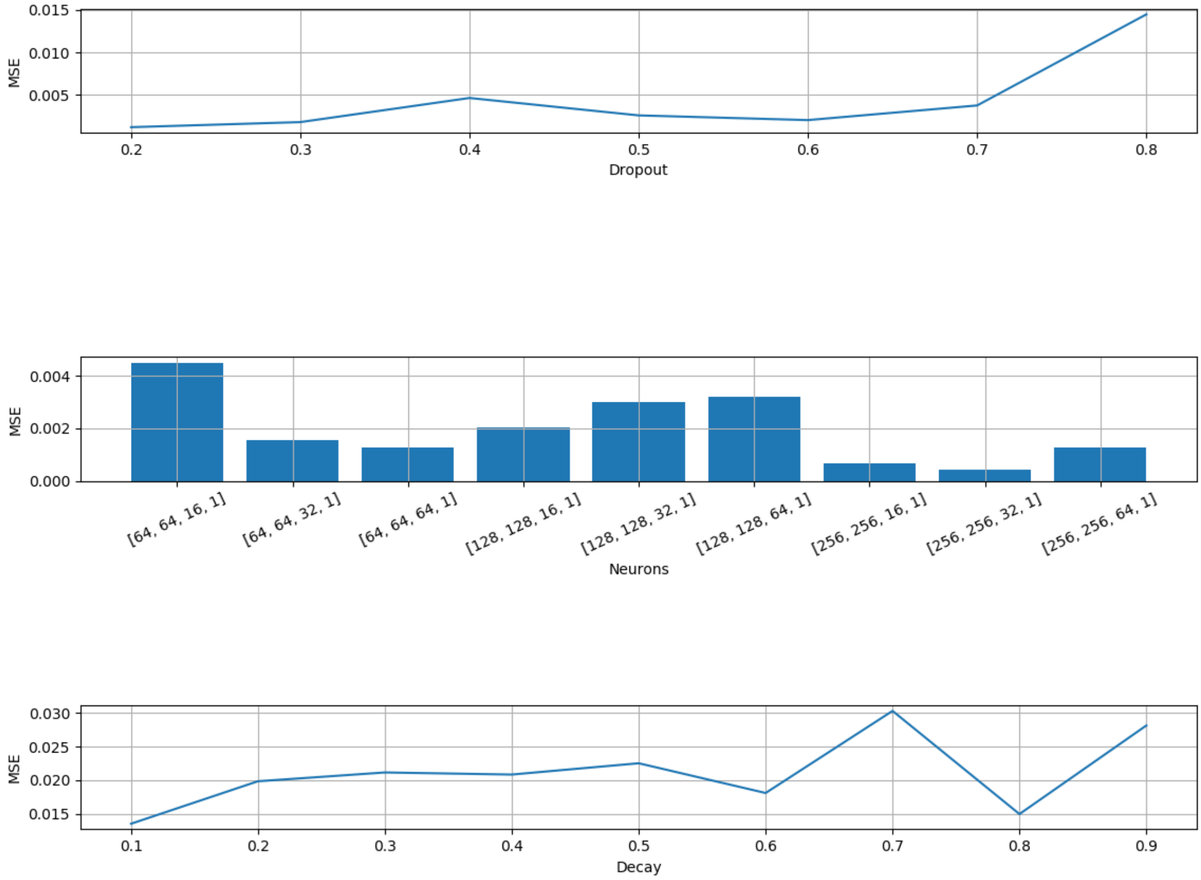
(b) Batch and Epochs

Figure 4.1: FFNN Hyperparameter Tuning

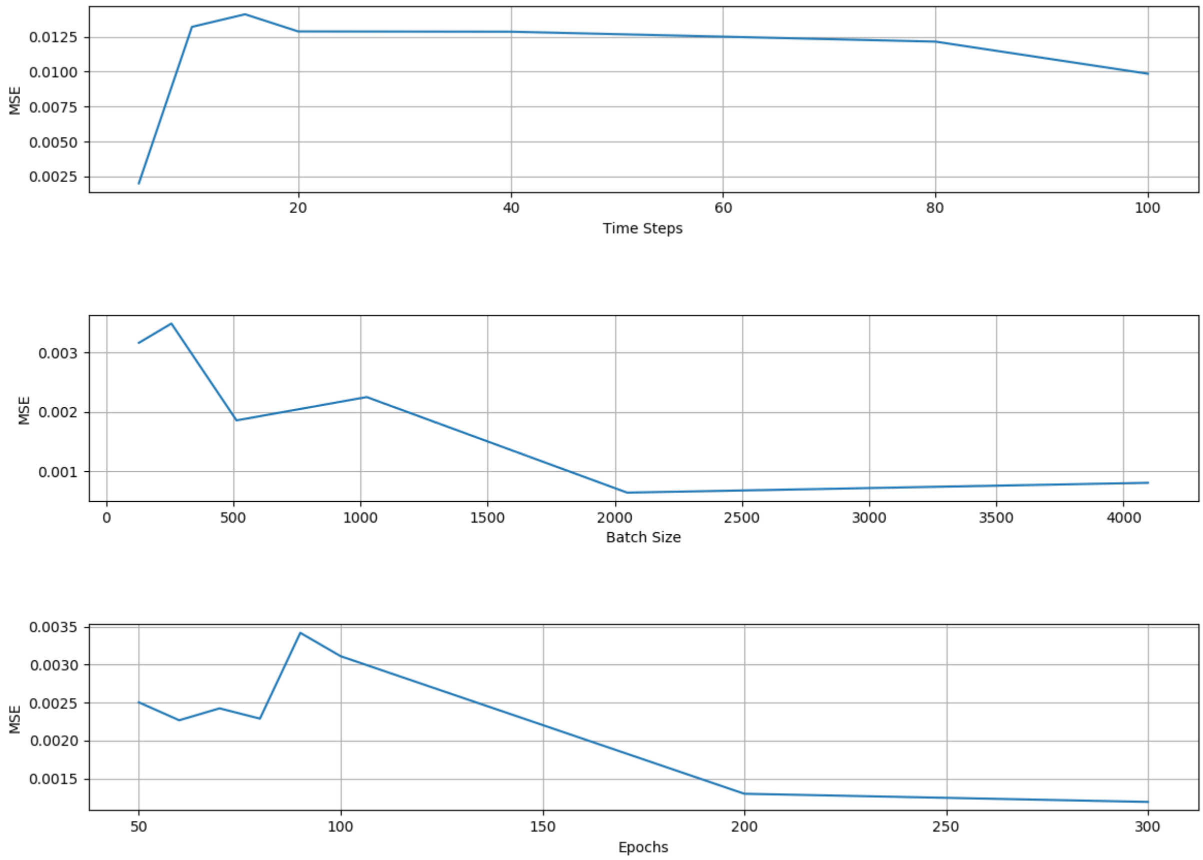
Hyperparameter	Optimal Value
Dropout	0.8
Neurons	[256, 256, 64, 1]
Decay	0.1
Batch Size	128
Epochs	200

Table 4.3: FFNN Optimal Hyperparameters

Although the grid search showed that the optimal dropout was 0.8. When running the experiments, a 0.8 dropout seemed excessive and did not yield good results. The dropout value used was 0.2. The neurons number in the FFNN gird search did not seem to have a big impact on the loss. The highest number of neurons in the selected values for the neurons hyperparameter did seem to achieve the minimum loss, but by a tiny margin. For the decay, batch size, and epochs hyperparameters, the grid search was quite decisive. The loss seemed to converge to its minimum with a decay of 0.1, a batch size of 128, and 200 epochs.



(a) Dropout, Neurons, and Decay



(b) Time Steps, Batch, and Epochs

Figure 4.2: LSTM RNN Hyperparameter Tuning

Hyperparameter	Optimal Value
Dropout	0.2
Neurons	[256, 256, 32, 1]
Decay	0.1
Time Steps	5
Batch Size	2048
Epochs	300

Table 4.4: LSTM RNN Optimal Hyperparameters

The LSTM RNN grid search showed that the optimal dropout was 0.2, which seemed like a reasonable value. With grid search, the hyperparameter tuning process will be limited, because after all only some values are tried out. It is impossible to try out all the possible values for each parameter. For the selected values for the neurons gird search, first two hidden layers with 256 nodes each, and the last with 32 nodes seemed to yield the minimum loss. Unlike the FFNN's neurons grid search, which opted for the maximum number of neurons in each hidden layer. A decay of 0.1 seemed reasonable. The time steps gird search showed that the optimal time steps value was 5. However, further tests showed that the as the number of time steps decreases, the loss decreases. Thus, the minimum number of time steps, which is 1, achieves the best results. For the batch size hyperparameter, the grid search was quite decisive. The loss seemed to converge to a minimum with a batch size of 2048. The epochs gird search was limited to a maximum of 300 epochs, because of limitation in the available computation power. The plot shows that the loss seems to continue moving downwards with more epochs. It is guaranteed however, that at some point more epochs will cause overfitting, and thus higher loss.

Chapter 5

Experiments and Results

In this chapter, the models are tested, assessed and analyzed. Multiple experiments are performed to help identify the behaviour of the models, how do the models compare with each other, and how does the performance of the models varies with different companies and different time periods.

5.1 Apple 2017 Stock Price Forecast

For the first experiment, each model provided a forecast of Apple's stock price during 2017. All models were trained with the same training dataset that goes back to when Apple first started trading publicly on 12/12/1980. This forecast is a next-trading-day price forecast, which means that the future gap is set to 1. From the metrics shown below in table 5.1, and the forecasts of the models shown below in figures 5.1 through 5.5, a few deductions can be made.

1. The LSTM RNN provides the best forecast.
2. The linear regressor comes as a close runner-up.
3. The kNN regressor provides an unreliable, highly volatile forecast.

Model	RMSE	NRMSE	MAE	R	R^2
Linear Regressor	0.0304	0.0532	0.0223	0.992	0.984
kNN Regressor	0.0500	0.0876	0.0367	0.984	0.956
FFNN	0.0531	0.0931	0.0408	0.990	0.951
LSTM RNN	0.0279	0.0490	0.0196	0.993	0.986

Table 5.1: Experiment #1 Evaluation

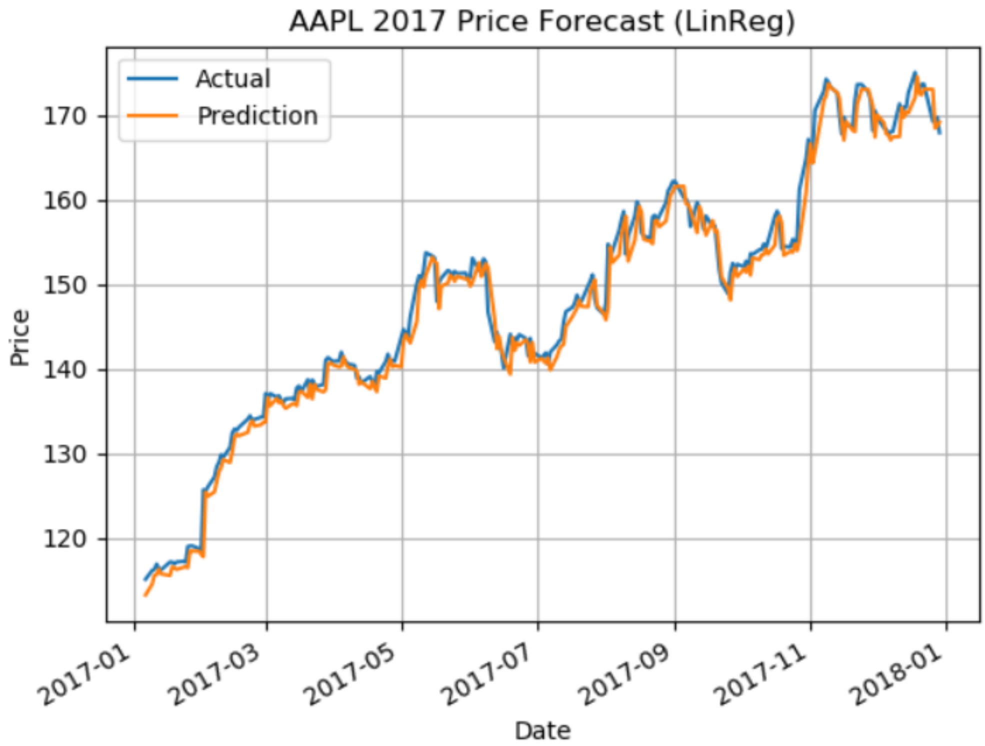


Figure 5.1: Linear Regressor Forecast

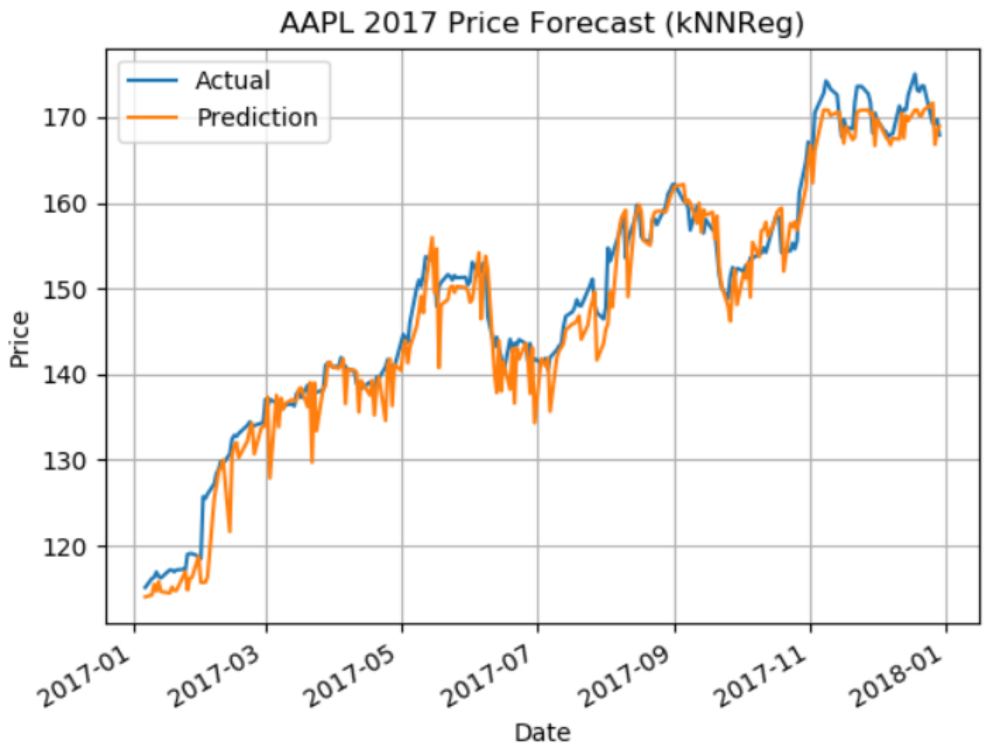


Figure 5.2: kNN Regressor Forecast

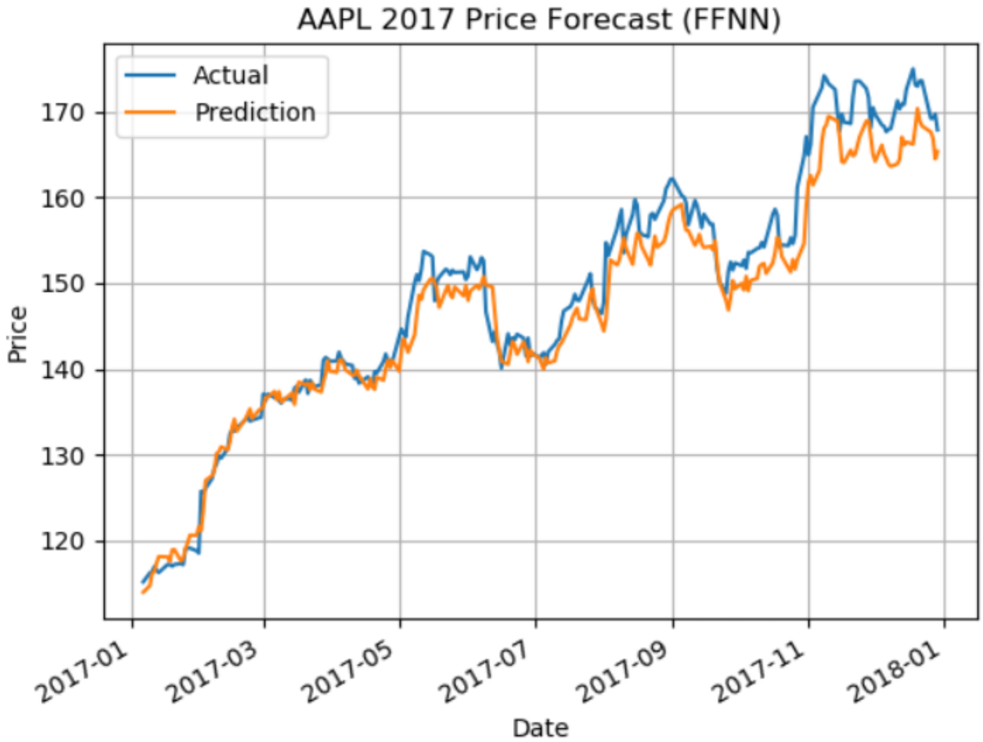


Figure 5.3: FFNN Forecast

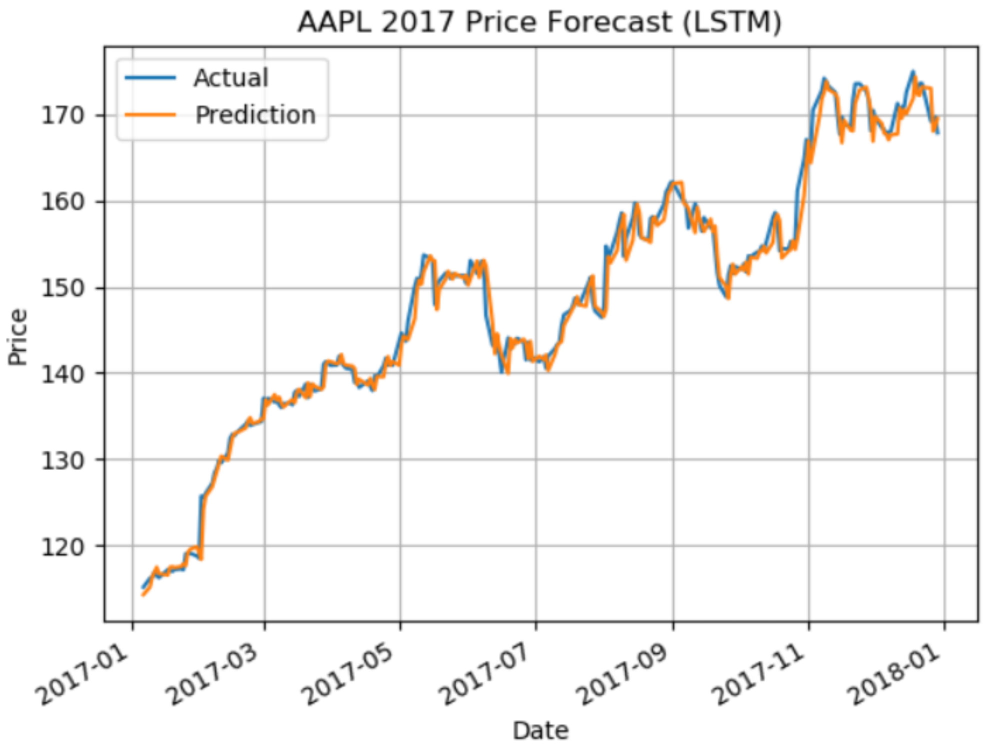


Figure 5.4: LSTM RNN Forecast

The evaluation metrics seem to confirm that the LSTM RNN achieves the minimum loss, and maximum correlation and determination. The difference between the LSTM RNN and the linear regressor evaluation metrics is quite small. Further comparison between these two models will be needed. All the metrics show that the FFNN achieved the worst evaluation.

Taking a look at the forecasts, the linear regressor forecast seems to capture the motion of the stock and the changes in the price. However, it is noticed that there is a slight shift to the right. The kNN regressor forecast seems to be unreliable and highly volatile. The kNN regressor is thus discarded from the next experiments. Even though the FFNN evaluation metrics were the worst, the forecast does not seem bad in terms of the motion of the stock and the changes in the price. However, the forecast confirms the metrics poor evaluation for the FFNN and shows a difference between the actual and predicted prices over multiple points. It also seems to worsen by the end of the forecast, as it did not start out as bad as noticed by the end of the forecast. In this experiment, the LSTM RNN achieves the best forecast, which captures the motion of the stock and the changes in the price quite well. The forecast also shows that the horizontal shift between the actual and predicted prices is less than the one in the linear regressor.

The linear regressor seems to be a good competitor to the LSTM RNN, which is supported by the metrics and the forecast as the best predictor among the models. A further more in-depth comparison is thus needed to further pinpoint the differences between the linear regressor and the LSTM RNN.

5.2 Linear Regressor vs LSTM RNN

In the previous experiment, the linear regressor and the LSTM RNN model provided the best forecasts. In this experiment, the linear regressor and the LSTM RNN forecasts are analyzed and compared.

To analyse the forecast and evaluate how fast does the model predict the closest price to the actual, a lag metric is created. The Prediction-Actual Lag (PAL) metric works as follows: a lookup window is to be selected; this window indicates the number of predicted prices each actual price should be compared with. The actual prices are then traversed and compared with the predictions, each actual price datapoint is compared against a number of the predicted data points, that number is specified by the lookup window, so if the lookup window is set to 5, then each actual datapoint is compared to the corresponding predicted datapoint and the 4 next to it.

Figures 5.5 and 5.6 show the frequency of when was the prediction closest to the actual price, the day lag indicates the number of days it took for the forecast to best match the actual price.

Figures 5.7 and 5.8 show the overlay of the daily lag over the forecast. The daily lag follows the timeline of the forecast to show day-by-day the lag of each prediction. Overlaying the daily lag over the forecast helps to identify the different points in the timeline that led to different lags.

Deductions that can be made from this experiment are:

1. It is important to have a metric which focuses on the horizontal comparison, rather than the vertical comparison performed by the evaluation metrics.
2. The evaluation metrics show that LSTM RNN surpasses the close runner-up, the linear regressor, by a modest difference. However, by comparing the forecasts and using the metric PAL, it is clear that the LSTM RNN provides a more accurate forecast.

Model	RMSE	NRMSE	MAE	R	R^2
Linear Regressor	0.0304	0.0532	0.0223	0.992	0.984
LSTM RNN	0.0279	0.0490	0.0196	0.993	0.986

Table 5.2: Experiment #2 Evaluation

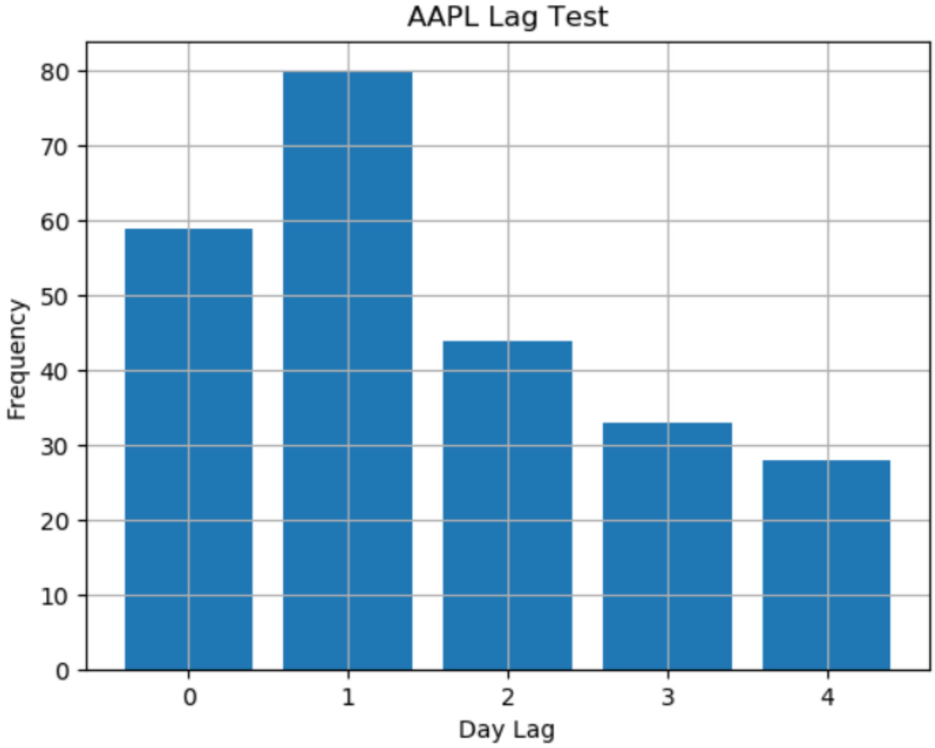


Figure 5.5: Linear Regressor PAL

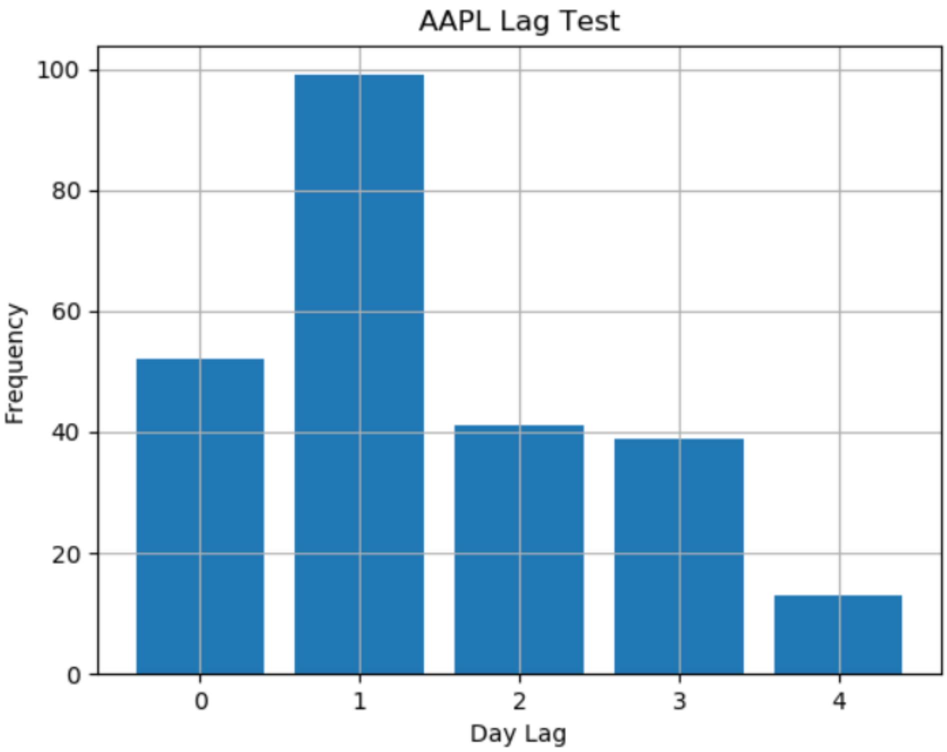


Figure 5.6: LSTM RNN PAL

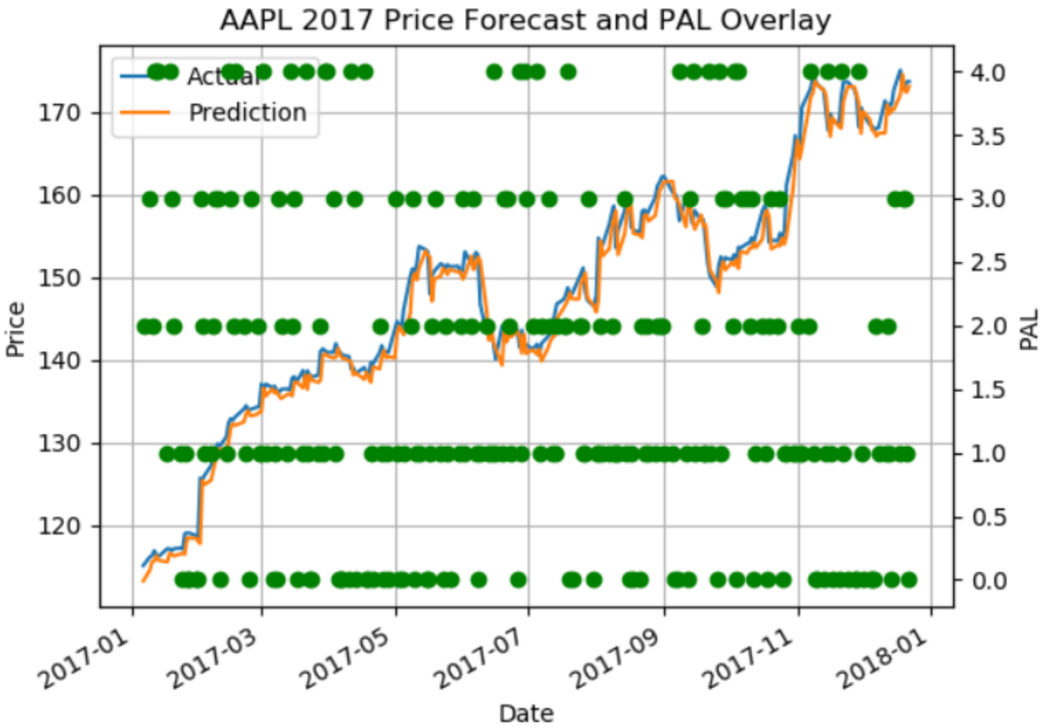


Figure 5.7: Linear Regressor Forecast and Daily PAL Overlay

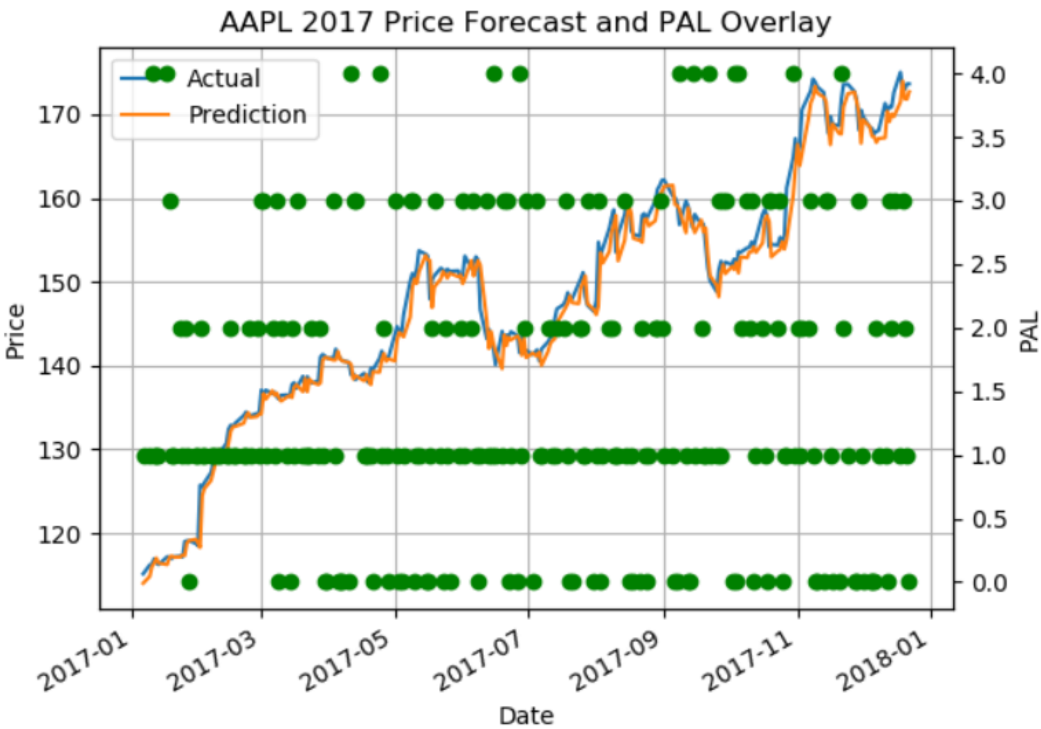


Figure 5.8: LSTM RNN Forecast and Daily PAL Overlay

Comparing figures 5.5 and 5.6, we see that for a 0 day lag, the linear regressor has about 58 instances against the LSTM RNN's approximately 46 instances. Before favouring the linear regressor just for the frequency of the 0-day lag, it is noticed that for all the remaining day lags, the LSTM RNN is better. For a 1-day lag, the LSTM RNN has a frequency of 100 against the linear regressor's 80. What is on the LSTM RNN supporting side is that most of the lags are concentrated on the 1-day lag, and do not push to the 2nd, 3rd, and 4th days as much as the linear regressor. Further day lag are essentially bad, because they do not only show an error or a delay at a given day, but they also affect all the subsequent predictions. The predicted forecast tends to shift more to the right, widening the horizontal gap between the actual and the predicted prices. It is thus concluded that the linear regressor lags more than the LSTM RNN.

Overlaying the daily lag over the price forecast as done in figures 5.7 and 5.8 displays the daily lag, which follows the same timeline as the forecast. This helps in identifying the points in a forecast that lead to a bigger lag. The figures show that the bigger lags are caused when the change in price is big, regardless of whether the price is increasing or decreasing. The fewer day lags are on the points in the forecast where the change in price is not so steep. There are also some anomalies spread throughout the daily lag. This conclusion motivates the next experiment. The stock price time series is to be analyzed to classify the different states and transitions of the prices. Furthermore, how does the forecast behave during these different states and transitions?

5.3 Sudden Changes vs Normal Movements

The purpose of this experiment is to analyze the behaviour of the model during two different periods a stock usually goes through; a normal movement, where the stock price fluctuates with no dramatic change, the other period is a sudden change period, where the stock moves violently either upwards, downwards, or up and down with high volatility.

The stock selected for this experiment is Tesla's, and the testing period is between 01/01/2013 and 01/06/2013. The experiment is performed using the LSTM RNN model.

Figure 5.9 shows the forecast, figure 5.10 shows the frequency of when was the prediction closest to the actual price, the day lag indicates the number of days it took for the forecast to best match the actual price. Figure 5.11 follows the same timeline of the forecast on the x-axis, against the lag on the y-axis. Figure 5.12 shows the overlay of the daily lag over the forecast to identify the different points in the timeline that led to different lags.

Deductions that can be made from this experiment are:

1. The model is capable of predicting the price and the fluctuations in price caused by the stock market movement.
2. When external events impact the stock price suddenly, the model naturally does not pick up on these events. The technical indicators upon which the model builds its predictive engine give indications on the movement of the stock price within the market. When external events such the one examined in this experiment occur, the model will lag on picking up a surge or a drop, because these foreign situations are not a part of the stock movement in the market.

RMSE	NRMSE	MAE	R	R^2
0.046	0.231	0.035	0.988	0.967

Table 5.3: Experiment #3 Evaluation

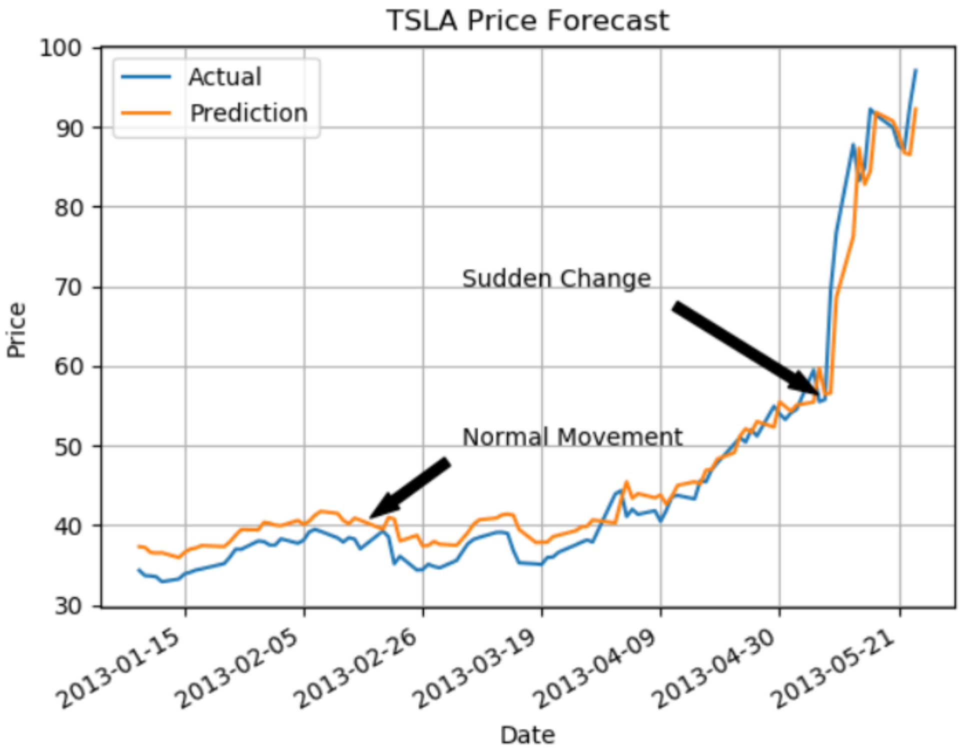


Figure 5.9: Price Forecast

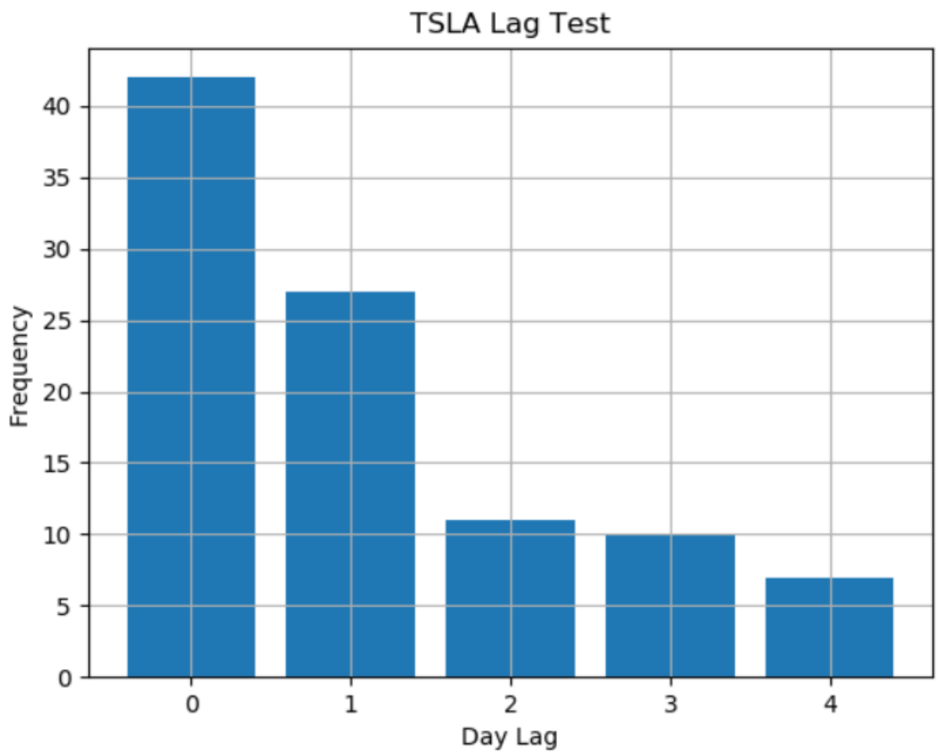


Figure 5.10: PAL

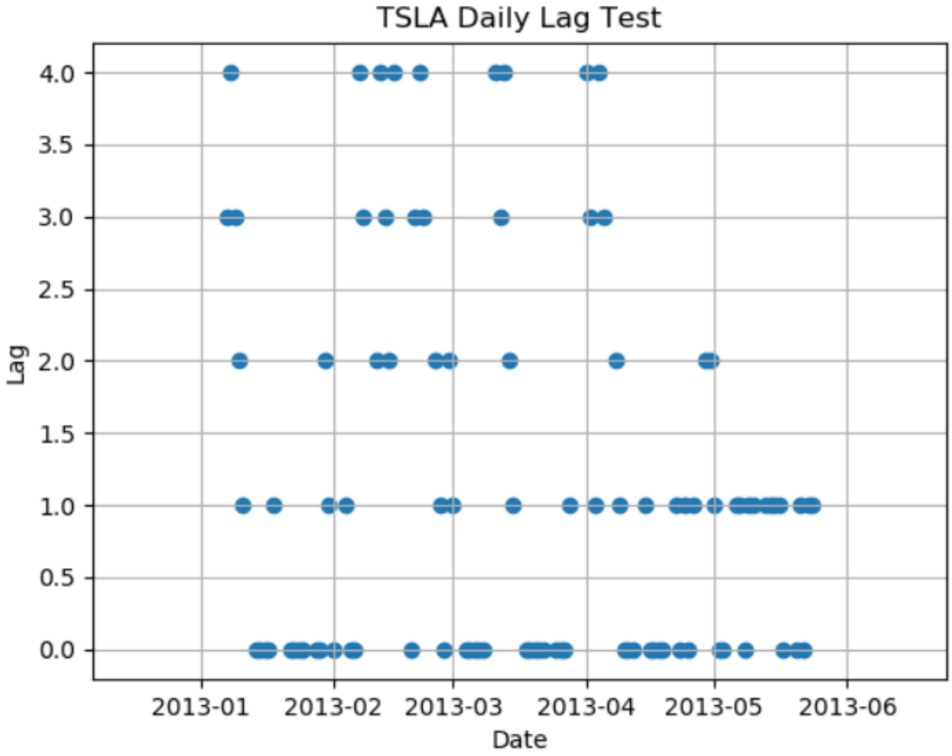


Figure 5.11: Daily PAL

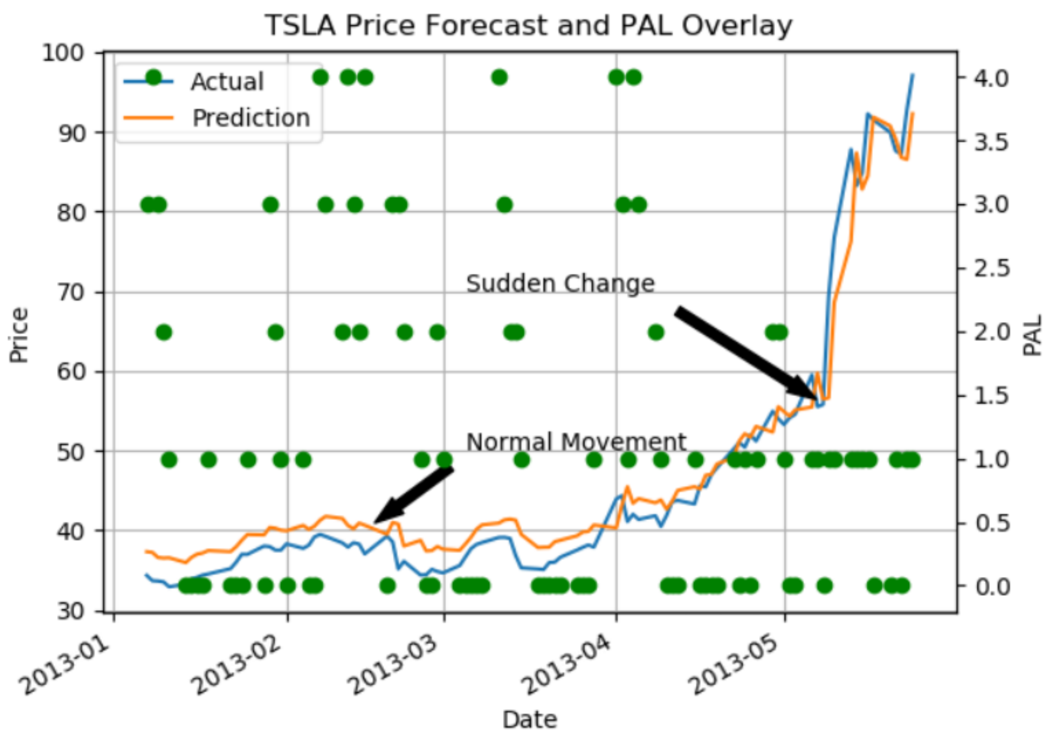


Figure 5.12: Price Forecast and Daily PAL Overlay

Figure 5.9 shows that up until 07/05/2013, the stock price movement exhibits a normal movement, no violent trajectories appear. This is when the model performs best. The forecast does not lag the actual price, and follows the same trend and movement of the actual price. Starting from 07/05/2013, the stock moves up with a steep trajectory, and during that sudden change is when the model performs poorly. Upon researching news about Tesla on May/2013, it was discovered that the company reported its first quarterly profit and its flagship at that time, the Model S, received the best review of any car in Consumer Reports magazine's history [21]. These positive news caused an unexpected and sudden surge in Tesla's stock price.

Figure 5.10 shows that the frequency of the 0-day lag is the highest, and the frequencies of the next days lags decrease as we go further to more days lags. This behaviour is excellent and reflects a good forecast with minimal horizontal shift.

Figures 5.11 and 5.12 show that the model finds the closest prediction to the actual price early on during the normal movement phase, and lags at the end of the timeline during the sudden change phase. The overlaying of the daily lag on the forecast shows that the lag increases at the sudden points of change, and that the lag is minimal at the points of normal movements. There are also some anomalies spread throughout the daily lag. Some points where the change is not too sudden show big lags.

5.4 External Events Impact on Companies' Stocks

The purpose of this experiment is to test different companies during different interesting time periods. The timelines of four companies were reviewed to look for events that affected the companies' stock prices. The LSTM RNN model was tested to find out how would it perform during these periods.

- Amazon

  Amazon started trading publicly on 15/05/1997. The period of interest is between September/2017 and February/2018. That is when Amazon exceeded its 2017 Q3 expectations. Amazon's Q3 reports showed an increase in profits, an acceleration in revenue growth, an increase in operating income, and the success of Alexa-enabled devices [22]. The forecast is shown in figure 5.13.

- Apple

  Apple started trading publicly on 12/12/1980. The period of interest is between September/2012 and June/2013. That is when Apple's stock price took an aggressive plunge. Apple faced multiple hardships during that period; earnings were no longer growing, low-priced phones were capturing most of the smartphone market share over the iPhone, and the company entered the "post-Steve Jobs" era where the company's next generation of leaders and products were in question [23]. The forecast is shown in figure 5.14.

- Facebook

  Facebook started trading publicly on 18/05/2012. The period of interest is between January/2018 and March/2018. That is during the Facebook—Cambridge Analytica data scandal. Amid the scandal and Mark Zuckerburg's public hearing, Facebook's stock price fell [24]. The forecast is shown in figure 5.15.

- Tesla

  Tesla started trading publicly on 29/06/2010. The period of interest is between September/2013 and November/2013. That is when Tesla reported disappointing third quarter financial results. In addition, a third widely-reported fire involving a Model S in just two months was putting Tesla under heat [25]. The forecast is shown in figure 5.16.

Company	RMSE	NRMSE	MAE	R	R^2
Amazon	0.0343	0.0896	0.0232	0.995	0.988
Apple	0.0417	0.0993	0.0303	0.990	0.976
Facebook	0.0931	0.1600	0.0666	0.926	0.849
Tesla	0.0840	0.1670	0.0582	0.948	0.896

Table 5.4: Experiment #4 Evaluation

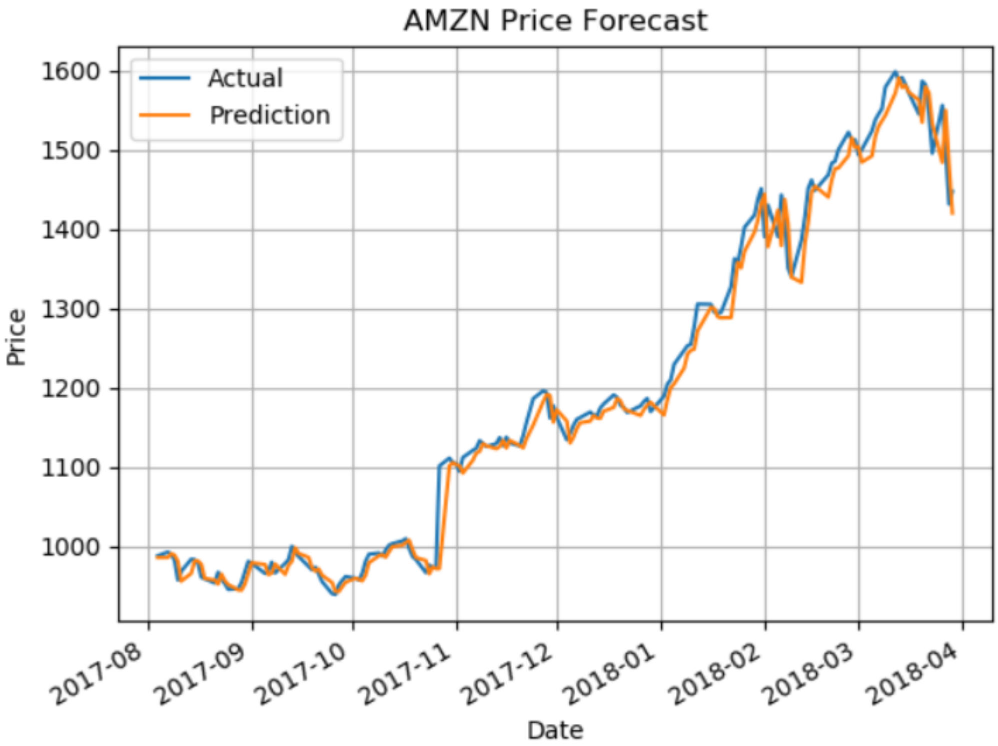


Figure 5.13: Amazon

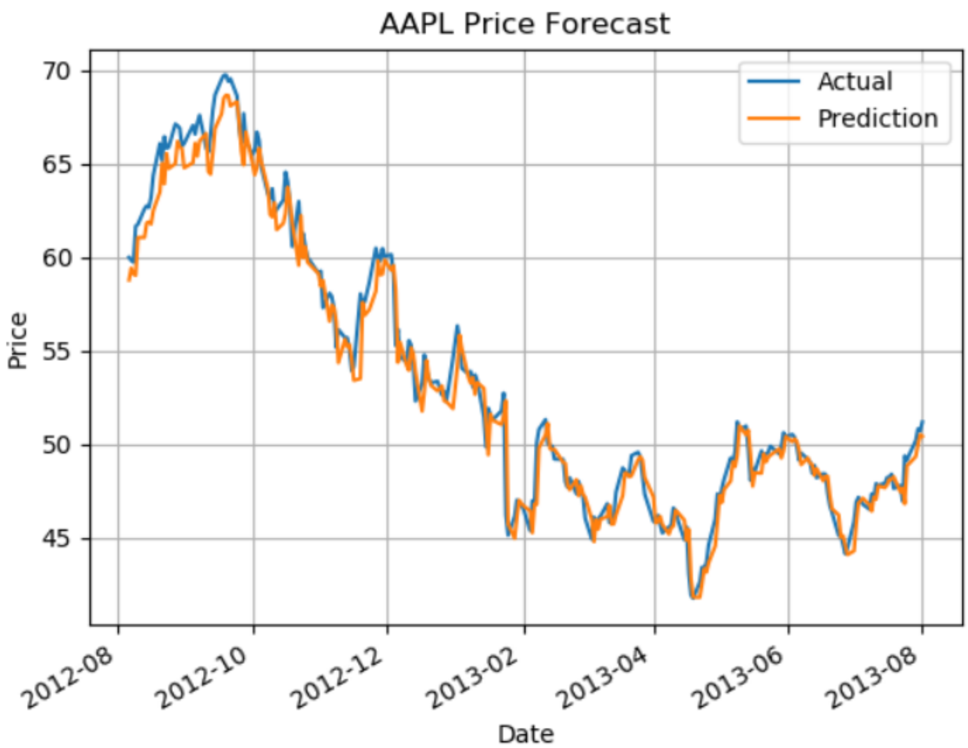


Figure 5.14: Apple

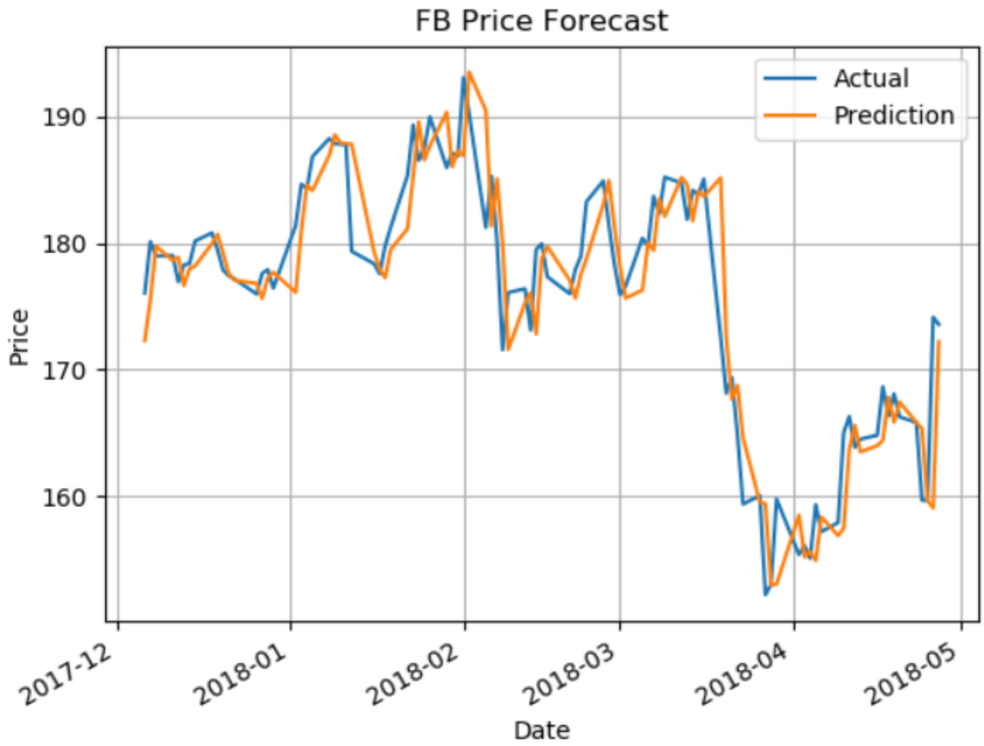


Figure 5.15: Facebook

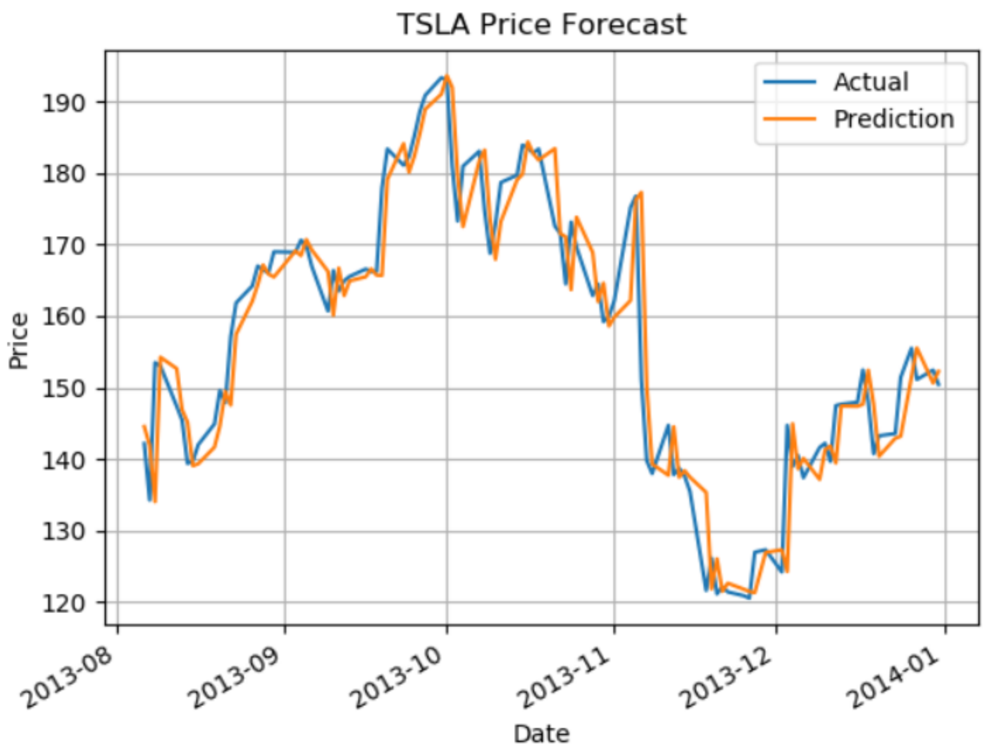


Figure 5.16: Tesla

The evaluation metrics for Amazon and Apple are acceptable. Facebook and Tesla evaluation metrics seem to show big errors. However, it should be noted that these test periods are periods where the stock price movement is heavily influenced by external events. The technical analysis approach integrated into the training of the models focus on analyzing the movement of the stock prices within the market and can not anticipate events foreign to the stock market.

Taking a look at the forecasts, Amazon and Apple can be considered as fairly informative. Facebook and Tesla show highly volatile and steep changes during the testing period, which explains their poor evaluation.

5.5 Future Gap

The purpose of this experiment is to analyze the behaviour of the models with different future gaps.

The stock selected for this experiment is Microsoft's, and the testing period is between 01/01/2017 and 01/01/2018. The experiment is performed using the LSTM RNN, linear regressor, and FFNN models. The forecasts of the LSTM RNN model are shown in figures 5.17 through 5.21.

Deductions that can be made from this experiment are:

1. Unsurprisingly, the models perform best in the next-day forecast with 1 day future gap, and the performance worsens as the future gap increases.

2. The LSTM RNN provides the best results. Furthermore, The RMSE difference between the LSTM RNN and the linear regressor is quite small in the next-day forecast. However, as the future gap increases, the gap between the LSTM RNN and the linear regressor RMSEs increases. Recall that RMSE has the advantage of penalizing large errors. This indicates that the linear regressor has larger errors than the LSTM RNN, which shows the resilience of the LSTM RNN over the linear regressor.

Future Gap	RMSE	NRMSE	MAE	R	R^2
1 Day	0.0273	0.0676	0.0184	0.995	0.991
2 Days	0.0369	0.0909	0.0254	0.992	0.983
3 Days	0.0437	0.1070	0.0314	0.989	0.976
4 Days	0.0496	0.1210	0.0363	0.985	0.969
5 Days	0.0568	0.1380	0.0421	0.981	0.959

Table 5.5: Experiment #5: LSTM Evaluation

Future Gap	RMSE	NRMSE	MAE	R	R^2
1 Day	0.0275	0.0679	0.0185	0.993	0.990
2 Days	0.0372	0.0917	0.0260	0.992	0.983
3 Days	0.0441	0.1080	0.0317	0.989	0.976
4 Days	0.0504	0.1230	0.0366	0.985	0.968
5 Days	0.0572	0.1390	0.0422	0.981	0.958

Table 5.6: Experiment #5: Linear Regressor Evaluation

Future Gap	RMSE	NRMSE	MAE	R	R^2
1 Day	0.0376	0.0931	0.0278	0.994	0.982
2 Days	0.0474	0.1170	0.0335	0.991	0.972
3 Days	0.0691	0.1700	0.0501	0.984	0.939
4 Days	0.0535	0.1310	0.0389	0.982	0.964
5 Days	0.0709	0.1729	0.0512	0.972	0.936

Table 5.7: Experiment #5: FFNN Evaluation

Figures 5.17 through 5.21 show that when the future gap increases, the forecast tends to shift to the right, which creates a big difference between the actual and predicted prices. It is this advised to use the next-day forecast with a future gap of 1 day. It seems to be quite tough, if possible, to train a model to predict too much into the future. This seems to be logical and justified.

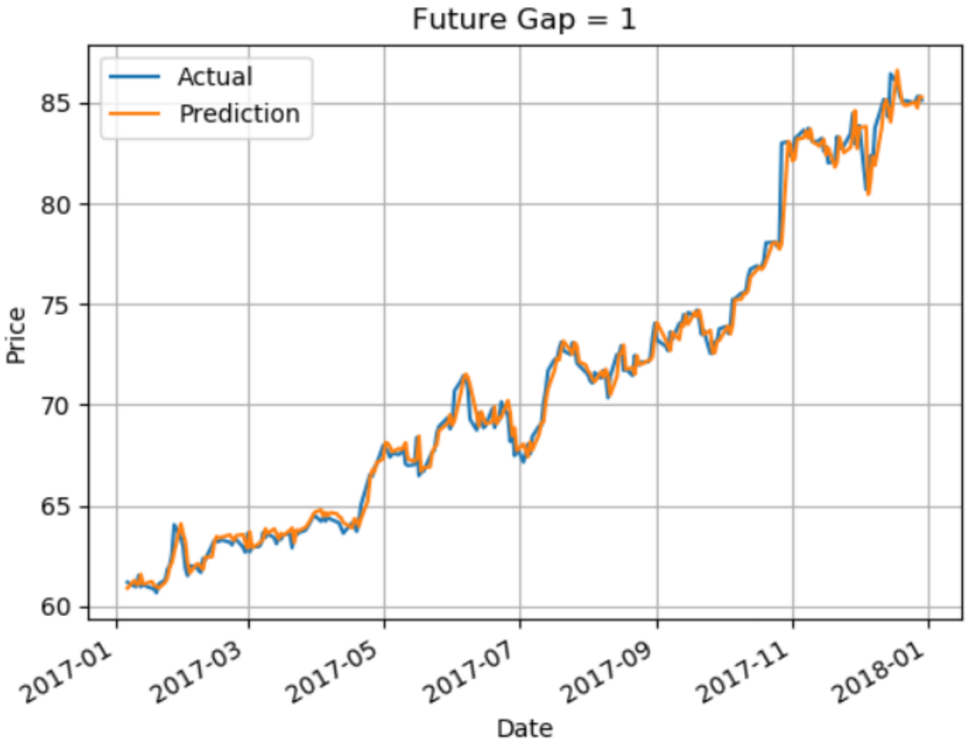


Figure 5.17: 1 Day Future Gap

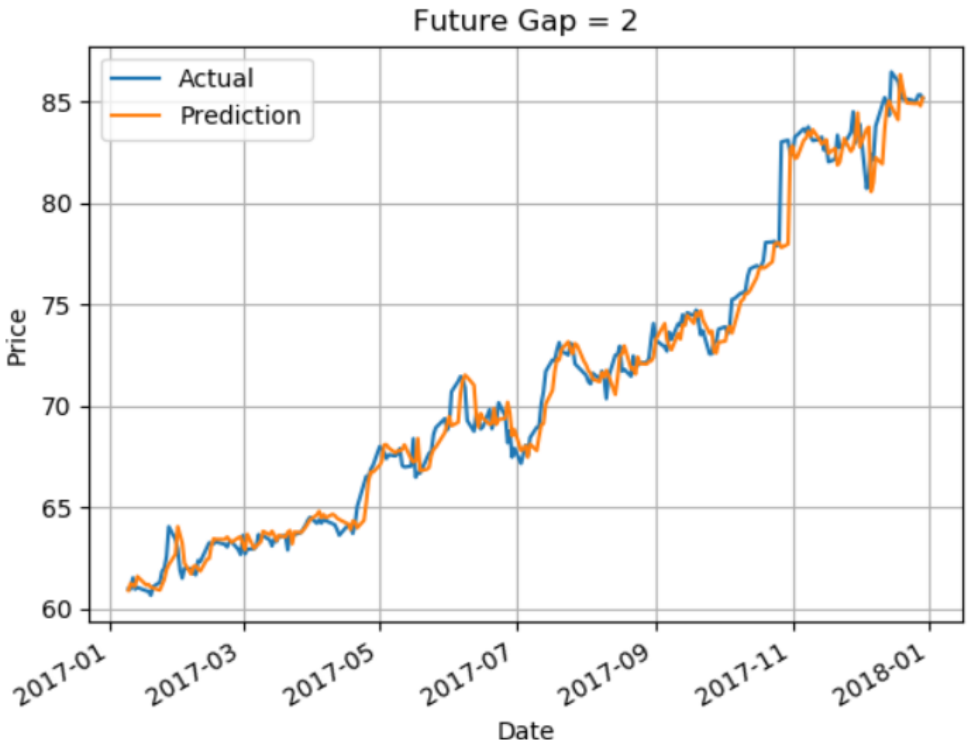


Figure 5.18: 2 Days Future Gap

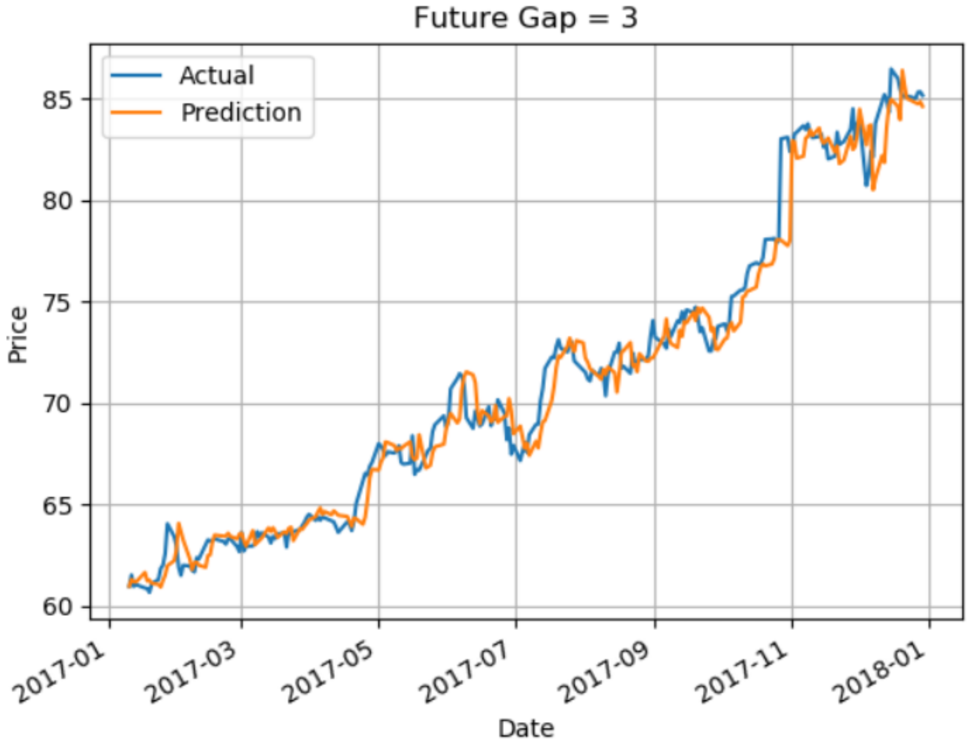


Figure 5.19: 3 Days Future Gap

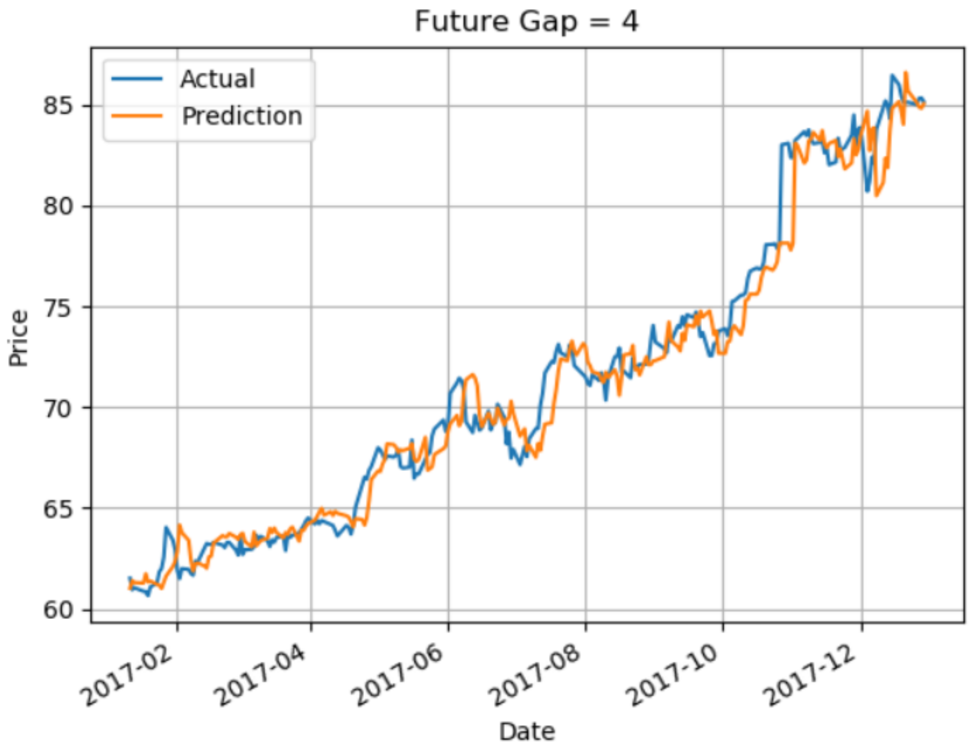


Figure 5.20: 4 Days Future Gap

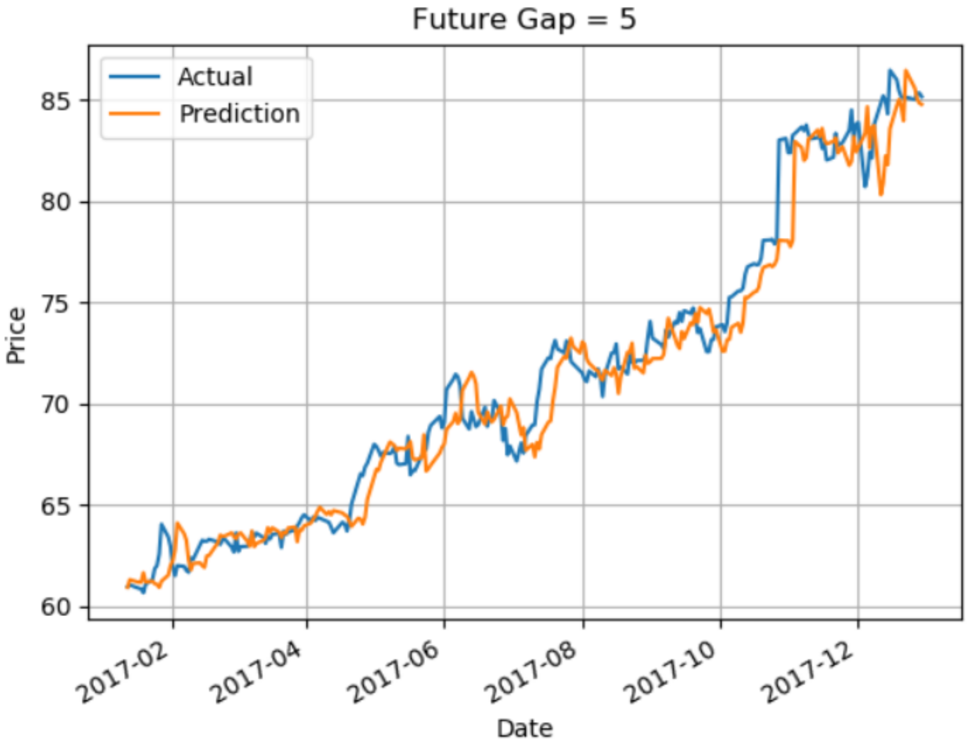


Figure 5.21: 5 Days Future Gap

Chapter 6

Conclusion

This chapter attempts to draw some conclusions from the experiments and results. Section 6.1 discusses the research major findings, and tries to find some meaningful answers to the research questions. Section 6.2 discusses intriguing related future work.

6.1 Findings and Research Questions

This research focused on the task of making use of machine learning in the field of stock trading. Four machine learning models namely linear regressor, kNN regressor, FFNN, and LSTM RNN were developed. The models were trained on a dataset that includes the price and technical indicators as features, and the future price as the target output. Several experiments were performed to test the predictive capabilities of the models. The models were compared with each other by using several error evaluation metrics, and a novel metric that focuses on comparing the horizontal error between the predicted and the actual values. Some of the major findings of this research are:

1. The LSTM RNN outperformed all the other models.
2. The LSTM RNN model is capable of accurately predicting the next-day price unless a major external event impacts the stock price suddenly.
3. The LSTM RNN model naturally lags on picking up on external events that impact the stock price suddenly.

After applying the experiments and studying the results, it is essential to take a look back at the research questions. The following answers to the research questions attempt to briefly sum up the research, and provide somewhat adequate answers to the investigations of this research.

1. *Can machine learning be used to predict future stock prices?*

   Machine learning models that can predict future stock prices are possible to develop, however they operate under a set of constraints. It is advised to use a next-day forecast, as predicting too much into the future is difficult, if possible. The forecast can be quite reliable, unless a sudden event that is external to the stock market affects a company's stock price. The technical analysis approach integrated into the training of the models focus on analyzing the movement of the stock prices within the market and can not anticipate events foreign to the stock market.

2. *How does the performance of different machine learning algorithms vary?*

   LSTM RNN has shown better results than the other models. That is supported by the error metrics that were the least with the LSTM RNN, and the other metrics that quantify the degree to which the predicted and actual prices are related that were the highest with the LSTM RNN. Comparing the forecasts visually and by the assist of a lag metric to assess the horizontal lag between the actual prices and the predicted prices also showed that the LSTM RNN outperformed the other models.

6.2 Future Work

6.2.1 Automated Trader

The next interesting step after having a forecast is to actually be able to automate stock trading decisions. Using a next-day-forecast for more reliable forecasts, some kind of an automated trader can be developed to place daily trading orders based on its presumed knowledge of the closing adjusted price of each day before the market closes. The automated trader can be designed to manage a portfolio consisting of a number of diverse stocks of different companies in different sectors and industries. It would be interesting to see to how would the automated trader perform given a starting capital, and whether it will be able to generate profits or sustain losses.

6.2.2 Reinforcement Learning

Reinforcement learning is a type of machine learning where an agent takes actions in an environment to maximize some form of aggregate reward. Reinforcement learners create policies that provide specific direction on which action to take. A reinforcement learner is defined by: (s, π, a, r). s is the current state of the learner's environment. a is the action taken by the learner, this action causes a state transition in the environment. r is the reward the learner gets when making an action at a specific state. π is the policy the learner uses to choose the optimal action that results in maximum reward for the current state. In terms of stock trading, the environment is the market, the actions are actions that can be taken in the market, like buying, selling, and holding. States are factors about stocks that might be observed, like technical indicators. Reward is the return from making proper trades. A Reinforcement learner performs actions throughout its learning process to deduce the optimal policy for maximizing rewards. It would be interesting to see it is possible to design a reinforcement learner tailored to the stock trading problem, and to see to how it would perform given a starting capital, and whether it will be able to generate profits or sustain losses.

Appendix

Appendix A

Lists

kNN	k-Nearest Neighbor
FFNN	Feedforward Neural Network
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
IPO	Initial Public Offering
OTC	Over-The-Counter
NYSE	New York Stock Exchange
TYO	Tokyo Stock Exchange
MSE	Mean Squared Error
ANN	Artificial Neural Network
SMA	Simple Moving Average
BBs	Bollinger Bands
VROC	Volume Rate of Change
RMSE	Root Mean Squared Error
ReLU	Rectified Linear Unit
NRMSE	Normalized Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

R	Coefficient of Correlation
R^2	Coefficient of Determination
PAL	Prediction-Actual Lag

List of Figures

List of Tables

Bibliography

[1] A. Dingli and K. S. Fournier. Financial Time Series Forecasting: A Machine Learning Approach. *Machine Learning and Applications: An International Journal (MLAIJ)*, 4, 2017.

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques [J]. *Expert Systems with Applications*, 2015.

[3] R. Dash and P. K. Dash. A hybrid stock trading framework integrating technical analysis with machine learning techniques [J]. *The Journal of Finance and Data Science*, 2, 2016.

[4] J. I. Larsen. Predicting Stock Prices Using Technical Analysis and Machine Learning. *Norwegian University of Science and Technology*, 2010.

[5] M. Olden. Predicting Stocks with Machine Learning. *University of Oslo*, 2016.

[6] John Ducas. Stock Market Investing for Beginners. https://www.udemy.com/the-beginners-guide-to-the-stock-market/. [Online; accessed May 2018].

[7] Investopedia. Market Capitalization. https://www.investopedia.com/terms/m/marketcapitalization.asp. [Online; accessed May 2018].

[8] Investopedia. Dividend. https://www.investopedia.com/terms/d/dividend.asp. [Online; accessed May 2018].

[9] Investopedia. Mutual Fund. https://www.investopedia.com/terms/m/mutualfund.asp. [Online; accessed May 2018].

[10] Andrew Ng. Machine Learning. https://www.coursera.org/learn/machine-learning. [Online; accessed May 2018].

[11] Tucker Balch. Machine Learning for Trading. https://udacity.com/course/machine-learning-for-trading--ud501. [Online; accessed May 2018].

[12] Sebastian Ruder. A Quick Introduction to Neural Networks. https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/. [Online; accessed May 2018].

[13] Christopher Olah. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Online; accessed May 2018].

[14] stockstotrade. Types of Technical Indicators INFOGRAPHIC. https://stockstotrade.com/types-technical-indicators-infographic/. [Online; accessed May 2018].

[15] Tushar Gupta. Deep Learning: Overfitting. https://towardsdatascience.com/deep-learning-overfitting-846bf5b35e24. [Online; accessed May 2018].

[16] Jason Brownlee. Dropout Regularization in Deep Learning Models With Keras. https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/. [Online; accessed May 2018].

[17] Janet Wesner. MAE and RMSE. Which Metric is Better? https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d. [Online; accessed May 2018].

[18] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. Kamaev. A Survey of Forecast Error Measures. *Applied Sciences Journal*, 24, 2013.

[19] Jessica McCallister. Pearson Correlation Coefficient. https://study.com/academy/lesson/pearson-correlation-coefficient-formula-example-significance.html. [Online; accessed May 2018].

[20] Emmanuelle Rieuf. How To Interpret R-squared and Goodness-of-Fit in Regression Analysis. https://www.datasciencecentral.com/profiles/blogs/regression-analysis-how-do-i-interpret-r-squared-and-assess-the. [Online; accessed May 2018].

[21] Chris Isidore. Tesla stock up 40% this week. http://money.cnn.com/2013/05/10/investing/tesla-stock. [Online; accessed May 2018].

[22] Daniel Sparks. 5 Ways Amazon.com, Inc. Crushed Its 3rd Quarter. https://www.fool.com/investing/2017/10/27/5-ways-amazoncom-inc-crushed-its-3rd-quarter.aspx. [Online; accessed May 2018].

[23] Henry Blodget. Meanwhile, Apple Stock Just Crashed To A New Low. http://www.businessinsider.com/apple-stock-low-2013-4. [Online; accessed May 2018].

[24] Lucinda Shen. Facebook Stock Is in the Red for the Year After the FTC Confirms Investigation. http://fortune.com/2018/03/26/

facebook-stock-ftc-investigation-cambridge-analytica/. [Online; accessed May 2018].

[25] Peter Valdes-Dapena. Third Tesla Model S catches fire after crash. http://money.cnn.com/2013/11/07/autos/tesla-fire/index.html. [Online; accessed May 2018].