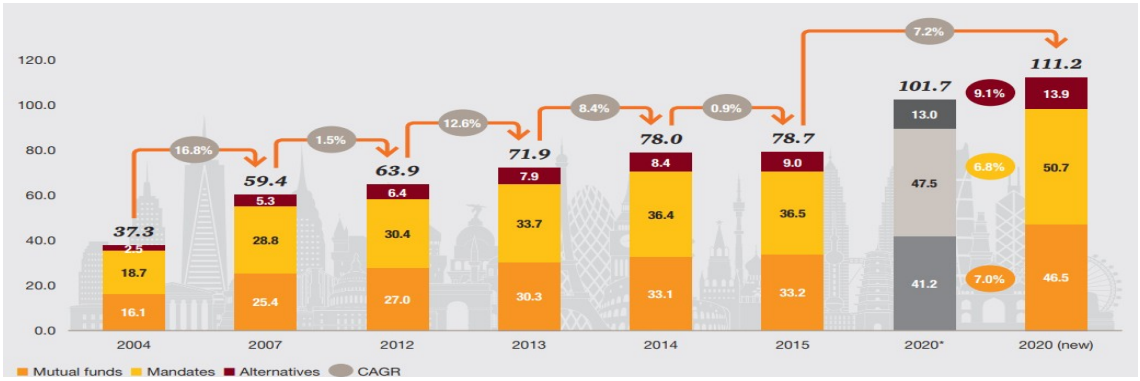
Neural Network in Portfolio Management

Introduction

As the amount of investable assets grows each year, the need to have a better return increase drastically (Exhibit 1). Moreover, there is a rising demand on data-driven management to maximize distribution opportunities because accurate market prediction is instrumental in maintaining and re-balancing investors' current portfolio in the field of portfolio management. Recently, proliferation of ETFs and passive investment indicates that investors become more sophisticated and risk-aware as investors do not wish to over-pay for the management fee. With machine learning applications, financial managers are able to analyze financial markets with statistically sound approach.

Exhibit 1: Industry Trend



Data Sourcing and Cleansing

However, before applying the financial data to sophisticated algorithms, it is critical for financial managers to properly source and clean the data. In this paper, the primary source of historical stock data and each stock's industry category would be gathered from Alpha Vantage, Google Finance, and Kaggle using Python library *Panadas.DataReader* to readily download the S&P 500 data.

Exhibit 2: Sample Data Overview – Individual Stock

	date	open	high	low	close	volume	Name
	2013-02-08	15.07	15.12	14.63	14.75	8407500	AAL
	2013-02-11	14.89	15.01	14.26	14.46	8882000	AAL
	2013-02-12	14.45	14.51	14.10	14.27	8126000	AAL
	2013-02-13	14.30	14.94	14.25	14.66	10259500	AAL
	2013-02-14	14.94	14.96	13.16	13.99	31879900	AAL

As shown in Exhibit 2, the original dataset includes the date, open price, high quote, low quote, closing price, trading volume, and the ticker symbol of the stock. The paper would focus on the opening and closing price of stocks ranging from 2013 to 2018. With over a five-year of data, over 600,000 rows need to be processed. In addition to the abovementioned data field, financial managers may want to merge each stock's industry sector (Exhibit 3) to the original dataset to easily recognize the compositions of the portfolio. We will use ticker symbol as a key to merge the GICS Sector and GICS Sub Industry to the original dataset. As each industry poses different returns and risks, financial managers can utilize this additional piece of information to further determine which group of stocks should be purchased and sold in order to maximize the return while tailoring to clients' risk appetite.

Exhibit 3: Sample Data Overview – Stock Industry Sector

	Ticker Symbol	Security	SEC filings	GICS Sector	GICS Sub Industry	
0	MMM	3M Company	reports	Industrials	Industrial Conglomerates	
1	ABT	Abbott Laboratories	reports	Health Care	Health Care Equipment	N
2	ABBV	AbbVie Inc.	reports	Health Care	Pharmaceuticals	N
3	ACN	Accenture plc	reports	Information Technology	IT Consulting & Other Services	
4	ATVI	Activision Blizzard	reports	Information Technology	Home Entertainment Software	
5	AYI	Acuity Brands Inc	reports	Industrials	Electrical Components & Equipment	
6	ADBE	Adobe Systems Inc	reports	Information Technology	Application Software	
7	AMD	Advanced Micro Devices Inc	reports	Information Technology	Semiconductors	S

Because the original data is listed by date, the dataset would need to be first grouped by ticker

symbol (the Name column from Exhibit 2) before any analysis is performed. Furthermore, any machine learning algorithms can only be applied after the data is cleansed. While complex mathematical models can help analyze data, it is often that the more sophisticated the models, the more sensitive they are to the presence of corrupt values. Therefore, it is crucial that suitable measure is taken. Exhibit 4 provides a general overview of how data cleansing can be done in different situations. Depending on the experience and final objective of the analysis, financial managers will use a variety of methods to handle corrupt data.

For corrupt values present in the data, financial managers can either remove or impute the corrupt values. There are three steps in cleaning financial data:

(1) Check for missing values,

(2) Check for impossible values and,

(3) Check for inconsistent or unlikely (outliers) values.

When dealing with missing values, financial managers can expect that there is always a value in the closing price column (note that the dataset in Exhibit 2 already excludes weekends and holidays). Because there are multiple open-sourced APIs available for S&P 500 data, any missing values can be quickly identified and filled with proper date and ticker symbol.

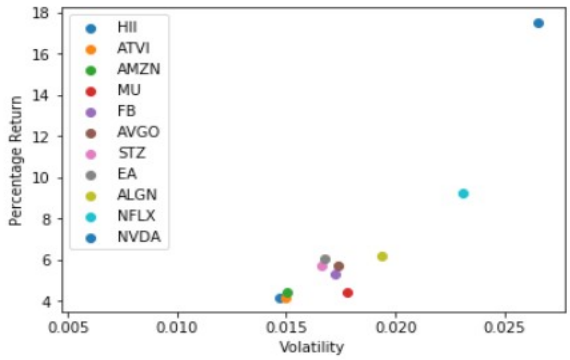
Impossible values may arise when data is not sourced and downloaded appropriately. In our case, share prices should always to greater than 0. No negative values should be present. Resolving inconsistent and unlikely data values, on the other hand, is a more time-consuming task. In our case, when the closing price of a particular stock increases or decreases for 10% or more in a day, some manual checks may be required. Given that most of the data is sourced from Alpha Vantage, a relatively easy way to check for this suspicious values would be to compare values in Google Finance with the exact date and ticker symbol. If the two values are the same,

the drastic movement is as it should be. However, if the values are different, the particular data cell would need to be examined and fixed manually.

Exploratory Data Analysis (EDA)

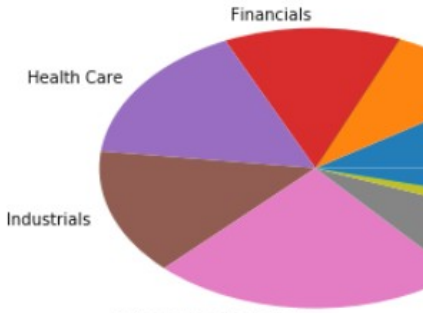
After the dataset is cleansed, financial managers can start perform a preliminary emanation of the dataset. From Exhibit 5, several stocks along with the stock market have witnessed a tremendous growth over the past 5 years.

Exhibit 5: Top 10 Stocks by Return



By taking a closer look, one can notice that stocks with a highly specialized technology component outgrew others. For instance, Align Technology (ticker symbol: ALGN), a global medical devise company specialized in aligners used in orthodontics, has appreciated over 600% since 2013 while the market (S&P 500) only increases over 80%. If financial managers sort the returns by industry sector, the information technology sector has the best performing stocks over the years while stocks in the energy sector actually depreciate (Exhibit 6). Therefore, in the later section of

Exhibit 6: Return by Industry Ret

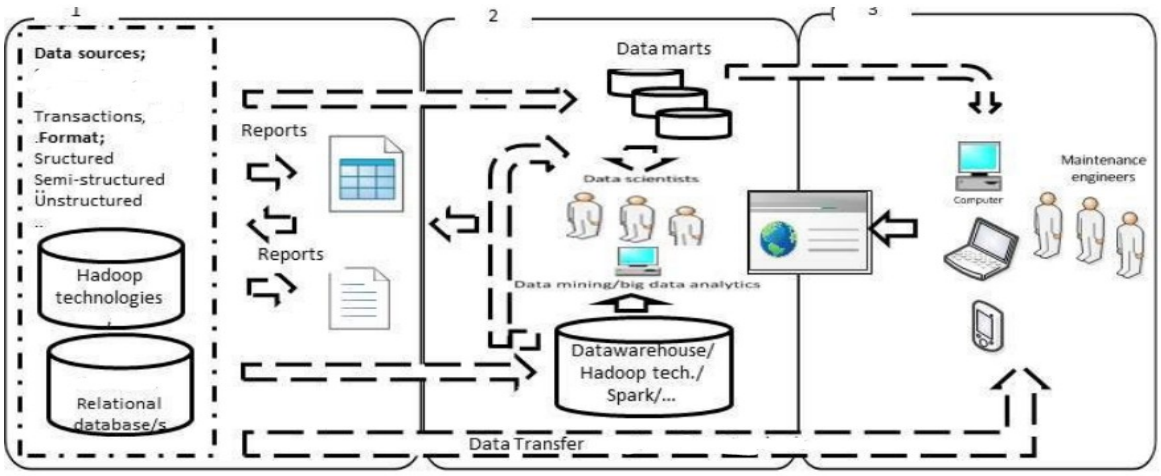


the paper, 10 stocks from various industries are selected to see if the selected algorithms can outperform the market.

Data Architecture

Given that there is a massive amount of data to be processed and more data will come in as time goes by, it is crucial that the storage and process system used is not only scalable, but can also handle the volume, velocity and variety of data. Although most data portfolio management is structured, we cannot rule out that there is a small portion of unstructured text data. Hadoop, therefore, is a promising platform because it overcomes many of the constraints, such as

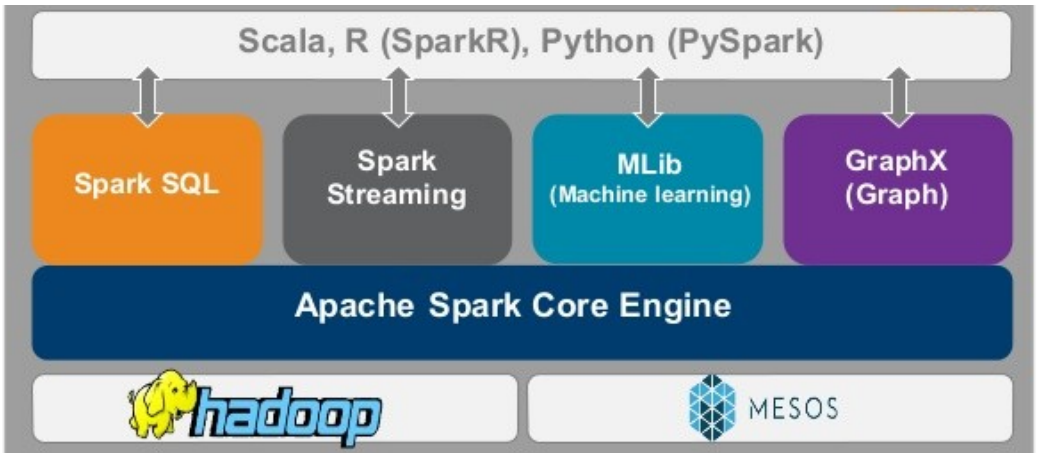
Exhibit 7: Architecture diagram



limitations of storage and capacities of computation of huge volume of data. Hadoop Distributed File System (HDFS) allows the data to be processed in parallel, which saves significantly amount of time. Additionally, it supports multiple data formats, which includes structured, semi-structured and unstructured data. Another advantage of using Hadoop is that data stored in traditional relational databases can also be integrated. Hive, a data warehouse software constructed on Hadoop, enables organizations and professionals to read, write, and manage data stored in distributed storage with the SQL support.

Exhibit 7 demonstrates how data can be seamlessly stored, processed and visualized. In the first stage, organizations can store financial data and pre-process the data in Hadoop or traditional relational databases. In the second stage, data analysis can take place in Spark. With the nature of financial industry, data has to be processed on-demand in a timely manner. Given

Exhibit 8: Spark – Data Processing Platform

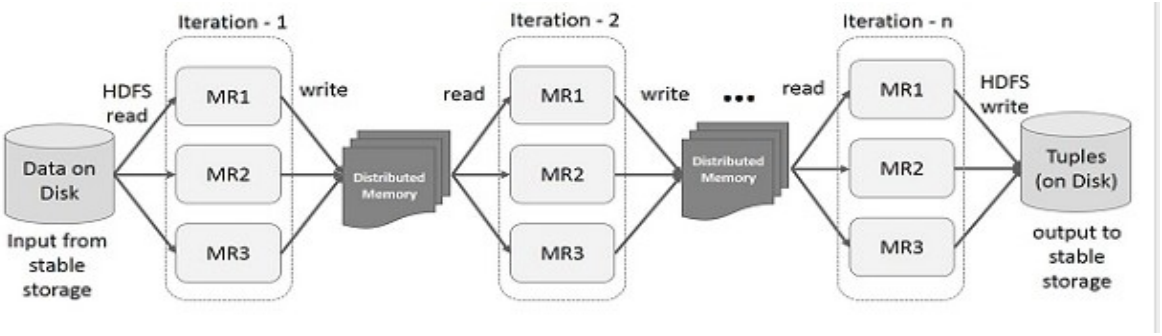


that Spark and Hadoop technologies can be utilized together, organizations can achieve economic storage, scalability, and fast data processing capacities using the suggested technologies in the architecture diagram. Spark is an in-memory, open source, unified cluster computing system that support various analytics applications in numerous programming languages (Exhibit 8). Efficiency and usability are two salient benefits when using Spark. With in-memory computing and data parallelism, Spark can process large scale data much more efficiently compare to Hadoop MapReduce, another distributed framework that analyze data in parallel. Moreover, because Spark has several mainstream programming in place and supports data streaming, machine learning library, SQL queries, and graph computing, Spark can be easily used and adapted by various organizations.

Spark's data structure is Resilient Distributed Dataset (RDD), an immutable, fault-tolerant collections of objects built through parallel transformations (Exhibit 9). The property of

fault tolerance means that Spark system can continue to function even if one of the components in the system fail. This fault-tolerant characteristic combined with the lazy evaluation of RDD allows Spark to maintain exceedingly efficient when processing an enormous amount of data. Lazy evaluation means that any transformations being done unto RDD is not executed until an action is called to see the results of transformations. Because the system only needs to calculate essential values, the overhead is reduced thus saving time. Given that there is a massive amount of financial data to be processed and analyzed, the iterative operations of Spark RDD and fault-tolerant characteristic allow financial managers to apply complex machine learning algorithms and receive diagnostic and prognostics analyses in a timely manner.

Exhibit 9: Spark RDD



It should be noted that Databricks is used in both second and third stage (Exhibit 7). Since Databricks not only permits organizations to use Spark, but it also provides a collaborative workspace in which colleagues can work together and visualization can be done instantaneously (Exhibit 5 and 6). Another advantage of Databricks is that financial managers can seamlessly switch between SQL, R and Python. Put simply, data exploration, data analysis, and data visualization can be done in one place. In our case, we first do a quick scan of the overall S&P 500 data with SQL (Exhibit 2), then explore, refine and visualize the data with Python(Exhibit 5 and 6). Then, predicative analysis can be performed to determine which stocks should be added

to the portfolio. With a unifying platform such as Databricks, financial managers can focus on assisting clients with data-driven results with handy visualizations instead of spending time on using different platforms to perform just a specific task.

Data Analytics - Algorithm

Given that the nature of portfolio management requires fund managers to constantly reallocate capital into different financial assets to maximize gain, numerous machine learning algorithms can be applied into this field. Before deciding to use neural network to build the portfolio model, some of the supervised learning algorithms, such as linear regression, logistic regression, decision trees and random forests, have been applied in determining the movement of portfolio (a group of individual stocks). Since all of the abovementioned algorithms can only be well-trained if the input features have high correlation to the output, obtaining appropriate features is paramount. Interest rates, number of cargo shipments for import and export, unemployment rates, daily moving averages (DMA), and on-balance volume (OBV) are selected and sourced as input features to help predict stock price movements.

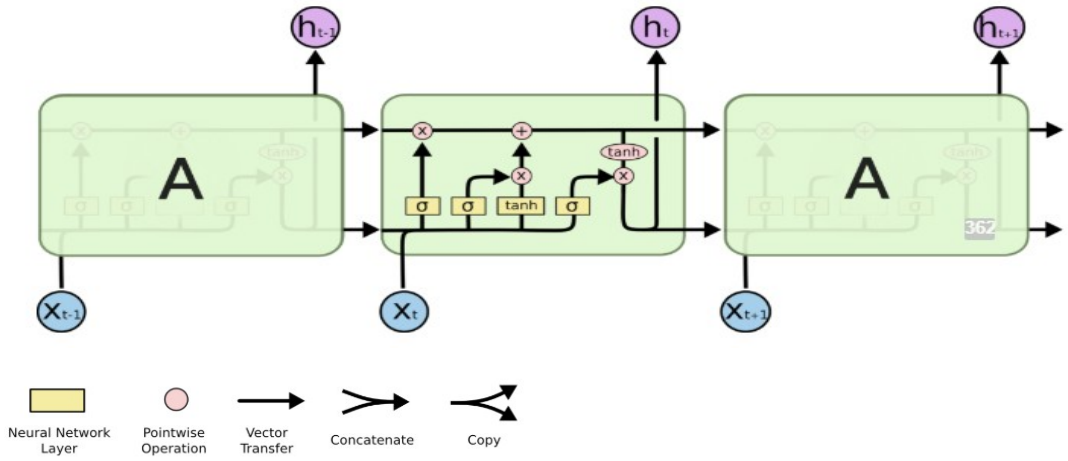
However, as most financial data has high noise and there tends to be "random walk movements" in the stock price (meaning that not all stock price movements are meaningful), the aforementioned algorithms fail to recognize a useful pattern to anticipate stock movements. For instance, when Brexit was first announced in the summer of 2016, the market overreacted even though the fundamental indicators for most companies were still sound. It took around 2 to 3 business days for the market to rebound. During the market rebound period, there was little insightful information the algorithms can gather from the features. Therefore, even if additional features are added, the algorithms need to be overfit in order for the outperform the baseline (S&P 500 index) in test datasets. In other words, the algorithms do not perform well for new

Exhibit 10: Convolutional neural network (CNN)

incoming data. For regression-based algorithms, the selected-features are simply not good enough; for classification algorithms, the results were similar to that of coin flip (around 50%).

As a result, neural network, more specifically reinforcement learning, is chosen to be a more appropriate method to use in determining the course of actions that need to be taken in order for the portfolio to maximize profits. Reinforcement learning is a technique that trains the algorithm (software agent) to take actions in a pre-described environment so as to produce the greatest desired reward. Under different circumstances, different neural networks would produce a variety of results. Convolutional neural network (CNN) and Recurrent neural network (RNN) are used to decide which network would be best the model to train. CNN is a connected multi-layered, feed-forward network which takes a fixed size inputs and produce fixed size outputs, generalize local patterns, and shines when data is sampled periodically (Exhibit 10). RNN, on the other hand, can handle arbitrary input and output size, connect previous information to the current task, and excels in situations where a sequence of time series data is involved. Long short-term memory (LSTM), a specialized RNN, differs from a standard RNN in that LSTM

Exhibit 11: Long short-term memory (LSTM)



utilizes multiple interactive layers to allow the algorithm to remember information for long periods of time (Exhibit 11). This would be useful when predicting similar stock movements.

In the environment of the reinforcement learning (Exhibit 12), stock price is normalized as followed: close price divides by open price, subtract from 1, then multiply by 100. A scaled numerical data would serve as a better feature because the model would give the same weight for all prices of stocks in any given time period, and thus avoid bias compare with a mere use of raw stock price. Note that price is the only input, no other feature is placed into the neural network models. In CNN, given that time is the only dimension, 1 x 3 kernel is found to work better after a series of trail and errors. For LSTM, the network is set to remember 20-day of trading period,

Exhibit 12: Reinforcement Learning Structure

which equals to about 1-month, so that recent trend can be used to maximize profit. Given the dataset only has limited data, trading pattern for over one month tends to overfit the dataset. Moreover, the algorithm is set to take an action in a fixed window length.

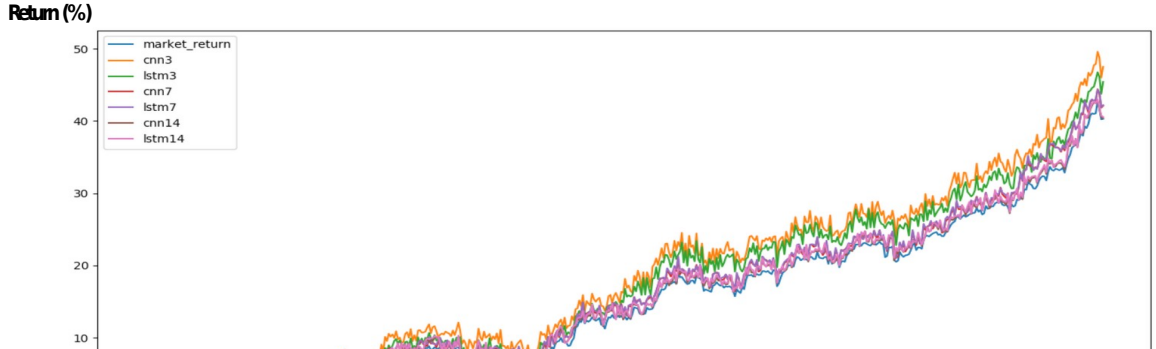
In our reinforcement learning model,

- 10 stocks from different sectors are chosen to form the portfolio
    - Apple, Blizzard, Comcast, Costco, CSX, Hasbro, Illumina, Intel, Marriott, Starbucks
    - During the EDA stage, we already know that if we place more stocks in the technology sector, the portfolio will general outperform the market. As a result, 10 representative stocks are chosen to eliminate the bias.
    - 2013-2015 (3 year) for training, and Jan, 2016 - Jan, 2018 for testing

- **Environment** in the model is the 10-stcok portfolio. Equal amount of share is purchase at the beginning (Price is normalized as discussed previously).
- **State** is the observation time that we allow the model to observe the market before making a trade and rebalancing the portfolio.
    - Since we are trying to mimic what would happen in a portfolio management setting, 3, 7, 14 observation-day are used to test the model.
- **Action** is what the model would do after a given observation period (i.e. the weight of each stock). The model will atomically decide to buy, sell, or hold after each observation period.
- **Reward** is profit derived from each action. The goal of the model is to maximize reward.
- Since price is the only input, our model generally rebalances (buy or sell) when a particular stock has been increasing or decreasing for 3 trading day.

Based on Exhibit 13, CNN-based models have better performance than LSTM-based models, and shorter action periods generally outperform longer periods (the best model, cnn3, outperformed the market by 7%). Given the temporal relationship among stock price movements, the performance of shorter action periods is expected. It should be noted that most models have similar performance within the first year. Furthermore, trading cost in the models is set at a negligible amount as most well-established financial firms have a low trading cost. In contrast, smaller and less-established organizations may not be able to enjoy such benefits and would need to take trading costs into account. All in all, neural network models do provide a better guidance than traditional machine learning algorithms in the field of portfolio management.

Exhibit 13: Trading Performance of Different Models



Future Work

From the result, it is evident that technical analysis can outperform the market by employing neural network. Given limited computing power, 1 input (normalized price) in a 10-stock portfolio is 10 variables that the model needs to handle. However, if computing power is not an issue, we can add more technical indicator such as rate of change for trading volume, daily and monthly moving average for individual stock price, moving average convergence divergence (MACD, an oscillating indicator for trending movement), and Bollinger Bands, a volatility and trend indicator. Additionally, Natural Language Processing (NLP) can be used to further find trends for a variety of stocks within the portfolio. With more input features, the model could have a better performance than the current one. All in all, portfolio managers would use all available variables to find patterns and continually reallocate its resources into different stocks to maximize profits.

References

Choundhry, R., and Garg, K. (2008). A Hybrid Machine Learning System for Stock Market Forecasting. Retrieved from http://waset.org/publications/8952/a-hybrid-machine-learning-system-for-stock-market-forecasting

Chowdhry, A. (2018, March 01). SAS COO And CTO Oliver Schabenberger Reflects On Staying Power Of Analytics. Retrieved from https://www.forbes.com/sites/amitchowdhry/2018/03/01/sas-coo-and-cto-oliver-schabenberger-reflects-on-staying-power-of-analytics/

Dunn, D. (2016, July 07). Clean, Accurate, Historical Stock Data. Retrieved from https://blog.quantopian.com/clean-accurate-historical-stock-data/

Ferguson, M. (2016, April 29). What is Spark? Retrieved from http://www.ibmbigdatahub.com/blog/what-spark

Hardik Patel. (2017, November 11). A Brief Review of Reinforcement Learning. Retrieved from https://www.hardikp.com/2017/11/11/reinforcement-learning-review/

Srivastava, P. (2017, December 23). Essentials of Deep Learning: Introduction to Long Short Term Memory. Retrieved from https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

Swalin, A. (2018, January 31). How to Handle Missing Data – Towards Data Science. Retrieved from https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

Tutorials Point. (2018, February 23). Apache Spark RDD. Retrieved from https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm