Machine Learning Engineer Nanodegree

Capstone Project Report

Egar Garcia

March 23rd, 2019

Stock Price Predictor

I. Definition

Project Overview

A stock market, also called equity market or share market, is a network of economic transactions, where the stocks of public companies can be bought and sold. The equity market offers companies the ability to access capital in exchange of a portion in the ownership of the company for interested outside parties.

In the stock market, other financial securities like exchange traded funds (ETF), corporate bonds and derivatives based on stocks, commodities, currencies, bonds, etc. can be traded. However, for purpose of this project only the exchange of stocks will be considered.

It is common to use stock market and stock exchange interchangeably, but the stock market is a general superset of the stock exchange. If someone is trading in the stock market, it means that it buys and sells stock/shares/equity on one (or more) of the stock exchange(s) that are part of the overall stock market.

The stock market offers an opportunity to investors to increase their income without the high risk of entering into their own businesses with high overheads and startup costs. On the other hand, selling stocks helps companies themselves to expand exponentially, when a company's shares are purchased it is generally associated with the increased in the company's worth. Therefore, trading on the stock market can be a win-win for both investor and owner.

The stock exchange that are leading the U.S. include the New York Stock Exchange (NYSE), Nasdaq, BATS and Chicago Board Options Exchange (CBOE). The Dow Jones Industrial Average (DJIA) is a price-weighted average of 30 significant stocks traded on the NYSE and the Nasdaq, it is the most closely watched market indicator in the world and it is generally perceived to mirror the state of American economy.

Projecting how the stock market will perform is a very difficult thing to do, there are so many factors involved in the prediction some of them emotional or irrational, which combined with the prices volatility make difficult to predict with a high degree of accuracy. Abundant information is available in the form of historical stock prices, which make this problem suitable for the use of machine learning algorithms.

Investment firms, hedge funds and individuals have been using financial models to better understand the market behavior and attempt to make projections in order to make profitable investments and trades.

Problem Statement

The purpose of this project is to build a stock price predictor, more specifically, the problem is to predict the closing price of a given company's stock in the trading days existing in a queried date range. For the scope of this project only the companies included in the Dow Jones Industrial Average are considered.

To address the problem a supervised learning approach is taken, the strategy to follow is to approach the problem as a particular case of forecasting in time series. As in supervised learning, a generalized function is tried to be inferred from the existing data which already contains the ground truth, the difference with this approach is that the existing data is in the past and the data to predict in the future, i.e. there is a clear separation of the predictor's values used for training and prediction, instead of being mixed and distributed along the possible set of values.

The different machine learning methods used in this project take historical stock data for a particular company over a certain date range (in the past) as training input, and outputs projected estimates for a given queried date range (in the future). The following ones are the methods taken in this project to address the problem of making predictions about how the closing stock prices will perform:

- ARIMA (AutoRegressive Integrated Moving Average): It is a very popular statistical method for time series forecasting that takes into account the past values to predict the future values. [1]
- Prophet: It is a time series forecasting library designed and pioneered by Facebook, that is claimed to be extremely simple to implement. [2] [3]
- LSTM (Long Short-Term Memory): It is a deep learning approach based in Recurrent Neural Networks (RNN) also used to make predictions in time series. [4] [5]

Metrics

To measure the performance of the predictions, the predicted value needs to be compared against the real value in the test and/or validation dataset, the predicted value is a (floating point) numerical value, thus using pure accuracy would not be convenient and not provide a useful description about how well the prediction is performing or improving through training iterations.

Instead, a metric that can provide some sort of average of the total error would be more effective, since the purpose of the prediction is to get a value as close as possible to the real one, but it does not need to be the same to be useful. For this reason the Root Mean Square Error (RMSE) is going to be used as evaluation metric for this project.

The formula to calculate the Root Mean Square Error is the following:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

Where N is the number of data points, y_i is the observed value or ground truth for the datapoint i and $\hat{y}_i$ is the predicted value for the data point i.

II. Analysis

Data Exploration

The original proposal contemplated to get the date from an open dataset called "EOD data for all Dow Jones stocks" in Kaggle, which contained the historical data for Dow Jones stocks, however for some reason this dataset is not longer available. Then the alternative and solution used in this project is to extract the data from the original source, which is an API provided by IEX Group Inc. (https://iextrading.com), it provides access to stock historical records to developers and engineers for free.

The API provided by IEX (which documentation can be found at https://iextrading.com/developer/docs/#chart) allows to retrieve historical stock price information for a maximum of 5 years back to the current date. The API can provide historical records for several companies, but for this project only the records for the ones in the Dow Jones are retrieved. An important aspect to notice is that the market is closed on weekends and some defined holidays.

The API retrieves the data in the in JSON format, which at the end contains records, where each record corresponds to the information of a trading day for a specific company, a company is identified by a ticker symbol (also known as stock symbol). For example to retrieve the historical stock prices for the ticker symbol MSFT (Microsoft Corporation) of the last 5 years, the call to the API would be https://api.iextrading.com/1.0/stock/aapl/chart/5y

The historic stock price records contain the following columns:

- `date`: Trading day
- `open`: Opening price
- `high`: Highest price
- `low`: Lower price
- `close`: Closing price
- `volume`: Number of shares traded
- `unadjustedVolume`: Number of shares traded also considering companies adjustments
- `change`: Change of closing price relative to the day before
- `changePercent`: Change in percent value
- `vwap`: Volume Weighted Average Price
- `label`: Formatted version of the date
- `changeOverTime`: Metric considered to measure the changes bases on weighted dates

The following is a fragment of how the historical stock looks for the thicker symbol APPL (Apple Inc.):

date	open	high	low	close	volume	change	changePercent	vwap
2014-02-21	69.9727	70.2061	68.8967	68.9821	69757247	-0.774858	-1.111	69.4256
2014-02-24	68.7063	69.5954	68.6104	69.2841	72364950	0.302061	0.438	69.1567
2014-02-25	69.5245	69.5488	68.4239	68.5631	58247350	-0.72101	-1.041	68.9153
2014-02-26	68.7667	68.9492	67.7147	67.9446	69131286	-0.618575	-0.902	68.1373
2014-02-27	67.917	69.4457	67.7738	69.2999	75557321	1.3553	1.995	68.8615
2014-02-28	69.4851	69.9671	68.571	69.1121	93074653	-0.187807	-0.271	69.2731
2014-03-03	68.7417	69.6913	68.6616	69.3117	59667923	0.199626	0.289	69.1371

The purpose of this project is to predict the future values of the closing price, which corresponds to the column `close`. To make predictions is necessary to chose a subset of columns which values can be known a priory and used for the prediction, unfortunately all the column values except date and label (which finally is a variant of the date) are unknown before the occurrence of the respective trading days. Then, the only data that can be used to predict the future closing prices is the past closing prices and the company itself.

Exploratory Visualization

In figure 1 it is presented a visualization of the historical closing prices for the 30 stocks of the companies in the Dow Jones, displaying the five years prior to March 23rd, 2019. The top image displays the prices along the time for all the companies, the bottom image is a comparison with the actual Dow Jones Industrial Average index (highlighted in black).

The visualization in figure 2 displays the historical prices of the stocks, but this time grouped by industry type, they are also contrasted with the Dow Jones Industrial Average index (identified with the symbol DIA and plotted in a dotted black line).

The Dow Jones Industrial Average is calculated with the stock prices of 30 selected public large companies, then it is not surprising that most of these stocks are behaving in a similar way to the DJIA. To calculate the DJIA, the prices are added and then divided by the Dow divisor, which is constantly modified.

In figure 3 is presented a visualization that shows the correlation of each one of the stocks in the Dow Jones against each other and against the actual DJIA. It can be observed that in most of the cases there is a high

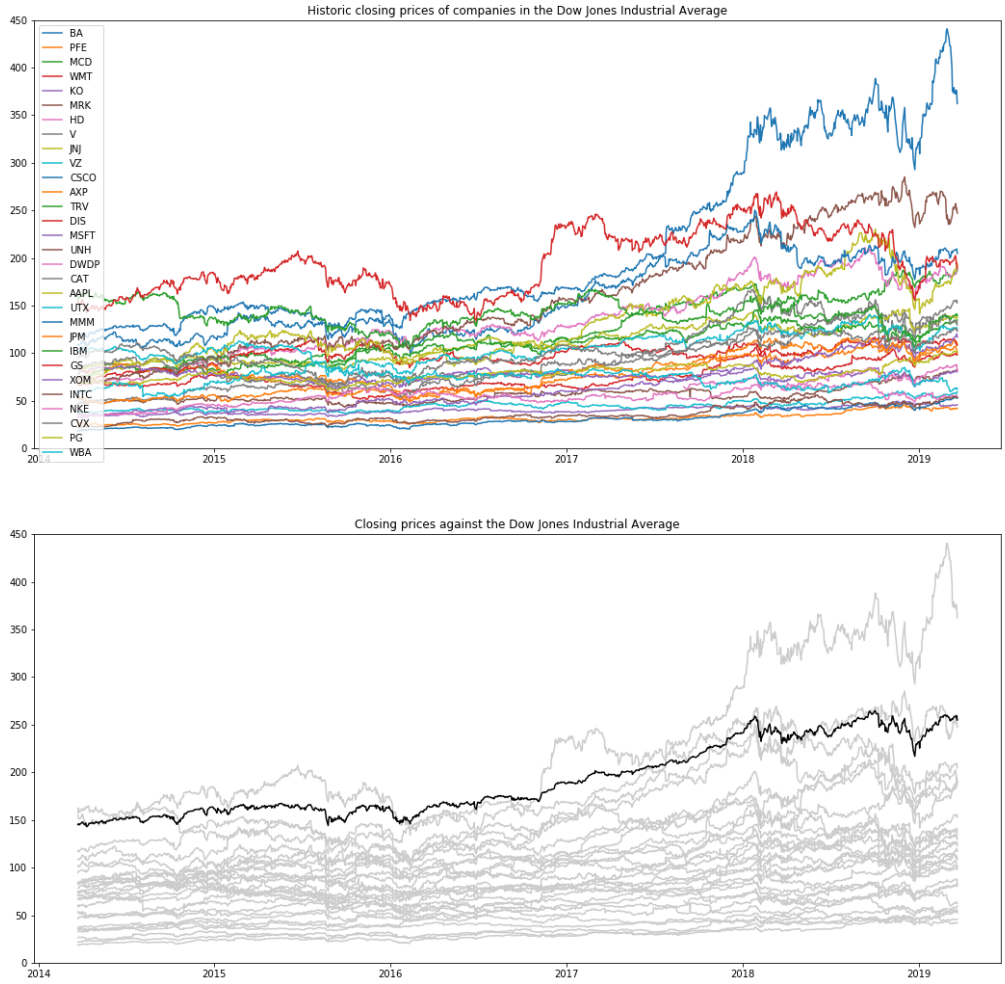

Figure 1: Dow Jones 5 year historic prices

Figure 2: Dow Jones 5 year historic prices per industry

correlation, observed by a dominance of the red color in the matrix (which is the color to indicate a high correlation, i.e. close to 1), only 4 companies seem no to follow the same tendency as the the general DJIA.

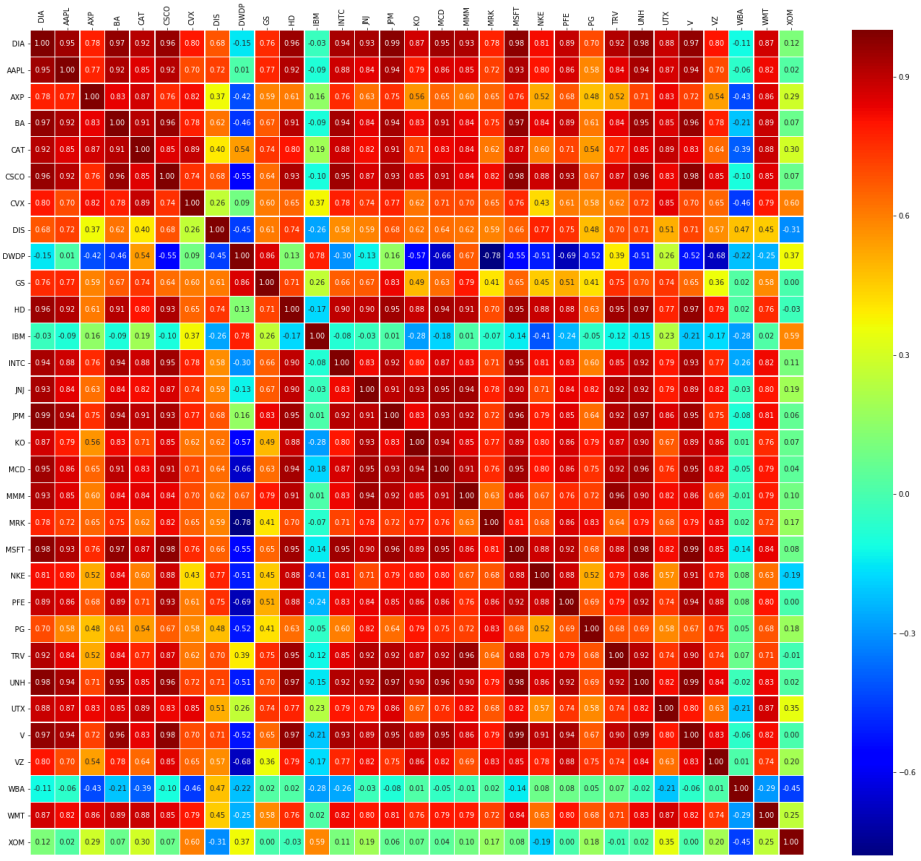

Figure 3: Correlation among Dow Jones stocks

Looking at the stocks one at a time, it looks like the stock prices fluctuates a lot and a clear pattern is not perceived, at least not at human comprehensible level, however several theories have been developed. This is understandable since the stock prices depend on a lot of different factors among them the company's financial health, economic supply-demand and even involving human emotions like trust, euphoria or panic.

At a macro level it can be observed that they are common events that seem to affect the stock prices as a whole, like a rise and sudden fall of prices around the beginning of 2018, or a drop of prices at the end of 2018. However these kind of events also do not seem to have a comprehensible pattern.

Algorithms and Techniques

A normal machine learning dataset is a collection of observations where time does not necessarily play a main role, predictions are made for new/unknown data, that might be considered as predicting the future, however

all the prior observations are pretty much treated equally, and the order or the observations is not taken in consideration, even is a good practice to shuffle the observations to perform the training.

A time series dataset is different, because they have an explicit dependence of the order between observations, thus the time plays a main role. For these datasets the time is both a constraint and also a structure that provides additional information.

During the exploration of the dataset it was realized that the only data that we can know a priory and use to make the predictions are the dates, then this characteristic makes the problem to forecast the stock closing prices to be a time series forecasting problem.

There are some known methods to address the problem of forecasting time series, the ones that are going to be used for the implementation of this projects are described bellow.

1. Linear Regression

Linear regression is the first and naive Machine Learning method to implement, the idea is just to obtain the regression line for the closing prices against the dates, this is the one that is going to be used to benchmark the other methods during the evaluation.

A variant of linear regression is also going to be explored for this project, the idea is that that the date components: year, month, day, week, day-of-week and day-of-year, might play a role in the determination of the closing price of a stock, then linear regression is applied using these components as predictors.

2. ARIMA (AutoRegressive Integrated Moving Average)

It is a very popular statistical method for time series analysis and forecasting, its acronym is descriptive, capturing the key aspects of the model itself [6]:

- AR (Autoregression): A model that uses the dependent relationship between an observation and some number of lagged observations.
- I (Integrated). The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.
- MA (Moving Average): A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

3. Prophet

It is an open source forecasting tool developed by Facebook, it is optimized for the business forecast tasks encountered at Facebook. They claim that the default settings produce forecasts that are often accurate as those produced by skilled forecasters, with much less effort. [2]

4. LSTM (Long Short-Term Memory):

A Recurrent Neural Network (RNN) can be thought of as multiple copies of the same network, each passing a message to a successor, aiming for them to learn from the past [7]. RNNs are good in handling sequential data but they have two main problems, the first one called "Vanishing/Exploding Gradient problem" presented as a result of the weights being repeated several times, and the second one called "Long-Term Dependencies problem" that happens when the context is far away [8].

Long Short-Term Memory networks are a special kind of RNNs (introduced by Hochreiter & Schmidhuber in 1997 [9]) with capability of handling Long-Term dependencies and also provide a solution to the Vanishing/Exploding Gradient problem [8]. They are currently used to address difficult sequence problems in machine learning and achieve state-of-the-art results [10].

Benchmark

To test the prediction's accuracy/performance for the machine learning methods used in this project, a couple of historical tests datasets for a ticker symbol are taken up to a given date and used to perform the training. Then the prediction is performed and benchmarked over validation sets for the following 1, 5, 10, 20, 40, 60 and 120 trading days, which is almost equal to predict for the next trading day and then 1, 2, 4, 8, 12 and 24 weeks, however there can be holidays in between.

To benchmark the performance of the different machine learning methods implemented in this project, they are compared against an initial naive solution which is produced via linear regression, where the predictor is the trading days' date (or a decomposition of it in day, month, year, week of year, day of week and day of year), and the predicted value the closing price. A result of the benchmarking can be expressed in terms of percentage of improving (or worsening) against the linear regression.

III. Methodology

Data Preprocessing

For this projects different ML methods/models are used, and the requirements for the inputs they receive are diverse, of course for all of them the outcome is the same, i.e. the closing price of the stock. It is also important to notice that all the methods performs the forecast for just one ticker symbol, then a common step is to filter the data set to get the records for the related ticker symbol (since the dataset contains the records for all the ticker symbols in the Dow Jones).

Each one of the methods/model addressed in this project requires its particular way to preprocess the data that is used in the training set, those are described bellow:

- Linear regression: The training set requires as a numeric representation of the date as a predictor (because dates are not supported), the preprocessing to prepare the training set consists in selecting just the `date` and `close` columns, then for the date its timestamp is calculated and using this value as a predictor instead of the actual date. The model is trained by using the timestamp as a predictor and the closing price as the outcome's ground truth.
- Linear regression using date components: In this approach also to prepare the training `date` and `close` columns are selected, then for the date the components: year, month, day, week, day-of-week and day-of-year are calculated. The model is trained by using the these components as predictors, they are numerical values so the linear regression can support them, the closing price is used as the outcome's ground truth.
- ARIMA: This model only needs a sequence of ground truth consecutive values in the time series to do the training, then to prepare the training set only the column `close` needs to be selected, however an important aspect is that the order must be preserved.
- Prophet: This models receives the date as predictor and the ground truth of the value to predict as outcome, however their columns should be named `ds` and `y` respectively. Then to prepare the training set for this model, the `date` and `close` columns are selected and then renamed.
- LSTM: The preprocessing mechanism for this model is the most complex of all, because LSTM takes as predictors sequences of a given length (known as time-steps) containing consecutive values in a time series and the outcome is the next value in the time sequence, i.e. the predictor is a subsequence of the time series instead of a single value. To prepare the training set only the values in the column `close` are needed, the date is not necessary since the important factor is the order, however the preprocessing for this method consists in creating the subsequences with the closing prices previous to the date of the respective outcome. Also as in most deep-learning approaches it's recommended that the values (to predict in this case) are normalized, then for this project the closing prices are scaled to the range from 0 to 1.

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section: - *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data? - Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution? - Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section: - *Has an initial solution been found and clearly reported? - Is the process of improvement clearly documented, such as what techniques were used? - Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section: - *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate? - Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data? - Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results? - Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section: - *Are the final results found stronger than the benchmark result reported earlier? - Have you thoroughly analyzed and discussed the final solution? - Is the final solution significant enough to have solved the problem?*

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section: - *Have you visualized a relevant or important quality about the problem, dataset, input data, or results? - Is the visualization thoroughly analyzed and discussed? - If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section: - *Have you thoroughly summarized the entire process you used for this project? - Were there any interesting aspects of the project? - Were there any difficult aspects of the project? - Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section: - *Are there further improvements that could be made on the algorithms or techniques you used in this project? - Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how? - If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?

VI. References

[1] A. O. A. Ayodele A. Adebiyi and C. K. Ayo, "Stock price prediction using the arima model," *International Journal of Simulation Systems, Science & Technology*, vol. 15, no. 4, pp. 105–111, 2014 [Online]. Available: http://ijssst.info/Vol-15/No-4/data/4923a105.pdf

[2] S. J. Taylor and B. Letham, "Prophet: Forecasting at scale." Facebook Research, 23-Feb-2017 [Online]. Available: https://research.fb.com/prophet-forecasting-at-scale

[3] R. Ritz, "Using facebook's prophet to predict mongolian stocks," 2018 [Online]. Available: https://medium.com/mongolian-data-stories/using-facebooks-prophet-to-predict-mongolian-stocks-cdf4feabd558

[4] Z. Pang X. and V. Chang, "Stock market prediction based on deep long short term memory neural network," in *Proceedings of the 3rd international conference on complexity, future information systems and risk (complexis 2018)*, 2018, pp. 102–108 [Online]. Available: https://www.scitepress.org/papers/2018/67499/67499.pdf

[5] A. C. M. P. David M. Q. Nelson and R. A. de Oliveira, "Predicting stock prices using lstm," *International Journal of Science and Research (IJSR)*, vol. 6, no. 4, pp. 1754–1756, 2017 [Online]. Available: https://www.ijsr.net/archive/v6i4/ART20172755.pdf

[6] J. Brownlee, "How to create an arima model for time series forecasting in python." Machine Learning Mastery, 09-Jan-2017 [Online]. Available: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

[7] C. Olah, "Understanding lstm networks." 27-Aug-2015 [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[8] M. Soni, "Understanding architecture of lstm cell from scratch with code." 21-Jun-2018 [Online]. Available: https://hackernoon.com/understanding-architecture-of-lstm-cell-from-scratch-with-code-8da40f0b71f4

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 6, pp. 1735–1780, 1997 [Online]. Available: http://www.bioinf.jku.at/publications/older/2604.pdf

[10] J. Brownlee, "Time series prediction with lstm recurrent neural networks in python with keras." Machine Learning Mastery, 21-Jul-2016 [Online]. Available: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/