Digital Signal Processing for Predicting Stock Prices

Okpor Margaret Dumebi
Department of Computer Science
Delta State Polytechnic, Ozoro
Delta State, Nigeria

ABSTRACT
With the substantial increase in the amount of data collected all over the world form various stock markets, it has become impossible to use traditional statistic and mathematical calculation for analyzing the data and using it for predicting trends like the opening and closing prices or highs and lows of the day for a company stock. Another crucial factor is the speed at which the data is generated. Hence it has become important to use concepts of digital signal processing for carrying out this type of analysis and prediction. Over the years many diverse types of filter and prediction algorithms have been developed which can, with a high amount of accuracy, predict stock market trends. This work will involve studying some of the popular filters and prediction algorithms used for stock market analysis and how they have been modified over the years to improve performance. This includes evaluating the techniques based on performance parameters such as speed, complexity and accuracy of prediction. In the next stage, some of the above algorithms will be implemented in Python and their performance will be analyzed using recent stock market data of Google finance. In the last section, an overall analysis of the results achieved will be discussed followed by conclusion

Keywords
Stock Market Prediction, DSP, Signal Processing

1. INTRODUCTION
Prediction algorithms have been a part of mathematical statistics since long and generally involved complex manual calculations that were time consuming, even for a small dataset. But from the late 1900's with the increase in use of computers for defense based applications, there was a need for prediction algorithms that were more suited to the recent technology. This was further motivated by the fact that there was a significant increase in the data being collected (especially from radars), thus making it impractical to use traditional statistics. With this data being available in digital form, it became apparent that there was a need for digital prediction techniques that could utilize the available computation resources [13 In 1960, [12] presented another direct expectation algorithm, presently known as the Kalman Filter. From that point forward a wide range of forecast calculations have been created, in view of the standards of likelihood and measurements from science, yet adjusted for handling considerable measures of advanced information. With the approach of globalization in the late 50% of twentieth century and development and digitization of budgetary business sectors everywhere on the world, these expectation

Calculations were applied for market investigation and anticipating securities exchange patterns. Moreover, since most stock market data can be seen as discrete signs (values examined at ordinary time spans like hourly, every day, month to month or yearly), it was anything but difficult to adjust existing channel strategies for financial exchange examination. In any case, it also his necessary the calculations to have the following two main characteristics:

• Accuracy: This was vital since most stock market predictions need to be done in real time and they need to be accurate and precise, else it could lead to major financial losses. To ensure this, it was necessary that the filters are not static and can adapt to numerous factors and changes in real time and accordingly make predictions. This meant that the prediction algorithms needed to use feedback and this lead to development of Adaptive Filters.

• Speed and Ability to process large data: The stock market trends depend on a large no. of factors like history of the stock prices, previous day closing prices, opening prices, day highs and lows as well as external factors like market trends of different countries, etc. All this translates to large amount of data that must be processed in a limited time to make effective predictions. The project will focus on the use of adaptive filters for analyzing stock market trends. Adaptive filters are recursive and use feedback to perform error correction. This makes them suited for making accurate predictions, as they can factor in the market volatility. The market data is generally available in the form of charts (histograms, pie charts, etc.) or in time series form, for a stock

Prediction of stock files has consistently been a tempting and charming field. Anticipating the financial exchange conduct through strategies and different techniques is a valuable instrument to help speculators to act with more prominent assurance, and facing the challenges, and unpredictability of a venture into thought and realize when to purchase the least expensive cost and when to offer to most exorbitant cost. Increase and abatement of financial exchange costs relies upon different factors, for example, measure of interest, conversion scale, cost of gold, cost of oil, political and monetary occasions, yet in the other view point we can consider the securities exchange cost variety as time arrangement and without documentation to the referenced variables, and just by finding the succession rules of cost train, make the cost expectation later on. For [14] ,the aim here is to achieve a fair degree of accuracy using two different prediction models and then using the mean squared error to provide an unbiased comparison of the results. The variable being predicted is today's closing price and opening price using either the previous day's opening, low, high and adjusted close price or using a time series of the closing prices. The first step is to acquire large amounts of historical data for analysis. The concept of data scraping has been used. The first algorithm is the Gradient descent/Steepest descent algorithm. This linear regression algorithm aims to generate a hypothesis function that will closely approximate the correct output signal. Linear regression is in essence curve fitting. Using the set of input training data, a curve is plotted, the equation of which is the hypothesis function. The cost work is the mean squared mistake between the anticipated yield and the real yield. The limited cost function helps in weight updation and improvement of the hypothesis function. The subsequent algorithm is an understanding of Prony's Normal Equation algorithm that can register the arrangement in a solitary advance while inclination plummet would require various emphases. The utilization of such all shaft frameworks has a computational bit of leeway over the more broad post zero models. To [11] the shafts are utilized to develop an increased Normal Equation. The posts are discovered with the end goal that they limit the squared mistake. [11] model is especially appropriate for transient price estimation.

Financial analysts investing in stock market usually are not aware of the stock market behaviour. They are facing the problem of trading as they do not properly understand which stocks to buy or which stock to sell in order to get more profit. In today's world, all the information pertaining to stock market is available. Analysing all this information individually or manually is tremendously difficult. This is where Data mining technique help.

Understanding that analysis of numerical time series gives close results, intelligent investors use machine learning techniques in predicting the stock market behaviour. This will allow financial analysts to foresee the behaviour of the stock that they are interested in and thus act accordingly.

This paper is aimed at prediction stock prices using python.

Hence, the objectives are to:

i. Generate input to our system with data from yahoo finance.
ii. Implement the system with python using open source libraries.

2. LITERATURE REVIEW

Since the 1970s researchers from the financial services industry have been employing a variety to technical analysis tools to identify trends in the market[8] attempted to build over the Capital Asset Pricing model by linking the movement of stock returns to the three fundamental variables, namely earnings yield, book-to-market ratio, and size. The model is based on the authors‟ research which suggested that value stocks outperform growth stocks and small capital outperform large capital stocks. [1] was the pioneer the development of logistic regression, which is often used when outcome to be predicted is binary in nature. [4] later used LR to construct the default prediction model. [6]) carried out an experiment which was led by [11] which utilized direct relapse line to produce new information from authentic information, and distinguished the examples that portray the stock pattern . The authors sounded an optimistic note regarding the inherent ability of linear regression to capture correlations and project them into the future. A bouquet of factors, for example, political stability (measured by an intermediary Worldwide Governance Index), quality of corporate administration and money related proportions are factors that have been considered for stock price forecasts [10]. Quite surprisingly the focused use of stock price data was minimal. While financial ratios such as Earnings per Share, Price/Earnings Ratio, etc. are good indicators of the core fundamentals of a stock, their ability to model real time stock movements in uncertain and imperfect markets could be questioned. This anomaly provoked our study to understand the predictive quality of stock price data.

The 1951 thesis of Peter Whittle, Hypothesis testing in time arrangement examination, was the first to investigate the general ARMA model. It was later advocated in the 1971 book Time Series Analysis: Forecasting and Control wrote by George. P. Box and Gwilym Jenkins. Furthermore, the time arrangement hypothesis gives us a few experiences into the sequential relationship impact [5]. [13] sums up that the greater part of the models in past examination considers the slacked returns, and his model isn't an exemption. A famous variation of the ARMA model, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Model was proposed by [7] and [3]. The GARCH cycle models the changes by an ARMA type measure. The model regularly presumes that instability is mean returning. Henceforth, utilizing GARCH (1,1), current unpredictability is anticipated as an element of a drawn out mean, one slacked term of change and one slacked term of squared return. An alluring element of the mode is its capacity to manage overabundance kurtosis in the bring dissemination back. Nonlinear GARCH models expand the fundamental commitments by [7] and [3]to join the lopsided effects of stuns or updates on equivalent greatness yet inverse sign on the restrictive change of advantage [2]. Another model developed from the ARMA model is the Autoregressive Integrated Moving Average (ARIMA). The overall exchange work model utilized by the ARIMA technique was talked about by [4] The ARIMA model changes non-fixed information into fixed information before preparing it. It's regularly used to demonstrate straight time arrangement information.

3. METHODOLOGY

The use of an appropriate device enhances effectiveness and efficiency in every research work. A "system development methodology (SDM) refers to the step-by-step procedure used to structure, plan, and control the process of developing an information system." We adopted the Object-Oriented Analysis and Design Methodology in analysis of this program. This methodology features the architectural design of the system.

4. DISCUSSION AND RESULT

Python was used for the implementation of this system.

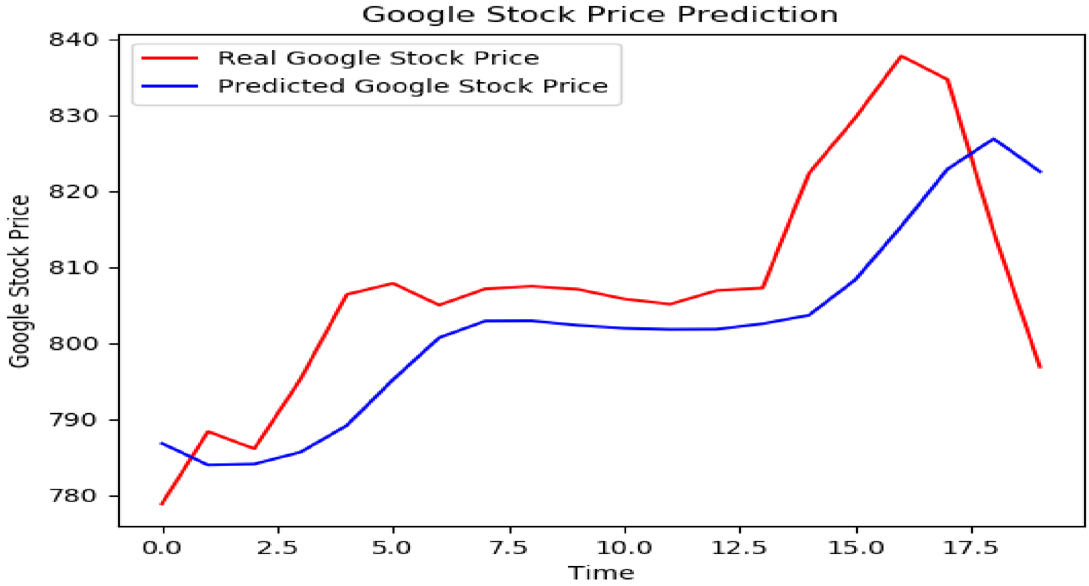
Python is an open-source object-arranged language. Its viability cuts over, Application Programming Interface (API), Platform autonomy, reproduction, low-level programming, object linker installing, network design and so forth. A Python program need not to be aggregated each time it is going to run once it is created.

TR AINED Dataset from Google Finanace

Date	Open	High	Low	Close	Volume
1/3/2017	778.81	789.63	775.8	786.14	1,657,300
1/4/2017	788.36	791.34	783.16	786.9	1,073,000
1/5/2017	786.08	794.48	785.02	794.02	1,335,200
1/6/2017	795.26	807.9	792.2	806.15	1,640,200

1/9/2017	806.4	809.97	802.83	806.65	1,272,400
1/10/2017	807.86	809.13	803.51	804.79	1,176,800
1/11/2017	805	808.15	801.37	807.91	1,065,900
1/12/2017	807.14	807.39	799.17	806.36	1,353,100
1/13/2017	807.48	811.22	806.69	807.88	1,099,200
1/17/2017	807.08	807.14	800.37	804.61	1,362,100
1/18/2017	805.81	806.21	800.99	806.07	1,294,400
1/19/2017	805.12	809.48	801.8	802.17	919,300
1/20/2017	806.91	806.91	801.69	805.02	1,670,000
1/23/2017	807.25	820.87	803.74	819.31	1,963,600
1/24/2017	822.3	825.9	817.82	823.87	1,474,000
1/25/2017	829.62	835.77	825.06	835.67	1,494,500
1/26/2017	837.81	838	827.01	832.15	2,973,900
1/27/2017	834.71	841.95	820.44	823.31	2,965,800
1/30/2017	814.66	815.84	799.8	802.32	3,246,600
1/31/2017	796.86	801.25	790.52	796.79	2,160,600

Stock Price Prediction



Code generated

```
# Recurrent Neural Network
# Part 1 - Data Preprocessing
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the training set

dataset_train = pd.read_csv('Google_Stock_Price_Train.csv')
training_set = dataset_train.iloc[:, 1:2].values

# Feature Scaling
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)

# Creating a data structure with 60 timesteps and 1 output
X_train = []
y_train = []
for i in range(60, 1258):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)
# Reshaping
X_train = np.reshape(X_train, (X_train.shape[0],
X_train.shape[1], 1))

# Part 2 - Building the RNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Dense
```

```
from keras.layers import LSTM
from keras.layers import Dropout

# Initialising the RNN
regressor = Sequential()

# Adding the first LSTM layer and some Dropout
regularisation
regressor.add(LSTM(units = 50, return_sequences = True,
input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

# Adding a second LSTM layer and some Dropout
regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a third LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a fourth LSTM layer and some Dropout
regularisation
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))

# Compiling the RNN
regressor.compile(optimizer = 'adam', loss =
'mean_squared_error')

# Fitting the RNN to the Training set
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)

# Part 3 - Making the predictions and visualising the results

# Getting the real stock price of 2017
dataset_test = pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price = dataset_test.iloc[:, 1:2].values
# Getting the predicted stock price of 2017
dataset_total = pd.concat((dataset_train['Open'],
dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) -
60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, 80):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1],
1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price =
sc.inverse_transform(predicted_stock_price)

# Visualising the results
plt.plot(real_stock_price, color = 'red', label = 'Real Google
Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted
Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
plt.show()
```

========**Out Put**===================

```
  32/1198 [..............................] - ETA: 1:16 - loss: 0.3795
  64/1198 [>.............................] - ETA: 39s - loss: 0.3600
  96/1198 [=>............................] - ETA: 26s - loss: 0.3213
 128/1198 [==>...........................] - ETA: 20s - loss: 0.2930
 160/1198 [===>..........................] - ETA: 16s - loss: 0.2600
 192/1198 [===>..........................] - ETA: 13s - loss: 0.2307
 224/1198 [====>.........................] - ETA: 12s - loss: 0.2042
 256/1198 [=====>........................] - ETA: 10s - loss: 0.1804
 288/1198 [======>.......................] - ETA: 9s - loss: 0.1683
 320/1198 [=======>......................] - ETA: 8s - loss: 0.1576
 352/1198 [=======>......................] - ETA: 7s - loss: 0.1479
 384/1198 [========>.....................] - ETA: 7s - loss: 0.1375
 416/1198 [=========>....................] - ETA: 6s - loss: 0.1291
 448/1198 [==========>...................] - ETA: 5s - loss: 0.1211
 480/1198 [===========>..................] - ETA: 5s - loss: 0.1148
 512/1198 [===========>..................] - ETA: 5s - loss: 0.1096
 544/1198 [============>.................] - ETA: 4s - loss: 0.1048
 576/1198 [=============>................] - ETA: 4s - loss: 0.1010
 608/1198 [==============>...............] - ETA: 4s - loss: 0.0964
 640/1198 [===============>..............] - ETA: 3s - loss: 0.0926
 672/1198 [===============>..............] - ETA: 3s - loss: 0.0892
 704/1198 [================>.............] - ETA: 3s - loss: 0.0858
```

```
 736/1198 [=================>............] - ETA: 2s - loss: 0.0829
 768/1198 [==================>...........] - ETA: 2s - loss: 0.0803
 800/1198 [===================>..........] - ETA: 2s - loss: 0.0779
 832/1198 [===================>..........] - ETA: 2s - loss: 0.0758
 864/1198 [====================>.........] - ETA: 1s - loss: 0.0736
 896/1198 [=====================>........] - ETA: 1s - loss: 0.0712
 928/1198 [======================>.......] - ETA: 1s - loss: 0.0692
 960/1198 [=======================>......] - ETA: 1s - loss: 0.0673
 992/1198 [=======================>......] - ETA: 1s - loss: 0.0655
1024/1198 [========================>.....] - ETA: 0s - loss: 0.0639
1056/1198 [=========================>....] - ETA: 0s - loss: 0.0623
1088/1198 [==========================>...] - ETA: 0s - loss: 0.0608
1120/1198 [===========================>..] - ETA: 0s - loss: 0.0594
1152/1198 [===========================>..] - ETA: 0s - loss: 0.0580
1184/1198 [============================>.] - ETA: 0s - loss: 0.0565
1198/1198 [==============================] - 6s 5ms/step - loss: 0.0559
```

5. CONCLUSION

The initial analysis shows significant correlation between news values and weekly stock price change. But we need to be careful when using this result since it is likely that the result is dominated by a number of influential observations and is not reflective of the general trend. In general the weekly stock price changes within different intervals of news values behave in the same way as what is expected. There are a number of limitations and shortcomings in our program and potential improvements can be made to our data collection. Further researches can be done with possible improvements such as more refined search data and more accurate algorithm to compute news values.

6. REFERENCES

[1] Altman, E. (1968) Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. Journal of Finance, 23, 589-609.

[2] Arowolo, W. (2013). Predicting Stock Prices Returns Using Garch Model. International Journal of Engineering and Sciences, 2(5) Pages 32-37

[3] Bollerslev, T. (1986), A Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics 3 (1), 37-45

[4] Box P., George E., and George C. (1975). Tiao, Intervention analysis with applications to economic and environmental problems, Journal of the American Statistical Association 70.349.

[5] Campbell, J. and Yogo M. (2006). Efficient tests of stock return predictability, Journal of Financial Economics, 81: 27-60.

[6] Dutta, Bandopadhyay, and Sengupta (2012). Prediction of Stock Performance in the Indian Stock Market Using Logistic Regression, International Journal of Business and Information, 7 (1)

[7] Engle R. (1982). The Use of ARCH/GARCH Models in Applied Econometrics. Journal of Economic Perspectives, 15(4), 157–168

[8] Fama E. and French K. (1992). The Cross-Section of Expected Stock Returns. The Journal of Finance, 47(2) 427 – 465.

[9] Gharehchopogh, F., and Khalifehlou, Z. (2012), A New Approach in Software Cost Estimation Using Regression Based Classifier, AWER Procedia Information Technology and Computer Science, Vol 2, pp. 252-256.

[10] Gharehchopogh F., Bonab M., and Khaze B. (2013). A Linear Regression Approach to Prediction of Stock Market Trading Volume: A Case Study, International Journal of Managing Value and Supply Chains (IJMVSC,) 4(3), September 2013

[11] Hayes, M. (2002), Statistical Digital Signal Processing and Modeling, Pg 144-145, Singapore: John Wiley & Sons (Asia) Pte. Ltd.

[12] Kalman, R.E., "A new approach to linear filtering and prediction problems", ASME Transactions, Series D, 82, 35-45,

[13] Martinelli R. (1995). “Market Data Prediction with an Adaptive Kalman Filter”, Haiku Laboratories, Retrieved on the 4th of November, 2019 from ww.haikulabs.com/mdpwakf.htm.

[14] Mojaddady, Nabi and Khadivi. (2008), Stock Market Prediction using Twin Gaussian Process Regression Modeling, Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

[15] Tiong, Ngo, and Lee Y. (2013). Forex Trading Prediction using Linear Regression Line, Artificial Neural Network and Dynamic Time Warping Algorithms, 4th International Conference on Computing and Informatics, ICOCI

[16] Vaisla B. 2010. An Analysis of the Performance of Artificial Neural Network Technique for Stock Market Forecasting, International Journal on Computer Science and Engineering, Vol 2 (6)