International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 22 (2018) pp. 16036-16041

Risk Analysis and Prediction of the Stock Market using Machine Learning and NLP

Sujay Lokesh, Siddharth Mitta, Shlok Sethia, Srivardhan Reddy Kalli, Manisha Sudhir

Department of Computer Sceince and Engineering, R.V College of Engineering, Banglore, Karnatka, India

Abstract

The stock market has been a source of income for many for the past 200 years and will continue to do so in the years to come. Predicting the rise and fall of the stock market has been the intention of many financial analysts and they have been trying to tackle that issue but with limited success. With the meteoric improvement in technology leading to higher processing power, increased storage and better algorithms it is now more possible than ever to do so. But to shatter the taboo that stock market is meant only for people well versed in finance, it is required that a new solution is developed to tell the interested user what is the risk associated and expected increase or decrease in value with respect to his or her investment. The proposed solution is an end product which is easy to use and understand. It uses the stock market dataset for training the model, sentiment analysis on tweets and risk calculation to overcome the existing barrier and making the riches of stock market investment available for all.

Keywords: Fintech, Machine Learning, Natural Language Processing, Predictive analysis, Risk calculation, Stock market

I. INTRODUCTION

Investing in the stock market is one of the most risky decisions made by a person as it may yield in complete loss or tremendous profit and it is upto the investor's skills to judge whether or not the investment will provide the desired results. It is important that the scope of human error in such decision making scenarios is reduced so that profit is maximized[2]. The chartist theories say there is some hidden information in the historical prices of a security that gives a clue to future price of that security. In [4], the researchers have used historical data to predict the position of stock market. The results of [4] prove that historical data has strong predictive ability. It is essential to understand that the stock value doesn't solely depend upon the historical values but also on the organisation's current proceedings and public sentiment of that organisation. Application of technologies like machine learning and sentiment analysis on a cloud platform decreases the risk quotient by informing the investor about the intricacies of the decision he or she is about to make[3][4].

Machine learning is the branch of technology which has become necessary in almost each and every field of science and by using this concept as an advantage the stock opening prices for a future date can be predicted by training the machine learning model by giving the stock prices for previous dates.

Sentiment analysis refers to parsing a given text and finding out the sentiment of the text in terms of score and magnitude, that is, if it's positive or negative and also the intensity of the statement in the text. When sentiment analysis is applied on tweets obtained from twitter regarding an organisation, public opinion on that organisation becomes evident which is an important factor in deciding whether to invest or not.

These aforementioned procedures when applied require processing power and due to the recent advances in cloud infrastructure, usage of services such as Google Cloud Services on the Google Cloud Platform results in optimal performance at low cost[5].

On the user end, the information which is obtained after processing all the data is to be displayed in a well defined and visually pleasing manner. To attain this, a mobile application was created which takes in the user input which are number of days of investment and amount to be invested, and gives out the sentiment regarding the organisations, the trend of the stock values in a graphical representation and also the risk and dollar amount associated with that company.

The proposed system inculcates all the previously mentioned functionalities in a compact and efficient manner. The usage of machine learning, sentiment analysis, cloud platform and a mobile application in a novel and ingenious way makes the system scalable and viable for practical implementation.

II. RELATED WORK

Extensive research has been done in the field of stock market analysis but required levels of accuracy have never been attained. Machine learning has been applied not only on the previous stock values but also on various other features such as volume of stock traded, average variance etc to boost the efficiency in predicting the values[3],[7]. There have been attempts to include sentiment analysis as well to improve the viability of the readings[13]. A lot of work has also been done in the field of making these predictors in a portable version for ensuring ease of access. Neural networks and sentiment analysis libraries are used in collaboration to obtain more accurate readings[3][8]. Very little research has been done to assign a risk to the investment as major focus is given to try and predict the precise value of the stock rather than associating risk based on trend and public sentiment. The system proposed in this paper assigns a risk value on the investment along with various other functionalities on a portable platform.

III. MOTIVATION

The existing systems cater to huge finance organisations and not to the common people[10]. The common man who is not

well equipped with the knowledge of the stock market deserves a solution which not only informs the investor regarding the historic and predicted trend of the company along with the public opinion but also suggests him or her investments by placing a risk percentage on each investment. There is a lack of compact and simple mobile platforms to view information regarding the stock market. The system proposed in this paper provides all of these functionalities at zero cost and is available easily. Any individual can make an informed decision with the help of this product which is ready for consumer use and easy to understand.

IV. IMPLEMENTATION

The implementation is based on a client-server architecture model. The client in this case is the mobile application which sends a request to the server whenever a user uploads an input to the firebase using the mobile application. The server continuously listens for requests from the user with the help of an event listener function which records any change in the firebase and performs the necessary operations after the change is recorded[14]. The mobile application was built using *Android Studio*[20] and it contains a Google sign in for the user as shown in Fig 1. No additional sign in information is required which reduces the possibility of data theft. The computation was performed on a remote VM instance set up on the Google Cloud Platform from the Singapore region[15]. This Compute Engine performs all the necessary computations required. By using a remote VM instance it was possible to reduce the size of the mobile application and avoid excessive resource utilization of the mobile device. Firebase was used as an intermediary between the mobile application and the server[16]. JSON parsing was used to traverse the firebase tree to obtain the required child and parent values in the firebase.

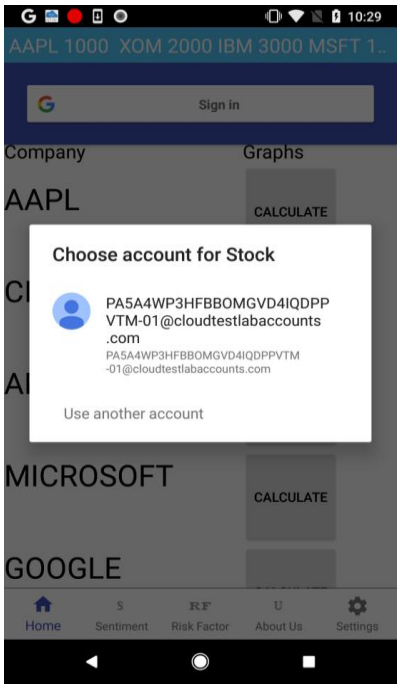

Fig 1: Google sign for the user

The implementation can be divided into three modules as they are sequential in nature.

Module 1: Prediction of stock values using polynomial regression

The first module corresponds to predicting the stock market values for future dates. This was done with the help of a machine learning model.

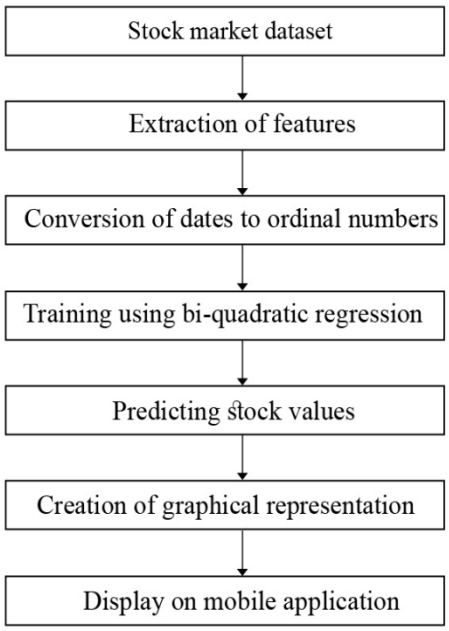

Fig 2: Block diagram for module 1 (Stock value prediction)

The training data was obtained from *Kaggle* where the opening stock prices were from the dates 2006-01-01 till 29-12-2017 for over 30 companies in the format YYYY-MM-DD. The dates and the stock opening values were chosen as the variables for the machine learning algorithm. To use the dates as a feature, it had to be converted into ordinal numbers which was done using python library *datetime* and the function *toordinal*. The ordinal numbers obtained as a result of these operations were then used as the independent variable and the stock value corresponding to the ordinal number as the dependent variable. Bi-quadratic regression was used as the model since it doesn't overfit or underfit the training data. 2000 days were taken as the training data for the regression model. This process is shown in Fig 2. The machine learning function was made available through Python's *sklearn* and *numpy* libraries[11],[12]. The trained model was then used to find the stock value for any date required. The formula used for bi-quadratic regression is shown in equation (1).

$$y_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 \qquad (1)$$
$$i = 1, 2, 3, ..., n$$

y : dependent variable

x : independent variable

i : number of independent variables

β0 : is the regression constant

β1, β2,....., βi : is the partial regression coefficient for the independent variables, 1, 2,...., i respectively.

The obtained graph after performing the curve fitting on the training data is plotted using Python's *matplotlib* library[11]. This computation was performed on a remote backend Google Cloud Server from Singapore region. The graphs produced were stored in the Google Cloud Storage Bucket which can be viewed from the app dynamically by making calls to the API. These graphs depict the trend of the stock value increase or decrease for the company in an easily comprehensible manner which allows the user to make an informed decision[21].

Module 2: Sentiment analysis of twitter data

Using Twitter API, all tweets related to the specific company were obtained. The call to the Twitter API results in an accumulation of 10,000 tweets related to that company in realtime. To perform sentiment analysis the obtained data must be pre-processed. The tweets were first checked for diacritics and all non-ascii characters were removed[6]. The removal of tweets from the official twitter handle of that respective company was then performed in order for the sentiment analysis results to be unbiased[4].

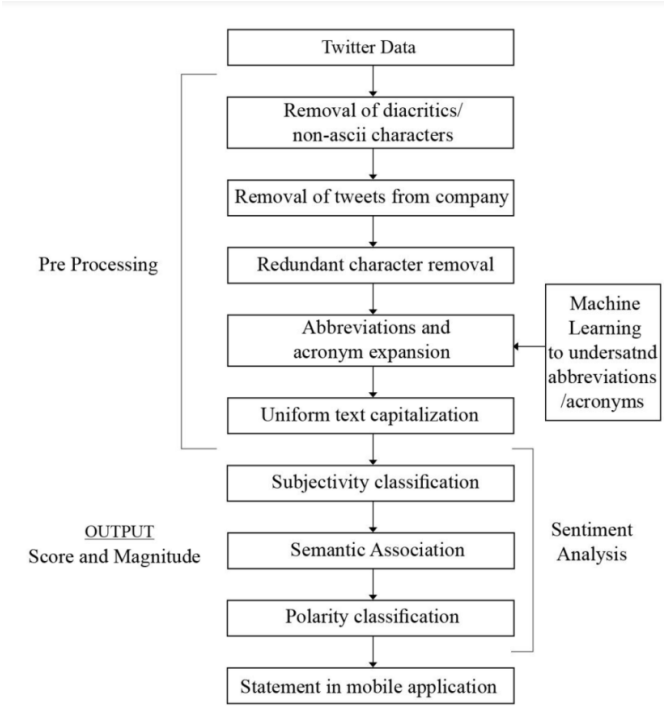

Fig 3: Block diagram for module 2 (sentiment analysis)

The subsequent steps in pre-processing include removal of hashtags, removal of redundant characters, expansion of abbreviations and uniform text capitalizations. The resultant file was then given to Google's Natural Language API where the subjectivity classification, semantic association and polarity classification was performed[10]. The output of this API was score and magnitude. The score reflects how positive or negative a set of tweets are and the magnitude refers to how opinionated the text is[17]. The score was then multiplied with the magnitude to obtain the sentiment as shown in equation 2. Based on the result of this operation there were statements displayed on the mobile application to inform the user about the public sentiment for that company. This process is shown in Fig 3. The statement was sent through firebase to the mobile application. The sentiment analysis is performed on live tweets so the sentiment obtained is very much up to date in terms of public opinion which leads to more accurate results. All of these steps were performed on a remote compute engine on the Google Cloud Platform[15].

$$Sentiment = Score * Magnitude \quad (2)$$

Module 3: Risk and change in stock price calculation

Risk calculation is performed for each of the 5 companies and the result is displayed on the mobile application. The number of days for the investment and also the amount to be invested are obtained as user inputs through the mobile application as shown in Fig 5.

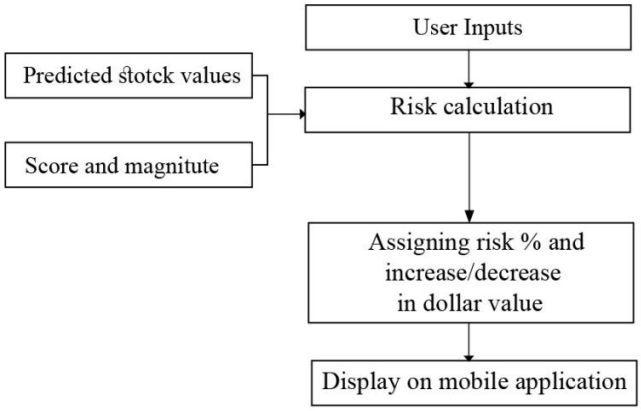

Fig 4: Block diagram for module 3 (Risk calculation)

The date for the current day, which is the day the investment period starts was obtained by using the python library *arrow*[11]. This date was converted into an ordinal number and the trained regression model was used to find the stock values for all days starting from this day till the investment period ends. These values were then compared with the stock value for the present day and a count was kept for each, the number of days where the stock value of the current day, was greater and lesser than the predicted values for the rest of the days. This operation was performed for all 5 companies. The previously calculated product of the score and magnitude values which were obtained by performing sentiment analysis on each of the 5 companies was found. These values when taken along with the count values formed an accurate metric

for risk prediction. Based on these two values a risk percentage range was assigned.

In order to find the dollar increase or decrease in value for each stock, mean of the predicted values was taken and subtracted with the present day value which will be the price of the stock the investor will be investing at. This difference if negative shows the decrease in the stock value or if positive shows the increase in the stock value. These values along with the risk percentage range are uploaded to the firebase from which the mobile application fetches the information to display to the user. This process is shown in Fig 4. These calculations are done on the google cloud server by creating a VM instance[15].

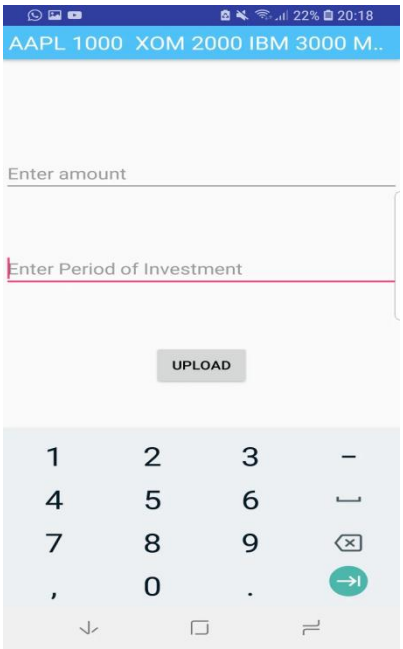

Fig 5: User input page for risk calculation

V. RESULTS

The results of the 3 modules can observed separately for clearer understanding.

First the machine learning aspect produces a graphical representation with the stock values on the y axis and the dates converted into ordinal numbers on the x axis. The red line depicts the curve fitting result obtained by performing bi-quadratic regression as shown in Fig 6 and 7. This graph is strictly to be used for observing the trend of the stock values over a period of time. This graph can be viewed by the user on the home screen of the mobile application by pressing the calculate button as shown in Fig 8.

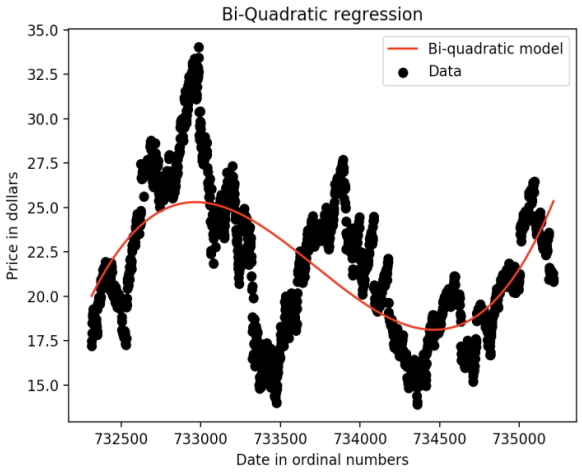

Fig 6: Graphical representation of Cisco stock

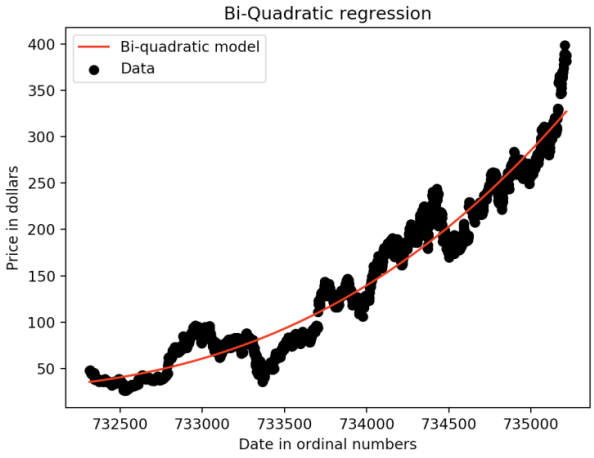

Fig 7: Graphical representation of amazon stock

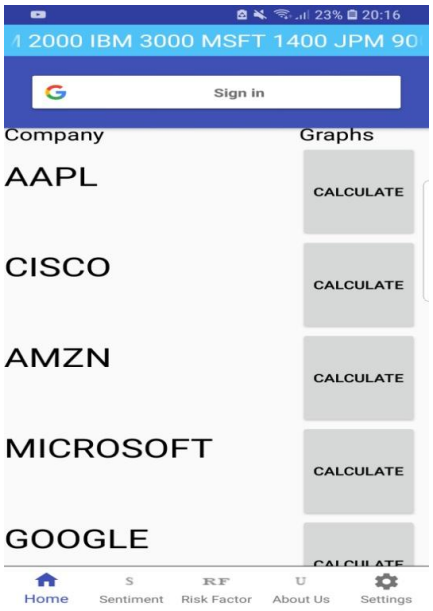

Fig 8: Home page where user can view the stock trends for a company

The next part involves the sentiment analysis of the tweets regarding the company. The user can view the public sentiment for a particular company by pressing the calculate button on the sentiment screen on the mobile application as shown in Fig 9. The sentiment calculated is displayed in terms of a statement in the mobile application along with some information regarding the company as shown in Fig 10.

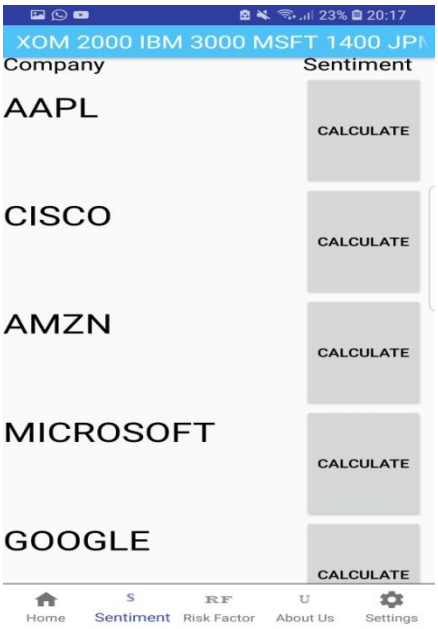

Fig 9: Sentiment page where user can choose to view the sentiment of required company

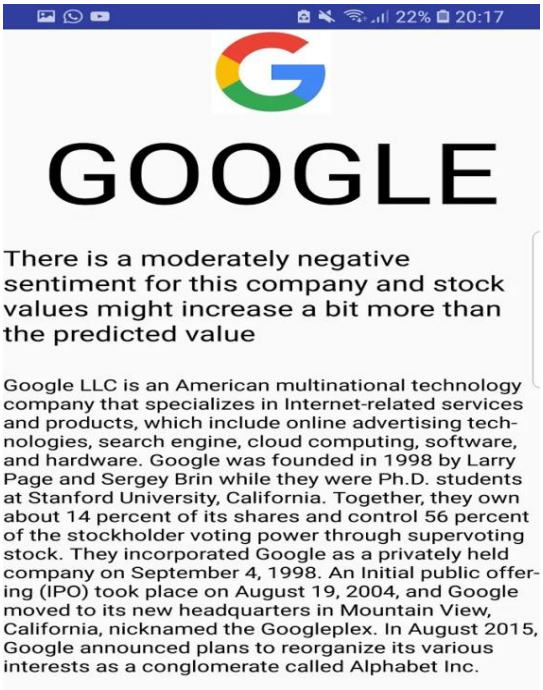

Fig 10: Predicted sentiment of Google as displayed in the mobile application

Module 3 results include risk calculation and increase or decrease in dollar value. These values are calculated after the user gives his or her inputs and are displayed as shown in Fig 11. The risk percentage is divided into 8 ranges where 100 is the highest and 0 is the least amount of risk involved in the investment. Since the risk calculation depends on the number of days it changes with each distinct user input. Also the predicted increase in value is displayed for each company in terms of dollars so that the investor can get an idea on his or her returns. This can be repeated any number of times to view the various returns on the investment.

Stability tests for the mobile application were run on the Google Cloud Platform using the service *Test lab* and the app passed all the required tests providing the users with a crash free experience.

Fig 11: Risk associated with each company

VI. CONCLUSION

From the obtained results, we can say that the risk factor calculated considering twitter data gives us a risk percentage range along with the growth per stock in dollar value instantaneously and with reasonable accuracy upon the user providing input.

In conclusion it can be confidently said that the proposed system differs from the existing system in the sense that, unlike other systems, it does not try to predict the accurate stock value of the company but rather assigns a risk percentage corresponding to each company based on sentiment analysis and machine learning results.

The stock value predictions varied within a range of plus or minus 2 dollars when bi-quadratic regression was used. But

these variations do not affect the risk percentage assignment by much since the risk also depends on the sentiment analysis aspect.

The final product created is a solid end product which is fully functional and ready to be used by the consumers. This product can be used by any common man who has very limited knowledge on the workings of the stock market and still make an informed decision.

VII. FUTURE WORK

Current implementation is restricted in terms of speed and efficiency due to constraints on computation power but to handle concurrent users and massive requests to the server, better hardware or processors can be used. The system can be made such that it can learn over time the habits of the user such as his or her investing schedule, preferences and willingness to take risk and then provide suggestions which are more suitable to his or her needs[18]. The 5 companies can be increased to all the companies registered on the stock exchange to provide more options to the user. The interface can be developed in vernacular languages so that people from all regions can fully utilize the benefits of the app. Localization of the processing can be done by performing machine learning and calculation on the local device itself by using the software *Tensorflow lite*[19]. This will enable the user to enjoy benefits of the app without internet connection as well.

REFERENCES

[1] S. Asur, & B. A. Huberman, "Predicting the future with social media," International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010.

[2] A. W. Lo, & A. C. MacKinlay, "Stock market prices do not follow random walks: Evidence from a simple specification test," Review of financial studies, vol. 1, no. 1, pp. 41-66, 1988.

[3] S. Shen, H. Jiang, & T. Zhang, "Stock market forecasting using machine learning algorithms, 2012.

[4] Efthymios Kouloumpis, Theresa Wilson, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," Proceedings of the fifth International AAAI Conference on Weblogs and Social Media.

[5] Chen, Tao, & Bahsoon, Rami. (2013). "Self-adaptive and sensitivity-aware QoS modeling for the cloud". Paper presented at the Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, San Francisco, CA, USA.

[6] A. Pak and P. Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320–1326.

[7] W. Huang Research paper: Forecasting stock market movement direction with support vector machine.Journal: Computers & Operations Research

[8] Yauheniya Shynkevich, T.M. McGinnity, Sonya Coleman, Ammar Belatreche, Predicting Stock Price Movements Based on Different Categories of News Articles, 2015 IEEE Symposium Series on Computational Intelligence

[9] Anurag Nagar, Michael Hahsler, Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams, IPCSIT vol. XX (2012) IACSIT Press, Singapore

[10] I. E. Fisher, M. R. Garnsey, and M. E. Hughes, "Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research," Intelligent Systems in Accounting, Finance and Management, 2016

[11] D. Albanese, G. Merler, S.and Jurman, and R. Visintainer. MLPy: high-performance Python package for predictive modeling. In NIPS, MLOSS workshop, 2008.

[12] C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. ACM Trans. on Intelligent Systems and Technology, 2(3):27, 2011.

[13] David Zimbra, M. Ghiassi and Sean Lee, "Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks", IEEE 1530-1605, 2016.

[14] Darrow, Barb (Dec 18, 2012). "Firebase secures its real-time back-end service".

[15] J. M. Emison, "Google Compute Engine: Hands-on review," http://www.informationweek.com/cloud-computing/infrastructure/ google-compute-engine-hands-on-review/240002899?pgno=1, 2012.

[16] Firebase. "Firebase/angularfire." GitHub. 09 Mar. 2017. Web. 04 Apr. 2017.

[17] K. Nishida, T. Hoshide, and K. Fujimura, "Improving tweet stream classification by detecting changes in word probability," in Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2012, pp. 971-980.

[18] Matsunaga, A., & Fortes, J. (2010). "On the Use of Machine Learning to Predict the Time and Resources Consumed by Applications". Paper presented at the Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM.

[19] Martin Wicke, Yuan Yu and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[20] Android Design Patterns: Interaction Design Solutions for Developers By: Greg Nudelman.

[21] Google,"Google Compute Engine instances," https://developers.google. com/compute/docs/instances, 2013.