ROYAL SOCIETY
OPEN SCIENCE

rsos.royalsocietypublishing.org

Research

Check for updates

Article submitted to journal

Subject Areas:
Computer Science

Keywords:
cryptocurrency, sentiment analysis, computational social science, machine learning, algorithmic trading

Author for correspondence:
Franco Valencia
e-mail: franco.avalencia@gmail.com
Alfonso Gómez-Espinosa
e-mail: agomeze@itesm.mx

Algorithmic Trading of Cryptocurrencies using Sentiment Analysis and Machine Learning

Franco Valencia and Alfonso Gómez-Espinosa

Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Ave. Epigmenio González 500, Fracc. San Pablo, Santiago de Queretaro, Queretaro 76130, Mexico

Cryptocurrencies are becoming increasingly relevant in the financial world and can be considered as an emerging market. The low barrier of entry and high data availability of the cryptocurrency market makes it an excellent subject of study, from which it is possible to derive insights into the behavior of markets through the application of sentiment analysis and machine learning techniques for the challenging task of stock market prediction. While there have been some previous studies, most of them have focused exclusively on the behavior of *Bitcoin*. In this paper, we propose the utilization of an algorithmic trading strategy based on machine learning and sentiment analysis for the prediction of *Bitcoin*, *Ethereum*, *Ripple* and *Litecoin* cryptocurrency markets. We compare the utilization of Neural Networks (NN), Support Vector Machines (SVM) and Random Forest (RF) while using elements from *Twitter* and market data as input features. The results show that it is possible to predict cryptocurrency markets using machine learning and sentiment analysis, where *Twitter* data by itself can be used to predict certain cryptocurrencies and that NN outperform the other models.

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

Although there are some studies that deal with both the task of predicting stock market price movements, as well as the development of profitable trading strategies based on those predictions, it is important to verify the applicability of such studies in new and emerging markets; in particular the cryptocurrency market.

This market is characterized by high volatility, no closed trading periods, relatively smaller capitalization and high market data availability. These characteristics have attracted a considerable amount of capital, however up to now there have only been a few studies that have attempted to create profitable trading strategies in the cryptocurrency market. [1]

Another point of interest in the cryptocurrency market is the large-scale of available public sentiment data, particularly from social networks. This data can presumably be used to infer future human behavior, and therefore could be used to develop advantageous trading strategies. [2] [3]

Stock market prediction has always been regarded as a highly challenging task that has attracted attention from both academia and investors. The complexity of the task can be attributed to the multiple factors and uncertainties that interact in the markets including economic and political conditions, as well as human behavior. Being able to consistently predict the market price movements is quite difficult, but not impossible. According to academic research, movements in the market prices are not random, but behave in a highly non-linear and dynamic behavior. Previous studies have also shown that it is not necessary to be able to foretell the exact value of the future price in order to make profit in financial predictions. In reality, predicting the market direction as compared to its value can result in higher profits [4]

Over the past decades, artificial intelligence and machine learning techniques have been used to predict the stock market. Neural Networks (NNs), Support Vector Machines (SVMs) and Random Forests (RFs) are by far the most widely used techniques. Most successful models treat stock market prediction not as a regression problem as one could expect, but as a classification problem. Significant progress has been made in the prediction of the price movement direction of the S&P 500 stock index futures on a daily basis. [5] [6] [7]

A more novel approach has been using social signals and sentiment analysis for the prediction of trading volumes and the prices of individual stocks. [8] Sentiment in social networks, particularly from *Twitter*, can be used to predict movements in stock indices [2]. While there is no evidence that predictions based on sentiment produce significant returns on stock trading [9], a study was able to obtain a trading strategy based on social media sentiment for *Bitcoin* cryptocurrency. [1]

In this paper we extend the application of algorithmic trading strategies based on machine learning techniques and sentiment analysis to cryptocurrency markets. While doing this, we compare three prediction models: NNs, SVMs and RFs by applying them to four different cryptocurrencies: *Bitcoin*, *Ethereum*, *Ripple* and *Litecoin*. We use three approaches for input to these models. The first approach trains the model exclusively with *social* data, the second trains the model exclusively with *market* data and the third combines both *social* and *market* data for training. Then we evaluate the performance of each prediction model, and test whether social media sentiment predicts the market price movements for the cryptocurrency in question.

The rest of this paper is arranged as follows: In section 2, we give a general introduction to the data, sentiment analysis and machine learning. In section 3 we present the obtained results with their interpretation. Conclusions and disclaimers are in section 4.

2. Materials and Methods

2.1 Market Data

Historical Market data was obtained from the top performing 65 cryptocurrency exchanges. The data was fetched from *cryptocompare.com* public API, which allows requesting up to 80 days of historical data from any tradable cryptocurrency for free and the complete historical data can be obtained by request. The data obtained can be requested with either an hourly or daily granularity and contains the opening price, highest price, lowest price, closing price and transaction volume for each timestep.

2.2 Social Data

Social data was obtained in the form of raw *tweets* from *Twitter*. *Tweets* were filtered so that only those containing either the name of the cryptocurrency in question (*i.e. bitcoin*) or its ticker symbol (*i.e. btc*) in their text fields or tags were selected.

Because of the lack of historical data from the *Twitter API*, *Tweets* had to be collected on a daily basis. This was done by fetching *tweets* from the *Twitter streaming API* and saving them in a timeseries database.

Averaging 345,000 *tweets* per day, it is expected to collect more than 20,700,000 *tweets* by the end of the collection timespan.

2.3 Sentiment Analysis

Sentiment was measured by applying Valence Sentiment Analysis to the text of the cryptocurrency related *tweets*. Valence quantifies the degree of pleasure or displeasure of an emotional experience. The state-of-the-art lexicon technique VADER (Valence Aware Dictionary and Sentiment Reasoner) was used to perform the measurements. This tool was selected because VADER is specifically attuned to sentiment expressed in social media, and incorporates a "gold-standard" sentiment lexicon that is specifically attuned to *Twitter-like* content.[10]

The result of applying VADER to a *tweet* text is a vector with a normalized value for the scores: positive sentiment, neutral sentiment, negative sentiment and compound sentiment.

Because most previous works on sentiment analysis for financial markets focus only on the dimensions of valence, mood or calmness, the phenomenon of polarization of opinions is often overlooked. For this reason, we calculate a polarization score for each hour of data by applying the geometric mean of the average of the positive sentiment and the negative sentiment of all the *tweets* that are in the timestep with the intention of using the polarization score as a complementary dimension to emotional valence.

2.4 Feature Vectors

A system was setup to gather all collected data from the different data sources, for the creation of a single dataset that includes both *market* and *social* data. Thus, given the market data and social signals, a feature vector V for a certain time period *t* is defined as:

$$V(t) = \begin{bmatrix} neu, \\ norm, \\ pos, \\ pol, \\ close, \\ high, \\ low, \\ open, \\ volumeto \end{bmatrix} \tag{2.1}$$

where,

neu = Average of neutral sentiment
norm = Sum of the valence scores of each word
pos = Average of positive sentiment
pol = Geometric mean of *pos* and *neg*
close = Closing asset price
high = Highest price
low = Lowest price
open = Opening asset price
volumeto = Trading volume for the time period

The target $Z(t)$ is defined as a binary classification with a value of 1 or -1. That represents whether there was an increase or a decrease in price between two time periods. An increment in the closing price between $V(t)$ and $V(t+1)$ would have a $Z(t)$ value of 1. A decrement in the closing price between $V(t)$ and $V(t+1)$ would have a $Z(t)$ value of -1.

The selection of this target is based on the previous knowledge that it is enough to know the direction of the market in order to obtain profit from a prediction[4], as it was previously stated in related research.

2.5 Multi-layer Perceptron

Multi-layer Perceptrons (MLPs) are a type of NN that consists of at least three layers of nodes. MLPs may use backpropagation and supervised learning for training. As such, they belong to the NN class of Back Propagation (BP). An MLP function can be simply stated as $F() = \mathbb{R}^m - > \mathbb{R}^o$ where *m* is the dimension size of the feature vector and *o* is the dimension size of the target.

This algorithm was selected because it is possible for an MLP to learn a non-linear function approximation for either classification or regression. How it differs from logistic regression is that it supports the existence of one or more non-lineal layers. The first layer consists of a set of inputs $x_i | x_1, x_2, \ldots x_m$ that represent the input features and are connected to the first layer of neurons, known as the input layer. Neurons from the hidden layers apply a lineal summation function $w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$ followed by a non-linear activation function to the values of the previous layers. For our model we selected a hyperbolic tangent activation function because of its popularity and good performance. The output layer transforms the values received from the last hidden layer into outputs.

There have been multiple studies that have shown the utility of BP algorithms in stock market prediction problems[11] [12], and how easily BP algorithms can outperform even the best regression models for this task.[13]

For the usage of any type of NN, it is required to design its architecture. This implies the selection of the number of layers for each type as well as the number of nodes in each of these layers. In order to prevent over-fitting in our NN model we applied the following heuristic, derived from several assertions and formulas from [14] to calculate N_h, the upper bound on the number of hidden layers.

$$N_h = \frac{N_S}{(\alpha * (N_i + N_o))} \tag{2.2}$$

N_s represents the number of samples in the training dataset, α is defined as an arbitrary scaling factor which usually ranges from 5 to 10, N_i is the number of input neurons and N_o is the number of output neurons.

2.6 Support Vector Machines

Support Vector Machines (SVMs) are a supervised learning algorithm that construct a hyperplane or set of hyper planes, in a high or infinite dimensional space, by the use of a kernel function. SVMs seek to maximize the distance of the hyperplane from the nearest training examples, by obtaining the training examples that are the closest to the maximum margin hyperplane which are denominated support vectors. SVMs can be used for classification or regression problems.

In the case that the data is not linearly separable, which applies to our model, the SVM transforms the inputs into a high-dimensional feature space by using a kernel function. The decision function is:

$$y = sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \rho) \tag{2.3}$$

where y is the classification label (1 or -1), n is the number of the training vectors, α is a Lagrange multiplier, $K(x_i, x)$ is the Kernel function and ρ is the intercept for the maximum margin decision boundary. As our kernel, we selected a Gaussian radial basis function $K(x; y) = exp(-1/\sigma^2(x-y)^2)$ because of its popularity for SVM classification problems.

Some notable features of SMVs are their effectiveness in high dimensional spaces and that unlike NN's SVMs are resistant to over-fitting. These features have made SVMs a popular choice for financial forecasting and stock market prediction. [15] [16] Some studies have even found that SVMs outperform other classification methods and as such are the best model for forecasting market movement directions. [17] However others have found that BP or SVMs superiority over each other is dependent on the market. [18]

2.7 Random Forests

Random Forests (RFs) are meta estimators that fit a number of decision trees on various sub-samples of the dataset. Just as other models, RFs can be applied for classification, using decision tree classifiers. RFs control predictive accuracy and over-fitting by using averaging for the decision of each tree.

A study comparing NNs, SVMs, RFs and Naive-Bayes performance for stock price index movement in Indian Stock Markets, found that RFs outperformed the other models, when the model was trained with ten technical parameters that were presented as continuous values. [19] In another study [20] the authors claimed that a Random Forest Classifier (RFC) also outperformed other models and algorithms found in the literature.

2.8 Training

The dataset consists of a time series of *market* and *Twitter* data. For training and testing the dataset was divided in a 70-30 split where 70% of the data is reserved for training and 30% is used for testing.

3. Results

The data used for this study was obtained from the sources mentioned in the previous section. We collected 60 days of data from February 16th, 2018 to April 21st, 2018. The market data had one hour granularity, and the twitter data was processed as previously mentioned to fit this granularity.

Our prediction models MLPs, SVRs and RFCs to foretell the daily market movements of *Bitcoin*, *Ethereum*, *Ripple* and *Litecoin*. For each cryptocurrency, we compared the performance of the model when using different subsets of the previously defined feature vector $V(t)$. *Twitter data* is comprised of the $V(t)$ elements *neu, norm, pos* and *pol* while *Market data* of *close, high, low, open* and *volumeto*.

All models were implemented using the *sci-kit learn* library. MLPs were trained multiple times and we took the results from the best performing networks.

To evaluate the robustness of each model we used accuracy, precision, recall and f_1 scores which are defined as follows:

$$Accurracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{3.1}$$

$$Precision = \frac{t_p}{t_p + f_p} \tag{3.2}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{3.3}$$

$$f_1 = 2\frac{precision * recall}{precision + recall} \tag{3.4}$$

where,
t_p = *Number of true positive values*
t_n = *Number of true negative values*
f_p = *Number of false positive values*
f_p = *Number of false negative values*

Accuracy measures the ratio of all testing samples classified correctly, precision is the ratio of relevant classified samples among the total retrieved samples, recall is the ratio of relevant classified samples among the total amount of relevant samples and F_1 score is the harmonic average of the precision and recall. Precision is considered the most important score, as it implies how many times we are correct in our prediction which would determine what type of market order a strategy would create.

Tables 1 to 4 show the scores of each of our models applied to the previously defined cryptocurrencies for predicting the market movement of the next day.

Model	Twitter Data	Market Data	Accuracy	Precision	Recall	F_1 Score
MLP	Yes	No	0.39	0.38	0.39	0.38
MLP	No	Yes	0.72	0.74	0.72	0.71
MLP	Yes	Yes	0.72	0.76	0.72	0.72
SVR	Yes	No	0.50	0.29	0.5	0.37
SVR	No	Yes	0.55	0.53	0.56	0.47
SVR	Yes	Yes	0.55	0.31	0.56	0.40
RFC	Yes	No	0.50	0.29	0.5	0.37
RFC	No	Yes	0.50	0.76	0.5	0.39
RFC	Yes	Yes	0.44	0.28	0.44	0.34

Table 1: Results of applying MLP, SVR and RFC using *Twitter* data, Market data or both for predicting daily market movements for Bitcoin.

As we can see in table 1, MLP was the best performing model for *Bitcoin*. Having an accuracy of over 0.72 and a precision of 0.76, this model is better than random by a large margin. Both SVR and RFC also managed to beat random when using Market data. *Twitter* data by itself could not be used to predict the market movement in any model, and its inclusion appeared to worsen the performance of the SVR and RFC models. However it improved the precision in the MLP model slightly.

Model	Twitter Data	Market Data	Accuracy	Precision	Recall	F_1 Score
MLP	Yes	No	0.39	0.44	0.39	0.38
MLP	No	Yes	0.39	0.44	0.39	0.35
MLP	Yes	Yes	0.44	0.56	0.44	0.39
SVR	Yes	No	0.39	0.15	0.39	0.22
SVR	No	Yes	0.39	0.15	0.39	0.22
SVR	Yes	Yes	0.39	0.15	0.39	0.22
RFC	Yes	No	0.33	0.14	0.33	0.19
RFC	No	Yes	0.39	0.15	0.22	0.22
RFC	Yes	Yes	0.39	0.15	0.39	0.22

Table 2: Results of applying MLP, SVR and RFC using *Twitter* data, Market data or both for predicting daily market movements for Ethereum.

For *Ethereum* the best performing model was MLP as shown in table 2. No model was able to perform significantly better than random. MLP was the only model that was able to obtain a slight edge in precision against random by including both Market and *Twitter* data. Neither *Twitter* data nor market data by themselves were able to predict the *Ethereum* market movements.

Model	Twitter Data	Market Data	Accuracy	Precision	Recall	F_1 Score
MLP	Yes	No	0.50	0.5	0.5	0.5
MLP	No	Yes	0.66	0.68	0.67	0.66
MLP	Yes	Yes	0.55	0.56	0.56	0.55
SVR	Yes	No	0.55	0.60	0.56	0.50
SVR	No	Yes	0.50	0.5	0.5	0.41
SVR	Yes	Yes	0.50	0.25	0.5	0.33
RFC	Yes	No	0.38	0.39	0.39	0.39
RFC	No	Yes	0.44	0.40	0.44	0.38
RFC	Yes	Yes	0.44	0.44	0.44	0.44

Table 3: Results of applying MLP, SVR and RFC using *Twitter* data, Market data or both for predicting daily market movements for Ripple.

In table 3 we can see how Ripple, MLP was again the best performing model obtaining a 0.66 accuracy and a 0.68 precision score beating random by a large margin. SVR also beat random by a small margin when using only *Twitter* data. RFC did not manage to beat random. *Twitter* data was able to beat random by itself when using the SVR model with 0.55 accuracy and 0.6 precision scores.

Model	Twitter Data	Market Data	Accuracy	Precision	Recall	F_1 Score
MLP	Yes	No	0.61	0.61	0.61	0.61
MLP	No	Yes	0.61	0.78	0.61	0.54
MLP	Yes	Yes	0.61	0.62	0.61	0.60
SVR	Yes	No	0.50	0.5	0.5	0.41
SVR	No	Yes	0.50	0.25	0.5	0.33
SVR	Yes	Yes	0.66	0.8	0.67	0.62
RFC	Yes	No	0.55	0.56	0.56	0.55
RFC	No	Yes	0.50	0.5	0.5	0.49
RFC	Yes	Yes	0.61	0.66	0.61	0.58

Table 4: Results of applying MLP, SVR and RFC using *Twitter* data, Market data or both for predicting daily market movements for Litecoin.

Table 4 shows how SVR was the best performing model for Litecoin, obtaining a 0.66 accuracy and a 0.8 precision score. RFC performed slightly better than MLP when using both *Twitter* data and market data. All models were able to beat random. *Twitter data* was able to predict the market by itself when using the MLP and RFC models.

4. Conclusion

In this paper, we proved that it is possible to create algorithmic trading strategies for the emerging cryptocurrency market using techniques that had been previously utilized for *Bitcoin*. These strategies use both machine learning and sentiment analysis to predict price market movements. We evaluated and compared the performance of three prediction models: MLPs, SVMs and RFCs for *Bitcoin, Ethereum, Ripple* and *Litecoin* using *Twitter* data, market data or both.

Our results show that for the *Bitcoin, Ethereum, Ripple* and *Litecoin* markets there is at least one model that can predict market movements by beating random chance. *Bitcoin's* best model was a MLP which using both *Twitter* and market data, obtained scores of 0.72 accuracy and 0.74 precision. *Ethereum's* best model was also a MLP that used both *Twitter* and market data to obtain 0.44 accuracy and 0.56 precision scores. In *Ripple*, once again, the best model was an MLP that only used market data, obtaining 0.66 accuracy and 0.67 precision scores. *Litecoin* was the only cryptocurrency where the SVR model performed the best, using both *Twitter* and market data it obtained 0.66 accuracy and 0.8 precision scores.

With the highest precision score, *Litecoin* was the most predictable market, followed by *Bitcoin* and *Ripple*. Only the *Ethereum* market had an accuracy score of under 0.50. MLP was the most successful model, managing to successfully predict market movement prices in all cryptocurrencies while outperforming the other models in 3 out of 4 cases. SVR was successful in predicting the markets for *Bitcoin, Ripple* and *Litecoin* while failing to predict *Ethereum's*. RFC was able to predict the *Bitcoin* and the *Litecoin* markets.

These results also make it possible to observe how the usage of exclusively *Twitter* data can be used by itself to predict the *Ripple* and the *Litecoin* markets, but it is not superior to the utilization of exclusively market data. The use of both *Twitter* data and Market data may bring slight improvements in scores, however in other cases it may also cause a reduction in the model performance. When using SVR models, it is theorized that this reduction in performance could be caused by the utilization of a single kernel function for different types of data. It is unknown why this problem occurs with NN and RF models and such question exceeds the scope of this study.

Further work should be done on improving the quality and the quantity of *social signals*. Quality could be improved by eliminating *Tweet* duplicates and content posted by *bot* accounts. Obtaining data from other social networks such as *Reddit* and *Facebook* could improve the quantity of the data and add a larger sample of opinions. Another area of interest might be the usage of separate models for *Twitter* and market data in order to improve models accuracy and precision scores. Finally it should be proven whether transforming these predictive models into profitable trading strategies is possible.

Ethics. This research is based on observational data shared publicly. No personal or individual information was retrieved, stored or analysed.

Data Accessibility. Data used for this article is publicly available either though the corresponding Application Programming Interface or it is available alongside the required scripts to reproduce these results at https://github.com/vanclief/algo-trading-crypto

Authors' Contributions. The research was conceived by both authors, F.V.A retrieved the data, applied the methodology and wrote the manuscript.

Competing Interests. We declare we have no competing interests.

Funding. There was no funding for this research.

Acknowledgements. We thank Benjamín Valdés and Ruben Stranders who provided insight and expertise that greatly assisted the research.

References

1. David Garcia, Frank Schweitzer. 2015. *Social signals and algorithmic trading of Bitcoin*. R. Soc. open sci. 2: 150288. (doi:10.1098/rsos.150288)
2. Johan Bollen, Huina Maoa, Xiaojun Zeng. 2010. *Twitter mood predicts the stock market*. ELSEVIER Journal of Computational Science. (doi:10.1016/j.jocs.2010.12.007)
3. Qing Li, TieJun Wang, Ping Li, Qixu Gong, Yuanzhu Chen. 2014 *The effects of news and public mood on stock movements*. ELSEVIER. (doi:10.1016/j.ins.2014.03.096)
4. Chen, A.S., Leung, M.T., and Daouk, H. 2003. *Application of Neural Networks to an Emerging Financial Market: Forecasting and Trading the Taiwan Stock Index.*. Computers and Operations Research.
5. E. Saad, D. Prokhorov, and D. Wunsch. 1996. *Advanced neural-network training methods for low false alarm stock trend prediction*. IEEE Int. Conf. Neural Networks, Washington, D.C..
6. Tsaih, R., Hsu, Y. and Lai, C.C. 1998. *Forecasting S&P 500 stock index futures with a hybrid AI system.* Decision Support Systems. 23 2, 1998, pp. 161–174.
7. Kohara, K., Ishikawa, T., Fukuhara, Y. and Nakamura, Y. 1997. *Stock price prediction using prior knowledge and neural networks.* International Journal of Intelligent Systems in Accounting, Finance and Management 6 1, pp. 11–22.
8. Bordino I, Battiston S, Caldarelli G, Cristelli M, Ukkonen A, Weber I. 2012. *Web search queries can predict stock market volumes*. PLoS ONE 7, e40014. (doi:10.1371/journal.pone.0040014)
9. Schoen H, Gayo-Avello D, Metaxas PT, Mustafaraj E, Strohmaier M, Gloor P. 2013. *The power of prediction with social media* Internet Res. 23, 528-543. (doi:10.1108/IntR-06-2013-0115)
10. C.J. Hutto and Eric Gilbert. 2014. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.* Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI.
11. Baestaens, D.E. and van den Bergh, W.M. (1995) *Tracking the Amsterdam stock index using neural networks*. Neural Networks in the Capital Markets, pp.149–161.
12. Tsibouris, G. and Zeidenberg, M. (1995) *Testing the efficient markets hypothesis with gradient descent algorithms*. Neural Networks in the Capital Markets, pp.127–136.
13. Refenes, A-P., Zapranis, A.D. and Francis, G. (1995) *Modeling stock returns in the framework of APT: a comparative study with regression models*. Neural Networks in the Capital Markets, pp.101–125.
14. Martin T. Haganm, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesús. 2014. *Neural Network Design*. Martin Hagan. ISBN-13: 978-0971732117
15. L.J. Cao and F.E.H. Tay. 2001. *Financial forecasting using support vector machines* Neural Computing Applications 10, pp. 184-192.
16. F.E.H. Tay and L.J. Cao. 2001. *Application of support vector machines in financial time series forecasting*. Omega 29, pp. 309–317.
17. Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang. *Forecasting stock market movement direction with support vector machine*. ELSEVIER Computers & Operations Research. (doi:10.1016/j.cor.2004.03.016)
18. Wun-Hua Chen and Jen-Ying Shih. 2006. *Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets*. Int. J. Electronic Finance, Vol. 1, No. 1.
19. Jigar Patel, Sahil Shah, Priyank Thakkar, K Kotecha. 2014. *Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques* ELSEVIER Expert Systems with Applications Volume 42, Issue 1. (doi:10.1016/j.eswa.2014.07.040)
20. Luckyson Khaidem, Snehanshu Saha, Sudeepa Roy Dey. 2016. *Predicting the direction of stock market prices using random forest* To appear in Applied Mathematical Finance.