Predicting Stock Market Indexes with World News

Yunsong Zhong
School of Software Engineering
Tongji University, Shanghai, China
Email:15yszhong@tongji.edu.cn

Qinpei Zhao
School of Software Engineering
Tongji University, Shanghai, China
Email:qinpeizhao@tongji.edu.cn

Weixiong Rao
School of Software Engineering
Tongji University, Shanghai, China
Email:wxrao@tongji.edu.cn

***Abstract*—With the rapid development of economy, people need to raise the accuracy of predicting prices considering late events. The main tackle in raising the accuracy is to fully use the information in daily reports. Unfortunately, most of the current solutions separate the time series statistics, the events reported in text data and prediction in data mining way from each other. As a result, although they predict the prices accurately in most of the days, they can not grasp the sudden change points of those time series.**

In this paper, we propose an ensemble framework to take advantage of the news text in predicting change points in stock market indexes as well as traditional prediction works, so that we can improve our prediction sufficiently. Our extensive experimental results shows that we reduce the loss of error predictions and enhance the good prediction results.

I. Introduction

In the era of big data, people tend to put various types of data together to enhance their learning, which provides a new approach in many areas. Predicting trend of stock market behavior is one of those tasks. For instance, [2] proposes a method to predict economic indicators by social media. This work introduces an event trigger in Latent Dirichlet Allocation model to predict food price by daily news. It is a good attempt, but analyzing stock market is more complicated because of the complexity of stock transactions.

The most important challenge of predicting economic by media is the weak dependency between media information and economic indicators. Reference [5] splits the values to three classes (upward, downward, expected) and classify them by news. Obviously, it can not fully use the news. There are many other similar recent works like [11] and [16], which regard the news data as a classification problem, however, traders in stock market always focus on the price itself instead of whether it rises or falls. What's more, a higher classification results can not give a instruction, an opposite but near prediction is more valuable to traders than a trend-accurate prediction which is far from the truth.

Unfortunately, existing methods separate the predicting task into two parts: time series prediction and text analysis. Few of them consider these together. When predicting time series, researchers begin from statistical methods such as ARIMA[1], VAR[12], etc. These methods focus only on the data itself, support their results by independence test. Further studies using SVR[21] and neural network[6] reduce the prediction error, but neither are they able to use exogenous data. On the other hand, when analyzing text data, plenty of works focus on the sentiment hidden in the text. For example, [14] builds stock networks and studies the labeled words to modify the prediction of VAR. Reference [18] calculates emotion scores from the news to find the causality between this kind of emotion and stock market. These works did well in finding relationships between human sentiment and market price, however, sentiment is not the kind of factors which can influence the trend at any time, in anywhere. It only occurs at special time.

In this paper, we propose a novel method to predict indicators in stock market using the news which contains the important events occurred worldwide. Our method consists of two parts: modeling the news and predicting the time series. We model the news by a temporal topic model proposed by [9]. This approach considers that news text is short and time-related, so we update the parameters according to the distribution of the last times. Then we combine the historical records and features extracted from topic model together to describe the situation of the days to be predicted. In this paper, compared to traditional predicting methods, our work has following contributions.

1) We conserve the advantage of statistical prediction by including the historical record of the time series as well as events contained in news text. Each kind of data has its own advantage. Historical record of the time series can ensure that the prediction is not far away from the previous days, therefore the average of predictions is near the truth. On the other hand, text data is sensitive to the events, so the sudden changes of data can be captured. Considering both of them, the prediction will be more accurate.
2) We predict the time series by history and topic model separately, and merge them by training the parameter of an autoregressive model. Our work use regression method instead of matrix factorization methods to predict the parameters in the autoregressive model, because the stock market data in not definite, similar historical records do not mean similar price, it depends on times, events and even random factors. However, if we use regression of random forest prediction, we can always capture the most important factors in the model.
3) Our experiment shows that regardless of most of the parameters, our method is apparently more accurate than predictions only by historical records or only by topic model. Our prediction reduces the shift of the single

978-1-5386-1107-4/17/$31.00 ©2017 IEEE

predictions, and make the prediction more smooth.

The rest of the paper is organized as follows. First we introduce the basic preliminaries of the methods we used in our work in Section II. Next, we give the details of our approach in Section III. After that, we evaluate in Section IV, and review related works in Section V. Finally in Section VI we conclude the paper.

II. Model Ensembles

We will first give the definitions of our problem in Section II-A. Then we will introduce our features for historical stock market records and topics for news text in Section II-B and Section II-C. Finally, we use ARX model to combine different prediction results in Section II-D together.

A. Preliminaries

In this work, we consider daily stock market indicators as a discrete time series $X = \{x_1, x_2, ..., x_N\}$ with the total number of times N. Our main task is to predict the value of x_{N+1} with the given X above.

After this, we begin to consider the events. We extract the events from daily world news. At the beginning, we transfer the raw news to a document-term matrix D. Each row of D represents for a day, and each column of D is the frequency of words occurred. Then we transfer D using Latent Dirichlet Allocation (LDA). LDA transfer D to two matrixes: document-topic matrix T and topic-term matrix. We use only document-topic matrix T for the following computation.

B. Features in Regression Model

We use data mining based method as the foundation of our prediction. These methods takes large amount of features of samples as input and find the relation between features and their ground truth labels. Most of these methods assume that samples whose labels are close to each other are more common in some features than else. For example, if a piece of news reporting the Iraq War and another piece reporting the Vietnam War both talk about the number of died soldiers, and they are both sorted into military topic, we can see that the number of died soldiers is useful in predicting war.

In our work, we use random forest model to make predictions. Random forest is a classification model based on decision trees. Each nodes on the tree represents a value or range of values in one feature, and values in the same feature are in one layer. The nearer the nodes are to the root, the more important that feature is in the classification problem.

Random forest is a set of such decision trees with sampling data and voting label. Firstly, when the number of features grows, it is not scalable if we give all those features to each single tree. Therefore, we split those features and the samples to a large number of decision trees. Secondly, we give the final results by summarizing the result given by each tree. The most efficient way is to vote, i.e. fetch the majority result of those trees.

In our work, we extract several features from the time series of stock market indices as well as text in documents. A brief introduction of those features is as below.

I. **Original Data** We list indexes from several days before as the features in prediction, which gives the criterion for the final result. For example, there are both Olympic Sports in 2008 and 2016, other features maybe close. However, in 2008 we have global financial crisis, so the market index in 2008 are low in average, and the original data of several days before are useful.

II. **First-Order difference** First-order difference is the difference of two values near by, like $x_2 - x_1, x_3 - x_2$ and so on. These kind of features can eliminate the constant part of values and focus on the small changes.

III. **Meaningful Combination** We also extract some other meaningful features, like difference between close price and open price in the same day. These features help the model to predict better when same extraordinary situations happen again.

IV. **Time-Related Value** We set several time stamps on the feature to identify the periodic part of the values. This kind of values are meaningful in many specific articles in economy, e.g. [13] introduces the weekend effect and January effect.

Practically, we put all of those features into the model and give them the same weight.

C. Topic Model

Latent Dirichlet Allocation (LDA) is a widely-used method which converts bunch of texts to their topic distribution according to their words, considering the special concerns of text data. For example, "the Golden State Warriors wins the champion" and "Durant leaves the Oklahoma City Thunder" do not have much word in common, but we know that they are related because there are many reports which these two expressions appear together. Practically, traditional LDA model takes document-term matrix as the input. Given the number of topics and the prior parameters of the distributions(α and β), this method transforms the text to topic distribution in a document θ and word distribution in topics$W$. The meaning of topic Z is not explicit, the method will assign words to topics automatically.

In our work, we use the dynamic LDA model introduced in [9] to change short news reports to time-varying features. They modify this model so that the parameter can change with time flows, as shown in Figure 1. Traditional LDA model doesn't consider time, i.e. it does not relate θ_{t-1} to θ_t. The dynamic LDA model multiplies α_t and θ_{t-1} to get θ_t, considering the changes of topics over time.

This modification is important since the topic itself cannot help us in prediction, and even if it can, that is not reasonable. However, the slight changes in topic is useful. For example, if a natural disaster appears, many report will follow. Since the disaster destroys some factories, the market will be influenced.

Essentially, this method gives a way to represent the words in a document by sampling documents and topics. Like traditional LDA models, we infer the hidden topics by the frequency of words. For example, if word "United" and "State" always come up together in the documents, they will always

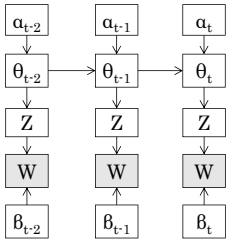


Fig. 1. Dynamic LDA model

have the same frequency. Thus, they will be distributed to the same topic. Although this LDA model does not consider the distance of words, which will lead to some mistakes (e.g. news reports often begins with the name of reporter and end with publisher, which does not mean the publisher is responsible for the accident), it is still a good way to conserve the context information when reducing the dimension from the number of words to the number of topics.

D. Autoregressive Combination

We combine our basic predictions in the thought of autoregressive model. Reference [20] offers a good idea on merging same kinds of time series together. Their work focuses on repairing the series using exogenous data, and our work focus on merging different kinds of predictions. The basic idea of this combination is to sum up the differences in training time stamps with proper linear weights so that in the real-time prediction we can reduce the offsets of any single predictions. ARX model (autoregressive model with exogenous inputs) fixes them together with the focus on the changes of difference between the prediction and the ground truth.

The original formula of the ARX model is Equation 1.

$$y_t = x_t + \sum_{i=1}^{p} \psi_i (y_{t-i} - x_{t-i}) + \epsilon_t \tag{1}$$

where y_t is the final prediction, p is the order, x_t is the observation value, ψ are parameters and ϵ is a white noise. Practically, we let ϵ obey a Gaussian distribution with the mean of 0 and a very little variance.

As Equation 1 shows, the ARX model refines the observation values x by the exogenous values y. For instance, if a thermometer reads like x at time t, and we know the exact temperature before time t is like y, then we can learn the parameters and get true temperature y_t from x_t and y. In the situation that the series to be predicted is related to other series, e.g. in our work, stock market indexes is related to topic changes, this model can be used to enhance the effect of exogenous variable (in our example the effect of topics).

We come up with the idea to change the usage of the ARX model step by step. At the beginning, we just merge the two predictions in a linear calculation in Equation 2.

$$y_t = \psi_t x_t^0 + (1 - \psi_t) x_t^1 \tag{2}$$

where x_t^0 and x_t^1 are two prediction values, ψ_t is parameter and y_t is the final prediction.

As long as we train ψ accurately enough, the prediction will be enough accurate despite of the errors in a single prediction. However, we find it difficult because we can not learn ψ well with only two values. Then we refine the formula as Equation 3, adding a sliding window along the time.

$$y_t = \psi \sum_{\omega} x_t^0 + (1 - \psi) \sum_{\omega} x_t^1 \tag{3}$$

where $\psi = \{\psi_\omega, \psi_{\omega+1}, ..., \psi_t\}$, ω is the size of sliding window.

And the result proves that this is not a bad idea. Thus, we turn to a new ARX model while conserving training ψ with random forest model instead of matrix factorization way introduced in [20].

III. Algorithm Design

In this section, we first describe our whole routine in Section III-A. It is followed by the details of three parts: initialization in Section III-B, regression model settings in III-C, and parameters in Section III-D.

A. Algorithm Description

Algorithm 1: Algorithm

Input: time series X, document-term matrix $D = D_0, D_1, ..., D_N$
Output: value x_{N+1}

initialize p;
$T_0 = dLDA(D_0)$;
for *t from 1 to N* **do**
  $X^0 = rf(X^t_{t-\omega})$; // Random Forest Regression
  $T_t = dLDA(D_t, T_{t-1})$; // Changing parameters in LDA model
  $X^1 = rf(T_t)$;
  while *not end of* T_j **do**
    if $diff_t^0 \leqslant diff_t^1$ **then**
      replace x_t^1 with x_t^0;
  $\psi_t = arxtrain(x_t^0, x_t^1)$;
$x_{N+1} = arx(x_N^0, \psi)$; // ARX model, Equation 3
return x_{N+1};

Algorithm 1 gives the pseudo-code of our approach. Our work begin with initializing the order p and applying dynamic LDA model $dLDA()$ to transform document-term matrix D to document-topic distribution T. The probabilities in T is used as features in following random forest predictions. For each sliding window from $t-p$ to t, we make random forest regressions(the function $rf()$) according to series from time $t-p$ to t as X^t_{t-p} and matrix T separately, and update the parameters in the LDA model. Then, we estimate the parameter ψ_t in function $arxtrain()$. We use the prediction from series x_N^0 as the observation in ARX model, and the prediction from topic x_N^1 as the refinement. Finally, we use the trained parameters from the second step to calculate the value of x_{N+1}. The entire flowchart is shown in Figure 2.

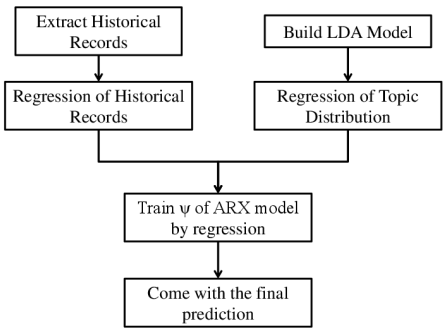


Fig. 2. Flowchart of Algorithm

B. Beginning Parameters

We need to set some parameters at the beginning of the algorithm. They are listed as below.

I. **Order** Order, which is p in Equation 1, is an important parameter in time series analysis. In our work, we analyzed the autocorrelation function of our data. The result shows that order 1 is enough for this series.

II. **LDA Priors** In LDA model, there are two prior parameters. At the beginning, we set them as the total number of the topics N_{topic} and the size of dictionaryN_{word}, which is $\frac{1}{N_{topic}}$ for α and $\frac{1}{N_{word}}$ for β(see Figure 1).

C. Regression Settings

Since random forest is a classification algorithm, we need to make some refinement to predict the exact prices in our work. Therefore, we apply a simple regression model on the result of classification. For those values, we divide the values into several intervals, and use the random forest classification model to predict the interval. When the interval is predicted, we choose the point which leads to the least mean squared error as the prediction result.

D. Parameters Estimation

We need to update parameters of LDA and ARX models when the time-stamp changes.

Firstly, we need to set the size of windows. The parameter ω in Algorithm 1 is the window size of random forest, which decides the time range we consider in one prediction. In dynamic LDA model, we also need to set the maximum days to consider. We set them as the same because of consistency and test the effect of window size in experiments.

Second, since we use another random forest regression model to estimate the ψs, we don't need to test different settings of this parameter. However, we find that since most of the time the prediction X^0 in Algorithm 1 is better than X^1, we set a tolerance to change some of the ψs. We will discuss that in Section IV-C.

Finally, we need to test the effect of the topic number. The umber of topics is an important part in LDA-like models because different divisions can make different meanings. Generally speaking, more topics means more calculation and more information in the text data.

IV. Evaluation

Based on the data in Section IV-A and experiment settings in Section IV-B, we conduct many experiments to show the performance of our work in Section IV-C and IV-D.

A. Data Description

We use two real stock market indexes: Dow Jones Industrial Average index and NASDAQ index for our prediction. For Dow Jones index, we use the index from August 8, 2008 to July 1, 2016. For NASDAQ index, we use the record from October 10, 2005 to October 12, 2015. Both of them have open price and close price in each day.

For the news text, we use the top 25 news from reddit.com world news area. The dataset only contains the title and description of these news, the discussion of readers are not included. We correspond the news to the close price of the same day, since the events happened on one day cannot have effect on the open price, because the events probably happened after the stock market is open.

Except for the algorithms mentioned in previous pages("**rf**" for Random Forest Regression, "**topic**" for dynamic LDA model, "**arx**" for our work), we also tested some other methods: **combine** simply joins the historical records feature and LDA distribution feature, **label-rf** and **label-topic** add date label like order of the week, order of the month, etc. for basic methods. The bolded letters are used in following figures.

B. Experiment Settings

We define two kinds of indexes to describe the performances of different works. We begin with evaluating the absolute errors to depict that our work can reduce the loss of miss-judging the trend. Then we use the accuracy to show how well our work can predict the trend.

We show the absolute distances between different methods to show the loss of our prediction. The meaning of these indexes is that even if we cannot predict the interval of the market indexes, we can still reduce the loss of investigators by approaching to truth nearer. For instance, if the market rises, less fall-amplitude prediction can let investigators sell less stocks, so that they will lose less money.

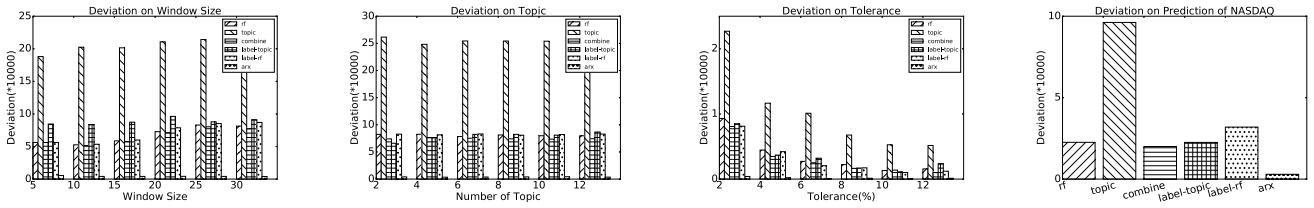
We use sum of an absolute deviation D of X^p and X^t as an evaluation index by Equation 4.

$$D = \sum_{i=1}^{N} |x_i^p - x_i^t| \tag{4}$$

where X^p stands for the prediction, and X^t stands for the ground truth.

D depicts the sum of differences between the prediction and the ground truth, while classification accuracy depicts how sensitive the algorithm is in predicting the situation. In the following sections we can see that our work reduce them explicitly. Then we introduce the concept of accuracy A in the evaluation of classification methods in Equation 5.

$$A = \frac{N_{good}}{N_{all}} \tag{5}$$



(a) (b) (c) (d)

Fig. 3. D on the parameters: (a) Window Size, (b) Number of Topics, (c) Tolerance, (d) NASDAQ index

TABLE I
PARAMETERS TO BE TESTED

	Window Size ω	Tolerance	Number of Topic
Minimum	2	0	2
Maximum	200	0.2	50
Default	200	0	10
Step	5	0.02	5

where N_{good} is the number of predictions which fall in the right interval, and N_{all} is the total number of predictions. We come up with this idea because people concern more about the rules like "when the index is less than 2000, I will sell half of my stocks". To realize this prediction, we separate the stock indexes into several intervals, and replace all of our regression models with traditional random forest classification model.

C. Experiments on Parameters

We conduct several experiments on several parameters, like the size of sliding window ω, the number of topics and the tolerance. We list the maximum numbers and the minimum numbers of these parameters in Table I.

Figure 3(a) shows the different results on window size. When the size of window grows, the error is also growing. Sliding window can ensure that every day which we have both historical values and news text of previous days can be predicted. For the LDA-based models, the error increases when window size is larger. This is related to the characteristics of news, which is strongly related to time, the same news in different weeks simply means different things. For example, the report of the first SARS case in 2003 brought a great panic, but after that period of time passed, a report of SARS case is not so surprising. Therefore, for news text, the prediction of a few nearest samples are closer than the truth.

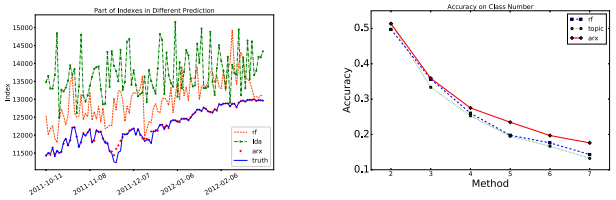
Figure 3(b) shows the experiment on the number of topics. The result shows that the number of topics affects the prediction little. It is probably because our news text dataset is not selected for a certain company, neither is the time series we want to predict. Therefore, the relationship between a certain event and a slight change is not so tight. For example, an announcement from the chairman of a company can certainly cause a great shock on the stock of this company, while transactions in a large amount of factories can only have a small effect on the average index of industry.

Figure 3(c) shows the deviation of different algorithms on the tolerance. Tolerance is a parameter which we use to control the number of samples when we learn ψ in ARX model. We set some of ψs more than 0 and put it into training part again, so that we can get a better result. For example, if we set the tolerance to 0.02, which mean that ψs in this percent of the whole window size are set to a small value, making sure that the factor of topic model is more than 0. We are glad to see that this effort is useful, with the value we set getting larger, the total error becomes smaller.

From the trend shown in those experiments, we conclude the default settings in Table I and conduct predictions in the following subsection.

D. Comparison of Predictions

We show part of our predictions of Dow Jones Indexes in Figure 4(a) to see the effect at the first glimpse, and the classification experiment on number of classes in 4(b). Both of them are tested in the default settings in Table I.



(a) (b)

Fig. 4. Comparison of Different Methods: (a) Time Series, (b) Number of Class

We noticed in Figure 4(a) that traditional random forest regression and LDA based methods have a shift in prediction, while our work eliminate this shift. Meantime, our prediction is much nearer to the truth than the others in most of the time. Even if we regularize them to ignore the shift, the trend of our prediction is still much better than any of those single predictions.

We also tested the classification results on the class number in Figure 4(b). We put all the true values in order and evenly cut them to pieces. For example, if class number is 2, we put all the value under the median into Class 1 and set others as Class 2. Then we replace all the regression part as random forest classification model. We can see that our work performs better than those single predictions in any number of class, and our work performs better when the number of class increases.

We also tested the effect on NASDAQ index, and the result is shown in Figure 3(d). The result shows that our method appropriate for many kinds of stock market indexes.

To make a summery, our work performs better than any single predictions and their simple combination concluded in our work. Our work is a constructive exploration in taking use of heterogenous data to predict time series, and we succeed in reducing the absolute deviations as well as increasing the accuracy of predicting the intervals.

V. Related Work

We come to the dynamic LDA model in [9] because this model is designed for short documents, and our news data does not contain much words. Besides, there are many other temporal topic models, and they are designed for some special purposes. Reference [15] designs Temporal Ailment Topic Aspect algorithm to detect change points and ailment. Their work is applied for health monitoring, in which detecting ailment is the most important problem. Reference [19] also proposes a temporal LDA model, and they focus on the conversation communications contained in the text.

There are many other works on predicting financial time series, such as prices of futures. Reference [3] studies the volatility of stock market and finds the investors related to the volatility. Reference [10] applies linear regression model to predict occupation market for occupation hazard prevention service. There are also some wavelet-based methods for stock markets[4] and other time series[17]. Most of these works are designed to give suggestions for services or throw alert when a certain pattern occurs.

Many works use random forest to solve other problems. Reference [8] uses random forest regression for magnetic resonance image synthesis, and [7] investigates land surface temperatures. Most of these works uses more than one random forest models to strengthen their result and reduce the negative effect of randomization.

VI. Conclusion

In this paper, we present a novel fusion of regression models to predict stock market indexes with news text. It is a hybrid regression model of predicting stock market indexes with its own historical data and events happened at the same time. We use a dynamic LDA model to turn news text into the change of events, considering the evolution of events. Then we build regression models two times, first time to generate the weight for ARX model and the second time to predict the time series. Our experimental shows that our hybrid regression method is better than any single one of the models we have integrated.

Our work considers the further relationships between the prices of market and the human activities. For example, a president election may influence the policy, thus influence the public attitude towards goods. We have explored a method to detect the relationships and take use of the relationships.

Acknowledgment

This work is partially supported by National Natural Science Foundation of China (Grant No. 61572365, 61503286), Science and Technology Commission of Shanghai Municipality (Grant No. 15ZR1443000, 15YF1412600).

References

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] S. Chakraborty, A. Venkataraman, S. Jagabathula, and L. Subramanian. Predicting socio-economic indicators using news events. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1455–1464. ACM, 2016.

[3] T. Dimpfl and S. Jank. Can internet search queries help to predict stock market volatility? 2016.

[4] Y. Fang, K. Fataliyev, L. Wang, X. Fu, and Y. Wang. Improving the genetic-algorithm-optimized wavelet neural network for stock market prediction. In *International Joint Conference on Neural Networks*, pages 3038–3042, 2014.

[5] G. Gidofalvi and C. Elkan. Using news articles to predict stock price movements. *Department of Computer Science and Engineering, University of California, San Diego*, 2001.

[6] S. Gupta and L. Wang. Stock forecasting with feedforward neural networks and gradual data sub-sampling. *Journal of Neurology Neurosurgery & Psychiatry*, 61(1):52–6, 2010.

[7] C. Hutengs and M. Vohland. Downscaling land surface temperatures at regional scales with random forest regression. *Remote Sensing of Environment*, 178:127–141, 2016.

[8] A. Jog, A. Carass, S. Roy, D. L. Pham, and J. L. Prince. Random forest regression for magnetic resonance image synthesis. *Medical image analysis*, 35:475–488, 2017.

[9] S. Liang, E. Yilmaz, and E. Kanoulas. Dynamic clustering of streaming short documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 995–1004. ACM, 2016.

[10] A. G. Mouhaffel, C. M. Domínguez, B. Arcones, F. M. Redonda, and R. D. Martín. Using multiple regression analysis lineal to predict occupation market work in occupational hazard prevention services. *International Journal of Applied Engineering Research*, 12(3):283–288, 2017.

[11] H. R. Patel and S. Parikh. Comparative analytical study for news text classification techniques applied for stock market price extrapolation. In *International Conference on Smart Trends for Information Technology and Computer Communications*, pages 239–243. Springer, 2016.

[12] A. M. Rather, V. Sastry, and A. Agarwal. Stock market prediction and portfolio selection models: a survey. *OPSEARCH*, pages 1–22, 2017.

[13] M. Schulmerich, Y.-M. Leporcher, and C.-H. Eu. *Stock Market Anomalies*, pages 175–244. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[14] J. Si, A. Mukherjee, B. Liu, S. J. Pan, Q. Li, and H. Li. Exploiting social relations and sentiment for stock prediction. In *EMNLP*, volume 14, pages 1139–1145, 2014.

[15] S. Sidana, S. Mishra, S. Amer-Yahia, M. Clausel, and M.-R. Amini. Health monitoring on social media over time. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 849–852. ACM, 2016.

[16] V. P. Upadhyay, S. Panwar, R. Merugu, and R. Panchariya. Forecasting stock market movements using various kernel functions in support vector machine. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, page 107. ACM, 2016.

[17] L. Wang, K. K. Teo, and Z. Lin. Predicting time series with wavelet packet neural networks. In *International Joint Conference on Neural Networks, 2001. Proceedings. IJCNN*, pages 1593–1597 vol.3, 2001.

[18] C. Wong and I.-Y. Ko. Predictive power of public emotions as extracted from daily news articles on the movements of stock market indices. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 705–708. IEEE, 2016.

[19] J.-F. Yeh, Y.-S. Tan, and C.-H. Lee. Topic detection and tracking for conversational content by using conceptual dynamic latent dirichlet allocation. *Neurocomputing*, 216:310–318, 2016.

[20] A. Zhang, S. Song, J. Wang, and P. S. Yu. Time series data cleaning: from anomaly detection to anomaly repairing. *Proceedings of the VLDB Endowment*, 10(10):1046–1057, 2017.

[21] M. Zhu and L. Wang. Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms. In *International Joint Conference on Neural Networks*, pages 1–5, 2010.