Use of Long Short-Term Memory Algorithms and Twitter Sentiment Analysis for Stock Index Prediction

Haoyuan Dong Weiran Lin Jiebo Luo

*Goergen Institute For Data Science Goergen Institute For Data Science Computer Science*

*University of Rochester University of Rochester University of Rochester*

Rochester, NY, 14627 Rochester, NY, 14627 Rochester, NY, 14627

[hdong5@ur.rochester.edu](mailto:hdong5@ur.rochester.edu) wlin25@ur.rochester.edu jluo@cs.rochester.edu

**Abstract:** In this report, we propose a framework to predict the stock index price based on time series data and sentiment analysis from Mr. President Donald Trump’s twitter. For Trump’s tweets analysis, we use LDA model to derive 20 topics from his tweets and use VADER model to get sentiment analysis scores of each tweet. For time series data, we use LSTMs models to train data and compare results from different models. Experiment results show that the revised LSTMs models can predict stock index prices form different countries, moreover, we verify the social media sentiment especially Mr. President Trump’s tweets make influence on stock index price by capturing the volatility from our models.

**Keywords:** LDA, VADER, Donald Trump’s Twitter, LSTM, stock index price prediction, sentiment analysis

1. **Introduction**

Stock market index can be decided by very compound factors plus random errors. One of the potential factors is information contained in Twitter platform. For stock market is somewhat sentiment driven, either the panic that precedes a crash and bursting of a bubble, or the excitement in public sphere regarding a new announcement and technology shapes the way the stock price evolves, and social media is often the first to display these trends.

In fact, there are many companies are actively using Twitter messages to trade in algorithmic trading industry[1]. The most prominent twitter is President Donald Trump’s twitter. As we know, he likes to use twitter to express his ideas and emotion. As a president, his tweets no longer represent himself, but become the symbolic authority of many policies from the White House. His tweets reflect his attitude in advance of official media and contain lots of implicit information that could have an impact on stock market index.

In this project, we want to understand how Trump’s tweets influence stock market index. And using LSTM model, we would like to make prediction about stock index based on time series data and sentiment analysis from Mr. President Donald Trump’s twitter.

1. **Related Work**
   1. Volfefe Index

The "Volfefe Index" created by JPMorgan bank analyzed more than 10,000 of Trump's tweets since he took office, with the aim of measuring their impact on US rates market volatility. They found strong evidence that tweets have increasingly moved US rates markets immediately after publication[2].

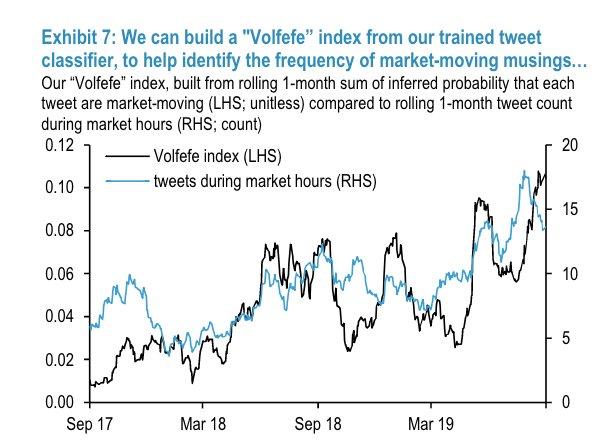


Fig. 1: JPMorgan’s Volfefe Index

From Fig.1 we can see that this index is a statistical aggregation of volume from Trump’s tweets, with which to monitor and quantify shifts in the rates market environment. What we do in this project is to analyze the topics and sentiment of Trump’s tweets to see its influence in the stock market.

* 1. Time Series Analysis

There have been many attempts to predict the stock market through conventional time-series analysis methods. Tang[3] studied the application of the ARMA-GARCH model extended by the AR-GARCH model in stock price prediction. In [4], the author constructed an autoregressive dynamic Bayesian network (AR-DBN) based on dynamic Bayesian network (DBN)[5] and inferred the market index, which improves the predictability of stock market volatility. The authors put the conventional ARMA model with SVMs, which was combined to give full play to the advantages of both to conduct stock market prediction, providing a model with better explanatory power. However, most of the conventional time-series analysis studies relied on the linear relationship between stock prices and were more suitable for sequences with stable trends and laws, which made them inadequate to handle more complex nonlinear relationships. Moreover, the stock market has many inﬂuential factors and the impact is complex, which is ignored in simple time-series analysis methods and makes the prediction less effective.

* 1. Long Short-Term Memory Model

LSTM is a long short-term memory network, a kind of time loop neural network, which is suitable for processing and predicting important events with relatively long interval and delay in time series. In this section, we mainly discuss works on the development of LSTM and its application in time-series prediction.

In the ﬁeld of deep learning, the conventional feedforward neural network represented by CNN has excellent performance in solving classiﬁcation tasks but cannot handle the complex time correlation between information. RNN introduces a directional loop, where the output of the neuron can be directly applied to itself at the next timestamp. This directional loop enables RNN to handle the problem of before and after the input. In order to solve the long-term dependence in neural networks and the disappearance and outburst of the conventional RNN model gradients, in 1997 Hochreiter and Schmidhuber[6] proposed LSTM, an RNN architecture that better stores and accesses information and serves as the basis for other models.

There has been a lot of work to prove that LSTM can achieve better results in time-series prediction. Shi et al.[7] used the LSTM improved by convolution for the time-series prediction problem of precipitation, and it is proved that the algorithm is superior to other existing precipitation prediction models. Ma[8] used LSTM to capture non-linear trafﬁc dynamics for short-term trafﬁc prediction, achieving the best predictive performance in terms of accuracy and stability. Liu[09] used LSTM to predict wind

speed. Ding et al.[10] constructed a deep convolutional neural network to study the relationship between the occurrence of events and stock prices over different time horizons. Fischer conﬁrmed that LSTM is more suitable for time-series analysis than the no-memory classiﬁcation method, such as random forest (RAF), deep neural network (DNN) and logistic regression classiﬁer (LOG).

1. **Methodology**
   1. LDA Model

In natural language processing, LDA(Latent Dirichlet allocation) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. If observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. LDA is one of the topic models that is often used in natural language processing.

LDA is a three-level hierarchical Bayesian model. Every word in a document is generated by first choosing a topic under some probability, then from that topic choosing certain word under some probability. The whole process can be interpreted in equation 1.

(1)

* 1. VADER Model

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of a sentiment lexicon that is a list of lexical features (e.g., words), which are generally labeled according to their semantic orientation as either positive or negative.

The text is analyzed based on sentiment metrics including positive, negative, neutral and overall compound score. Beside word itself, the factors that influence the sentiment metric include punctuation, capitalization, degree modifiers, conjunctions, preceding trigram according to VADER. It also performs very well with emojis, slangs and acronyms in sentences.

* 1. Autoregressive Integrated Moving Average

Autoregressive Integrated Moving Average (ARIMA) model is widely used to predict linear time series data. Comparing with traditional ARMA model, it effectively transforms the non-stationary time series data, such as financial index, to stationary series.

* 1. LSTM

As mentioned before, the model is a LSTM network which is a type of Recurrent Neural Network (RNN). RNNs are used for time-series data because they keep track of all previous data points and can capture patterns developing through time. Due to their nature, RNNs many time suffer from vanishing gradient, that is the changes the weights receive during training become so small, that they don’t change, making the network unable to converge to a minimal loss (The opposite problem can also be observed at times — when gradients become too big. This is called gradient exploding, but the solution to this is quite simple — clip gradients if they start exceeding some constant number, i.e. gradient clipping). Two modifications tackle this problem — Gated Recurrent Unit (GRU) and Long-Short Term Memory (LSTM). The biggest differences between the two are: (1) GRU has 2 gates (update and reset) and LSTM has 4 (update, input, forget, and output), (2) LSTM maintains an internal memory state, while GRU doesn’t, and (3) LSTM applies a nonlinearity (sigmoid) before the output gate, GRU doesn’t.

In most cases, LSTM and GRU give similar results in terms of accuracy but GRU is much less computationally intensive, as GRU has much fewer trainable params. LSTMs, however, and much more used.

The basic structure of a LSTM cell and associate math equations is shown as Fig.2

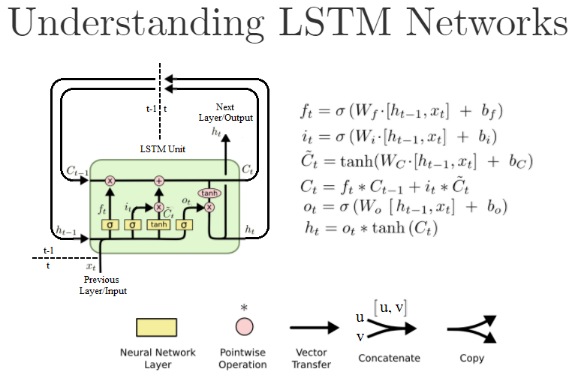


Fig.2 LSTM Networks

For all , the two activation functions are:

1. **Experiment**

In this section, we discuss the whole experiment in detail, followed by detailed discussions on the three key modules: topic generation and sentiment analysis of Mr. President Trump’s twitter; data selecting and preprocessing; fitting and tuning LSTM models.

* 1. NLP Preprocessing:
     1. Data Acquisition and Cleaning

This project focuses on the tweets of President Donald Trump during his presidency, from November 01, 2016 to October 31, 2019. Generally we could get our data from Twitter public API, but for the limitation of queries per time, we decided to use online Trump’s tweets archive[11]. We exported their archive, and got tweets.json file with 12295 tweets from @realDonaldTrump during his presidency.

In order to match our tweets with everyday stock index, we need to adjust our tweets. First we label each tweet with a timestamp: if the tweet is posted on 12am-4pm from Monday to Friday, we label the tweet the date it posted; if the tweet is posted after 4pm from Monday to Thursday, we make the date it posted plus one; if the tweet is posted after 4pm on Friday, we make the date plus three; if the tweet is posted after 4pm on Saturday, we make the date plus two; if the tweet is posted after 4pm on Sunday, we make the date plus one. That is because the stock market is closed after 4pm during weekdays and is closed during weekend. Then we combine the tweets of the same timestamp into one document. Therefore we match the tweets with stock index day by day with 775 tweet documents.

We clean our tweet documents in 2 ways. For topic generation, we clean the document’s text into bag of words so that we could apply LAD model; for sentiment analysis, beside the text, we also keep the VADER influence factors such as punctuations (exclamation marks) and capitulations.

After cleaning, we calculated the most frequent words on Trump’s twitter. The top 30 is [great, president, people, trump, democrats, country, news, thank, big, fake], [border, new, america, never, media, american, good, china, house, job], [vote first, bad, military, made, wall, trade, deal, crime, win, security, russia, collusion]. And the word cloud is shown in Fig.3.

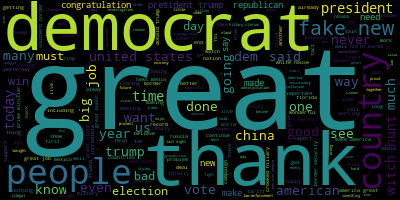


Fig. 2 Word Cloud of most frequent words of Trump’s Twitter

* + 1. Topic Generation

We apply LDA model using genism package. But there is a issue: each topic we generated contains “great”, “thank” with high P (word|document) probabilities which contribute nothing to our topic exaction. We need to lower the importance of those meaningless words and raise the importance of true topic words in our corpora.

We apply TF-IDF model to our corpora first. TF-IDF(Term Frequency–Inverse Document Frequency) model is a numerical statistic that is intended to reflect how important a word is to a document in a corpus. The TF–IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. By applying TF-IDF model to our corpora, we made the words that contribute to topic exaction stand out.

Next step is to decide the number of topics to be generated. The general method is to calculate perplexity of different models under different number of topics and to select the turning point of low perplexity. It is worth noting that perplexity is a measurement of how well a probability distribution or probability model predicts a sample and it does not necessary suggest a proper number of topics. This coincides with the view of genism package author Radim[12]. The more relevant metrics are topic coherence and topic exclusivity. The coherence measures how often the topic words appear together in the corpus so that the same topic would not spread out. The exclusivity measures the overlap of topics. We want to maintain the high topic sematic coherence at the same time make the overlap of topics as small as possible. We tried topic numbers from 10 to 50, finally selecting 20 as our topic number.

The result is showed on appendix part of this report. I selected top 30 words from each topic, after eliminate meaningless words, extracted topic from those words. The topic 15 “Presidency” is the most frequent topic Trump has mentioned, this topic is formed by the words list [president, border, democrats, trump, jobs, fake, news, country, people, America, vote, china, new, big, wall, media, tax, Russia, trade, great, security, …].

We also noticed three issues from our results: Topic 5 and topic 10 involves two kinds of topic words which is correlated to our exclusivity issue; “Industry” related top words is scattered in topic 5, 14 and 16 which is correlated to our coherence issue; we cannot extract clear topic from topic words of 17 and 18, therefore we denote them as “ambiguous”. We could not extract clear topic from topic words is due to three possible reasons: there are either too many descriptive and sentiment words like proper, dynamic which could possibly describe any topic; there are some proper nouns like names and places which are hard to classify; the last reason is just the coherence issue, some words does not cohere tightly to form a interpretable topic.

* + 1. Sentiment Analysis

We apply VADER model to our cleaned 775 tweets and got a list of sentiment scores that each tweet has a positive, negative, neutral and compound score. Part of the result is shown in Fig.4. We found that out of 775 compound score, only 130 are scored overall negative, which means Trump’s twitter’s sentiments are generally positive. We extracted the compound score as a feature of our stock price to feed in our LSTM model in our next step.

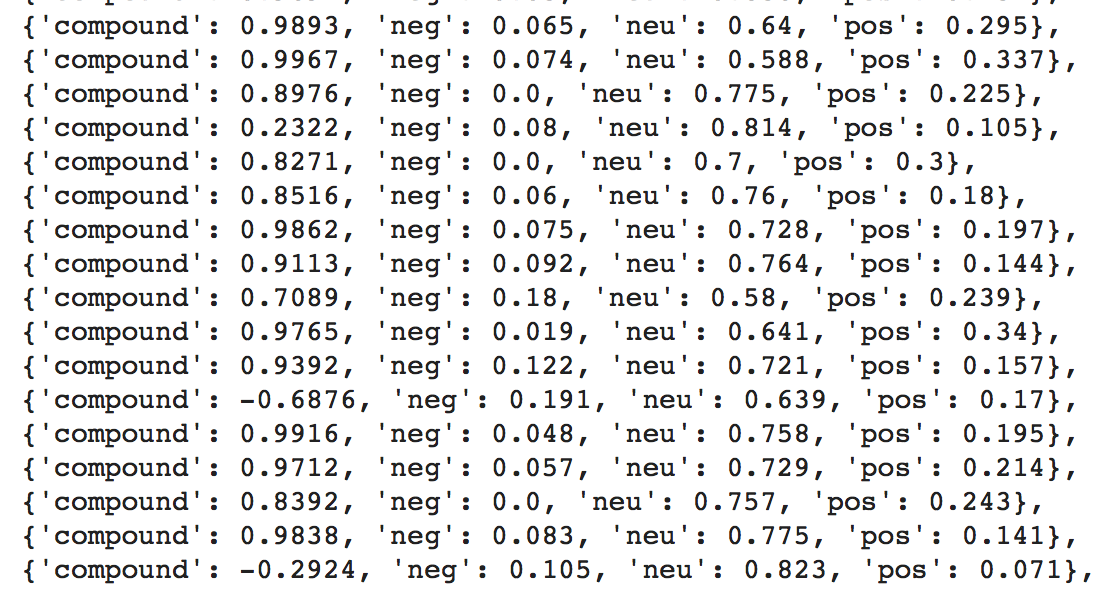


Fig. 4 Part of Sentiment Analysis Results

* + 1. Data Preprocessing
       1. Features selecting

In term of historical stock index data, we choose two features, including the closing price and the trading volume.

* 1. Dataset
     1. Data Selecting

Our dataset is divided into two parts: the historical stock index dataset and the sentiment index of Mr. President Trump’s tweets. As given in Table 1, many financial websites provide historical stock data, e.g., Yahoo or Google Finance. We obtain S&P500, NASDAQ, DJI ( Dow Jones Industrial Index) and HIS (Hangseng Index, Hong Kong, China) from Yahoo Finance, the Web’s #1 Finance site, which guarantees the accuracy of the data from the data source. We also get SZCZ, an important stock index in China, from SINA.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Explanation** | | |
| ***Features*** | ***Size*** | ***Function*** |
| S&P500 | Date, Open, High, Low, Close, Volume | 755 | Predicting stock prices |
| NASDAQ |
| DJI |
| Russell2000 |
| HSI | 740 |
| SZCZ | 710 |
| Tweets | Compount sentiment scores | 775 |

Table 1 Dataset

* + 1. Data Preprocessing
       1. Features selecting

In term of historical stock index data, we choose two features, including the closing price and the trading volume.

* + - 1. Features extension

We need to understand what affects whether stock index price will move up or down, therefore we need to incorporate as much information as possible. As a high-level overview, the features we use are:

* Correlated assets: These are other assets, the stock index depends on other stock index too.
* Technical indicators: A lot of investors follow technical indicators. We include the most popular indicators as independent features, e.g., 7 and 21 days moving average, exponential moving average, momentum, Bollinger bands, MACD.
* Fourier transforms: Along with the daily closing price, we create Fourier transform in order to generalize several long and short-term trends. Using these transforms we eliminate a lot of movement. Having trend approximations can help the LSTM network pick its prediction trends more accurately.
* Autoregressive Integrated Moving Average (ARIMA): This was one of the most popular techniques for predicting future values of time series data. We add it and see if it comes off as an important predictive feature.
* Compound sentiment score of each tweet document
  + - 1. Feature importance

Having so many features, we have to consider whether all of them are really indicative of the direction stock index will take. There are many ways to test feature importance, but the one we apply uses XGBoost, because it gives one of the best results in both classification and regression problems.

* + - 1. Data normalization.

For the purpose of training fast and convergence, we need to normalize all features before feeding them into neural network. Noteworthily, with different magnitudes, we should better separate them to different groups to normalize, otherwise, those features with smaller magnitudes will all nearly close to 0, and the algorithm won’t learn anything. Another crucial thing is we should keep the label alone and normalize it separately whatever the magnitude it is. The reason is we are going to deformalize the label back after training and we need the original ratio to address the backward operation.

* + - 1. Data Splitting

Consider for the moment a standard feed-forward network. These networks expect input and output data that is two-dimensional. That is, data with “shape” [numExamples, inputSize]. This means that the data into a feed-forward network has “numExample” rows/examples, where each row consists of “inputSize” columns. A single example would have shape [1, inputSize], though in practice we generally use multiple examples for computational and optimization efficiency. Similarly, output data for a standard feed-forward network is also two dimensional, with shape [numExamples, outputSize].

Conversely, data for RNNs are time series. Thus, they have 3 dimensions: one additional dimension for time. Input data thus has shape [numExamples, inputSize, timeSeriesLength]. This means that our stock index data is laid out such that the value at position (i, j, k) is the kth value at the jth time step of the ith example in the minibatch. This data layout is shown below. In our experiment, we set time frame into two chunks, 5 days and 22 days, to represent short term (week) and long term (month). For the example of S&P500 with 755 observations and 42 features, the dimension of input data should be [749, 5, 42] and [732, 22, 42], correspondingly, the input dimension of label should be [750] and [733].

Then we split the dataset into train, validation and test sets according to the proportion of 70:15:15.



Fig.5 Explanation of time padding

* + 1. Training Strategy

We select two stock index samples from US and China, and feed long and short-term training and validation dataset into RNNs and LSTMs models respectively. During the tuning hyperparameters session, we apply the procedure below.

* + - 1. Learning rate scheduler

One of the most important hyperparameters is the learning rate. Setting the learning rate for almost every optimizer (such as SGD, Adam, or RMSProp) is crucially important when training neural networks because it controls both the speed of convergence and the ultimate performance of the network. One of the simplest learning rate strategies is to have a fixed learning rate throughout the training process, but this comes at the expense of limiting the initial speed of convergence. Changing the learning rate over time can overcome this tradeoff.

Recent papers, such as [13], show the benefits of changing the global learning rate during training, in terms of both convergence and time. The figure of learning rate scheduler changing with epoch is shown below.



Fig.6 Learning rate scheduler

* + - 1. Manners of preventing overfitting and the bias-variance trade-off

Having a lot of features and neural networks we need to make sure we prevent overfitting and be mindful of the total loss. We use several methods for preventing overfitting (not only for LSTM, but also in the Neural Network)

* Regularization (or weights penalty). The two most widely used regularization techniques are LASSO (L1) and Ridge (L2). L1 adds the mean absolute error and L2 adds mean squared error to the loss. Without going into too many mathematical details, the basic differences are: L1 does both variable selection and parameter shrinkage, whereas L2 only does parameter shrinkage and end up including all the coefficients in the model. In presence of correlated variables, L2 might be the preferred choice. Also, L2 works best in situations where the least square estimates have higher variance. Therefore, it depends on our model objective. The impact of the two types of regularizations is quite different. While they both penalize large weights, L1 regularization leads to a non-differentiable function at zero.

L2 regularization favors smaller weights, but L1 regularizations favors weight that go to zero. So, with L1 regularization we can end up with a sparse model (i.e., one with fewer parameters). In both cases the parameters of the L1 and L2 regularized models “shrink”, but in the case of L1 regularization the shrinkage directly impacts the complexity ( the number of parameters) of the model. Precisely, L2 regression works best in situations where the least square estimates have higher variance. L1 is more robust to outliers, is used when data is sparse, and creates feature importance. We use L1 regularization to tune our models.

* Dropout. Each time when using a batch update the weights of neural network, we temporarily remove a percentage of neurons from the network. After the update is complete, then add new neurons to temporarily remove. That’s the main idea of dropout. It causes effectively train the network as if it were small (each update will be computed using a small network), however at inference should use all the neurons, gaining the advantage of a larger network. Another reason states that neurons will not learn to depend highly on just a couple previous neurons, but on many, thus when some neurons are not available (due to noise), it will still function. In this experiment, we set this parameter below 0.15.
* Early stopping. As a popular approach, early stopping can control overfit easily. During tracking validation loss each epoch, once it stops improving (or worse starts increasing), stop training. Ideally, we would like a way to stop our model from overfitting while still allowing it to train and find an optimal solution, so not a perfect solution on its own. However, we almost never be able to get network to completely stop overfitting, therefore it is often wise to implement some form early stopping in addition to any other measures.
* Bias-variance trade-off. Another important consideration when building complex neural networks is the bias-variance trade-off. Basically, the error we get when training nets is a function of the bias, the variance, and irreducible error-ε (error due to noise and randomness). The simplest formula of the trade-off is:

Error = bias2 + variance + ε (4)

Bias measures how well a trained (on training dataset) algorithm can generalize on unseen data. Variance measures the sensitivity of the model to changes in the dataset. High variance is the overfitting.

1. **Performance Evaluation**
   1. Performance evaluation metric

In this project, our basic evaluation indicators of the regression model include root mean square error (RMSE) and R-square (R2). Their equations are shown as follows:

(5)

(6)

Here, m is the total number of samples; and represent the actual and predicted value of the test set, respectively; and represents the mean of real values of the test set. The normal range of R2 is [0, 1]. The closer this value to 1, the stronger the ability of the equation to interpret y, and the better the model fits the data.

We also use Welch’s t-test to make comparison mean of predicted and true values. In statistics, Welch’s t-test, or unequal variances t-test, is a two-sample location test which is used to test the hypothesis that two populations have equal means. Comparing to the Student’s t-test, it is more reliable when the two samples have unequal variance and/or unequal sample sizes. These tests are often referred to as “unpaired” or “independent samples” t-tests, as they are typically applied when the statistical units underlying the two samples being compared are non-overlapping. In our project, we assume the null hypothesis H0 is the means between predicted and true values are same (i.e. μ1 = μ2) under the significant level α = 0.05 and implement the Welch’s t-test which defined by the following formula:

(7)

Where , and are the 1st sample mean, sample variance and sample size, respectively. Unlike in Student’s t-test, the denominator is not based on a pooled variance estimate. The degrees of freedom v associated with this variance estimate is approximated using the Welch-Satterthwaite equation:

Here , the degrees of freedom associated with the first variance estimate. , the degrees of freedom associated with the 2nd variance estimate.

After we calculate the t value, we compute the p-value to compare to the significant level, if p-value > α, we false to reject the null hypothesis, that is the means between predicted and true values are same, vice versa.

Appendix 2 gives the detailed results of the evaluation indicators for each model and every stock index.

* 1. Examining predicted performance by residuals and quantile-quantile plots

Residual is the difference between the true value and the predicted value. It’s an important indicator to measure the quality of a regression model. We use the residual plot to illustrate intuitionally.

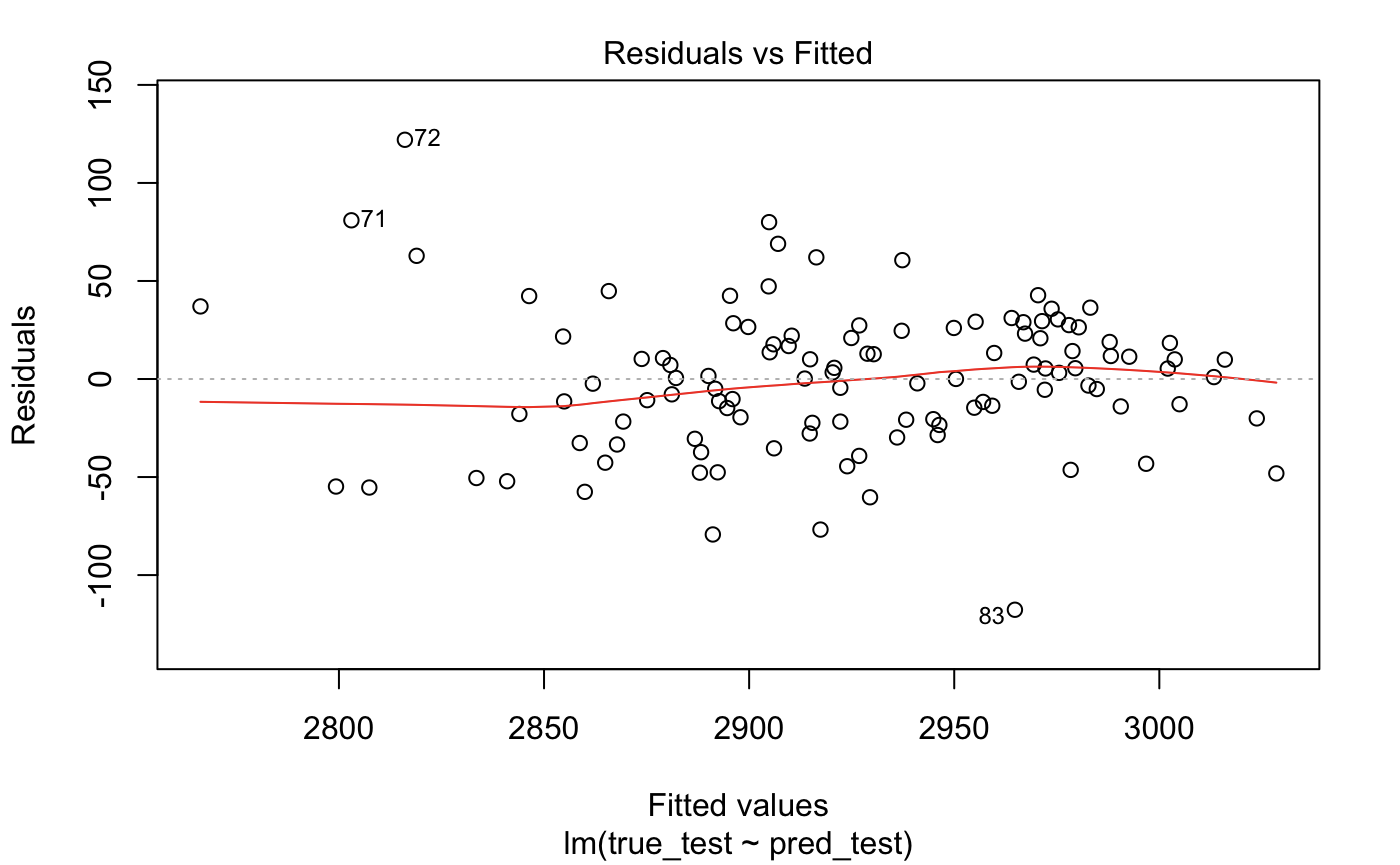


Fig. 7. Residuals figure

The predicted values lie on the x-axis, and the residuals on the y-axis. We can see from this figure that both positive and negative points are pretty symmetrically distributed, tending to cluster towards the middle of the plot, and are clustered around zero of the y-axis, it means our models are less likely to bias.

We also plot a quantile-quantile figure to show the distributions of predicted and true values. A quantile-quantile figure is a scatterplot created by plotting two sets of quantiles against one another. It is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential.

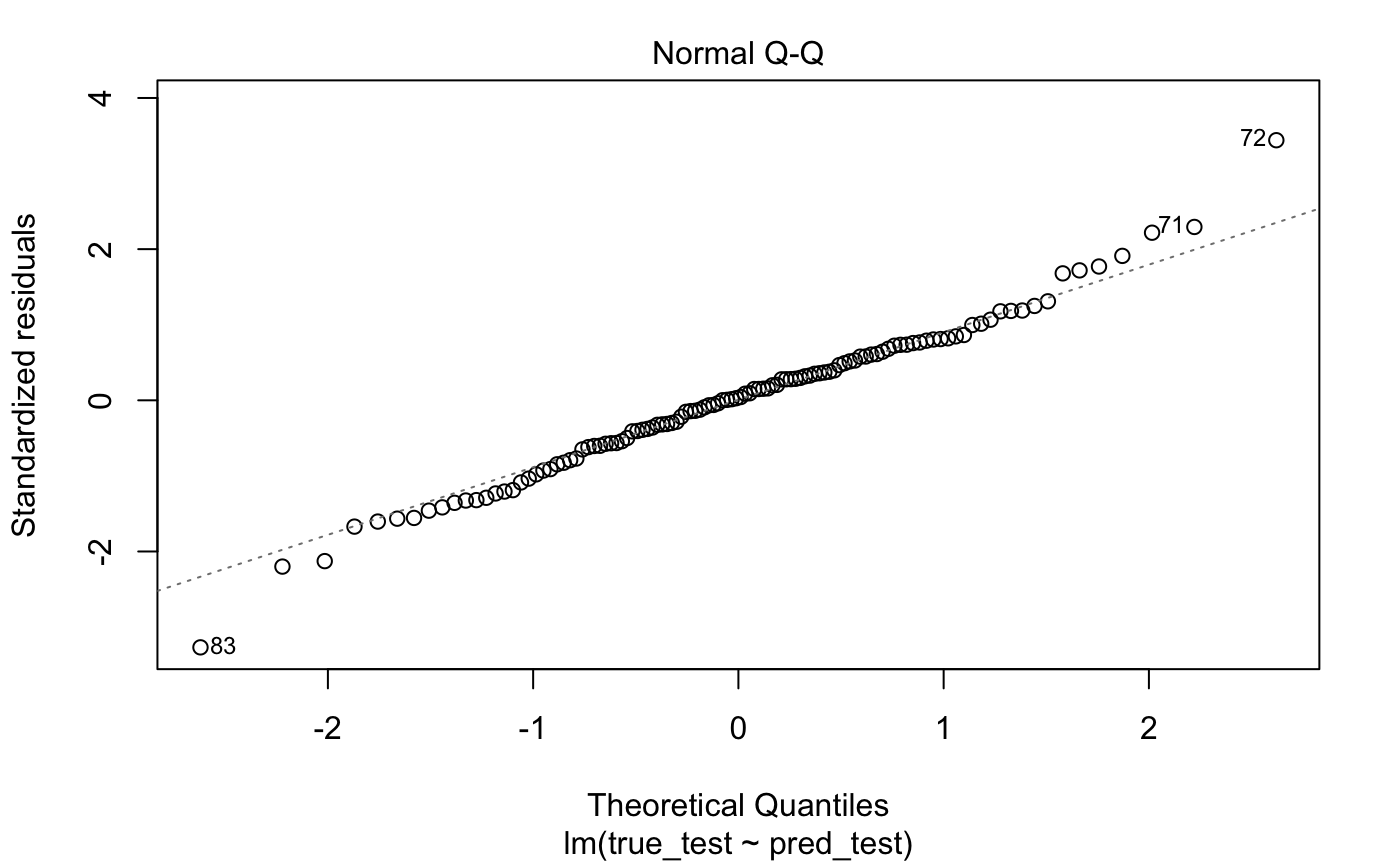


Fig.8. Quantile-quantile plot

As shown in this figure, we see the points forming a line that’s roughly straight, that is both sets of quantiles came from the same distribution, the predicted and true values follow the linear relationship.

* 1. Evaluate the effectiveness of models by stock index price fitting lines

We use the fitting line figure to assess the effectiveness of models. The result of fitting S&P 500 is shown in Fig.9.

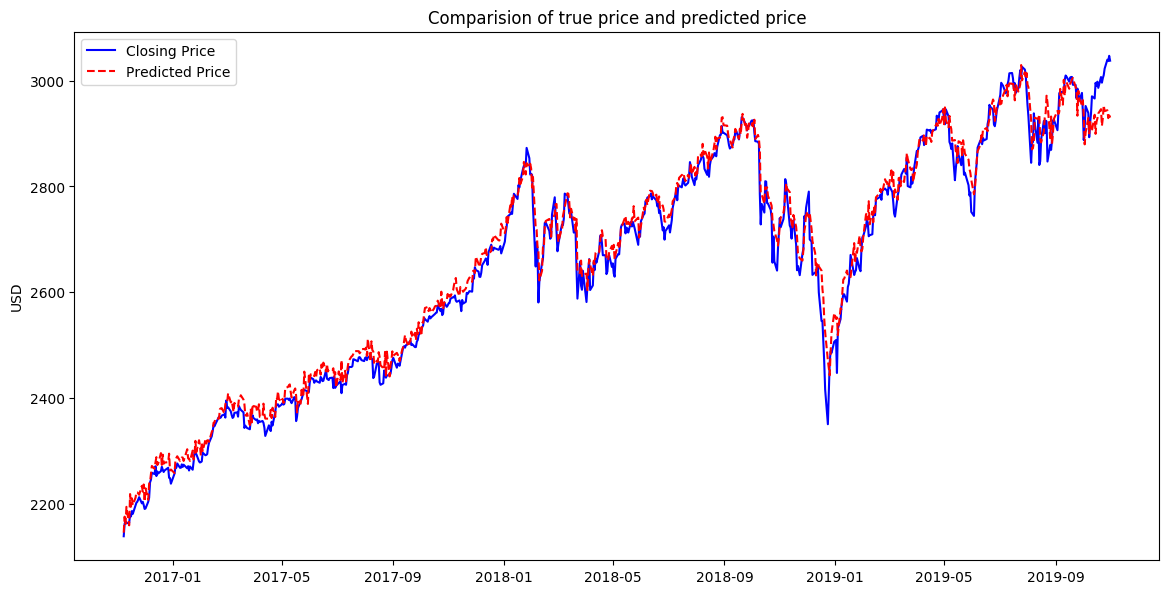
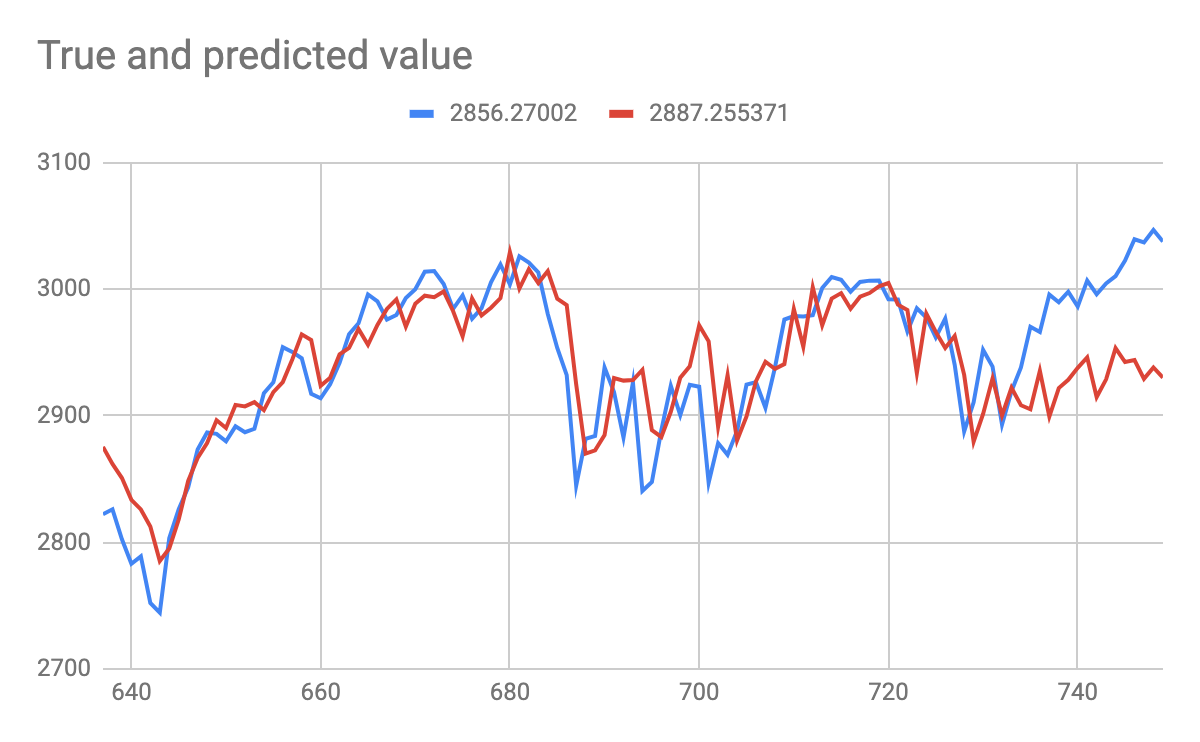


Fig. 9 Fitting result between true and predicted values of S&P500.

In Fig.9, the x-axis and the y-axis represent the date and the closing price of the S&P500, respectively. The blue line labeled “Closing Price” represents the real stock index closing price, the red line labeled “Predicted Price” is the predicted closing price of model. As shown in Fig. 9, compared to the baseline of true price, the result of prediction fits well.

In Fig.10, we zoom in the fitting result specifically on the test set.



True price

Predicted price

Fig.10. Fitting result between true and predicted values on test set of S&P500.

Like the last figure, the blue line represents the true value and red line represents the predicted value. Even in the test set the fitting quality is creditable, except the last part, where true value goes increasingly but the predicted value doesn’t follow it promptly.

We also compare the result of China’s index SZCZ to test whether our mode are adaptable other country’s stock index, and the result is acceptable.

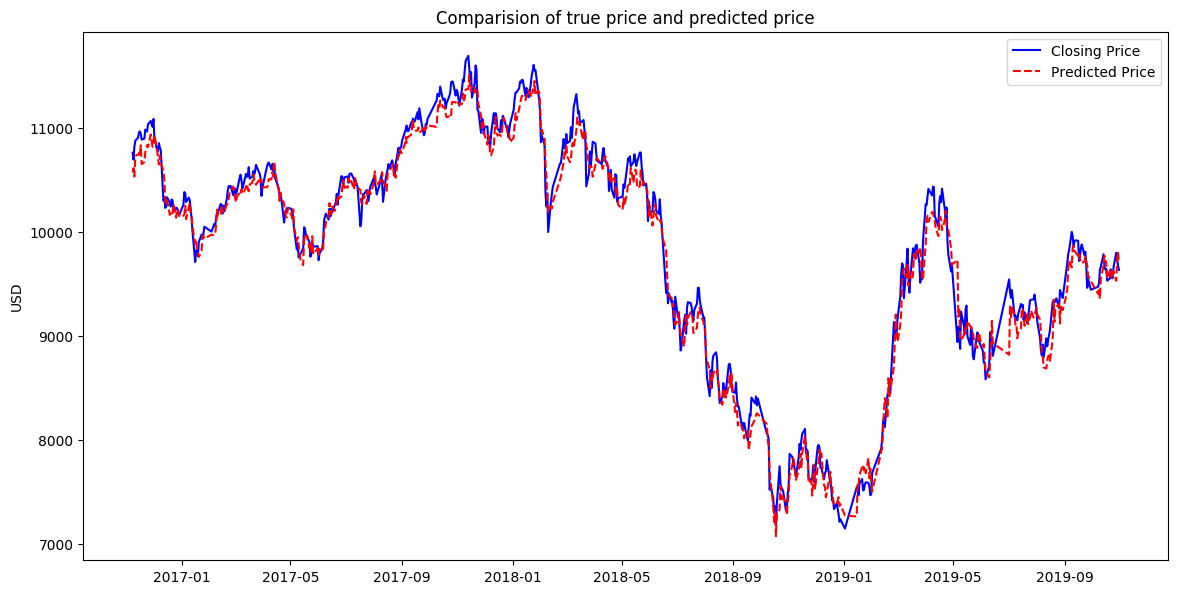


Fig.11. Fitting result between true and predicted values of SZCZ

* 1. Evaluate the effectiveness of sentiment analysis

In this project, we assume the sentiment analysis of social media especially from Mr. President Trump’s tweets can affect the stock index price. Therefore, we make an experiment to test the influence of sentiment analysis.

We combine the result of predicted values with and without sentiment analysis attributes and crop a part of fitting result as the Fig. 12.

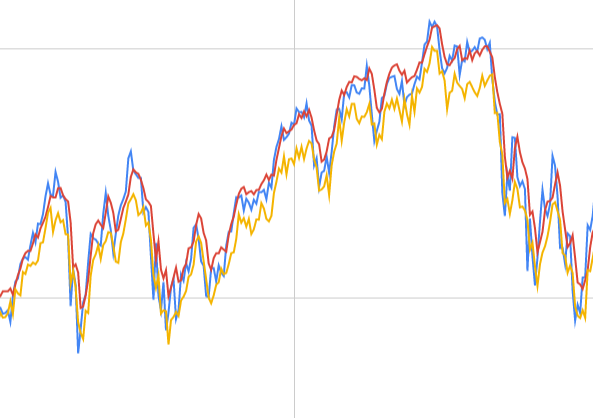
 

Fig.12. Comparison of predicted values between with and without sentiment analysis.

The blue line represents true value, the red line represents the predicted value without sentiment analysis, and the yellow line represents the predicted value with sentiment analysis. We train two samples under the same situation, e.g., same model structure, same hyperparameters, we can see the red line fits the true value better totally, and the values of yellow line are lower than true value. It means with more features, the model with sentiment analysis need to be tuned deeply. However, in some local part, the yellow line can reflect true value’s volatility better than comparison group, the red line. It shows our models can capture the influence on stock index from the sentiment analysis.

1. **Conclusion**

In this project, a LSTM-based model is proposed for stock index prediction. Expressly, for one thing, sentiment index is used to take the social media emotional tendency into consideration.

According to experiments conducted on the dataset of several stock index in both US and China, the performance of the proposed scheme has been verified. The experimental results show that the proposed scheme outperforms the comparison schemes consistently. The models from this project are acceptable for predicting stock index price from many countries. This project also verifies the influence on stock index from sentiment analysis can be captured by our models.

Last but not least, in this work, we adopt sentiment analysis scores from 20 topic, however, we couldn’t specify the implied meaning of these topics, that causes the feature selecting and model tuning lack of interpretation from the practical analysis. In the future work, we will try to extract more refine and reasonable features of sentiment analysis.

1. **Bibliography**
2. Tong Yang and Yuxin Yang, December 16, 2017, “Predict Effect of Trump’s Tweets on Stock Price Milestone”. Retrieved December 12, 2019, from:

<http://cs229.stanford.edu/proj2017/final-reports/5232886.pdf>

1. Yusuf Khan, September 09, 2019, “A JPMorgan bot analyzed 14,000 Trump tweets and found they're having an increasingly sharp impact on markets”. Retrieved December 12, 2019, from:

<https://markets.businessinsider.com/news/stocks/jpmorgan-volvefe-index-on-impact-of-trump-tweets-on-market-volatility-2019-9-1028508499>

1. Tang H, Chiu KC, Lei X (2003) In: Proceedings of 3rd international workshop on computational intelligence in economics and ﬁnance (CIEF’2003), North Carolina, USA, September 26–30, pp 1112–1119
2. Duan T (2016) Auto regressive dynamic Bayesian network and its application in stock market inference. In: IFIP international conference on artiﬁcial intelligence applications and innovations. Springer, Berlin
3. Hinton G, Deng L, Yu D et al (2012) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
4. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
5. Xingjian SH, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems 2015, pp. 802–810
6. Ma X, Tao Z, Wang Y et al (2015) Long short-term memory neural network for trafﬁc speed prediction using remote microwave sensor data. Transp Res Part C Emerg Technol 54:187–197
7. Liu H, Mi X, Li Y (2018) Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. Energy Convers Manag 159:54–64
8. Ding X, Zhang Y, Liu T, Duan J (eds) (2015) Deep learning for event-driven stock prediction. In: International conference on artiﬁcial intelligence
9. Trump’s tweets archive. Retrieved December 12, 2019, from:

<http://www.trumptwitterarchive.com>

1. Google Group: Finding number of topics using perplexity. Retrieved December 12, 2019, from:

<https://groups.google.com/forum/#!searchin/gensim/perplexity%7Csort:relevance/gensim/TpuYRxhyIOc/98kCvxcpDLcJ>

1. Bo-Yang Hsueh, Wei Li, Stochastic Gradient Descent with Hyperbolic-Tangent Decay on Classification

Appendix

|  |  |  |
| --- | --- | --- |
| 1 | Tour | Fort Myer, SOTU, Panama, hurricane, blessing, Colombia, square, overwhelming, forest, unlikely, quoting, CEOs, backwards |
| 2 | Politics | Troy Balderson, WEF, brave, dropped, shut, directed, colleagues, Jeff Flake, liability, unmasking, affordable, raging, bail out, roar, withdraw, ovations |
| 3 | Livelihood | Obamacare, imploding, Kaye West, picked, birthright, citizenship, suspect, tears, murder, deeds, picks, comprehend, Dept. |
| 4 | World’s Affairs | Helsinki Finland, anti-Semitism, commencement, UNGA, Rivlin, electric, restoration, pursuit, danger, barrier, overturned |
| 5 | Industry/  Impeachment | premiums, industries, deductible, manufacturing, subpoena, oath, treat, spying, mentioned, motors, proposal, James Clapper, tears, clue, cooperation, opposition |
| 6 | Public Figure | Woodward book, transition, Omarosa Newman, comedian, continued, preserving, wish, correspondents, strengthening, whatsoever |
| 7 | Care | women, pretend, aggrieved, complaining, hypocrites, humanity, frantically, celebrities, Laura Ingraham, detailed, drain the swamp, proactive, ashamed |
| 8 | Immigration | lottery, chain, barrier, migration, reviews, ultimate, plausible, visa, random, inflow |
| 9 | Military | crime, Iran, DACA, business, federal, talk, steel, veterans, plan, service, judge, angry, responders, freedom, Vietnam, decades, across, police, citizens |
| 10 | Economic/  Police | Ronald Reagan, Elizebeth Warrenm, informant, google, mass, steelworkers, fellow, IACP, memo, shootings, surge, dropped, wounded, exposed, cash, law enforcement |
| 11 | Religion | Pope, holiness, pastor, despair, frontiers, Vatican, boldly, pursue, Italy, omnibus |
| 12 | Event | anniversary, summit, ISIS, POTUS, professionals, kicking, dedicated, cemetery |
| 13 | Identification | POTUS, Riyadh summit, drain the swamp, remarkably, Doral, statute, bother, bid, serious, unemployed, opponents, identification |
| 14 | Industry | poised, retreat, reception, surge, industrial, dealings, caravans, production, equal, plummet |
| 15 | Presidency | president, border, democrats, trump, jobs, fake news, country, people, America, vote, china, new, big, wall, media, tax, Russia, trade, great, security |
| 16 | Industry | owners, protester, automobile, uniform, unwavering, sold, executives, losses, campaigned, powering, dreamers, plants, cellphone, smaller, cars |
| 17 | Ambiguous | inauguration, England, memorial, Brussels, grow, agony, treasure, proper, solidarity, untrue, challenge, governing, manufacturing, concert, tones, dynamic, journalist |
| 18 | Ambiguous | divide, adjusting, arguing, bonds, intentions, imports, shatter, Google, dynamics, disbar, strong, airline, communications |
| 19 | Technology | cyber attack, hack, fiction, deplorable, gift, irony, negotiator, deductible, skyrocket |
| 20 | Congress | joint address, joint session, delivered, delegation, annual, censure, roll, express, superior, currencies |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Short term (5 Days)** | | | | **Long term (22 Days)** | | | |
| **Stock** | **RNN** | **LSTM** | **LSTM with peephole** | **GRU** | **RNN** | **LSTM** | **LSTM with peephole** | **GRU** |
| **S&P 500** | **RMSE: 42.93**  **t-test: 0.3301**  **R^2: 0.7841** | **RMSE: 40.99**  **t-test: 0.4155**  **R^2: 0.6906** | **RMSE: 53.07**  **t-test: 0.0074**  **R^2: 0.6767** | **RMSE: 41.79**  **t-test: 0.6327**  **R^2: 0.6383** | **RMSE: 66.61**  **t-test: 0.00372**  **R^2: 0.4165** | **RMSE: 74.51**  **t-test: 0.0005**  **R^2: 0.3639** | **RMSE: 39.96**  **t-test: 0.1764**  **R^2: 0.669** | **RMSE: 40.99**  **t-test: 0.1192**  **R^2: 0.4993** |
| **NASDAQ** | **RMSE: 121.95**  **t-test: 0.2659**  **R^2: 0.7416** | **RMSE: 131.67**  **t-test: 0.05913**  **R^2: 0.7182** | **RMSE: 159.78**  **t-test: 0.1025**  **R^2: 0.6786** | **RMSE: 171.26**  **t-test: 0.0002**  **R^2: 0.6836** | **RMSE: 126.8**  **t-test: 0.04417**  **R^2: 0.6905** | **RMSE: 212.0**  **t-test: 0.0077**  **R^2: 0.3791** | **RMSE: 156.96**  **t-test: 0.5645**  **R^2: 0.5405** | **RMSE: 149.7**  **t-test: 0.9293**  **R^2: 0.573** |
| **Dow Jones** | **RMSE: 425.01**  **t-test: 0.2145**  **R^2: 0.6568** | **RMSE: 399.45**  **t-test: 0.296**  **R^2: 0.7019** | **RMSE: 479.34**  **t-test: 0.0005**  **R^2: 0.6441** | **RMSE: 479.53**  **t-test: 0.0003**  **R^2: 0.6645** | **RMSE: 449.1**  **t-test: 0.0006**  **R^2: 0.6965** | **RMSE: 423.8**  **t-test: 0.284**  **R^2: 0.6645** | **RMSE: 536.3**  **t-test: 0.00001**  **R^2: 0.6941** | **RMSE: 432.4**  **t-test: 0.019**  **R^2: 0.5102** |
| **SZCZ** | **RMSE: 180.48**  **t-test: 0.07194**  **R^2: 0.7463** | **RMSE: 162.16**  **t-test: 0.1409**  **R^2: 0.7866** | **RMSE: 314.87**  **t-test: 0.00003**  **R^2: 0.6565** | **RMSE: 314.52**  **t-test: 0.00002**  **R^2: 0.6673** | **RMSE: 211.36**  **t-test: 0.1451**  **R^2: 0.7003** | **RMSE: 202.9**  **t-test: 0.6751**  **R^2: 0.6584** | **RMSE: 227.99**  **t-test: 0.01878**  **R^2: 0.6844** | **RMSE: 176.9**  **t-test: 0.1111**  **R^2: 0.7943** |