Proceedings of the 26th International Conference on Automation & Computing, University of Portsmouth, Portsmouth, UK, 2-4 September 2021

Market-Aware Sentiment Analysis for Stock Microblogs

Trung Thanh Dang[1,2], Yongqiang Cheng[2], Ken Hawick[2]
[1] University of Economics, The University of Da Nang, Vietnam
[2] Department of Computer Science and Technology, University of Hull, United Kingdom
{Thanh.DT@due.edu.vn; Y.Cheng@hull.ac.uk; K.A.Hawick@hull.ac.uk }

Abstract—**In financial markets, public sentiment acquired from microblogs allows understanding of traders' attitudes, and hence, it can be utilized in market analysis and prediction. Current works on sentiment analysis for financial microblogs only focus on the microblogging messages themselves but tend to ignore their corresponding securities exchange data in the finance market when the messages were created. This study proposes an approach to utilize the contextual information extracted from the stock market data to improve sentiment classification performance for stock-related microblogging messages. Specifically, pre-trained LSTM encoders are employed to interpret and transform the end-of-day and intraday stock data into vectors which are then incorporated into the sentiment prediction model. A 3-step training strategy is proposed to improve the convergence and accuracy of the multi-input models. Results from experiments indicate that contextual information from the stock market data improves the prediction accuracy of the sentiment classification by about 2.7%, attributed to both the end-of-day and intraday stock market data.**

Context-aware sentiment analysis; bi-directional LSTM; LSTM autoencoder; stock microblogs

I. INTRODUCTION

Sentiment extraction for microblogging messages has attracted great attention among researchers for years due to the widespread social media platforms and the challenges in interpreting such a special kind of text. Microblogging messages as a specialized type of text with loads of jargon, abbreviations, slang, misspelling, and out-of-vocabulary words. They are also relatively short and usually related to the context where they are created and discussed. In many cases, the texts contain only several words, or even a word referring to a subject only. Thus, to improve performance in sentiment classification for such kind of text, studies tend to focus on pre-processing techniques [1], utilizing specific linguistic characteristics and particular elements in the text [2] [3], or considering the social context of microblogs, such as the interactions between the authors, and the users' interactions with the messages [4][5].

Various sentiment analysis models have been proposed for financial microblogs. Still, there has not been any model taking into account the relationship between the microblogs and the exchanged securities in the market. The majority of these studies have only focused on making use of information from within the texts [6][7][8][9][10]. They tend to ignore the fact that the fluctuations in financial markets have effects on the sentiment expressed in social media posts, which were indicated in recent studies [11][12][13]. The volatility of the financial market impacts traders' attitude and trading decisions, and then these opinions can be expressed on microblogging messages. Therefore, considering the related securities' exchange information in the markets when the financial microblogs were created is particularly promising to improve sentiment analysis accuracy for these kinds of messages.

The main challenges in utilizing market data for the sentiment analysis model come from financial time-series data characteristics, leading to difficulties in interpreting and aggregating the data. The time-series stock market data are complex, highly noisy, intrinsically non-stationary, non-linear, non-parametric, and chaotic [14][15]. It requires complicated techniques in pre-processing and representing the data. Besides, the financial data can be analyzed at different frequencies from different viewpoints because of the nature of time-series data. While day traders working with intraday data to focus on price trend at minutes intervals, the others can use lower frequency information from the data, such as daily, or weekly, to figure out the overall trends of securities and the market. These traders can also use different techniques to analyze the market data. Moreover, the stock market data appeared by fixed time interval, but the microblogging messages are published randomly over time. Hence, special mechanisms are required to transform the stock market data to enable meaningful integration of such different types of data.

This paper proposes an approach to extract contextual information from stock market data to improve sentiment classification performance for stock-related microblogs. A deep neural network model containing various long short-term memory (LSTM) layers to interpret and aggregate both the textual microblogging and stock market data is proposed to classify the sentiment of the microblogging messages. In the model, bi-directional LSTM with attention mechanism is used in processing textual microblogging data to extract relationships between tokens (i.e. words, characters, or subwords) in the text in both directions and focus on the important parts of the text. Pre-trained LSTM encoders are employed to extract latent information from the time-series stock market data and represent the data in the form of vectors to be combined with vectorized textual data in the sentiment analysis model. A 3-step training strategy is applied to train the multi-input model.

The main contributions of the study include:

- A novel approach to improve sentiment classification performance for stock-related microblogging messages by utilizing contextual information from the stock market. In this approach, both end-of-day and intraday stock market data are considered.

- The combination of LSTM encoders and the 3-step training strategy help increase the accuracy of the model by improving the interpretation and integration of the time-series stock market data and the microblogging data.

The rest of this article is structured as follows. Section II will review the related work on sentiment analysis for financial microblogs and feature extraction for stock market data. Section III explains the methodology, including the model architecture and the model training strategy. Section IV describes the experimental settings, followed by discussions and the experiment results in section V. Finally, section VI concludes the paper.

II. Related Work

In financial market, sentiment analysis of microblogs allows the understanding of investors' perspectives, and then the information can be used for market forecasting and decision making in trading. Lexicon-based analysis and machine learning are the two common approaches to extract sentiment from financial microblogs. In the lexicon-based approach, the sentiment of microblogs messages can be calculated using the existing lexica. The lexica can be general-domain, such as Harvard IV-4 [16], or domain-specific, such as Loughran–McDonald [17]. In 2016, Oliveira et al. proposed a method to create lexica specifically for stock market microblogs, and the method was used in later studies [6][7]. As for machine learning approaches, sentiment analysis models are trained with labelled datasets. Recent achievements in textual data vectorization such as Word2Vec [18], GloVe [19], and BERT [20] have been applied in the machine learning approaches for predicting the sentiment of financial microblogs [8][10]. In [8], two word-embedding approaches, including Word2Vec and GloVe, and three deep neural network models, including CNN, LSTM, and GRU, were compared in investor sentiment classification with a set of StockTwits messages. In the experiment, the emojis have shown good contributions to improve sentiment classification. In [10], FinALBERT, an ALBERT-based model, was proposed. It was compared with BERT and other models in sentiment classification using a set of StockTwits messages. The BERT-based models usually outperform other models. However, this study assumes that the sentiment of all the messages depends on the stock prices only and tend to ignore the sentiment labels assigns by the messages' authors. To the best of our knowledge, there has not been any literature exploiting context information from exchanged securities mentioned in financial microblogs for classifying sentiment. Thus, in this study, we will utilize this kind of data to improve sentiment classification performance.

The stock market data are represented in time series, and they can contain prices and exchanged volume as well as a wide range of technical indicators. Therefore, it requires an appropriate data representation mechanism to keep both the temporal links and the relationships among the indicators. Several approaches have been proposed and applied in earlier studies to reduce the number of input features in stock data analysis, such as principal component analysis (PCA) [21][22], Restricted Boltzmann Machine (RBM), and Autoencoder (AE) [21]. In [21], three feature extraction methods were used in intraday stock returns prediction models, but either the number of reduced features is insignificant, or the prediction accuracy of the models decreased notably. In [22], PCA - a commonly-used feature reduction method in multivariate analysis, was employed on a 52-variable stock dataset and improved ensemble models. However, PCA only reduces the number of indicators and ignores the relationships between time points. Besides, the training for PCA only focuses on improving the data reconstruction accuracy instead of the relationship with the output of the prediction models. Similarly, the stacked AE in [28] is used separately with LSTM, which leads to the ignorance of the relationships of features in different time points. To extract these relationships, the combination of LSTM in the form of AE can be a solution. Sagheer and Kotb [23] proposed a method to train layer by layer of an LSTM network in an unsupervised manner. Each layer of the multi-LSTM layer network (called LSTM stacked AE) is trained in a 2-layer LSTM-AE, and the encoder of the earlier one will be used to prepare input data for the next LSTM-AE. After training all single LSTM layers, the model is fine-tuned where all the pre-trained layers are retrained. The experiment results indicate that the model achieves faster convergence and higher accuracy. However, this approach has not been applied to stock market data and multi-input models. Besides, the layer-by-layer training can take a considerable amount of time. This study will use stacked LSTM encoder to extract features from stock market data, but the training strategy will be improved to fit a multiple input sentiment analysis model.

III. Methodology

A. Contextual sentiment analysis model

The overall architecture of our proposed sentiment analysis model for financial microblogs, namely TDI-LSTM, is shown in Fig. 1. The model inputs a microblog message and corresponding end-of-day and intraday stock market data to classify its sentiment into positive (bullish) or negative (bearish). The input data are interpreted and represented by three different parts before concatenated and processed by the fully connected layers to yield the results. While the textual data are vectorized by word embeddings and interpreted by a bi-directional LSTM with attention mechanism, the end-of-day and intraday stock market data are represented by an LSTM encoder. Below, the details of each part of the model and the training strategy will be described.

B. Textual data process

Before being interpreted by prediction models, the microblogging texts are vectorized by a word2vec word embeddings module. Unlike recent studies, usually using a pre-trained word2vec with a large corpus leading to a nontrivial size of the model, this study uses a word2vec model specifically trained using the microblog dataset collected by ourselves. The trained word embeddings model is relatively small and focuses on the domain-specific vocabulary used in stock-related microblogs. Hence, this compact model can vectorize the texts faster and represent the data more precisely.

The vectorized textual data are then interpreted by a bi-directional LSTM with attention mechanism proposed by [24]. The bi-directional LSTM ensures that both forward and backward relationships between words in the text are considered. After being processed by the bi-directional LSTM layers, corresponding to each word, the output includes a pair of vectors, one from the forward-LSTM layer and one from the backward-LSTM. An attention vector is then calculated from the output of the bidirectional LSTM layers. This weighted vector identifies the importance of each word in the text.

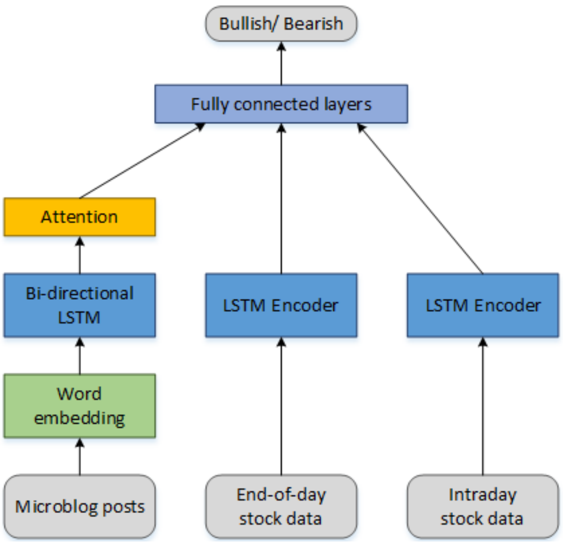

Figure 1. Structure of the TDI-LSTM sentiment classification model

Let $H \in \mathbb{R}^{N\times 2d}$ be a matrix consisting of hidden vectors yielded by the BiLSTM layers $[h_1, \ldots, h_N]$. Where h_i $(i = 1..N)$ is the combination of $\overleftarrow{h_i}$ and $\overrightarrow{h_i}$, d is the size of $\overleftarrow{h_i}$ and $\overrightarrow{h_i}$. N is the input message length. The attention mechanism produces an attention vector α, which is calculated with equations (1) and (2):

$$\alpha = softmax(\omega^T M) \qquad (1)$$

$$M = tanh(H) \qquad (2)$$

Where, ω^T is a transpose of weights vector ω.

Next, the bidirectional LSTM's output is multiplied by the attention vector, the weighted attention feature vector c is calculated with (4):

$$c = H\alpha^T \qquad (3)$$

Finally, the c is combined with the stock exchange data and fed into the fully connected layers.

C. *Feature extraction for stock market data*

Data from the stock market provides contextual information at the time the authors created social media messages. The related market data can be identified based on the timestamps of each message. This time-series data is represented as a 2D matrix where each row contains information of a time point, including stock prices, exchange volumes, and technical indicators. While the stock data only provide contextual information for the sentiment classification, the set of features is still large. Hence, an encoder is proposed to reduce the features but keep enough valuable information of the stock market. LSTM layers are employed in this encoder because of their ability in capturing the relationship of the market status among different time points of the time-series data.

The encoder is pre-trained in the LSTM AE using end-of-day or intraday stock market data. This approach inherits the general idea from [23] that using pre-trained LSTM layers can lead to quick convergence and improve the performance of the deep LSTM model. However, in our approach, each LSTM layer is not trained independently using stacked AE. Instead, both the LSTM layers are trained simultaneously. It reduces the training time but still ensures a good fit of the LSTM layers' parameters with the stock market data. The LSTM autoencoder architecture is illustrated in Fig. 2. The encoder has two LSTM layers, where the second layer has fewer neurons than the first one.

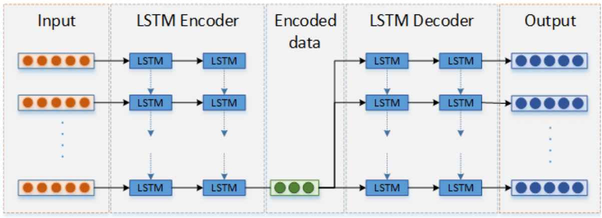

Figure 2. The LSTM AutoEncoder strucure

D. *Training strategy*

The model contains several parts taking data from different sources, processing and then combining them to produce the final prediction. It is not easy to reach the convergence state, where all parameters of these parts are optimized. To achieve the optimal state, we train the model part by part before fine-tuning all the parameters. The three steps of training and fine-tuning the model are as below:

Step 1 - Train the word2vec word embeddings model and the LSTM encoders: The word embeddings model is trained with the whole set of our collected microblogging text corpus to ensure having all the vocabulary. Two LSTM AEs containing the two LSTM encoders are trained by two stock market datasets (i.e., end-of-day and intraday). This training step allows the LSTM encoders to represent the stock data with minimum loss of information.

Step 2 - Train the sentiment analysis model: In this step, all the parameters of the two LSTM encoders are kept unchanged. This helps the training focus only on optimizing the bi-directional LSTM with attention mechanism and the fully connected layers for interpreting the textual data.

Step 3 - Fine-tune the entire model: All the model parameters are calibrated to reach the optimal state. In this step, the LSTM encoders are optimized to focusing on the task-specific information.

IV. EXPERIMENT

A. *Data description*

A dataset containing social media data and associated stock market data is prepared for training and testing the models. Each record has a StockTwits message along with corresponding daily and intraday stock data and an associated label of Bullish or Bearish assigned by the authors when they created the messages. The messages were collected from September 10th, 2018, to September 9th, 2019, and each of them mentions at least one stock in the Standard and Poor 500 (S&P 500) list.

1) Social media data

The StockTwits messages were collected by using the APIs provided by the social network. Over 710 thousand labelled messages related to stocks in the S&P 500 list were collected. Each of these messages contains at least one hashtag (#) related to the stocks. In the collected dataset, the number of positive messages is more than the negative ones. Therefore, random sampling was used to have a balanced dataset. The messages with identical contents but created by different authors or at different time by the same author were kept because the messages can have different labels depending on the market's status or other factors, despite the contents are the same. The final dataset contains 500,000 records. This dataset is split into two sub-sets, including 80% (400,000 records) for training and validating and 20% (100,000 records) for testing the models. In all the sets, the number of Bullish messages is equal to that of the Bearish messages.

The selected messages were then cleaned and tokenized. The cleaning and tokenizing process are similar to the approach used in [6] and [7]. The texts are converted to lower case, and out-of-vocabulary words are corrected. Then cash-tags, hyperlinks, numbers, and usernames are replaced by corresponding labels "CSH", "LNK", "NUM", and "USR". Positive and negative emotions are replaced by "EMOPOS" and "EMONEG" labels to differentiate these are two broad categories of emoticons. However, all labels are treated uniformly in our model without emotional contexts. Similarly, the negative words, such as "not", "no", and "none", are replaced by the label "NEG". Punctuation marks are removed, except exclamation points (!), question marks (?), and their repetitions because they can indicate the author's attitude. All the repetitions longer than two are replaced by two, (!!) or (??). Stemming is also applied, so all the words are represented in the original form. Finally, the messages were tokenized.

2) Stock market data

The end-of-day and 15-minute interval stock market data are collected from the Bloomberg database via Bloomberg terminals. Only data in the trading time (between 9:30 AM to 4:00 PM Eastern time) of the stock exchanges are used in the experiment. Therefore, each typical trading day has 26 x 15-minute time points.

Both kinds of stock data are pre-processed in the same way. All the missing time points are filled. The main reason for the missing values is that there is no transaction related to the stock at the time points; thus, the stock prices at these time points are the close price at the nearest prior time points, and the exchange volume is zero (0). Next, corresponding to each StockTwits message, two sequences of the stocks mentioned in the message, one for end-of-day and one for intraday data, are prepared. The last time point of each sequence is the time point right before the created time of the message. The length of daily stock sequences is 30, and that of intraday stock sequences is 26. At each time point of the end-of-day stock sequence, besides 5 fundamental stock indicators, including open, close, high, and low price, and exchange volume, 15 features of 11 technical indicators are calculated. As for the intraday stock sequence, 17 features of 14 technical indicators are calculated. The technical indicators used in each kind of sequence are listed in Table I. The Python library FinTA [25] is used for calculating the technical indicators. The values in these sequences are calculated and scaled into [0,1] range to eliminate effects of the non-stationary and chaotic characteristics. This is done independently for each sequence by using MinMaxScaler in Sci-kit Learn library [26]. If a message mentions more than one stock, the data of these stocks will be averaged in each time point and each feature so that each message has only one end-of-day sequence and one intraday sequence.

TABLE I. LIST OF TECHNICAL INDICATORS USED IN THE END-OF-DAY AND INTRADAY STOCK DATA

Name of indicator	End-of-day data	Intraday data
Average Directional Index	X	
Average True Range	X	X
Commodity Channel Index	X	X
Exponential Moving Average	X (5, 20) [1]	X (5, 20) [1]
Market Momentum		X
Moving Average Convergence Divergence	X	X
Percentage Price Oscillator		X
Rate-of-Change	X	X
Relative Strength Index	X	X
Simple Moving Average	X (5, 20, 50, 200) [1]	X (5, 20, 50) [1]
Stochastic oscillator %D	X	X
Stochastic Oscillator %K	X	X
Triple Exponential Moving Average		X
Ultimate Oscillator		X
Williams %R	X	X

1. In the bracket, number of time points (or days)

B. *Tested models*

The proposed model is compared with two baseline sentiment analysis models inputting textual data only and a model inputting both textual and stock market data. The first model uses the bidirectional LSTM with attention mechanism (T-LSTM), similar to the textual processing part of our proposed model. The second one is a convolutional neural network (T-CNN) which was proposed by Ye Zhang and Byron Wallace [27] and the modified version by Mathieu Cliché [28]. The third model, namely TDI-CNN, is an extended version of T-CNN in which the LSTM-autoencoders in our proposed model are employed to make use of the stock market data for the sentiment classification. Besides, to analyze the effect of end-of-day and intraday market data on the classification performance, two reduced versions of our proposed model are also used. The TD-LSTM uses the textual data along with end-of-day stock market data only, and the TI-LSTM uses the textual data along with intraday stock market data.

All the sentiment classification models and LSTM AE were built and trained by using Tensorflow [29] and Keras [30]. The models utilizing stock market data were trained with our proposed training strategy.

C. Evaluation metrics

Classification accuracy and F1 score of both positive and negative sentiment classification are used to evaluate the models' performance. They are calculated with equations (4) to (7).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (7)$$

Where TP, FP, TN, and FN are number of true positives, false positives, true negatives, and false negatives.

V. RESULTS AND DISCUSSIONS

A. Classification accuracy

TABLE II. CLASSIFICATION PERFORMANCE OF SENTIMENT CLASSIFICATION MODELS

Model	Accuracy	F1 Score		
		Bullish	*Bearish*	*Average*
T-CNN	78.05%	0.777	0.783	0.780
T-LSTM	79.59%	0.790	0.801	0.796
TD-LSTM	81.95%	0.819	0.820	0.819
TI-LSTM	81.45%	0.812	0.817	0.815
TDI-CNN	80.75%	0.806	0.809	0.808
TDI-LSTM	**82.32%**	**0.823**	**0.824**	**0.823**

The overall prediction performance of the sentiment analysis model is presented in Table II. According to the table, the tested models yield a good balance between positive and negative values. The difference in the F1 score of the bullish class is more or less equal to that of the bearish class. The accuracy and the average F1 scores of the benchmarked models range from 78.05% to 82.32% and from 0.780 to 0.823. These models' accuracy is higher than other machine learning and lexicon-based sentiment analysis models using balanced StockTwits datasets recorded in recent studies [7][9].

The results also show the role of contextual information from stock market data as the models using market data achieve higher accuracy than those using textual data of microblogs only. The accuracy and average F1 score of TDI-CNN and TDI-LSTM are higher than those of T-CNN and T-LSTM with about 2.7% accuracy and 0.27 F1 score on average. These differences indicate that the contextual information does contribute to improving the classification performance of sentiment analysis.

As for the influences of end-of-day and intraday market data, all models using textual and stock data TD-LSTM and TI-LSTM outperform the model using textual data only T-LSTM. TD-LSTM achieves 81.95% accuracy and 0.819 F1-score, which are slightly higher than those recorded by TI-LSTM with 81.45% and 0.815, respectively. Therefore, the end-of-day market data seems to have more influence on the public sentiment expressed in microblogs. However, it needs more experiments to be confirmed.

B. Impact of training strategy

The performance of the sentiment analysis models is influenced not only by the model itself and the inputted data but also by the training strategy. Specifically, the fine-tuning step significantly contributes to the improvement of the model's performance. Table III shows the test results for the models before and after they were fine-tuned. Accuracy and F1 scores of the fine-tuned versions are usually higher than the versions before being fine-tuned for about 1% and 0.01. Significantly, the model TDI-LSTM, after being fine-tuned, achieves over 1.8% accuracy improvement and 0.018 F1-score higher than its version without being fine-tuned.

TABLE III. PERFORMANCE OF THE MODEL BEFORE AND AFTER BEING FINE-TUNED

Model	Before fine-tuned		After fine-tuned	
	Accuracy	*F1 Score*	*Accuracy*	*F1 Score*
TD-LSTM	80.24%	0.802	81.95%	0.819
TI-LSTM	80.40%	0.804	81.45%	0.815
TDI-CNN	79.45%	0.795	80.75%	0.808
TDI-LSTM	80.49%	0.805	82.32%	0.823

C. Prediction accuracy for short messages

To deeper understand the contextual information's effect in classifying different kinds of texts, we analyze the relationship between the classification accuracy and the length of messages. Specifically, the accuracy of the four models, including T-CNN, T-LSTM, TDI-CNN, and TDI-LSTM, by the number of tokens per message will be compared. According to Fig. 3, TDI-CNN and TDI-LSTM, which utilize stock market data, have significantly higher accuracy than T-CNN and T-LSTM for the messages having ten tokens or fewer. Significantly, for the message having 1 to 5 tokens, TDI-CNN and TDI-LSTM can predict correctly at about 77%, while those of T-CNN and T-LSTM are only around 69% to 72%. The outperformance of TDI-LSTM and TDI-CNN also remains for the longer messages (i.e., 11 to 40 tokens) with about 2%.

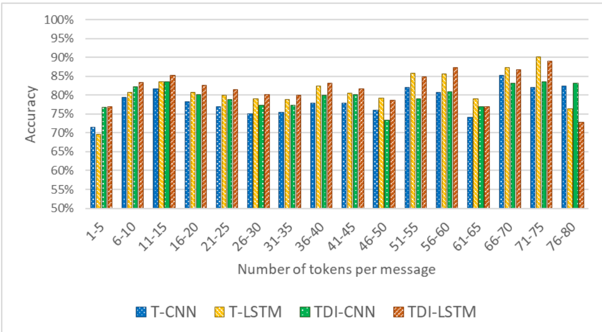

Figure 3. Classification accuracy by the number of tokens per message

VI. CONCLUSION

In this paper, an approach to improve sentiment classification performance for stock-related microblogs by utilizing contextual information from stock market data is proposed. The approach includes the stock market data interpretation and integration, and the training strategy for the model. The experiment results indicate that the contextual information has improved the sentiment classification accuracy for stock-related microblogs. It is especially good for short messages, where the sentiment analysis models using textual data only usually underperform. The experiment results also indicate that both end-of-day and intraday stock market data contribute to the improvement. Moreover, the training strategy applied for the multi-input models optimizes the parameter quicker and increases classification accuracy.

This experiment still has a limitation, where the dataset period is relatively short, with 12 months. Although in the selected years 2018-2019, the US stock market is in normal condition, it cannot represent all the characteristics in the financial market because trader behavior changes over time. In the near future, we intend to test the models with datasets having a longer time range. We also plan to use BERT for the textual data modelling.

REFERENCES

[1] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Syst. Appl.*, vol. 110, pp. 298–310, Nov. 2018.

[2] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015.

[3] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.

[4] F. Wu, Y. Huang, and Y. Song, "Structured microblog sentiment classification via social context regularization," *Neurocomputing*, vol. 175, no. PartA, pp. 599–609, 2016.

[5] J. F. Sánchez-Rada and C. A. Iglesias, "Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison," *Inf. Fusion*, vol. 52, pp. 344–356, Dec. 2019.

[6] N. Oliveira, P. Cortez, and N. Areal, "The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices," *Expert Syst. Appl.*, vol. 73, pp. 125–144, May 2017.

[7] T. Renault, "Intraday online investor sentiment and return patterns in the U.S. stock market," *J. Bank. Financ.*, vol. 84, pp. 25–40, Nov. 2017.

[8] N. Mahmoudi, P. Docherty, and P. Moscato, "Deep neural networks understand investors better," *Decis. Support Syst.*, vol. 112, pp. 23–34, Aug. 2018.

[9] T. Renault, "Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages," *Digit. Financ.*, pp. 1–13, Sep. 2019.

[10] M. Jaggi, P. Mandal, S. Narang, U. Naseem, and M. Khushi, "Text mining of stocktwits data for predicting stock prices," *Appl. Syst. Innov.*, vol. 4, no. 1, pp. 1–22, 2021.

[11] K. Shutes, K. McGrath, P. Lis, and R. Riegler, "Twitter and the US stock market: The influence of micro-bloggers on share prices," *Econ. Bus. Rev.*, vol. 2 (16), no. 3, pp. 57–77, Sep. 2016.

[12] A. Mudinas, D. Zhang, and M. Levene, "Market Trend Prediction using Sentiment Analysis: Lessons Learned and Paths Forward," in *Proceedings of the 7th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM'18)*, 2018.

[13] F. J. Garcia-Lopez, I. Batyrshin, and A. Gelbukh, "Analysis of relationships between tweets and stock market trends," *J. Intell. Fuzzy Syst.*, vol. 34, no. 5, pp. 3337–3347, May 2018.

[14] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, "Computational intelligence and financial markets: A survey and future directions," *Expert Syst. Appl.*, vol. 55, pp. 194–211, 2016.

[15] W. Long, Z. Lu, and L. Cui, "Deep learning-based feature engineering for stock price movement prediction," *Knowledge-Based Syst.*, vol. 164, pp. 163–173, Nov. 2018.

[16] H. K. Sul, A. R. Dennis, and L. I. Yuan, "Trading on Twitter: Using Social Media Sentiment to Predict Stock Returns," *Decis. Sci.*, vol. 48, no. 3, pp. 454–488, 2017.

[17] T. Loughran and B. McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *J. Finance*, vol. 66, no. 1, pp. 35–65, Feb. 2011.

[18] "Word2Vec." [Online]. Available: https://code.google.com/archive/p/word2vec/.

[19] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1532–1543, 2014.

[20] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018.

[21] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[22] B. Weng, L. Lu, X. Wang, F. M. Megahed, and W. Martinez, "Predicting Short-Term Stock Prices using Ensemble Methods and Online Data Sources," *Expert Syst. Appl.*, vol. 112, pp. 258–273, Jun. 2018.

[23] A. Sagheer and M. Kotb, "Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems," *Sci. Rep.*, vol. 9, no. 1, pp. 1–16, Dec. 2019.

[24] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Short Papers*, 2016, pp. 207–212.

[25] "FinTA." [Online]. Available: https://pypi.org/project/finta/ .

[26] "Scikit-learn." [Online]. Available: https://scikit-learn.org.

[27] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," 2015.

[28] M. Cliche, "Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," *arXiv Prepr. arXiv1704.06125*, 2017.

[29] "TensorFlow." [Online]. Available: https://www.tensorflow.org/.

[30] "Keras: The Python Deep Learning library." [Online]. Available: https://keras.io/.