Stock Movement Classification from Twitter via Mogrifier based memory cells with Attention Mechanism

Tian Qin
Department of Statistics
University of Illinois at Urbana-Champaign
Champaign, IL, US
tianq2@illinois.edu

ABSTRACT

A growing number of financial market participants use the deep learning models to predict the future market movement via exploiting the news and social media messages. Memory cells like long short-term memory (LSTM) and gated recurrent unit (GRU), though play indispensable roles in sequence prediction, are criticized for lacking the interaction between inputs and previous context. The Mogrifier LSTM cell is designed to enhance such relationship. We employ this novel cell in our hybrid attention prediction network. The new structure enriches the communication between the previous message embedding and the current signal, hence a better learning process representation compared with the neural network without such mechanism. Experiment on real-world stock market data shows the Mogrifier based cell can outperform the venerable ones across different market sectors while still achieves lower prediction volatility.

KEYWORDS

Stock market prediction, Deep Learning, Natural language processing, LSTM, Mogrifier LSTM, GRU, Mogrifier GRU

1 Introduction

The stock movement prediction, as a branch of financial time series forecasting, is a highly empirical filed which tries to quantify the relationship between the market trend and the latent variables based on historical data. One fundamental premise of series forecasting is the efficient market theory (EMH) first proposed by [1], it categorized market efficiency into 'weak', 'semi-strong' and 'strong'. Any analysis based on historical prices will be effortless under the EMH. It still remains disputable whether market is efficient or not. One widely spread view is proposed by [2], it claims while individual stocks may somehow be efficient, the theory performs badly for the aggregate stock market.

The inefficiency for the market as a whole furnishes prediction models theoretical background. From single, linear statistical model such as auto regressive moving average (ARMA) to multiple sources, nonlinear deep neural network (DNN), the evolutionary of the prediction model reflects the evolving of the artificial intelligence. As pointed out by [3], a rising star in forecasting stock movement is the combination of deep learning models and natural language processing techniques to mimic the behavior of human learning process. Sequence model, one of the largest branches in the deep learning family has a proven track record in portraying the temporal nature of stock indexes and semantic sequences. Recurrent cell is the very first widely used mechanism in the sequence model, LSTM[12] and GRU[13] are later designed to tackle the gradient vanishing problem occurred by recurrent cell. Though being popular in handling the series data task such as stock movement prediction, these cells' performance are hampered by their additive building block, namely, different gate computations inside the cells. State vectors from different steps are combined through simple summation hence the gradient descent is incapable of updating of the learning system[4]. In its language processing branch, sequence model with above cells comes down with the brittleness of generalization[5]. [21] aims at alleviating this long haunted problem over the natural language processing community by adding a gating step before the LSTM cells, thus a contextualized representation vector of the current information flow is input into the memory unit rather than being thrown into the LSTM cell without any modification. This mechanism enriches the additive operation in the cells and hence closes the performance gap between LSTM and attention based models. One of the original incentives for this optimization is to intensify both the striking and weaken features of the current input. This makes the Mogrifier LSTM adept at detecting tiny changes in the temporal data, thus an ideal candidate for the stock movement prediction model. Given its practicability in processing temporal data like semantic sequences, we decide to adopt this novel cells together with attention mechanisms used in the state-of-the-art transformer models as the main tools to capture not only the sentiment lurking behind the tweets but also the temporal dependence among different trading days.

This paper's contribution is summarized as follows: 1) Word-level attention mechanism is leveraged to the Hybrid Attention Network (HAN) designed by [20]. Humans pay different attention to different words in one sentence. The contribution of different words to human emotions can be quantitatively determined by importing attention mechanism. 2) MOGRIFIER LSTM are considered in the prediction network. Performance comparison between MOGRIFIER LSTM and LSTM cell are implemented through multi-stage rolling window tests on different market sectors. 3) MOGRIFIER GRU, a parsimonious version of

MOGRIFIER LSTM, is further explored in our prediction framework.

The remainder of this paper is organized as follows: Theoretical background for related work is investigated in Section2. Section 3 introduces the proposed model for stock movement prediction in this study. Section 4 demonstrates the empirical results and analysis. Section 5 includes conclusion and potential future work.

2 Related Work

Many early forecasting models were proposed by economists and statisticians. These models focus on describing the statistical correlation between the target variable and historical variables. Auto-regressive (AR) and moving average (MA) models are two typical representatives. The AR model assumes the current return rate has a linear relationship with its past observations. The moving average model, on the other hand, presumes the dependent variable relies on the past white noise terms linearly. [6] proposed the ARMA model, this model captures the features of both the AR and MA model while still manages to reduce the number of parameters. The ARMA model conforms to the 'parsimonious rule' in statistics, thus the over-fitting problem can be significantly relieved. Based on ARMA, the ARIMA model was used to fit the non-stable time series. Furthermore, Seasonal ARIMA helps researchers to depict the periodicity of the economic data. The parameter estimations and significance test of the above models can all be achieved through the Box-Jenkins method [7]. The rate of return on financial assets is not only related to its past prices but also related to historical market variable data. The ARMAX model introduces exogenous variables so that the target variables can be better predicted [8].

In the past decades, statistical learning models and deep learning models have been introduced into the prediction task. These frameworks usually outperform the earlier ones. Unlike the auto-regressive models, classification problems have been incorporated into the statistical learning regime since its inception. Thus it can be used to determine the direction of stock market movement. [9] used CART and SVM with RBF kernel to classify the trend of the US stock market, empirical study supports the advantage of these nonlinear models. Two other common classification frameworks, XGBoost and LightGBM are also used in classification prediction. Both of them can be considered as the highly optimized extension of tree-based gradient boost machine. [10] studied the performance of tree-based models in stock price prediction and confirmed the better performance of nonlinear tree-based models. Deep learning models, on the contrast, have their own superiority compared with statistical learning models. A majority of them were originally designed to handle the temporal data. This type of model is usually called a sequence model. [11] introduced the prototype of the Recurrent Neural Network (RNN), one of the very first artificial neural network (ANN). The inner mechanism can combine the past information flow with the current new input and later send them into the next step. A final prediction will then be reproduced based on all information. However, while innovative, it is very hard to train the model parameters and the model suffers from the problem of gradient explosion and disappearance which leads to the invalidation of gradient descent methods. To tackle this problem, [12] proposed the LSTM structure. This smart design ensures the merge of past and current input without losing the efficiency of the gradient descend algorithm too much. [13] simplified the structure based on [12] while still obtaining similar new performance. There are many researches about using ANN to predict financial time series. [14, 15] use the concept of convolutional neural network (CNN) to build a price prediction system. These papers all support the view that ANN performs better than statistical learning models. Generally speaking, the deep learning model is efficient in learning nonlinear relationships between variables due to the larger amount of model parameters.

Non-historical technical data such as stock news and trend images give people a new way to do predictions. Information extracted from social media furnish analysts with new insights. Natural language processing techniques play an important role on making the model to better understand the language. [16] Innovatively extracts the weights of the shallow neural network as the vector representation of words. This representation method can better reflect the gap between words and words, [17] obtains word vectors through dimension reduction of the co-occurrence matrix while speeding up the process with the help of parallelization. Both of these models take the context of words into account, which is different from the previous practice of simply counting words frequencies as text vectors. On the other hand, [18] first proposed an attention mechanism, which allows the neural network to better understand the meaning of the input sentence by giving different weights to different words in the same sentence. This attention mechanism was later extended to fields other than natural language processing, [19] explored how to use this mechanism in multivariate time series. All approaches above will appear in our stock prediction classifiers.

3 Method

In this paper, stock movement prediction is a binary classification problem intends to forecast stock trend (UP , $DOWN$). Section 3.1 introduces the metrics used for our classification task. Section 3.2 shows the model details.

3.1 Preliminaries

Stock movement prediction is a binary classification problem. We use the vectorized twitter sentences as model input and get a label prediction as the output. Given a stock s at time t, its log return r_t can be calculated as:

$$r_t = log \frac{Closed\_Price_t}{Closed\_Price_{t-1}}, \quad (1)$$

We then label $l_{s,t}$ as 1 if $r_{t,s}$ is greater than or equal to 0, else -1.

Our task is to predict the label $l_{s,t}$ as accurate as possible based on the past N days tweets. We use cross entropy (CE) loss as

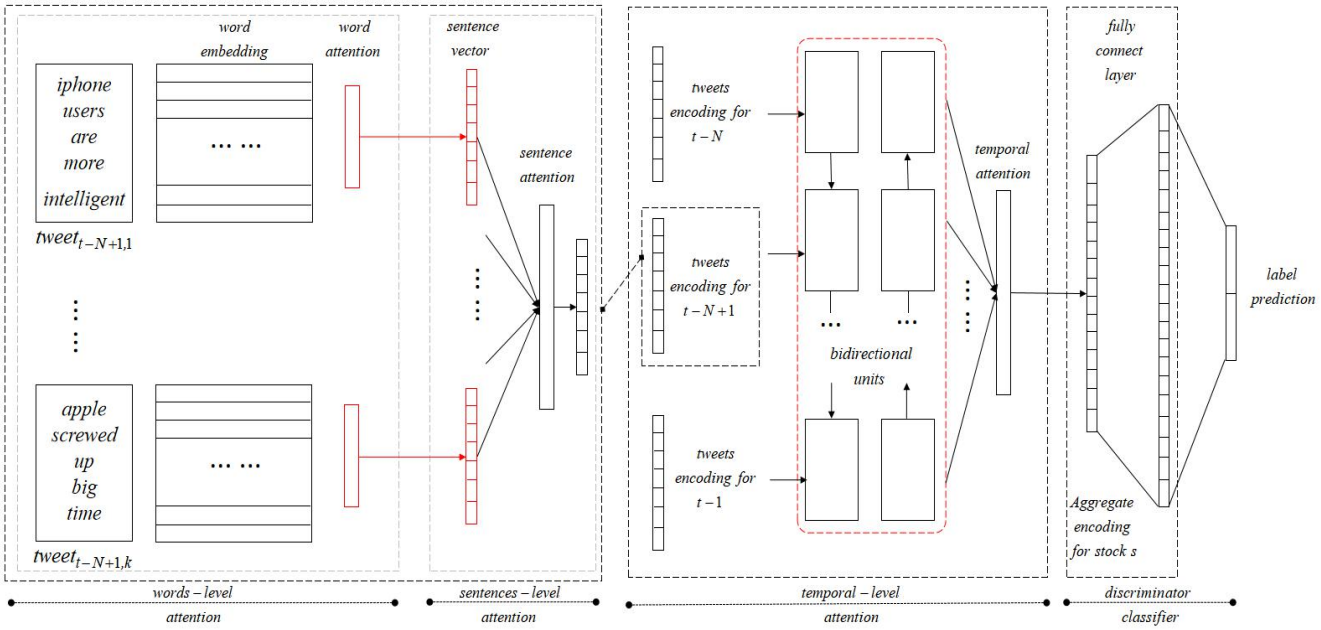

Figure 1: The framework of HAN with words-level attention. This prediction network contains three attention mechanisms: words-level, sentences-level, temporal-level. Each level is used to capture different information. Red parts represent this paper's modification. Different types of memory cells can be used as bidirectional unit in the box with red dashed line.

performance measurement to train the neural network. Under binary classification problem, the cross entropy loss is as follows:

$$CE = -\sum_{s=1}^{S} l_{s,t} \log(\hat{p}_{s,t}) - (1-l_{s,t}) \log(1-\hat{p}_{s,t}) \quad (2)$$

where $\hat{p}_{s,t}$ is the prediction probability for $l_{s,t}$ and S is the number of stocks.

We use the classification accuracy when report the model's final performance since it is well understood by human compared with cross entropy loss.

3.2 Model Structure

We designed our neural network based on the Hybrid Attention Networks (HAN). We first add a words-level attention layer and use the Mogrifier LSTM cell as the major component in the temporal-level attention layer.

For any stock s at time t with a given window size N, we first collect all related tweets generated from time t-N to t-1. We denote these twitter as $tweets_{t-N}, tweets_{t-N+1}, \cdots, tweets_{t-1}$. $tweets_{t-k}$ is the set of all tweets generated on day t-k. The i^{th} tweet on day t-k is denoted as $tweets_{t-k,i}$. Suppose the task is to predict the movement of stock s at time t. For a given sentence $tweets_{t-k,i}$, the word embedding layer converts each word in the sentence into a numerical vector which can later be used by the model. We only select first K words as the representation of the whole sentence. This is a reasonable simplification considering a normal tweet is usually short and the first K words have revealed enough information. All these word vectors then go through the word attention layer as follows:

$$u_{t-k,i,j} = sigmoid(U_{word} w_{t-k,i,j} + b_{word}) \quad (3)$$

$$\alpha_{t-k,i,j} = soft\,max(u_{t-k,i,j}) \quad (4)$$

$$s_{t-k,i} = \sum_{j} \alpha_{t-k,i,j} w_{t-k,i,j} \quad (5)$$

where $w_{t-k,i,j}$ is the j^{th} word vector in i^{th} tweet produced on day t-k, $\alpha_{t-k,i,j}$ is the weight for word $w_{t-k,i,j}$ and $s_{t-k,i}$ is the vector encoding of i^{th} tweet on day t-k.

Subsequently to word level attention layer, we repeat the same operation to all tweets in $tweets_{t-k}$. Note we only consider the first S sentences. Again, the randomly selected S sentences unveils sufficient information. Next, the sentence attention layer will give different weights to different messages as follows:

$$v_{t-k,i} = sigmoid(V_{sentences} s_{t-k,i} + b_{sentence}) \quad (6)$$

$$\beta_{t-k,i} = soft\,max(v_{t-k,i}) \quad (7)$$

$$d_{t-k} = \sum_{i} \beta_{t-k,i} s_{t-k,i} \quad (8)$$

where $\beta_{t-k,i}$ is the weight for tweet $s_{t-k,i}$ and d_{t-k} is the vector encoding for all tweets from day t-N to day t-1. These tweets by all means have different influences on market sentiments due to

variation of user account features such as the number of followers. Thus a useful prediction network should have the ability to distinguish the impressible tweets from the rest.

Next we choose the Bi-directional MOGRIFIER LSTM layer to capture the temporal dependency. This novel LSTM cell is proposed by [21]. Two inputs x and h_0 modulate one another in an alternating fashion before being put into the traditional LSTM or GRU thus adds a more robust interaction between previous information and current inputs. Figure 2 illustrates the interaction between x and h_0.

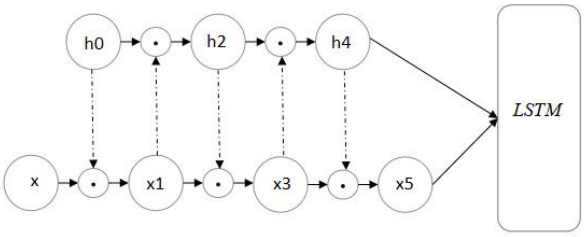

Figure 2: Structure of MOGRIFIER LSTM where x is the input and h0 is the hidden state from the previous cell.

This extended version of LSTM achieves state-of-the-art results on many sequential language modeling tasks. Thus it is an ideal candidate to handle the temporal data. Specifically, in our framework, the MOGRIFIER mechanism is as follows:

$$d_{t-k}^{i} = 2sigmoid(Qh_{prev}^{i-1}) \circ d_{t-k}^{i-2} \quad i \in [1,3,5] \tag{9}$$

$$h_{prev}^{i} = 2sigmoid(Rd_{t-k}^{i-1}) \circ h_{prev}^{i-2} \quad i \in [2,4] \tag{10}$$

where h_{prev}^{0} is the hidden state from last MOGRIFIER LSTM cell and d_{t-k}^{-1} is the tweets encoding at time *t-k* produced by (8).

These two vectors are then input into the original LSTM cells, which later output hidden states h_{t-k+1}^{0} and cell states c_{t-k+1} for the round *t-k+1*. An LSTM cell is shown in Figure 3.

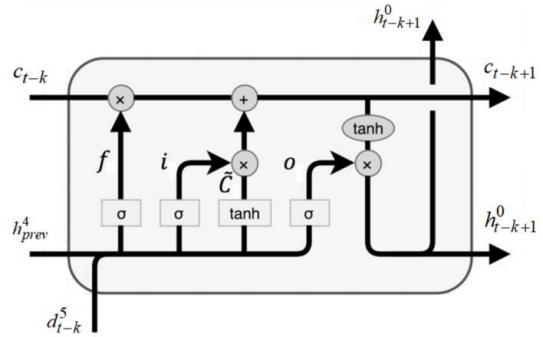

Figure 3: An LSTM cell at time *t-k*.

To simplify the notation, the above procedure are denoted as:

$$c_{t-k+1}, h_{t-k+1}^{0} = LSTM(c_{t-k}, h_{prev}^{4}, d_{t-k}^{5}) \tag{11}$$

Since we use the bidirectional units rather than the single one, we concatenate the hidden state vectors to get the vector as:

$$\vec{c}_{t-k}, \vec{h}_{t-k}^{0} = MOGRIFIERLSTM(\vec{c}_{t-k-1}, \vec{h}_{t-k-1}^{0}, d_{t-k-1}) \tag{12}$$

$$\overleftarrow{c}_{t-k}, \overleftarrow{h}_{t-k}^{0} = MOGRIFIERLSTM(\overleftarrow{c}_{t-k+1}, \overleftarrow{h}_{t-k+1}^{0}, d_{t-k+1}) \tag{13}$$

$$h_{t-k}^{c} = [\vec{h}_{t-k}^{0}, \overleftarrow{h}_{t-k}^{0}] \tag{14}$$

The author [20] also mentioned the feasibility of MOGRIFIER GRU. We hereby use Figure 4 to illustrates the GRU :

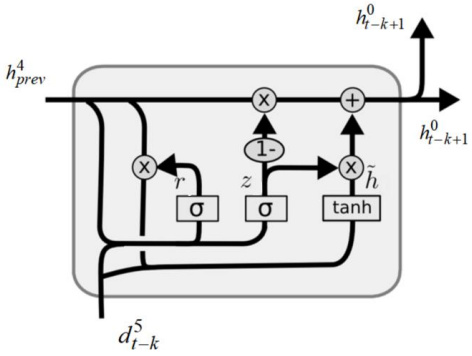

Figure 4: A GRU cell at time *t-k*.

The above procedure can be denoted as:

$$h_{t-k+1}^{0} = GRU(h_{prev}^{4}, d_{t-k}^{5}) \tag{15}$$

Figure 5 illustrates the difference between MOGRIFIER LSTM and MOGRIFIER GRU. In the latter cell, we don't have cell state *C*.

Similar to the MOGRIFIER LSTM, (25) to (27) give us the bidirectional version:

$$\vec{h}_{t-k}^{0} = MOGRIFIERGRU(\vec{h}_{t-k-1}^{0}, d_{t-k-1}) \tag{16}$$

$$\overleftarrow{h}_{t-k}^{0} = MOGRIFIERGRU(\overleftarrow{h}_{t-k+1}^{0}, d_{t-k+1}) \tag{17}$$

$$h_{t-k}^{c} = [\vec{h}_{t-k}^{0}, \overleftarrow{h}_{t-k}^{0}] \tag{18}$$

The temporal attention mechanism is employed to detect the tweets' sentiment trend along the window period. This is based on

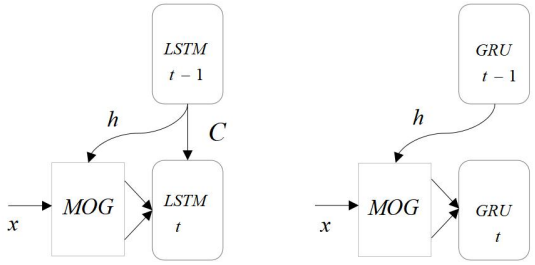

Figure 5: A graph illustration for the difference between MOGRIFIER LSTM and MOGRIFIER GRU

the assumption that though both past news and current news have impacts on stock movement, the intensity is different.

$$r_{t-k} = sigmoid(R_{temporal} h_{t-k}^{c} + b_{temporal}) \tag{19}$$

$$\gamma_{t-k} = soft\,max(r_{t-k}) \tag{20}$$

$$s_t = \sum_k \gamma_{t-k} r_{t-k} \tag{21}$$

where s_t includes all integrated information extracted by different attention mechanisms.

A fully connect layer then takes the synthesized s_t as input and finally makes a prediction of the stock movement direction.

4 Experiments

In this section, we evaluate the modified HAN based on different cells. GRU cell and LSTM cell are chosen as the baseline model and they are used to compare with the MOGRIFIER LSTM and MOGRIFIER GRU.

4.1 DataSets

This paper uses the data set provided by [22]. There are totally 87 stocks in the whole set which can be further divided into 9 different sections. We first choose the tweets records from 2014/01/01 to 2015/08/01 as the training set. To prevent the over-fitting problem, we do cross validation on the period from 2015/08/01 to 2015/10/01. Finally, we test the model performance from 2015/10/02 to 2016/01/01. To ensure the model's robustness, rolling window period test will be employed on nine different sectors. 4.2 will provide an overview on this method. There are 178239 tweets in total. Figure 6 show the distribution of return of different stocks from 2014/01/01 to 2016/01/01.

There exists label imbalanced problem in our data set. If we transfer the return into -1 and 1 without trimming, then the label will be imbalanced and the model prediction performance will be biased toward one side. We thus set -0.0046 as the lower bound and 0.0052 as the upper bound. Any value falls into this range will be discarded. For those data larger than 0.0052, we set them as 1 and for those lower than -0.0046, we set them as -1.After the data trimming, there are 13819 positive labels and 13769 negative labels, the ratio of the amount of two labels is 1.0036. This data process thus eliminates the potential data bias.

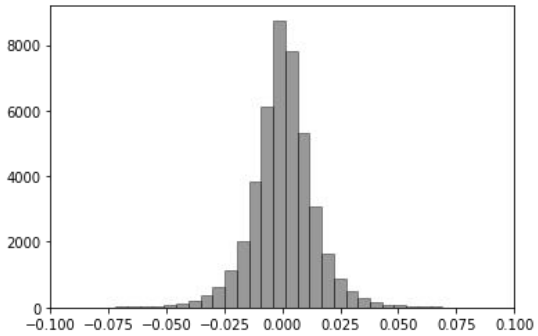

Figure 6: Distribution for return of stocks

4.2 Model Setup

We consider first 30 words appear in one tweet. Any word after that will be ignored. We also pick 40 tweets for a single stock on one day. This number is a balance between the total number of stocks and information completeness. The window size N is set up as 5. It is enough for the stock market to incorporate the information into its price in 5 days even though the market is inefficient. We use the glove.twitter.27b.50d as the word embedding. Pre-training of these word vectors is performed on aggregated global word-word co-occurence from Twitter. Thus it is a good start point for training the whole prediction network.

We choose Adaptive Moment Estimation (Adam) as our optimizer by setting exponential decay rates for the moment estimates $\beta_1 = 0.99$ and $\beta_2 = 0.999$ with $\varepsilon = 1\times10^{-6}$ .The learning rate is chosen as 1×10^{-3} . Early stopping rule with threshold equals 1 is adopted to avoid over-fitting problem. The optimizer will stop working if the cross entropy loss of the validation data set in current epoch is larger than the previous epoch. Finally, the batch size is chosen to be 128 so that model can be running on a single machine.

To better evaluate the model under different scenarios, we adopt the rolling window test on different market sectors. We set the first round training period as [2014-01-01, 2015-08-01), validation period as [2015-08-01, 2015-10-01), test period as [2015-10-01,2016-01-01). Then we shift all periods right by 1 to start the second round. After nine rounds, we calculate the average test set prediction accuracy for different market sectors as the final result.

Our model is trained on a NVIDIA RTX 2070 Graphics Processing Unit.

4.3 Experimental Result

We compare the LSTM based neural network with the Mogrifier one. Since the GRU has almost the same function as the LSTM, we also evaluate the difference between GRU model and Mogrifier GRU. An overview of the comparison are shown in Figure 7.

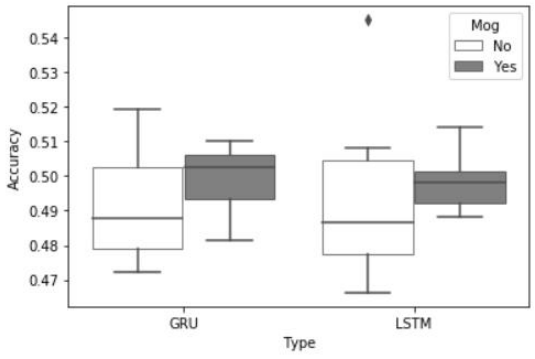

Figure 7: An overview of the rolling window test accuracy for different models

Figure 7 suggests LSTM and GRU cells with MOGRIFIER achieve higher prediction accuracy while still maintaining overall lower standard deviation. The average accuracy for MOGRIFIER based GRU is 0.499 with a standard deviation of 0.010 while the plain GRU is 0.491 with a standard deviation of 0.016. Considering the stock movement prediction is a hard task with tiny performance improvement can lead to considerable profit gain, this improvement is conceivable. On the other hand, the average accuracy for MOGRIFIER based LSTM is 0.498 and the plain LSTM is 0.493. The standard deviation for prediction accuracy of MOGRIFIER LSTM is 0.0078, a significant improvement compared with the original LSTM whose standard deviation is 0.0235. Table 1 exhibits the performance comparison between 2 mechanisms.

	Average Accuracy	Standard Deviation
GRU	0.491	0.016
MOGRIFIER GRU	0.499	0.010
Improvement	1.63%	37.5%
LSTM	0.493	0.024
MOGRIFIER LSTM	0.498	0.008
Improvement	1.01%	66.7%

Table 1: Prediction performance for four models with or without MOGRIFIER

By conditioning the input embedding on the recurrent state, the neural network exploits the information with higher efficiency. In the traditional LSTM cell, though input and previous hidden state are used to calculate different gates, they don't have a direct interactive effect on each other. Mogrifier based cells make up for the relationship that LSTM fails to seize.

5 Conclusions

This paper has demonstrated the Mogrifier cells together with attention mechanism outperform the plain LSTM in the stock movement prediction task by importing additional interaction between the input vector and hidden state from the past. It optimizes the simulation of learning process by amplifying both the salient and minor turbulence in the input layers through a modulated fashion. This enhancement makes the model to be more sensitive to tiny sentimental changes in the tweets and avoid the potential of being deceived by seemingly calm comments. Attention mechanism is also benefited from this better representation hence can determine the temporal dependency in a more precise way. These altogether contributes to the lower standard deviation of prediction accuracy. Our robust rolling window test over different market sectors supports this speculation.

In the future, we plan to use Mogrifier LSTM to extract the stock information from the combination of technical data and social media messages. A multi-source network may provide more insightful investment suggestions.

REFERENCES

[1] Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The journal of Finance, 25(2), 383-417.

[2] Jung, J., & Shiller, R. J. (2005). Samuelson's dictum and the stock market. Economic Inquiry, 43(2), 221-228.

[3] A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing

[4] Wu, Y., Zhang, S., Zhang, Y., Bengio, Y., & Salakhutdinov, R. R. (2016). On multiplicative integration with recurrent neural networks. In *Advances in neural information processing systems* (pp. 2856-2864).

[5] Jia, R., & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

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

[7] Kihoro, J., Otieno, R. O., & Wafula, C. (2004). Seasonal time series forecasting: A comparative study of ARIMA and ANN models.

[8] Baillie, R. T. (1980). Predictions from ARMAX models. *Journal of Econometrics*, *12*(3), 365-374.

[9] Golmohammadi, K., Zaiane, O. R., & Díaz, D. (2014, November). Detecting stock market manipulation using supervised learning algorithms. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 435-441). IEEE.

[10] Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552-567.

[11] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (No. ICS-8506). California Univ San Diego La Jolla Inst for Cognitive Science.

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[13] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015, June). Gated feedback recurrent neural networks. In *International conference on machine learning* (pp. 2067-2075).

[14] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.

[15] Sim, H. S., Kim, H. I., & Ahn, J. J. (2019). Is deep learning for image recognition applicable to stock market prediction?. *Complexity*, *2019*.

[16] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[17] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[18] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[19] Shih, S. Y., Sun, F. K., & Lee, H. Y. (2019). Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, *108*(8-9), 1421-1441.

[20] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. Y. (2018, February). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 261-269).

[21] Melis, G., Kočiský, T., & Blunsom, P. (2019). Mogrifier lstm. *arXiv preprint arXiv:1909.01792*.

[22] Xu, Y., & Cohen, S. B. (2018, July). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1970-1979).