# KIER DISCUSSION PAPER SERIES

## KYOTO INSTITUTE OF ECONOMIC RESEARCH

Discussion Paper No.1082

"Efficient Market Hypothesis Test with Stock Tweets and Natural Language Processing Models"

Bolin Mao, Chenhui Chu, Yuta Nakashima, Hajime Nagahara

September 2022


KYOTO UNIVERSITY

KYOTO, JAPAN

# Efficient Market Hypothesis Test with Stock Tweets and Natural Language Processing Models

Bolin Mao[*], Chenhui Chu[†], Yuta Nakashima[‡], Hajime Nagahara[§]

**Abstract**

The efficient market hypothesis (EMH) plays a fundamental role in modern financial theory. Previous empirical studies have tested the weak and semi-strong forms of EMH with typical financial data, such as historical stock prices and annual earnings. However, few tests have been extended to include alternative data such as tweets. In this study, we use 1) two stock tweet datasets that have different features and 2) nine natural language processing (NLP)-based deep learning models to test the semi-strong form EMH in the United States stock market. None of our experimental results show that stock tweets with NLP-based models can prominently improve the daily stock price prediction accuracy compared with random guesses. Our experiment provides evidence that the semi-strong form of EMH holds in the United States stock market on a daily basis when considering stock tweet information with the NLP-based models.

*Keywords:* Efficient Market Hypothesis Test, Daily Stock Price Prediction, Stock Tweet, Natural Language Processing

*JEL classification:* C4; C5; G1

[*]Corresponding author. Kyoto Institute of Economic Research, Kyoto University. Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8317, Japan. mao.bolin.5m@kyoto-u.ac.jp.

[†]Graduate School of Informatics, Kyoto University. Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan.

[‡]Institute for Datability Science, Osaka University. Techno-alliance bldg. C503, 2-8, Yamadaoka, Suita, Osaka, 565-0871, Japan.

[§]Institute for Datability Science, Osaka University. Techno-alliance bldg. C503, 2-8, Yamadaoka, Suita, Osaka, 565-0871, Japan.

# 1 Introduction

The efficient market hypothesis (EMH), formalized by Fama (1970), states that the financial market is efficient; that is, security prices at any time reflect all available information. It provides the theoretical foundation for modern financial economics and implies the unpredictability of financial asset prices and returns. The EMH is usually tested in three different forms: the *weak* form, which only considers the information contained in historical prices; the *semi-strong* form, which considers further publicly available information, such as announcements of annual earnings and stock splits; and the *strong* form, which also considers private information that has not yet been revealed to the public.

In an attempt to test the EMH, many empirical studies have utilized typical economic and financial data, such as historical prices, annual earnings, and monetary policy announcements. With the development of information technologies, the financial market, like other industries, has also advanced into the *big data* era where not only the size but also the type of market-related data have been widely extended. Among the various types of non-typical market-related data, Twitter[1] has been considered to be a promising information source for decision-making with regards to financial market investments. This is because Twitter is a channel where financial market participants can express their opinions about economic and company-wide events, which influence the financial market. For example, a well-known expert's opinions may be widely accepted by individual investors and cause buy/sell actions with an unignorable trading volume, which can therefore change security prices.

Moreover, a tweet message has abundant attributes, called *meta-information*. This includes the number of Twitter followers. Meta-information is also helpful in decision-making in the financial market. For example, famous financial analysts' tweets should have a broader impact than those of ordinary people. Therefore, it is worth investigating whether a group of famous Twitter users can improve tweet-based stock price predictions. A dataset that focuses on tweets published by famous and influential Twitter users is desirable.

In addition to the data aspect, recent fast-developing deep learning (DL) technologies provide potential approaches to exploit information from unstructured data, including tweets. Although DL is often criticized for its black-box nature compared to most other statistical and econometric

[1] https://twitter.com/

approaches, its outstanding prediction capability in many complicated tasks — such as computer vision and natural language processing (NLP), is expected to assist decision-making in the financial market.

In this study, we test the semi-strong form of the EMH in the scenario where tweet data containing hundreds of thousands of tweet messages and the number of followers of the corresponding Twitter users, is available for daily stock price prediction. More specifically, we use two tweet datasets containing entire *tweet objects*, each of which includes the message content and the number of followers of the Twitter user (i.e., the author of the tweet message). One of these datasets is available as a part of *StockNet* (Xu and Cohen, 2018), and we collected the other, called *BigName*, by ourselves. The BigName dataset focuses on tweets authored by famous Twitter users in the financial market, whereas the StockNet dataset was collected without noting the Twitter users.

We propose eight DL models with NLP technologies, ranging from relatively old to state-of-the-art. With another NLP-based DL benchmark model StockNet, we used the nine models in total to capture the relationship between *stock tweets* and the closing price movement (either increase or decrease) of 81 representative stocks in the United States stock market. If any NLP-based DL model shows a significant improvement in the prediction performance compared with random guesses, we can say that the model exploits the latent relationship between prices and stock tweets. This is not accounted for in the market — suggesting that the market may be inefficient. Otherwise, we cannot reject that the EMH holds for stock tweets and daily price change predictions in the United States stock market.

Our experiment shows that there is no large enough improvement in stock prediction accuracy for any of the models. Consequently, the result supports the notion that the United States stock market is efficient in terms of the semi-strong form of EMH for the daily basis stock price change prediction.

Our contributions are threefold:

- We collected the BigName dataset. To the best of our knowledge, this is the first study to focus on tweets authored by famous Twitter users in the stock market.
- We propose eight DL models for stock price prediction.
- Our experimental results support the semi-strong form EMH.

# 2 Related Work

## 2.1 Efficient Market Hypothesis

The EMH argues that financial asset prices reflect all available information. This is because investors will perform buy or sell actions immediately after company-related information becomes publicly available. Therefore, trading actions cause the asset's price to approach a new equivalent level within a short time period. Consequently, it is impossible to predict asset returns and constantly beat the market.

Many researchers and practitioners have supported or challenged the EMH by providing positive or negative evidence. Early works attempted to find the unpredictability of financial assets, for example, Ball and Brown (1968); Fama et al. (1969); and other more recent later works additionally built some return predictors for the financial market — such as Rosenberg et al. (1985); Campbell and Shiller (1988); Jegadeesh and Titman (1993). Currently, the EMH has become the fundamental theoretical hypothesis among academics.

## 2.2 Stock Price Prediction with DL models

Many DL models for stock price prediction use only numerical data, such as historical prices, technical indicators, and economic indexes (Atsalakis and Valavanis, 2009; Ballings et al., 2015; Gu et al., 2020; Henrique et al., 2019).

Meanwhile, with the development of NLP technologies, some studies employed NLP-based models to further utilize unstructured text data — such as news and tweets (Ding et al., 2014, 2015, 2016, 2019; Li and Shah, 2017; Duan et al., 2018; Chen et al., 2019b). One of the most state-of-the-art language models, BERT (Devlin et al., 2018), has demonstrated its capability for semantic analysis in many general NLP tasks, such as translation and question answering. Previous stock prediction works also leveraged BERT in their models. Chen et al. (2019a) proposed a BERT-based hierarchical aggregation model to summarize financial news for foreign exchange movement prediction. Yang et al. (2019) used BERT for semantics embedding of 30 search terms related to the financial and economic attitudes revealed by search (Da et al., 2015), known as FEARS. They then combined the embedding with the search volume indices and price changes and applied a self-attention layer for the S&P 500 Index return prediction.

## 2.3 Stock Price Prediction with Tweets

Some pioneering works employed tweets as an alternative input to the NLP-based model for stock price prediction. Bar-Haim et al. (2011) proposed a framework to find expert investors according to their tweet posts and historical stock movement. Then, they used expert investors' tweets as the basis for future stock price movement prediction. Si et al. (2013) first implemented a continuous Dirichlet process mixture model to learn topics from tweets on a daily basis. They derived the sentiment for each topic according to the opinion word distribution and consequently built a sentiment time series. They then regressed the stock market index on the sentiment time series. In addition, Si et al. (2014) proposed the use of a graph to encode relationships among stocks, where each stock is treated as a node and their relationships are identified as edges based on the topics. They showed that the graph could improve stock price predictions when regressing the stock price time series based on the topic-sentiment time series. Xu and Cohen (2018) proposed a generative model, *StockNet*, based on the variational autoencoder (Kingma and Welling, 2014). The StockNet model exploits tweets and historical prices for daily stock price predictions. Liu et al. (2019) proposed a model based on transformers (Vaswani et al., 2017) and the capsule network. These are used for extracting semantic features of stock tweets and capturing relationships among them, for stock price movement prediction.

However, looking back on the previous works on stock price prediction with tweets (Bar-Haim et al., 2011; Si et al., 2013, 2014; Xu and Cohen, 2018; Liu et al., 2019), none of them answered the question: *Can we treat all tweets by different users equally?* Previous studies only considered the content of tweet messages and ignored Twitter users' identity or attributes, which can provide additional signals to the importance of tweet messages.

## 2.4 Stock Tweet Datasets

Due to Twitter policy, most stock tweet datasets are no longer available, even though they had been available before. Nevertheless, two stock tweet datasets were still available at the start of our experiment. One of them, *CHRNN*, introduced by Wu et al. (2018), includes tweets from January 2017 to November 2017, and is related to 47 companies from Standard & Poor's 500 candidates. The other, StockNet, introduced by Xu and Cohen (2018), includes tweets from January 2014 to December 2015, covering 87 companies with the largest market capitalization in 9 sectors in

the United States stock market.[2] The CHRNN dataset contains only tweet messages and their published times. In contrast, the StockNet dataset consists of the complete *tweet objects*—that is, tweet messages and the associated *meta-information* available from Twitter.

# 3 Dataset and Task

Our task is to predict the movement of the upcoming daily closing price compared to the previous business day's (i.e., increase or decrease) by using stock tweets. In this section, we first show the concept of a stock tweet and then introduce the process of collecting the BigName dataset, comparing the characteristics of the BigName and StockNet datasets. Finally, we provide a formal definition of the task.

## 3.1 Stock Tweet

With the growing use of social networking services — such as Twitter and Facebook — everyone can broadcast, for example, what they think. Such messages may have a significant impact on financial circumstances, including stock markets. In this work, we chose Twitter, one of the most popular social networking services, as our testbed to test the EMH. The following example illustrates a typical tweet message that mentions the stock market:

```
Stock Futures Bounce, Now lets Trade This Action!  $WMT, $CSCO, $BABA, $CGC
&amp; More In Play...  https://t.co/NCLx1t3h8K
```

Formally, we denote the text in a tweet as a *tweet message* that may contain several types of content. A typical tweet message mentioning the stock market includes four types of content:

**Theme** is the primary and most informative sentence(s) in a tweet message. `Stock Futures Bounce, Now lets Trade This Action!` is the theme in the aforementioned tweet example.

**Ticker Symbol** is a series of letters assigned to a security for the trading purposes. To discriminate from other components in a tweet message, a ticker symbol is affixed with a `$` mark at the beginning, such as `$WMT` and `$CSCO`.

**Emoji** can give additional cues on the author's attitude, and it is encoded in Unicode format. A typical example is given by `\u263A`, which is a smiling face.

[2]They included 88 companies in the experiment in total, but one company did not have any tweet data.

**URL** provides further relevant information on the tweet message, but not directly through text, and therefore it is likely to become noise when it is fed into the model. `https://t.co/NCLx1t3h8K` is a URL in the above tweet example.

Moreover, a tweet message is enclosed in a *tweet object*, which also contains many other attributes associated with the message, for example, the author's name and number of followers. We collectively call these attributes *meta-information*. A brief illustration of a tweet object is shown in Figure 1.

(Insert Figure 1).

As mentioned before, because the CHRNN dataset only consists of tweet messages and their published times, meta-information cannot be obtained from it. Although the StockNet dataset consists of tweet objects, the collected tweets appear to be sampled without being aware of the author's identity. However, tweets by users with a smaller number of followers may not have a significant impact on the market and thus serve as noise for the prediction.

To incorporate the number of followers of Twitter users as an additional signal for the EMH test (something which is not considered in the *StockNet* dataset), we created the BigName dataset, which consists of tweets authored by well-known Twitter users who are supposedly working in the financial market or related sectors.

## 3.2 Building the BigName Dataset

We collected tweets using the Twitter API v1.1. We used the following method to create our *BigName* dataset:

The first stage of building the BigName dataset involved determining the candidate companies and identifying a set of well-known financial market-related Twitter accounts. To make a comparison with as well as to reuse the StockNet dataset, we chose the same list of companies the StockNet dataset contains as our candidate companies. There are 87 companies in the United States that have the largest market capitalization in their corresponding sectors. To create a set of Twitter accounts, we took the Twitter accounts in the StockNet dataset that had no less than 10k followers into consideration. Then, we added some other famous Twitter accounts at the suggestion

of online reports by Forbes[3] and Marketwatch.[4] Consequently, the total number of Twitter accounts in our set was 1,042.

The second stage used the Twitter Timeline API to collect tweets authored by Twitter accounts in the set, and to identify the tweets related to our candidate companies. More specifically:

**Querying** Our query to Twitter Timeline API retrieves tweets by each account in our set.

**Filtering** We use a regular expression to identify the candidate company-related stock tweets by ticker symbols.

**Repeating** We repeat the querying and filtering process a few times within around three months since the API only allows us to fetch the latest 3,200 tweets at most, from the time of query.

We downloaded the historical daily stock prices of companies on our list from *Yahoo! Finance*.[5]

Because the published time of a tweet object and the recorded time of a stock price are originally in the UTC and EST time zones, respectively, we converted the published time to the EST time zone.

For each tweet message, we used a regular expression to filter out URL strings, as they can be hardly comprehensible and thus are likely to become noise in the prediction.

Two consecutive closing prices sometimes remain unchanged; thus, we formulated our task as a binary classification problem, and we discarded the samples whose prices remained unchanged. Consequently, 0.58% of the samples in the BigName dataset and 1.03% in the StockNet dataset were dropped, and the ratios of the *increase* samples in these datasets became 48.55% and 48.41% for BigName and StockNet, respectively.

### 3.3 Dataset Characteristics

Because multiple tweets can be posted on the same day, and since our task is to predict stock closing price movements on a daily basis, we define a batch of tweets that have the same prediction target as a *sample*. A specific example is shown in Figure 2.

(Insert Figure 2).

[3]https://www.forbes.com/sites/alapshah/2017/11/16/the-100-best-twitter-accounts-for-finance/?sh=dd89b9d7ea0a

[4]https://www.marketwatch.com/story/finance-twitter-the-50-most-important-people-for-investors-to-follow-2018-12-13

[5]https://finance.yahoo.com/

### 3.3.1 Dataset Scale

Since 6 out of 87 companies' historical stock prices cannot be completely fetched by Yahoo! Finance, we excluded them from the BigName dataset.

The size of BigName is 124,357 in terms of the number of tweets; and 32,408 in terms of the number of samples. Each daily price movement is associated with 3.84 stock tweets on average. In contrast, the size of the StockNet dataset is 99,919 in terms of the number of tweets, and 18,996 in terms of the number of samples for the same 81 stocks. Each daily price movement is associated with 5.26 stock tweet messages on average.

### 3.3.2 Follower Number Distribution

Figure 3 shows the number of tweets vs. the author's follower number. It is clear that tweets in the BigName dataset are mostly published by the accounts whose follower numbers are in the [10k,100k] range; while tweets in the StockNet dataset are published by the accounts whose follower numbers are in the [10,10k] range.[6]

(Insert Figure 3).

Figure 4 shows the number of publishers vs. the publisher's follower number. We can see that most publishers' follower numbers in StockNet range from 0 to 1k, whereas those of BigName range from 10k to 100k. Moreover, the total publisher number in StockNet is significantly greater than BigName; that is, StockNet collects tweets from a large group of Twitter users without being aware of the user's follower number. In contrast, BigName collects tweets from a small group of Twitter users who have many followers.

(Insert Figure 4).

### 3.3.3 Chronological Distribution

The chronological distributions of the tweet objects and task samples are shown in Figure 5. Because StockNet only collected stock tweets in 2014 and 2015, we plotted them using dots in the figure.

(Insert Figure 5).

[6]Because the StockNet dataset was collected at an earlier point in time, the publisher's follower number may have changed. Thus, accounts in the StockNet dataset have a greater or lesser follower number than the same accounts in BigName and *vice versa*.

### 3.3.4 Company Distribution

Figure 6 and Figure 7 show the distributions of the number of companies with respect to the number of tweet objects and samples. From the two figures, it can be noted that very few companies have many tweets.

(Insert Figure 6 and Figure 7).

## 3.4 Prediction Task

To test stock price predictability —that is, the implication of the stock market's inefficiency — our task is formulated on top of StockNet (Xu and Cohen, 2018), and we use the follower number as an additional input for each tweet message. More specifically, we formulate stock price movement prediction as a binary classification task, where the output $y$ is whether the closing price of stock $s$ increases or decreases compared to that of the previous business day (i.e., $y \in \{\texttt{increase}, \texttt{decrease}\}$). The input for this prediction is $T = \{(t_i, n_i) \mid i = 1, \ldots, I\}$, which is the set of pairs of stock tweet messages $t_i$, with the stock symbol associated with $s$ and the publisher's follower number $n_i$, tweeted between the closing time of the last business day and the closing time of the current day. This is shown in Figure 8, where $I$ is the number of tweet messages in this period. Given this, our goal is to learn mapping $f$ from $T$ to $y$; that is,

$$y = f(T), \tag{1}$$

from the dataset collected. Note that the input of StockNet (Xu and Cohen, 2018) contains the set of tweet messages, $\{t_i \mid i = 1, \ldots, I\}$, but without $n_i$'s.

(Insert Figure 8).

# 4 Model

Figure 9 shows our basic pipeline for the stock price prediction task, which consists of *word representation*, *follower number representation*, *tweet message representation*, and *prediction* modules. We prepared several implementations for each module and constructed the models by combining them all for results comparison. They ranged from old-fashioned to state-of-the-art.

(Insert Figure 9).

## 4.1 Word Representation

The input $T = \{(t_i, n_i)|i = 1, \ldots, I\}$ of our pipeline contains $I$ tweet messages $t_i$, each of which is a sequence of words. To represent these words for later use, following the convention, we first use a tokenizer to normalize the word sequence and then compute the word embedding. More specifically, tweet message $t_i$ passes through a tokenizer. The tokenizer converts a word sequence into a sequence $t'_i = [t_{i1}, \ldots, t_{iM}]$, where $M$ is the number of tokens in $t'_i$ ($M$ differs for $i$, but we omit subscript $i$ for notation simplicity). The $m$-th token, $t_{im}$, is then converted into a word representation, which forms a sequence $\mathbf{e}_i$ of word representations —that is, $\mathbf{e}_i = [\mathbf{e}_{i1}, \ldots, \mathbf{e}_{iM}]$, where $\mathbf{e}_{im} \in R^{D_e}$. Here, we employed two combinations of tokenizer and word embedding, WordPiece+BERT and NLTK+GloVe+BiGRU, for word representation.

### 4.1.1 WordPiece+BERT

One of the state-of-the-art word embedding method is BERT (Devlin et al., 2018), which is a transformer-based approach. Here, we used a popular pre-trained model (`bert-base-uncased`) by Hugging Face.[7] It has 12 transformer layers, 12 attention heads, and 768-dimensional hidden states (i.e., $D_e = 768$). This pre-trained model uses *WordPiece* as a tokenizer.

### 4.1.2 NLTK+GloVe+BiGRU

The relatively old-fashioned method is a combination of the NLTK tokenizer,[8] GloVe (Pennington et al., 2014) for word embedding and BiGRU for contextualization. We first apply the NLTK *TweetTokenizer* and then use *GloVe* and BiGRU layers to obtain the sequence $\mathbf{e}_i$ of word representations. The details of our implementation are illustrated in Figure 10.

(Insert Figure 10).

## 4.2 Follower Number Representation

We represent the follower number in a $K$-dimensional vector $v_i$, each of which is either 0 or 1 and corresponds to one of the binary digits of the follower number; i.e., the $\kappa$-th element of $v_i$ is the $\kappa$-th binary digit of the follower number. The follower number is truncated to $2^K - 1$ if the number of followers is larger than $2^K$.

[7] https://huggingface.co/transformers/index.html

[8] https://nlp.stanford.edu/projects/glove/

### 4.3 Tweet Message Representation

Typically, a classifier (or our prediction module) takes a fixed-length vector as the input. Therefore, we reduce $\{\mathbf{e}_i\}_i$ into a single vector. Intuitively, the importance of words differs depending on the content, and the same applies to tweets in $T$. We adopt two methods for this reduction: one method is simply average pooling, computed over word representations and then tweets in $T$, and the other method uses hierarchical attention layers.

#### 4.3.1 Average Pooling

Average pooling (AP) is a naive method for reduction, without considering the semantics of words and tweets. For each sequence $\mathbf{e}_i$ derived from $T$, we take the average $\mathbf{s}_i$ within it; that is,

$$\mathbf{s}_i = \frac{1}{M} \sum_m \mathbf{e}_{im}. \tag{2}$$

Then, we concatenate $\mathbf{s}_i$ with follower number representation $\mathbf{v}_i$ for tweet $t_i$ as

$$\mathbf{h}_i = [\mathbf{s}_i^\top, \mathbf{v}_i^\top]^\top. \tag{3}$$

Finally, the average is computed over $T$ to obtain a collective tweet message representation $\mathbf{h}'$ as

$$\mathbf{h}' = \frac{1}{I} \sum_i \mathbf{h}_i. \tag{4}$$

#### 4.3.2 Hierarchical Attention

Because different words in message $t_i$ and different messages in $T$ contribute differently to the prediction, we introduce hierarchical attention layers (HALs) (Yang et al., 2016) to compute $\mathbf{h}'$.

Specifically, for the $m$-th word representation $\mathbf{e}_{im}$ in tweet $t_i$, the word-level attention $\alpha_{im}$ is given by

$$\mathbf{u}_{im} = \tanh(\mathbf{W}_w \mathbf{e}_{im} + \mathbf{b}_w), \tag{5}$$

$$\alpha_{im} = \frac{\exp(\mathbf{u}_w^\top \mathbf{u}_{im})}{\sum_{m'} \exp(\mathbf{u}_w^\top \mathbf{u}_{im'})}, \tag{6}$$

where $\mathbf{W}_w$ is a trainable matrix, $\mathbf{b}_w$ and $\mathbf{u}_w$ are trainable vectors, and $\mathbf{u}_w$ is in $R^{D_w}$. We can see $\mathbf{u}_w$ as a vector to find (or represent) informative words for the prediction. The output $\mathbf{s}_i$ of this word-level attention layer is a representation of $t_i$, given by

$$\mathbf{s}_i = \sum_m \alpha_{im} \mathbf{e}_{im}. \tag{7}$$

We then apply the tweet-level attention layer because, similar to the word-level attention layer, different tweets can contribute to the prediction differently. For this level, the follower number is also considered, because a tweet may have a greater impact on the market and thus the prediction, if it is authored by a Twitter user whose follower number is large. We concatenate $\mathbf{s}_i$ and $\mathbf{v}_i$ to obtain $\mathbf{h}_i$ in the same manner as Eq. 3. Then, we use a similar attention layer on the word-level as with the tweet-level attention:

$$\mathbf{u}'_i = \tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_s), \tag{8}$$

$$\alpha'_i = \frac{\exp(\mathbf{u_s}^\top \mathbf{u}'_i)}{\sum_i \exp(\mathbf{u}_s^\top \mathbf{u}'_i)}, \tag{9}$$

$$\mathbf{h}' = \sum_i \alpha'_i \mathbf{h}_i, \tag{10}$$

where $\mathbf{W}_s$ is a trainable matrix, $\mathbf{b}_s$ and $\mathbf{u}_s$ are trainable vectors, and $\mathbf{u}'_i$ is in $R^{D_s}$.

### 4.4 Prediction

The representation $\mathbf{h}'$ of $T$ goes through an MLP with two fully connected (FC) layers, and the activation function between the FC layers is a hyperbolic tangent function. Then, the output of the MLP is fed into a softmax function to obtain the prediction $z \in R^2$; that is,

$$\mathbf{z} = softmax(MLP(\mathbf{h}')). \tag{11}$$

## 5 Experimental Settings

### 5.1 Dataset Splits

We used both the BigName and StockNet datasets for our experiment. We divided the datasets into training, validation, and test splits based on the time periods, as shown in Table 1. The split of the StockNet dataset followed Xu and Cohen (2018).

(Insert Table 1).

### 5.2 Models for Experiment

We list all combinations of modules to be evaluated in Table 2. StockNet at the bottom is the baseline model presented in Xu and Cohen (2018).

(Insert Table 2).

### 5.3 Implementation

For the StockBERT and StockGVBR models, the maximum token size of each tweet message was limited to 50, with excessive tokens being truncated. Meanwhile, we left the limit as 40 for the StockNet model following (Xu and Cohen, 2018). The maximum number of tweets in each sample was set to 30, and the earliest 30 tweets were retained.

The hidden state dimension of BERT is 768 ($D_e = 768$), as mentioned previously. The dimension of the GloVe embedding is 50, which is transformed in the first layer of BiGRU to 100, and we used this also for its output; therefore, for the BiGRU-based model, $D_e = 100$. All variants of our model used $R^{D_w} = R^{D_s} = 100$. The MLP's hidden state size (i.e., the dimensionality of the first FC layer's output) was 16. The follower number representation parameter $K$ was 20.

We used *cross entropy* as our loss function and *Adam* as the optimizer. We followed the two-stage fine-tuning[9] method to train the StockBERT variants; the method first freezes the parameters in BERT and trains the rest of the model with a relatively larger learning rate. The parameters in BERT are then unfrozen, and the entire model is trained with a relatively small learning rate. We set five epochs for the first fine-tuning stage and another five epochs for the second fine-tuning stage. For the StockGVBG and StockNet models, we set the number of training epochs to 10. We used early stopping for all models. The learning rate was set to $5 \times 10^{-4}$ and $5 \times 10^{-6}$ for the first and the second fine-tuning stages for StockBERT, $5 \times 10^{-4}$ for StockGVBG, and $1 \times 10^{-3}$ for StockNet. The mini-batch size was set to 8. To evaluate the stability of these models, we trained them 10 different times and evaluated each trained model with the test split. The performance is given as the average and standard deviation of the test accuracy scores over 10 repetitions.

## 6 Results and Discussion

### 6.1 Results Overview

Table 3 lists the accuracy scores of our models and StockNet. All accuracy scores were approximately 50%.

[9] https://www.tensorflow.org/guide/keras/transfer_learning

(Insert Table 3).

Furthermore, we used the Student's t-test (hereinafter referred to as the t-test) to show the statistical significance of the performance differences between the models and random guesses. The null hypothesis is that *the mean of the prediction accuracy of the model is equal to that of random guesses*. The $t$-test was performed in a two-tailed manner, and the results are shown in Table 4.

(Insert Table 4).

The results show that when we use the BigName dataset, three models — StockBERT HN, StockGVBG HN, and StockBERT AN — outperformed random guesses with statistical significance, but the differences were very small, as shown in Table 3. For the StockNet dataset, none of the models could beat random guesses.

## 6.2 Comparative Analysis

Again, we used the t-test to compare the performances between the models that are the same except for one module.

### 6.2.1 With vs. Without Follower Numbers

Table 5 lists the t-test results of comparing the performance of the model with and without the follower numbers. For the cases without the follower numbers, we used $\mathbf{h}_i = \mathbf{s}_i$ instead of Eq. 3. The results of, *StockBERT HE vs. StockBERT HN* and *StockGVBG AE vs. StockGVBG AN* show statistically significant differences over the BigName dataset. This was against our expectation: the models that did not consider follower numbers outperformed the models that considered follower numbers. This implies that the extra information, that is, follower numbers, did not help or was even harmful to the prediction. A possible explanation is that (i) the models cannot make full use of this extra information due to, for example, insufficient training samples, and (ii) the follower numbers themselves are inherently useless because the regime of financial market can be variable.

(Insert Table 5).

### 6.2.2 *WordPiece+BERT* vs. *NLTK+GloVe+BiGRU*

Table 6 lists p-values between the models with WordPiece+BERT and NLTK+GloVe+BiGRU. Only the result of StockBERT AN vs. StockGVBG AN show a statistically significant difference,

but the *negative* marks (−) suggest that the advanced embedding method does not help improve the performance as expected.

(Insert Table 6).

### 6.2.3 *Hierarchical Attention* vs. Average Pooling

Table 7 lists p-values between the models with AP and HAL. Only the result of *StockBERT AN vs. StockGVBG AN* shows statistical significance with a *positive* mark, which implies that by introducing HAL, the performance improves compared with AP.

(Insert Table 7).

## 6.3 Discussion

Stock price prediction in the financial market using DL models is attractive — however, it has many pitfalls. Compared to traditional fields where DL models have already achieved great success, the financial market has its own data property. Therefore, we need to consider model development similar to many orthodox ML works, and we also need to consider data quality. In this section, we note our thinking about DL work in the financial market based on our experiments and results.

Many studies, such as ours, apply a methodology that directly uses the DL model and financial data to predict asset prices. Although some of the works claimed good performance, we still need to be aware that the hidden structure of the financial market can suddenly change. Thus, a DL model with millions of parameters can easily fall into an overfitting problem.

In addition to the constantly changing market, the prediction frequency may also heavily influence model performance. Our problem formulation employs daily basis prediction, and it hardly outperforms random guess. This result supports the EMH for this prediction frequency rate. However, we still think that it is interesting to see whether DL models can perform well when applied to higher frequency prediction or when predicting the instant price change immediately after new information is released.

# 7 Conclusion

In this study, we test the EMH with daily stock price prediction using tweet messages, taking the follower numbers into account. According to our experiment, all variants of our models fail to

prominently outperform random guesses, and thus we cannot reject that the stock market is efficient based on our data and method. We will further proceed to increase the prediction frequency rate so that the prediction can be performed before the fluctuation of the stock price in response to the release of new information converges.

# References


Atsalakis, G. S. and Valavanis, K. P. (2009). Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36:5936–5941.

Ball, R. and Brown, P. (1968). An empirical evaluation of accounting income numbers. *Journal of Accounting Research*, 6(2):159–178.

Ballings, M., den Poel, D. V., Hespeels, N., and Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42:7046–7056.

Bar-Haim, R., Dinur, E., Feldman, R., Fresko, M., and Goldstein, G. (2011). Identifying and following expert investors in stock microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, page 1310–1319.

Campbell, J. Y. and Shiller, R. J. (1988). The dividend-price ratio and expectations of future dividends and discount factors. *The Review of Financial Studies*, 1(3):195–228.

Chen, D., Ma, S., Harimoto, K., Bao, R., Ren, Y., Su, Q., and Sun, X. (2019a). Group, extract and aggregate: Summarizing a large amount of finance news for forex movement prediction. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, pages 41–50.

Chen, D., Zou, Y., Harimoto, K., Bao, R., Ren, X., and Sun, X. (2019b). Incorporating fine-grained events in stock movement prediction. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, page 31–40.

Da, Z., Engelberg, J., and Gao, P. (2015). The sum of all fears investor sentiment and asset prices. *The Review of Financial Studies*, 28(1):1–32.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, X., Liao, K., Li, Z., and Duan, J. (2019). Event representation learning enhanced with external commonsense knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, page 4894–4903.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2014). Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1415–1425.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, page 2327–2333.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2016). Knowledge-driven event embedding for stock prediction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, page 2133–2142.

Duan, J., Zhang, Y., Ding, X., Chang, C.-Y., and Liu, T. (2018). Learning target-specific representations of financial news documents for cumulative abnormal return prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, page 2823–2833.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*, 10(1):1–21.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33:2223–2273.

Henrique, B. M., Sobreiro, V. A., and Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems With Applications*, 124:226–251.

Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes.

Li, Q. and Shah, S. (2017). Learning stock market sentiment lexicon and sentiment-oriented word vector from stocktwits. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, page 301–310.

Liu, J., Liu, X., Lin, H., Xu, B., Ren, Y., Diao, Y., and Yang, L. (2019). Transformer-based capsule network for stock movements prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 66–73.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Rosenberg, B., Reid, K., and Lanstein, R. (1985). Persuasive evidence of market inefficiency. *The Journal of Portfolio Management*, 11(3):9–16.

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., and Deng, X. (2013). Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, page 24–26.

Si, J., Mukherjee, A., Liu, B., Pan, S. J., Li, Q., and Li, H. (2014). Exploiting social relations and sentiment for stock prediction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1139–1145.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wu, H., Zhang, W., Shen, W., and Wang, J. (2018). Hybrid deep sequential modeling for social text-driven stock prediction. In *CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1627–1630.

Xu, Y. and Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, page 1970–1979.

Yang, L., Xu, Y., Ng, T. L. J., and Dong, R. (2019). Leveraging bert to improve the fears index for stock forecasting. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 54–60.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016*, pages 1480–1489.

```
{
  “id”: “1161986571907817472”,
  “created_at”: “Thu Aug 15 13:02:42 +0000 2019”,
  “text”: “Stock Futures Bounce, Now ... ”,
  “user”: {
            “id”: “18616722”,
            “followers_count”: “7590”,
            …
  },
  …
}
```


Figure 1: Tweet object example.


```
{
  “stock_id”: 0,
  “stock”: “CSCO”,
  “label”: “-1”
  “tweet_from_time”: “2020-05-28 16:30:00 -0400EDT”,
  “tweet_to_time”: “2020-05-29 16:30:00 -0400EDT”,
  “tweeted_time”: [ “2020-05-28 20:43:11 -0400EDT”,
                    “2020-05-29 07:41:56 -0400EDT”,
                    …
                  ],
  “tweet”: [ “RT @russeltoc: @petenajarian and $CSCO ... ”,
             “$CSCO (+0.6% pre) Cisco acquires ... ”,
             …
           ],
  “follower”: [ 157647, 53752, …],
}
```


Figure 2: Sample example.

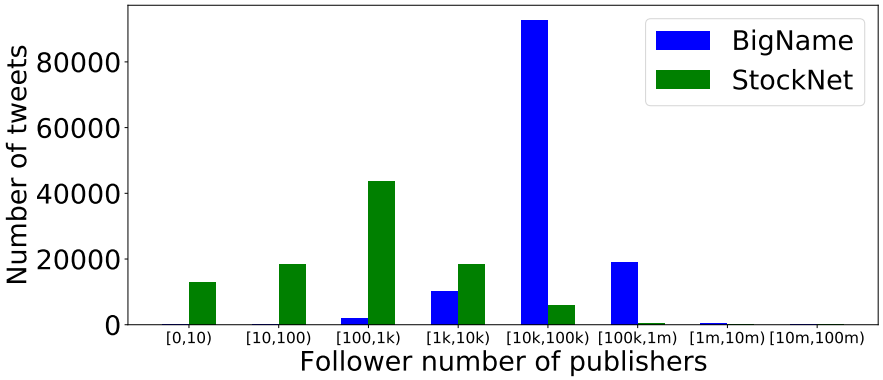


Figure 3: Distributions of the number of stock tweet messages with respect to the number of followers in BigName and StockNet.

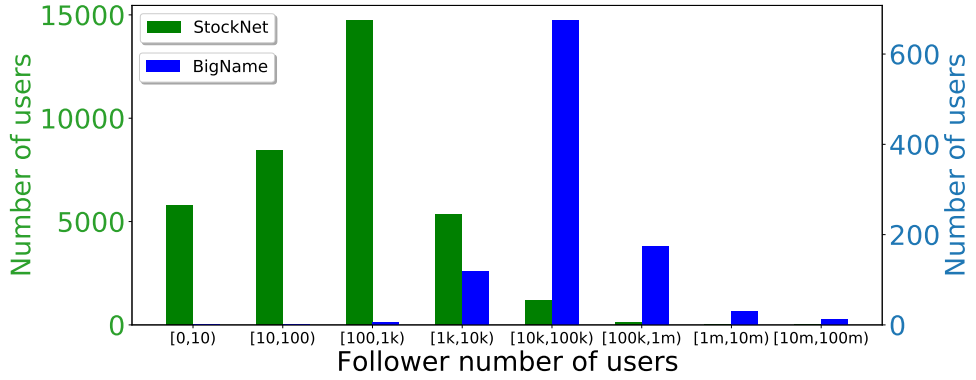


Figure 4: Distributions of the number of Twitter users with respect to the number of followers in the BigName and StockNet datasets.

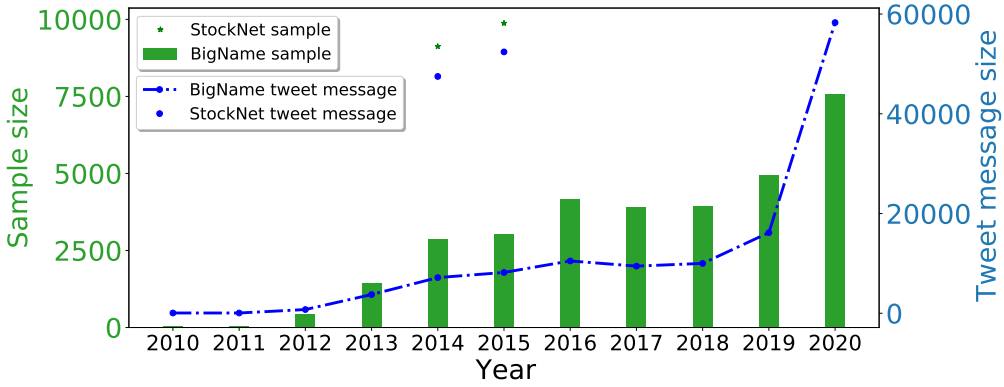


Figure 5: Distributions of the numbers of samples per year in BigName and StockNet.

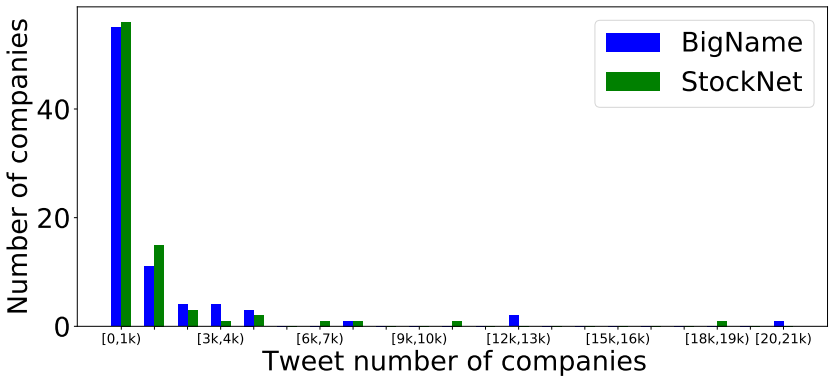


Figure 6: Distributions of the number of stocks with respect to the number of stock tweet messages.

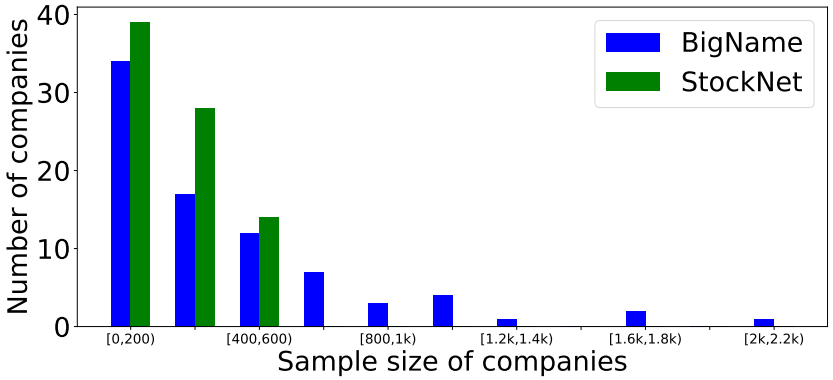


Figure 7: Distributions of the number of stocks with respect to the number of samples.

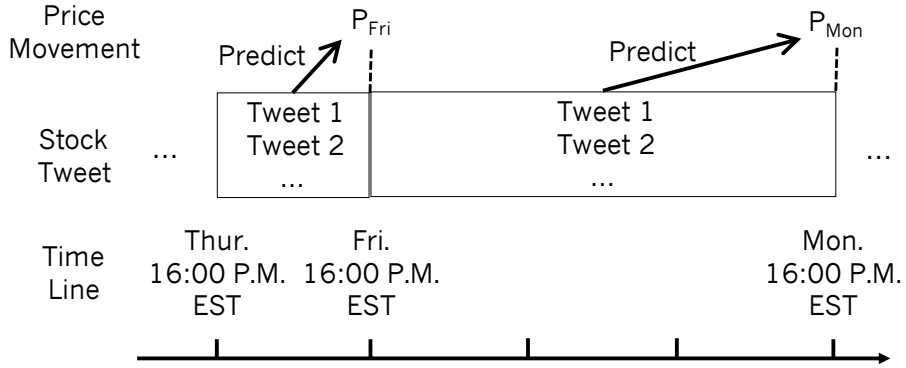


Figure 8: Illustration of our task. Tweet contains tweet message and the optional publisher's follower number.

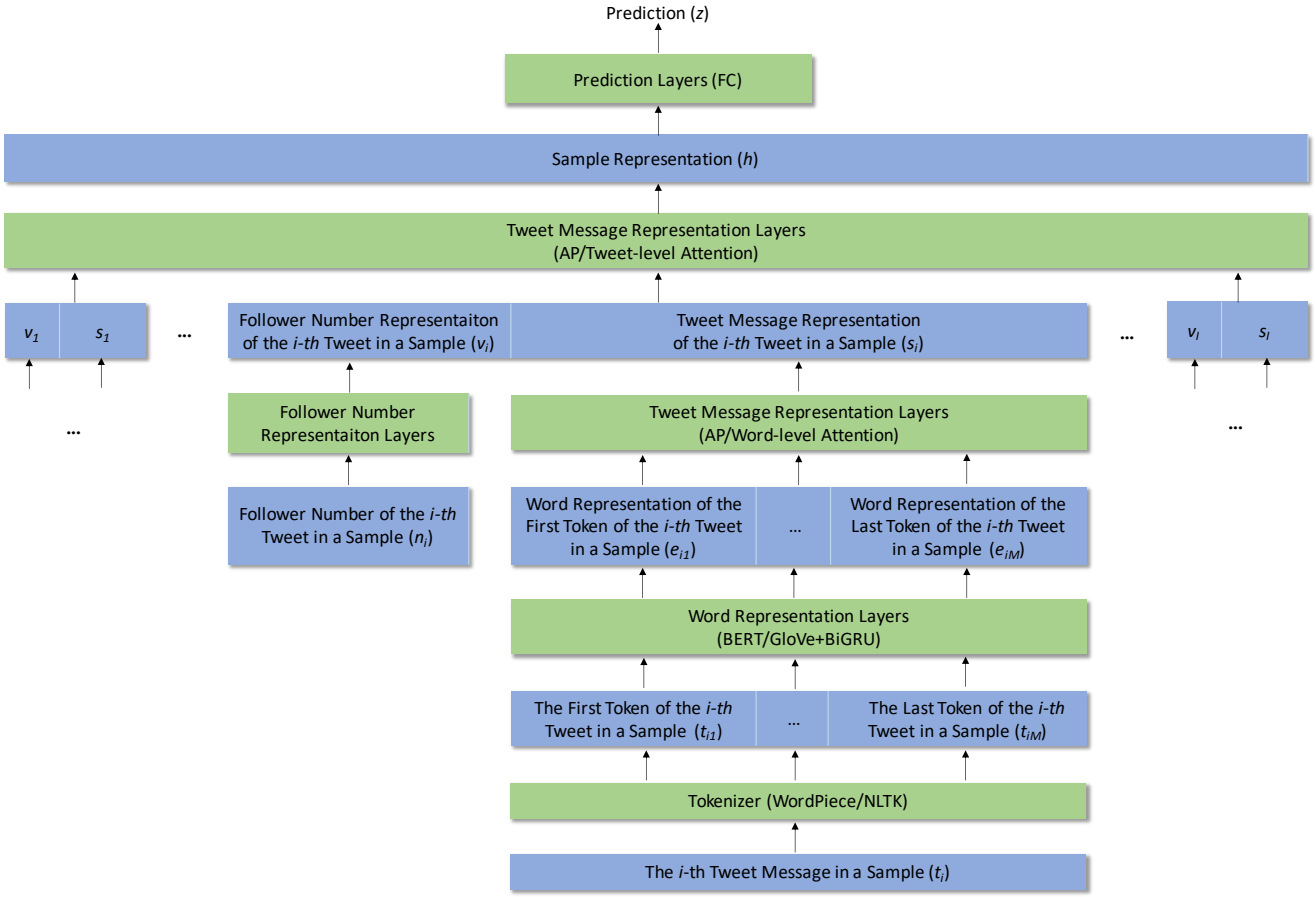


Figure 9: Illustration of our pipeline.

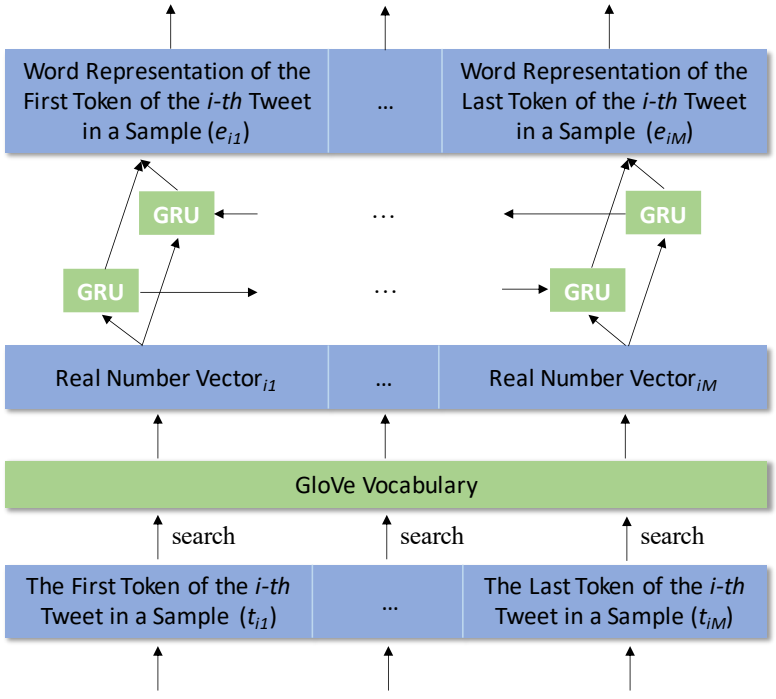


Figure 10: Illustration of GloVe+BiGRU.

Table 1: Dataset splits for BigName and StockNet.

| Dataset | | Training | Validation | Test |
|---|---|---|---|---|
| BigName | Start | 2010-01-01 | 2020-04-01 | 2020-07-01 |
| | End | 2020-03-31 | 2020-06-30 | 2020-09-30 |
| | Size | 26,530 | 2,557 | 2,703 |
| StockNet | Start | 2014-01-01 | 2015-08-01 | 2015-10-01 |
| | End | 2015-07-31 | 2015-09-30 | 2015-12-31 |
| | Size | 14,737 | 1,624 | 2,440 |

*Notes*: Size is measured by the number of task samples.

Table 2: Models used in the experiment.

| Model name | Tokenizer & word embedding | Word/sentence relation | Encoded follower number |
|---|---|---|---|
| *StockBERT HE* | WordPiece & BERT | HAL | With |
| *StockBERT HN* | WordPiece & BERT | HAL | Without |
| *StockBERT AE* | WordPiece & BERT | AP | With |
| *StockBERT AN* | WordPiece & BERT | AP | Without |
| *StockGVBG HE* | NLTK TweetTokenizer & GloVe+BiGRU | HAL | With |
| *StockGVBG HN* | NLTK TweetTokenizer & GloVe+BiGRU | HAL | Without |
| *StockGVBG AE* | NLTK TweetTokenizer & GloVe+BiGRU | AP | With |
| *StockGVBG AN* | NLTK TweetTokenizer & GloVe+BiGRU | AP | Without |
| *StockNet* | NLTK TweetTokenizer & GloVe+BiGRU | - | Without |

*Notes*: We use the *Fundamental StockNet*, one of the four StockNet (Xu and Cohen, 2018) variants.

Fundamental StockNet only uses tweet messages as input and can predict the next stock price movement.

The StockNet model has a different model structure and model components.

Table 3: Average and standard deviation of prediction accuracy.

| Model | BigName dataset | StockNet dataset |
|---|---|---|
| *StockBERT HE* | 50.50% ± 0.70% | 49.14% ± 1.98% |
| *StockBERT HN* | 51.09% ± 0.69% | 49.84% ± 1.75% |
| *StockBERT AE* | 50.17% ± 0.78% | 49.21% ± 2.11% |
| *StockBERT AN* | 50.61% ± 0.43% | 49.69% ± 2.06% |
| *StockGVBG HE* | 50.53% ± 0.74% | 50.86% ± 2.77% |
| *StockGVBG HN* | 50.90% ± 0.33% | 49.17% ± 2.48% |
| *StockGVBG AE* | 50.21% ± 0.65% | 50.10% ± 2.57% |
| *StockGVBG AN* | 51.02% ± 0.57% | 51.04% ± 2.49% |
| *StockNet* | 49.60% ± 0.00% | 47.25% ± 0.00% |
| *Random Guess* | 50.19% ± 1.12% | 50.13% ± 0.62% |

*Notes*: Results are based on experiments repeated 10 times.

Table 4: *P*-value of *t-test* for model performance comparison.

| Model | BigName dataset | StockNet dataset |
|---|---|---|
| *StockBERT HE* | 0.4548 | 0.1308 |
| *StockBERT HN* | 0.0345** | 0.6199 |
| *StockBERT AE* | 0.9641 | 0.1844 |
| *StockBERT AN* | 0.2659 | 0.5071 |
| *StockGVBG HE* | 0.5000 | 0.3406 |
| *StockGVBG HN* | 0.0578* | 0.2303 |
| *StockGVBG AE* | 0.9651 | 0.9759 |
| *StockGVBG AN* | 0.0414** | 0.2195 |
| *StockNet* | 0.0951* | 0.0000*** |

*Notes*: The null hypothesis is *the mean of the prediction accuracy of the model is equal to that of random guesses.* The star marks ***, **, and *, represent significance levels at 0.01, 0.05, and 0.1, respectively.

Table 5: Results of *t-test* for model performance comparison.

| Model | BigName dataset | StockNet dataset |
|---|---|---|
| *StockBERT HE* vs *StockBERT HN* | − 0.0710* | − 0.6168 |
| *StockBERT AE* vs *StockBERT AN* | − 0.1419 | − 0.4120 |
| *StockGVBG HE* vs *StockGVBG HN* | − 0.1026 | + 0.4182 |
| *StockGVBG AE* vs *StockGVBG AN* | − 0.0081*** | − 0.1435 |

*Notes*: The null hypothesis is *the mean of the prediction accuracy of the with-follower-number model is equal to that of the without-follower-number model.*

The star marks ***, **, and *, represent significance levels at 0.01, 0.05, and 0.1, respectively.

The positive mark + means that the with-follower-number model outperforms the without-follower-number model, while the negative mark − indicates the opposite.

Table 6: Results of *t-test* for model performance comparison.

| Model | BigName dataset | StockNet dataset |
|---|---|---|
| *StockBERT HE* vs *StockGVBG HE* | − 0.9312 | − 0.1041 |
| *StockBERT HN* vs *StockGVBG HN* | + 0.4409 | + 0.4940 |
| *StockBERT AE* vs *StockGVBG AE* | − 0.9119 | − 0.4095 |
| *StockBERT AN* vs *StockGVBG AN* | − 0.0833* | − 0.2029 |

*Notes*: The null hypothesis is *the mean of the prediction accuracy of the WordPiece+BERT leveraged model is* equal to that of the *NLTK+GloVe+BiGRU leveraged model.* The star marks ***, **, and *, represent significance levels at 0.01, 0.05, and 0.1, respectively.
The positive mark + means that the *WordPiece+BERT* leveraged model outperforms the *NLTK+GloVe+BiGRU* leveraged model, while the negative mark − indicates the opposite.

Table 7: Results of *t-test* for model performance comparison.

| Model | BigName dataset | StockNet dataset |
|---|---|---|
| *StockBERT HE* vs *StockBERT AE* | + 0.3443 | − 0.9381 |
| *StockBERT HN* vs *StockBERT AN* | + 0.0732* | + 0.8582 |
| *StockGVBG HE* vs *StockGVBG AE* | + 0.4111 | + 0.4913 |
| *StockGVBG HN* vs *StockGVBG AN* | − 0.5828 | − 0.1105 |

*Notes*: The null hypothesis is *the mean of the prediction accuracy of the hierarchical attention leveraged model is equal to that of the average pooling leveraged model.*

The star marks ***, **, and *, represent significance levels at 0.01, 0.05, and 0.1, respectively.

The positive mark + means that the *hierarchical attention* leveraged model outperforms the *average pooling* leveraged model, while the negative mark − indicates the opposite.