Asia Pacific Journal of Information Systems
Vol. 30 No. 4 (December 2020), 719-740

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
https://doi.org/10.14329/apjis.2020.30.4.719

Predicting Stock Prices Based on Online News Content and Technical Indicators by Combinatorial Analysis Using CNN and LSTM with Self-attention

Sang Hyung Jung[a], Gyo Jung Gu[b], Dongsung Kim[c], Jong Woo Kim[d,*]

[a] *Undergraduate student, Business Administration at the School of Business, Hanyang University, Korea*
[b] *Undergraduate student, Department of Finance at the School of Business, Hanyang University, Korea*
[c] *Postdoctoral researcher, Business Administration at the School of Business, Hanyang University, Korea*
[d] *Professor, School of Business, Hanyang University, Korea*

ABSTRACT

The stock market changes continuously as new information emerges, affecting the judgments of investors. Online news articles are valued as a traditional window to inform investors about various information that affects the stock market. This paper proposed new ways to utilize online news articles with technical indicators. The suggested hybrid model consists of three models. First, a self-attention-based convolutional neural network (CNN) model, considered to be better in interpreting the semantics of long texts, uses news content as inputs. Second, a self-attention-based, bi-long short-term memory (bi-LSTM) neural network model for short texts utilizes news titles as inputs. Third, a bi-LSTM model, considered to be better in analyzing context information and time-series models, uses 19 technical indicators as inputs. We used news articles from the previous day and technical indicators from the past seven days to predict the share price of the next day. An experiment was performed with Korean stock market data and news articles from 33 top companies over three years. Through this experiment, our proposed model showed better performance than previous approaches, which have mainly focused on news titles. This paper demonstrated that news titles and content should be treated in different ways for superior stock price prediction.

Keywords: Stock Price Prediction, Online News, CNN, LSTM, Technical Indicators

Ⅰ. Introduction

Stock price forecasting has been actively studied as a practically and academically important topic. As the big data era has emerged, stock price prediction research using big data has become an area of active study. Various studies have predicted stock behaviors through machine learning methods, such as re-

*Corresponding Author. E-mail: kjw@hanyang.ac.kr gression, k-nearest neighbor (k-NN), and support vector machine (SVM) analysis(Ren et al., 2015) using numerical data or technical indicators. Furthermore, due to advances in artificial intelligence and increased internet and social media data, studies are being actively conducted to increase accuracy in analyzing the stock market.

With advances in deep-learning methods, unstructured data sources such as online news or social networking service (SNS) content related to enterprises have increased vastly. These data sources provide real-time information on industry trends and individual corporations and are valuable for stock price prediction. Particularly because stock prices are affected by information from various resources (Malkiel et al., 2013), analysis of online news is important in predicting stock prices because it provides relatively accurate information. By analyzing online news, various research techniques have emerged, such as extracting sentimental information(Jeong et al., 2015) or creating event-embedded vectors(Ding et al., 2015) for stock price prediction.

Some current studies have proposed hybrid models that utilize numerical data or technical indicators with online news(Liu, 2018; Vargas et al., 2018). These studies focus on development of effective models, such as hybrid models, that concatenate data in final layers or on collecting highly explainable data before training. Through these approaches, some studies show that consideration of technical indicators or numerical data via online news may be better for stock price prediction than consideration of online news alone.

However, existing research mainly uses titles from online news, following methodologies that purport that usage of titles only is better for stock prediction(Ding et al., 2014). In using only the titles of online news, this approach misses the real purpose of hybrid models to properly reflect all information through technical indicators. Furthermore, some online news-based research attaches titles and content in training sets within a single model. This approach neglects the different characteristics of news titles and content. In this paper, we divide online news data into titles and content, training each component separately with different deep learning models.

We propose a hybrid deep learning model that appropriately analyzes titles and content derived from online news based on technical indicators. We aim to determine the best match between each data component and deep learning method. In our best model, titles are analyzed through self-attention-based LSTM, while content is analyzed by self-attention-based CNN and technical indicators with bi-LSTM. Through an experiment with a Korean stock market dataset, our best model shows superior performance to baseline models, which implies that online news content is a good data source for stock price prediction.

The rest of the paper is arranged as follows. We discuss related research and background context in Section 2. Our proposed methodologies for stock price prediction are introduced in Section 3. Section 4 shows experimental settings and results, and Section 5 addresses our market simulation strategy and results. Section 6 discusses our experiment results and the contributions of this study to existing research. Section 7 comprises conclusions and issues for future research.

Ⅱ. Background

This section discusses previous studies on stock price prediction with technical indicators and text, especially online news. Background methodologies are introduced, including deep learning methods and self-attention mechanisms. Furthermore, our reasons for selecting self-attention-based CNN (selfAttn/CNN) and self-attention-based LSTM (selfAttn/LSTM) for analyzing online news are discussed.

2.1. Stock Prediction with Technical Indicators

In past research, stock price prediction has been a very interesting topic for researchers in various fields(Shen et al., 2012). Much research has been performed by companies and universities, and various indicators have been used to more accurately predict stock prices.

Two primary approaches are mainly used. The first approach, called "fundamental analysis," focuses on financial statement and potential business viability of companies as opposed to tracing stock prices. In contrast, analysis with technical indicators is about stock price movement based on mathematical calculations of stock prices or volume. With explosive growth in data volume, it is easier to predict market trends or stock price movement with mathematical calculations through machine learning techniques than with fundamental analysis. Various technical indicators have been utilized to predict stock prices, and as machine learning techniques have emerged, much research has analyzed price patterns and stock index prediction(Shah, 2007).

Various studies use technical indicators. One study classifies various indicators such as highest price, lowest price, real strength index (RSI), and moving average using decision tree technique(Nair et al., 2010). In addition, a study was performed to predict the type and movement of stock prices using indicators such as stochastics and RSI and using k-NN for information on the movement of indicators (Teixeira et al., 2010). In addition, research on stock price prediction has been conducted using SVM techniques along with techniques such as technical indicators, macroeconomic indicators, and principal component analysis (PCA)(Choudry et al., 2008; Ren et al., 2015; Shen et al., 2012; Yu et al., 2014). Because stock price is affected by various factors including macroeconomic indicators, however, prediction accuracy is relatively low.

In recent years, as deep learning technology has developed, research has proceeded using various information that affects stock prices in addition to that in traditional studies(Chen et al., 2015; Yoshihara et al., 2014). Representative information that affects stock prices includes domestic and international economic conditions, news information, and company-related disclosures(Bank et al., 2011). This information can be obtained in online news and corporate disclosure systems. This paper uses text data from online news and forecasts stock prices in combination with 19 technical indicators comprising open, high, low, adjusted-close, and volume indicators. <Table 1> in section 4.1 shows previous research based on the technical indicators utilized herein.

2.2. Stock Prediction with Text Mining

With the development of natural language processing technology, research to predict stock price fluctuations has been conducted by analyzing texts from SNS and online news. In the case of SNS texts, some research attempts to predict fluctuations in stock prices by analyzing the emotions of investors, assuming that actions of the investors are based on their emotions(Bollen et al., 2011). To analyze emotions, some studies extract subjects in texts through latent Dirichlet allocation (LDA)(Si et al., 2013) for analysis with constructed emotional dictionaries(Mittal et al., 2012). In SNS texts, however, there is much slang and many typos, which cause difficulties in predicting stock price fluctuations with extracted emotions because subjects are weak and sometimes incorrect.

On the other hand, online news data are constant in amount, contain a lot of information in comparison to SNS text data, and have a correlation with changes in stock prices(Fu et al., 2008). In addition, because online news comprises official articles, few words are unclear, such as slang or typos. For this reason, recent studies have analyzed online news texts to predict stock price fluctuations. There are studies to predict the direction of stock prices via construction of emotional dictionaries from online news texts(Yu et al., 2014), as well as a study to predict the direction of stock prices with construction of separate emotional dictionaries per company(Jeong et al., 2015; Yu et al., 2013). In addition, text is analyzed by the bag-of-words model, noun phrases, and individual names, while stock price prediction is conducted with SVM(Schumaker and Chen, 2009).

Recently, research has emerged using deep learning techniques, which are based on short-term and long-term events through text data to learn stock price predictions CNN(Ding et al., 2015). On the other hand, long short-term memory (LSTM) networks or gated recurrent unit (GRU) networks have been two main methods in research in stock prediction fields. This paper proposes two main models, CNN and LSTM, both of which are based on self-attention and are more powerful than basic CNN and LSTM.

2.3. Stock Prediction with Multiple Sources

To strengthen predictions, recent research utilizes both text data and technical indicators. Research has extracted features from online news text data, used technical indicators, and predicted stock prices through SVM(Zhai et al., 2007). In one study, technical indicators are labeled along with text data to perform stock price predictions through multiple kernel regression analysis(Li et al., 2014).

Recently, studies have emerged with deep learning architecture of CNN for analysis of text data and LSTM for analysis of technical indicators(Liu, 2018; Vargas et al., 2018). On the other hand, some studies have applied LSTM not only for text data, but also for numerical data. However, most existing studies have utilized only titles in stock prediction with deep learning techniques. In this paper, we not only show that content is useful for stock prediction, but we also try to identify the deep learning technique suitable for each set of data.

2.4. Difference between CNN and LSTM in NLP

There are many ways to obtain information from text data. Because deep neural networks are in the spotlight for natural language processing (NLP), several methods have been applied for text classification or word embedding and other tasks. In particular, CNN(Lecun and Bengio, 1995) and LSTM (Hochreiter and Schmidhuber, 1997), which are main structures in deep neural networks, have shown remarkable achievements in various tasks, including NLP(Kalchbrenner et al., 2014; Kim, 2014; Zhang et al., 2015).

Because these two main types of neural networks have different structures, however, a lot of research compares the two architectures in NLP(Yin et al., 2017). Previous research shows that CNN is better when extracting semantic meaning from natural language data and to recognize key phrases from data. On the other hand, LSTM is better than CNN for overall tasks in NLP, relational classification, and text entailment and has shown outstanding results in performing tasks to identify sequential information. In conclusion, CNN weakly consider context information but extract key phrases for target features. In contrast, LSTM strongly consider context information and are therefore better for overall NLP tasks, although the architecture has vanishing gradient problems.

This paper utilizes online news content and titles separately for stock prediction because we assume different characteristics. Through an experiment, we compare various baseline models with our final model, determining which architecture is better suited for news content and titles.

2.5. Self-attention Mechanism

Self-attention mechanisms were first introduced in machine translation research(Vaswani et al., 2017). Because self-attention mechanisms have been mixed with several architectures with proven performance, many trials have successfully used self-attention with main deep learning architectures based on CNN or LSTM(Lin et al., 2017; Park et al., 2018; Shen et al., 2018; Woo et al., 2018).

Self-attention mechanisms utilize feature maps from deep-learning architecture. There are a lot of ways to use self-attention, but the objective begins with a concept of weighting the things that are considered more important, using a feature map in CNN or a hidden state in LSTM. This paper applies self-attention for CNN and LSTM, respectively, when analyzing online news content and online news titles. In CNN, we used two self-attention module, channel attention and spatial attention which use feature map with different ways. In LSTM, we used set of summation weight vectors for LSTM hidden state, dotted with the final hidden state. Although it is not described in this paper, we progressed several experiments and we can check when use deep learning architecture with self-attention, it showed significant result than naïve architecture.

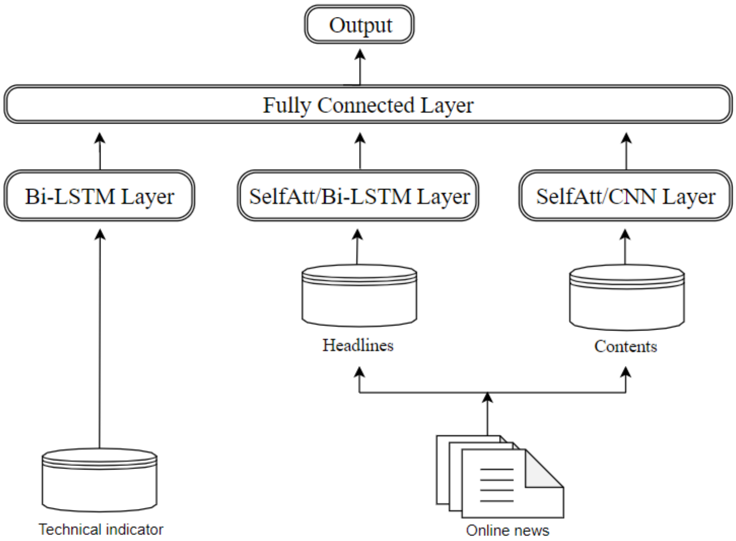
Ⅲ. Model

This section introduces our model architecture, which consists of three parts. The first part, selfAttn/CNN, has a convolution layer based on a self-attention module. The goal of the first part is to analyze online news content, which are characterized by relatively long sentences. The second part is selfAttn/LSTM, which is a bi-LSTM model based on a self-attention module. This part utilizes online news titles as inputs, which are shorter than online news content. In both parts, basic forms of self-attention modules are used to determine the type of method best for each type of data. The third part is bi-LSTM with 19 technical indicators, and this part is fixed throughout our experiment. All parts trained together because the components are merged in fully connected layers and predict stock movement through output layers. <Figure 1> shows an abbreviated form of our model architecture.

3.1. SelfAttn/CNN

In the selfAttn/CNN layer, we borrow the self-attention module from the convolution block attachment module (CBAM)(Park et al., 2018; Woo et al., 2018). The CBAM has shown significant results in image classification and detection with using CNN based on self-attention and is treated as the basic form of the self-attention module when using CNN. We modified this module to match NLP.

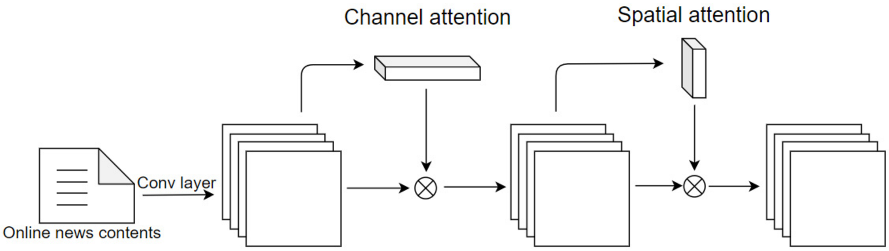
There are two self-attention modules in this part, channel attention and spatial attention. Input data



<Figure 1> Model Architecture

consist of N words from online news content. Input data are converted into three-dimensional data through the convolution layer, with C filters of size $k \times 300$(k = number of words to convolute at once, 300 = word embedding size). A feature map, with a size of $(N-k) \times C \times 300$, is obtained through the channel attention module. Each channel performs max pooling and average pooling and the size of results is . These results are concatenated to and go through simple MLP which returns size of vectors. This vectors are divided to two 1XC vectors and get sum to make up channel attention in <Figure 2>. Channel attention shows which channel is meaningful for each corpus of words. Channel attention is multiplied to feature map and then goes through spatial attention module. In the spatial attention module, the feature map inputs either max pooling or average pooling for each corpus of words with $k \times 300$ filters, comprising the spatial attention as shown in <Figure 2>. Although all k in the convolution layer and spatial attention module can have different values, we used k = 2 throughout our experiment.

The CBAM was used for image recognition and showed remarkable performance but could not be implemented to NLP in our research, especially for long texts. We modified the selfAttn/CNN part with



<Figure 2> SelfAttn/CNN Architecture

a self-attention module from CBAM in a way that contributes to utilization of online news content.

3.2. SelfAtt/LSTM

The selfAttn/LSTM layer is a bi-LSTM method with self-attention. We borrow the self-attention module from the Transformer(Vaswani et al., 2017) bi-LSTM self-attention module, which is regarded as a baseline of the self-attention mechanism. We modified this module slightly for our purposes.

Input data consists of N words from titles of online news. Input data goes through the bi-LSTM layer, and we extract the final hidden state of bi-LSTM to produce the self-attention map. With the last hidden state, we calculate dot-product self-attention, referring simply to the scalar-product of all hidden states from the LSTM layer with the final hidden state. This computation makes up self-attention module, length of N. This module goes through softmax layer and receives the element product with the feature map of LSTM.

Even if LSTM is better at the vanishing gradient problem than vanilla RNN(Sundermeyer et al., 2012), it also has vanishing gradient problem. So selfAttn/LSTM also has vanishing gradient problem because it is based on LSTM and the self-attention map reflects content in the final hidden state. Our experiment shows that this method is more powerful when used for short sentences than is selfAttn/CNN. <Figure 3> shows our selfAtt/LSTM architecture.

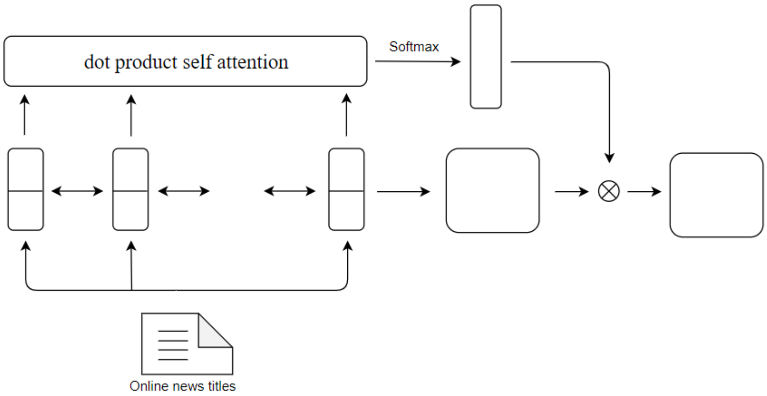
3.3. bi-LSTM

For technical indicators, we used a bi-LSTM layer. Although technical indicators already imply past information, when predicting stock prices, it is important to consider information from the distant past and from the near past. Accordingly, we chose bi-LSTM rather than vanilla-LSTM.

In all our experiments, we have utilized the input data with 19 technical indicators for the past 7 days.

Ⅳ. Experiment

In this section, we first introduce datasets and training method. Then, we introduce our alternative models in comparison to our model. Finally, we show our experiment results and demonstrate the effectiveness of our proposed model via t-tests.



<Figure 3> SelfAttn/LSTM Architecture

4.1. Data and Training

For our experiment, we collected online news data about 33 companies from a Korean representative news portal. Our aim was to use the news to predict stock prices, so we had no choice but to target companies that often appear in the news. So, we chose the top 3 companies in the market capitalization ranking in 11 industries that the Korea Exchange (KRX) divided in the Korea Composite Stock Price Index (KOSPI200). With 33 companies, we can evaluate whether the model works well throughout the industries. The collection period was from January 1, 2014 to December 31, 2018. The number of news articles per a company was at least 3,000 and the total number of news articles in our datasets was 597,456. We utilized data from January 1, 2014 to December 31, 2016 for the training set, data from January 1, 2017 to December 31, 2017 for a validation set and data from January 1, 2018 to December 31, 2018 for the test set. In addition, we collected technical indicators for each company in the same period. We utilized 19 technical indicators, calculated by open, high, low, adjusted-close, and volume indicators. <Table 1> shows formula for each indicators and previous research based on each indicator.

Throughout experimentations, we merged all online news with technical indicators for each article. When training, we split articles into three parts which are titles, content, and technical indicators, and use them separately or together. For text data, we used KHAIII (KaKao Hangul Analyzer III) API for morphological analysis and built own embedding model through the Word2Vec(Mikolov et al., 2013) method. In terms of online news, we were not able to use all texts, so we set thresholds of 15 characters for titles and 150 characters for content.

Model training was conducted by stock price prediction per online news but the final results of each model were evaluated by stock price prediction per day. When making predictions, if the numbers of predictions in up and down directions are equal, then predictions are classified by sum of softmax scores for each prediction.

<Table 1> Technical Indicators

Technical Indicators	Formula	Reference
RSI-14 (Relative Strength Index)	$y = \frac{AU}{(AU + AD)}$ $AU = average\ ups\ for\ 14\ days$ $AD = average\ downs\ for\ 14\ days$	Mizuno et al., 2012; Nair et al., 2010; Song, 2018
CCI (Commodity Channel Index)	$y = (M - m)/\ (d\ X\ 0.015)$ $M = (high + low + close)/\ 3$ $m = moving\ average\ of\ 7\ days\ of\ M$ $d = (\lvert M - m \rvert)/n$	Hsu et al., 2011; Lee et al., 2011; Patel et al., 2015
Momentum	$y = \frac{(Closing\ price)\ X\ 100}{(Closing\ price\ 7\ days\ before)}$	Bruni, 2017; Nair et al., 2010; Zhai et al., 2007
ADX (Average Directional Index)	$y = \frac{\lvert PDI - MDI \rvert X\ 100}{(PDI + MDI)}$ $PDI = high - high\ 1\ day\ before$ $MDI = low\ 1\ day\ before - low$	Bruni, 2017; Song, 2018
Slow Stochastic K	$y = moving\ average\ of\ fast\ \%K$ $fast\ \%K = \frac{(close - lowest\ among\ 14\ days) X 100}{highest\ among\ 14\ days - close\ lowest\ among\ 14\ days}$	Choudry et al., 2008; Liu, 2018; Zhai et al., 2007

<Table 1> Technical Indicators (Cont.)

Technical Indicators	Formula	Reference
Slow Stochastic D	$y = moving\ average\ of\ slow\ \%K$	Choudry et al., 2008; Liu, 2018; Zhai et al., 2007
Stochastic-RSI K	$y = moving\ average\ of\ fast - rsi\ \%K$ $fast - rsi\ \%\ K = \frac{(rsi14 - lowest\ rsi14\ among\ 14\ days)\ X\ 100}{highest\ rsi14\ among\ 14days - lowest\ rsi14\ among\ 14\ days}$	Bharathi et al., 2017; Bruni, 2017
Stochastic-RSI R	$y = moving\ average\ of\ slow - rsi\ \%K$	Bharathi et al., 2017; Bruni, 2017
Williams Percent Range	$y = \frac{highest\ among\ 14\ days - close}{highest\ among\ 14\ days - lowest\ among\ 14\ days}$	Liu, 2018; Nair et al., 2010; Zhai et al., 2007
MACD	$y = moving\ average\ of\ 12\ days -$ $moving\ average\ of\ 26\ days$	Bruni, 2017; Li et al., 2014; Nair et al., 2010
ROC (Rate of change)	$y = \frac{(close - close\ before\ 12\ days) X\ 100}{(close\ before\ 12\ days)}$	Li et al., 2014; Nair et al., 2010; Zhai et al., 2007
Exponential Moving Average – 8	$y = (close * ep\ ) + (y\ 1\ day\ before\ X\ (\ 1 - ep)$ $ep = \frac{2}{Length + 1}$ $Length = 8$	Bharathi et al., 2017; Bruni, 2017; Li et al., 2014
Exponential Moving Average – 20	$y = (close * ep\ ) + (y\ 1\ day\ before\ X\ (\ 1 - ep)$ $ep = \frac{2}{Length + 1}$ $Length = 8$	Bharathi et al., 2017; Bruni, 2017; Li et al., 2014
Exponential Moving Average – 200	$y = (close * ep\ ) + (y\ 1\ day\ before\ X\ (\ 1 - ep)$ $ep = \frac{2}{Length + 1}$ $Length = 8$	Bharathi et al., 2017; Bruni, 2017; Li et al., 2014
A/D (Accumulation/Distribution)	$y = \frac{(close - lowest) - (highest - close)\ X\ Volume}{highest - lowest}$	Liu, 2018; Vargas et al., 2018; Zhai et al., 2007
On-Balance Volume	$if\ close > close\ 1\ day\ before:$ $y = y\ day\ before + volume$ $elif\ close < close\ 1\ day\ before:$ $y = y\ day\ before - volume$ $elif\ close == close\ 1\ day\ before$ $y = y\ day\ before$	Nair et al., 2010; Vargas et al., 2018
Bollinger Bands upper	$y = moving\ average\ of\ 20\ days$ $+ (standard\ deviation\ of\ 20\ days\ X\ 2)$	Nair et al., 2010; Teixeira et al., 2010
Bollinger Bands middle	$y = moving\ average\ of\ 20\ days$	Nair et al., 2010; Teixeira et al., 2010
Bollinger Bands lower	$y = moving\ average\ of\ 20\ days$ $-$ (standard deviation of 20 days X 2)	Nair et al., 2010; Teixeira et al., 2010

4.2. Experiments Design

To evaluate our model, we compared several baseline models in, consideration of differing ways of data usage and number of cases in the model selection. We designed a total of nine models in four cases.

SVM In the first case, we only used technical indicators for predicting stock price. We used SVM instead of bi-LSTM which is our part of final model, because the amount of technical indicator data is too small for a deep-learning method. We selected SVM because, traditionally, SVM has been used most in several methods(Lin et al., 2013; Ren et al., 2015; Yu et al., 2014), and we compared it with our other baseline models.

LSTM_T In this model, we utilized online news titles and technical indicators for stock prediction. We used bi-LSTM with self-attention for titles and bi-LSTM for technical indicators, calling the model LSTM_T. This baseline model is similar to a model from previous research(Liu, 2018) showing significant performance in the S&P 500 index dataset. We modified the model to fit our approach.

CNN_T We changed the method for analyzing online news titles from selfAttn/LSTM to selfAttn/CNN in LSTM_T, as previous research(Woo et al., 2018). We modified the model by discarding the LSTM layer for a day prediction level and added self-attention for news prediction level. Both LSTM_T and CNN_T are the main baseline models for baseline model for our experiments. They use only titles and technical indicators, while discarding content data from online news when performing stock prediction

LSTM_TC For the fourth model in <Table 2>, we used online news title and content together. This approach is the most common when utilizing online news. We used selfAttn/LSTM for online news, which contains titles and content together and we used bi-LSTM for technical indicators.

CNN_TC In this model, we utilized online news in the same way as in LSTM_TC, but we changed the model from selfAttn/LSTM to selfAttn/CNN. Comparisons between CNN_T and CNN_TC, LSTM_T and LSTM_TC reveal that which is better for longer texts and whether online news content has meaningful information for stock price prediction.

LSTM_T/LSTM_C Between the sixth model and the final model in <Table 2>, we split titles and content from online news and applied other methods for each dataset. In this model, we used selfAttn/LSTM for titles and content and bi-LSTM for technical indicators.

CNN_T/CNN_C In this case, we used selfAttn/CNN for either titles or content, and we used bi-LSTM for technical indicators.

CNN_T/LSTM_C In this case, we applied different methods for titles and content data. We applied selfAttn/CNN for titles and selfAttn/LSTM for content.

LSTM_T/CNN_C For our final model, we used online news titles and content separately and applied selfAttn/LSTM for titles, selfAttn/CNN for content and bi-LSTM for technical indicators. These methods are concatenated in fully connected layers. By comparing the sixth and ninth models, we determine

<Table 2> Model Description

Data MID	Technical indicators	Title	Content	Title+ Content
SVM	SVM	-	-	-
LSTM_T	LSTM	SelfAttn/LSTM	-	-
CNN_T	LSTM	SelfAttn/CNN	-	-
LSTM_TC	LSTM	-	-	SelfAttn/LSTM
CNN_TC	LSTM	-	-	SelfAttn/CNN
LSTM_T/LSTM_C	LSTM	SelfAttn/LSTM	SelfAttn/LSTM	-
CNN_T/CNN_C	LSTM	SelfAttn/CNN	SelfAttn/CNN	-
CNN_T/LSTM_C	LSTM	SelfAttn/LSTM	SelfAttn/CNN	-
LSTM_T/CNN_C	LSTM	SelfAttn/CNN	SelfAttn/LSTM	-

which is better for each dataset, comprising short and long texts. Furthermore, we can identify suitable ways to utilize online news data for stock price prediction

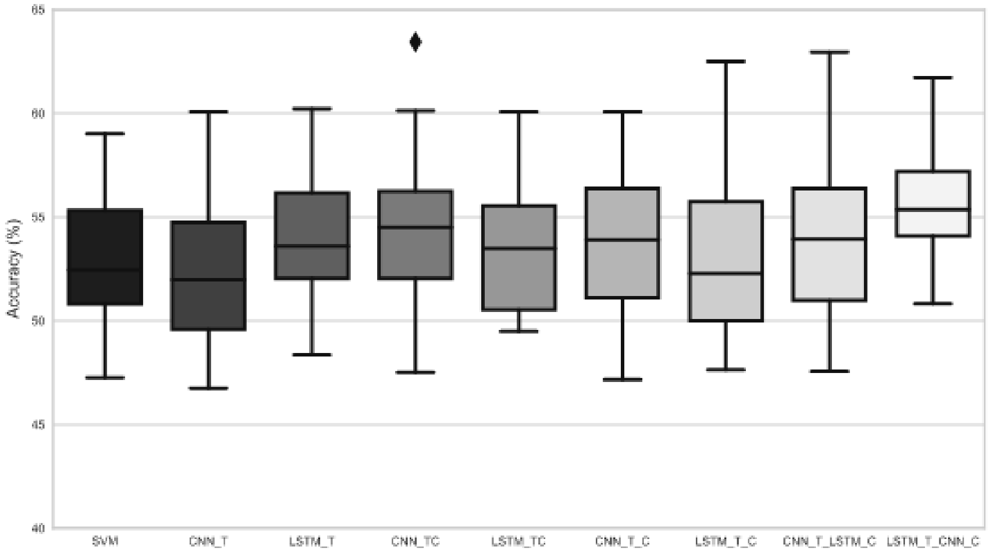
4.3. Experiments Result

Our final results is shown in <Table 3>. Each model was evaluated with 33 companies. The summarization of our final results are shown in <Table 4> and <Figure 4>. <Table 4> represents accuracy of final results. However, in stock prediction, precision and recall indicators are also important when evaluating models. The results of precision and recall are shown in <Appendix A.1> and <Appendix A.2>.

When using only technical indicators with SVM, despite showing better average accuracy then CNN_T, the highest accuracy is only 59.02%. This is the lowest accuracy among the models.

When using online news titles with technical indicators, as in CNN_T and LSTM_T, the main baseline models, selfAttn/LSTM shows much more effective results than selfAttn/CNN. This finding is in line with the findings of previous research(Liu, 2018). It shows that selfAttn/LSTM is a much more powerful method especially for stock price prediction with short texts. Furthermore, selfAttn/CNN shows even lower performance than usage of technical indicators with SVM only.

On the other hand, in CNN_TC and LSTM_TC, which use long texts as input data, selfAttn/CNN outperforms the other models. Overall, the CNN_TC model performed in second place among the models. Its highest accuracy was 63.44% which was the best accuracy among all models. It seems that selfAttn/CNN, a mechanism inherited from CBAM has great performance not only for images but also for long texts. Through the CNN_TC model, we conclude that online news content has useful information. However, even though titles and content are included in the data, LSTM_TC shows lower performance than LSTM_T. When attempting to use online news content, LSTM is not the best method. Although not included in our experimental design, vanilla CNN



<Figure 4> Model Comparison

yields the worst performance. When used with self-attention, CNN may be best for stock prediction. The method filters the more important words for stock price prediction and discards other words.

Between the sixth and seventh models, we gained some insight about online news data and the differing

<Table 3> Experiments Results for Each Model

MID Company	SVM	CNN_T	LSMT_T	CNN_TC	LSTM_TC	CNN_T/ CNN_C	LSTM_T/ LSTM_C	CNN_T/ LSTM_C	LSTM_T/ CNN_C
CJ LOGISTICS	55.74%	55.95%	56.55%	58.93%	57.14%	58.93%	62.50%	60.12%	60.12%
GS E&C	52.46%	50.33%	54.25%	60.13%	50.98%	57.52%	52.29%	50.98%	56.86%
Kakao	59.02%	60.08%	51.44%	60.08%	60.08%	60.08%	60.49%	62.96%	63.20%
KB Financial Group	53.28%	55.61%	57.65%	58.67%	49.49%	57.65%	50.00%	49.49%	54.59%
KT&G	58.20%	48.30%	57.39%	56.25%	50.00%	47.16%	47.73%	49.43%	59.09%
LGH&H	53.28%	50.83%	53.04%	47.51%	51.93%	50.28%	50.83%	49.17%	50.83%
LG ELECTRONICS	55.33%	56.79%	58.02%	55.97%	58.02%	56.38%	59.26%	57.20%	57.20%
LGCHEM	55.33%	53.60%	53.60%	51.80%	54.51%	53.60%	54.50%	55.86%	54.05%
NAVER	51.64%	59.75%	59.75%	57.68%	59.75%	59.75%	59.75%	61.41%	60.17%
POSCO	49.18%	49.59%	55.74%	52.05%	54.92%	53.28%	53.69%	55.74%	55.74%
S-Oil	56.15%	46.75%	59.09%	52.60%	55.84%	53.90%	52.60%	57.79%	53.15%
SKTelecom	52.05%	49.38%	49.38%	51.03%	52.27%	51.85%	52.26%	53.09%	53.91%
SK Innovation	54.10%	49.73%	53.01%	56.83%	50.00%	53.55%	49.73%	49.73%	54.10%
SK hynix	51.64%	54.73%	54.73%	55.97%	54.12%	54.73%	54.73%	56.38%	55.97%
KorZinc	55.33%	53.76%	60.22%	63.44%	50.54%	59.14%	56.99%	58.06%	55.91%
KiaMtr	48.36%	51.07%	52.36%	55.36%	49.79%	53.65%	47.64%	51.93%	55.36%
DaelimInd	49.18%	50.55%	48.35%	50.55%	49.73%	50.55%	51.65%	51.10%	54.40%
DSME	51.64%	47.03%	49.19%	52.97%	50.27%	49.19%	48.11%	47.57%	53.51%
SAMSUNG LIFE	54.92%	52.63%	52.11%	55.79%	53.69%	53.16%	52.11%	55.26%	60.53%
Samsung Elec	51.64%	48.36%	51.23%	48.36%	51.03%	48.36%	52.05%	52.46%	54.10%
Samsung HvyInd	48.36%	53.95%	53.95%	56.58%	53.62%	53.95%	53.95%	53.95%	53.95%
Celltrion	56.15%	55.74%	55.74%	55.74%	55.74%	55.74%	55.74%	55.74%	56.15%
ShinhanGroup	48.77%	53.69%	52.46%	54.51%	50.21%	56.15%	54.51%	52.87%	54.92%
S-1	56.97%	50.89%	49.11%	52.68%	52.24%	50.89%	50.00%	50.00%	53.57%
Yuhan	47.26%	47.26%	52.05%	54.11%	53.77%	47.26%	47.95%	51.37%	54.11%
KEPCO	52.05%	56.12%	57.65%	52.04%	56.38%	56.12%	56.63%	57.14%	56.63%
HanmiPharm	52.87%	54.17%	54.76%	54.76%	54.76%	57.74%	54.76%	56.55%	59.18%
HyundaiEng&Const	49.59%	46.81%	56.17%	54.04%	56.60%	56.17%	56.60%	56.17%	56.60%
HYUNDAIGLOVIS	52.05%	53.90%	51.06%	54.61%	51.77%	53.19%	49.65%	55.32%	57.45%
Mobis	53.69%	51.98%	52.97%	51.49%	53.47%	54.46%	49.01%	52.97%	54.61%
HYUNDAI STEEL	50.82%	56.35%	54.70%	52.49%	55.53%	56.91%	55.80%	55.25%	57.46%
KSOE	56.56%	51.12%	51.12%	49.78%	50.45%	51.12%	52.02%	49.78%	53.04%
HyundaiMtr	49.59%	47.13%	52.87%	47.54%	53.49%	49.59%	51.64%	48.77%	54.10%

methods. First, the LSTM_T/LSTM_C comparison shows better performance for some companies, even though it has lower average accuracy than LSTM_T and LSTM_TC. We conclude that a method such as LSTM should be applied to short texts. The application of LSTM to long texts, can yield lower results.

Second, CNN_T/CNN_C, which show lower performance than CNN_TC, even selfAttn/CNN identifies meaningful features in online news content, application of selfAttn/CNN to titles tends to lower the performance. We can verify that each method returns different features, so methods should be selected in careful consideration of data properties.

Lastly, CNN_T/LSTM_C, which is the final baseline model, we found better results when using selfAttn/CNN for titles and selfAttn/LSTM for content, than sixth and seventh models with higher average accuracy and higher accuracy in some companies. This suggests that each method analyzes different types of features and that these features contribute differently to stock price prediction. Nevertheless, this baseline model show lower accuracy than our final model, since selfAttn/CNN and selfAttn/LSTM are not the proper methods for predictive analysis in titles and content.

Based on these results, we designed our final model, LSTM_T/CNN_C. We chose selfAttn/LSTM for online news titles and selfAttn/CNN for online news content. Even though the model with the best accuracy of results among companies is CNN_TC as shown in <Figure 4>, the average accuracy per day in all companies is highest in LSTM_T/CNN_C. In addition, <Figure 4> shows that our final model has the lowest variance, implying that it may be applied universally. Our model shows almost 3.5% and 2% higher accuracy than our main baseline models, which were designed from previous methods, CNN_T and LSTM_T. Also, our model demonstrated higher accuracy per company except for 3 and 8 companies for each method.

To verify that our final model is the best method, we compared all the other models through paired sample t-tests against our model. The alternative hypothesis is reliable within 95% confidence, confirming that our final model performed better than other models. Each p-value can be checked in <Table 4>, and the t-test hypothesis can be written as:

$$\begin{gathered}\textit{Null Hypothesis } H_0 : u_{LSTM\_T/CNN\_C} = u_A \quad (A = \textit{other model}) \\ \textit{Alternative Hypothesis } H_1 : u_{LSTM\_T/CNN\_C} > u_A \quad (A = \textit{other model})\end{gathered}$$

<Table 4> Experiments Results for Each Model

Models	Max accuracy among all companies	Average Accuracy among all companies	p-value in paired t-test	p-value in proportions test
SVM	59.02%	52.82%	0.000	0.000
CNN_T	60.08%	52.24%	0.000	0.000
LSTM_T	60.22%	53.99%	0.002	0.011
CNN_TC	63.44%	54.31%	0.003	0.018
LSTM_TC	60.08%	53.39%	0.000	0.005
CNN_T/CNN_C	60.08%	54%	0.001	0.014
LSTM_T/LSTM_C	62.5%	53.25%	0.000	0.002
CNN_T/LSTM_C	62.96%	53.99%	0.000	0.020
LSTM_T/CNN_C	63.20%	55.89%	-	-

We also confirm the differences of accuracy of the final model and other models using proportions test (refer the last column of <Table 4>). The results are consistent with those of the paired sample t-test.

Ⅴ. Market Simulation

In this section, the performance of our model is evaluated through market simulation. First, our market simulation strategy will be introduced. And our model performance will be compared with LSTM_T and KOSPI index. In the process of simulation, we have considered transaction fees and tax for each transaction. We applied transactions fees for 0.015% and securities transaction tax as 0.1% since we only buy stocks that in KOSPI200.

5.1. Simulation Strategy

In. the market simulation, we used the 2018 data set, which were the test set of our experiment. For appropriate comparison, our model is compared with two models. First model is LSTM_T, which has shown the highest performance in previous researches(Liu, 2018) that utilized only news titles with selfAttn/LSTM. Second model is KOSPI index which is our baseline model.

Our stock trading strategy is as follows. Our model and LSTM_T predict the rise and fall of the stock price of the next day. In order to make the best use of our model, we chose a strategy to buy at opening price and to sell at the closing price of companies among 33 companies. The strategy can be specified by two conditions. First, we purchased stocks of the top- companies which have high ratio of articles predicting that stock prices would rise next day, and we bought stocks of all companies that are tied with the . However, if there are less than companies whose ratio of articles that predict to rise exceeds 50 percent, we buy only stocks of companies with a ratio of more than 50 percent. Second, if there are too few articles per company on that day, the ratio of articles that predict to rise can be possible to rise closely to 1. Furthermore, it can make a problem of reliability since rise or fall is not determined by one or two online news. So, we set a threshold of the number of news articles and in our simulation, we buy a stocks of companies that have at least 4 articles on that day.

5.2. Simulation Results

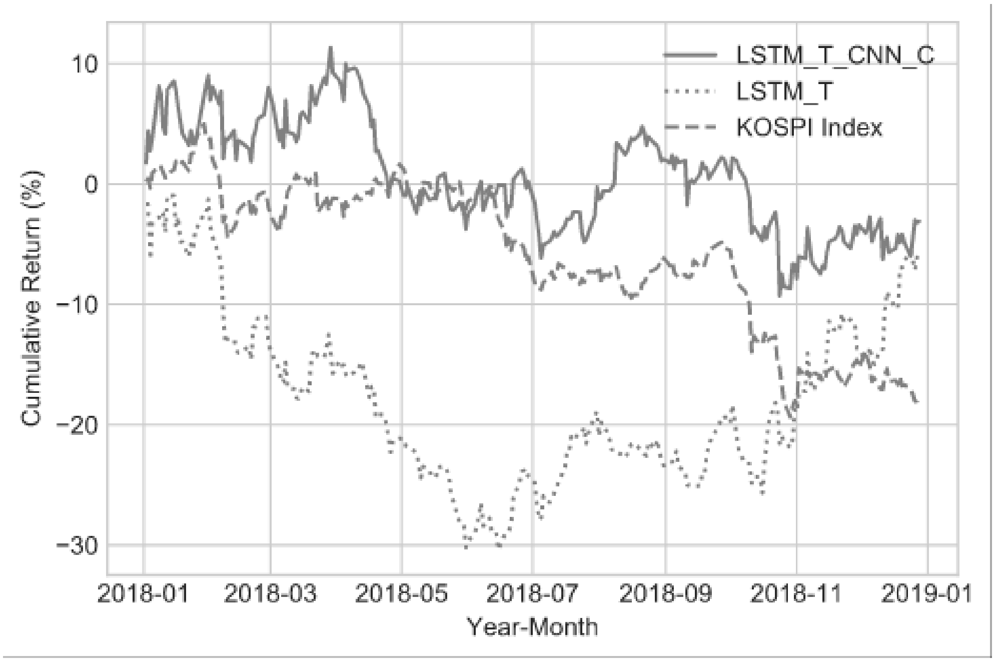
Through the simulation in stock market, we can confirm again that our proposed model is better than LSTM_T and KOSPI index, the baseline model. Simulation result when set N from 3 to 10 is shown in <Table 5> and simulation result when is visualized in <Figure 5>. In all cases except top-4, our proposed model performed better.

In <Figure 5>, despite start with lower rate of return in stable stock market, LSTM_T has shown better cumulative return than KOSPI index. However, LSTM_T/CNN_C, which was our final model shown much better performance. It rated 11.59% return rate in slightly ascending stock market, in 2018.02 to 2018.04 periods. Furthermore, our model has shown stable performance in dramatically falling market in 2018.06 to 2019.01. Finally, cumulative return of it was 14.11% higher than KOSPI index and 1.75% higher than LSTM_T.

Based on our strategy, the number of stocks of companies can be an important factor. LSTM_T buys and sells 523 times in 2018, in contrast, LSTM_T/CNN_C buys and sells 974 times which are almost double. However, stock market in Korea had shown dramati-

<Table 5> Simulation Results for Each Case

Model / Top-N	Kospi Index	LSTM_T	LSTM_T_CNN_C
Top-3	-17.53%	-8.82%	-6.65%
Top-4	-17.53%	-5.85%	-8.96%
Top-5	-17.53%	-5.17%	-3.42%
Top-6	-17.53%	-5.54%	-3.10%
Top-7	-17.53%	-5.54%	-4.42%
Top-8	-17.53%	-5.54%	-4.00%
Top-9	-17.53%	-5.54%	-3.86%
Top-10	-17.53%	-5.54%	-3.86%



<Figure 5> Market Simulation Result (Top - 5)

cally falling trends in 2018. So it is better not to buy stock, but LSTM_T/CNN_C shows stable performance through finding better companies in falling stock markets. Furthermore, even though transactions costs are considered, except for top-4, our proposed model haven shown better performance with higher number of transactions. As Korea stock market had shown ascending market in 2017 which were the validation set in our experiments, LSTM_T/CNN_C would be expected to show much better performance in ascending market.

Ⅵ. Discussion

This section discusses about main contributions of this paper. First, differences in properties between online news titles and content are discussed. Subsequently, we show why we applied selfAttn/CNN for content and selfAttn/LSTM for titles.

6.1. Differences between Titles and Content

Through the experiment results, we can conclude that online news content and titles have different characteristics. The main purpose of online news is information delivery. In the case of titles, information is compressed because the topics of documents must be represented accurately and succinctly. On the other hand, online news content, characterized by long sequences, may contain not only detailed information but also unnecessary information for any given topic.

If the purpose is to analyze online news, then the characteristics of online news must be considered. Rather than attaching content and titles, it is better to utilize the components separately for superior analysis. As the experimental results show, attaching content and titles, represented in CNN_TC and LSTM_TC, is the worse approach. To obtain better performance, it is important to apply techniques that match the properties of the data.

6.2. Differences between CNN and LSTM

As discussed in Section 2.4, CNN and LSTM have different characteristics. Because online news content tends to contain much unnecessary information, it is critical to weaken the influence of unnecessary information for purposes of predictive analysis. While LSTM strongly consider context information(Yin et al., 2017), the results can contain a lot of noise. As our experiments show, LSTM with online news content result in lower performance. Accordingly, we applied CNN architecture to online news content. Even though CNN weakly consider the context information, CNN are considered good at extracting local information(Zhou et al., 2015) and extracting semantic information(Yin et al., 2017). In conclusion, online news contents have much unnecessary information, the techniques extract key phrases and semantic meaning of key phrases very well.

On the other hand, LSTM architecture, which strongly considers context information, tends to obtain all information in data. Because online news titles have compressed information, it is necessary to extract features from titles for stock prediction. However, in a simple comparison of CNN_T and LSTM_T, CNN shows worse performance not only in a previous research(Liu, 2018), but also in our experiments. Thus, we conclude that LSTM architecture is more suitable than CNN for stock price prediction with online news titles.

Ⅶ. Conclusion

This paper contributes to our understanding of how to use online news data for stock price prediction. Previous studies have typically used only news titles. In this paper, however, we tried to use news articles content with news titles to stock prediction. Also, we proposed to use different deep-learning architectures for news titles and content separately. Through the experiments with Korea stock market data from 2014 to 2018, we revealed that even if it seems news titles and news contents similar information, each should be analyzed differently. We found that our proposed approach working better than previous approaches, because different deep-learning architectures for titles and content can extract proper features.

This paper also figured out how to use CNN and LSTM properly by applying them to Title and Contents data respectively. Title, who has relatively short with condensed information, said it is better to use LSTM than CNN, while contents, which has relatively long with unnecessary information, revealed that using CNN can bring stronger performance than LSTM. Our final proposed model consists of three parts, selfAttn/LSTM for titles, selfAttn/LSTM for content and bi-LSTM for technical indicators.

Nevertheless, this paper leaves some issues for further research. This paper mainly focuses on ways to utilize online news data and not on optimizing the performance of the models. Accordingly, we need to more work is needed to optimize the hyper-parameters of the proposed models. Also, there is room to improve prediction performance through more elaborate pre-processing, such as not only with word embedding, morphological analysis, and filtering of online news data, but also with preparing longer periods of data, especially for appropriate comparison to model that predicts only with technical indicators. To generalize the proposed approach, we need to perform experiments with stock markets data of other countries.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [2016-0-00562(R0124-16-0002), Emotional Intelligence Technology to Infer Human Emotion and Carry on Dialogue Accordingly].

<References>

[1] Bank, M., Larch, M., and Peter, G. (2011). Google search volume and its influence on liquidity and returns of German stocks. *Financial Markets and Portfolio Management, 25*(3), 239.

[2] Bharathi, S., Geetha, A., and Sathiynarayanan, R. (2017). Sentiment analysis of twitter and RSS news feeds and its impact on stock market prediction. *International Journal of Intelligent Engineering and Systems, 10*(6), 68-77.

[3] Bollen, J., Mao, H., and Zeng, X. (2010). Twitter mood predicts the stock market. *Journal of Computational Science, 2*(1), 1-8.

[4] Bruni, R. (2017). Stock market index data and indicators for day trading as a binary classification problem. *Data in Brief, 10*, 569-575.

[5] Chen, K., Zhou, Y., and Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. *In 2015 IEEE International Conference on Big Data(Big Data)*, IEEE, 2823-2824.

[6] Choudhry, R., and Garg, K. (2008). A hybrid machine learning system for stock market forecasting. *World Academy of Science, Engineering and Technology, 39*(3), 315-318.

[7] 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)*, 1415-1425.

[8] Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. *In Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2327-2333.

[9] Fu, T. C., Lee, K. K., Sze, D., Chung, F. L., and Ng, C. M. (2008). Discovering the correlation between stock time series and financial news. *In 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, 1, 880-883.

[10] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

[11] Hsu, C. M. (2011). A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. *Expert Systems with Applications, 38*(11), 14026-14036.

[12] Jeong, J. S., Kim, D. S., and Kim, J. W. (2015). Influence analysis of Internet buzz to corporate performance: Individual stock price prediction using sentiment analysis of online news. *Journal of Intelligence and Information Systems, 21*(4), 37-51.

[13] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). *A convolutional neural network for modelling sentences*. arXiv preprint arXiv:1404.2188.

[14] Kim, Y. (2014). *Convolutional neural networks for sentence classification*. arXiv preprint arXiv:1408.5882.

[15] LeCun, Y., and Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks*. MIT Press.

[16] Lee, S. H., and Lim, J. S. (2011). Forecasting KOSPI based on a neural network with weighted fuzzy membership functions. *Expert Systems with Applications, 38*(4), 4259-4263.

[17] Li, X., Huang, X., Deng, X., and Zhu, S. (2014). Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information. *Neurocomputing, 142*, 228-238.

[18] Li, Z., Yang, D., Zhao, L., Bian, J., Qin, T., and Liu, T. Y. (2019). Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 894-902.

[19] Lin, Y., Guo, H., and Hu, J. (2013). An SVM-based approach for stock market trend prediction. *In The 2013 International Joint Conference on Neural Networks(IJCNN)*, IEEE, 1-7.

[20] Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., and Bengio, Y. (2017). *A structured self-attentive sentence embedding*. arXiv preprint arXiv:1703.03130.

[21] Liu, H. (2018). *Leveraging financial news for stock trend prediction with attention-based recurrent neural network*. arXiv preprint arXiv:1811.06173.

[22] Malkiel, B. G. (1999). *A random walk down Wall Street: Including a life-cycle guide to personal investing*. WW Norton and Company.

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

[24] Mittal, A., and Goel, A. (2012). *Stock prediction using twitter sentiment analysis*. Standford University, CS229. Available at: http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

[25] Mizuno, H., Kosaka, M., Yajima, H., and Komoda, N. (1998). Application of neural network to technical analysis of stock market prediction. *Studies in Informatic and Control, 7*(3), 111-120.

[26] Nair, B. B., Mohandas, V. P., and Sakthivel, N. R. (2010). A decision tree-rough set hybrid system for stock market trend prediction. *International Journal of Computer Applications, 6*(9), 1-6.

[27] Park, J., Woo, S., Lee, J. Y., and Kweon, I. S. (2018). *Bam: Bottleneck attention module*. arXiv preprint arXiv:1807.06514.

[28] Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*(4), 2162-2172.

[29] Ren, G., Hong, T., and Park, Y. (2015). Multi-class SVM+ MTL for the prediction of corporate credit rating with structured data. *Asia Pacific Journal of Information Systems, 25*(3), 579-596.

[30] Schumaker, R. P., and Chen, H. (2009). A quantitative stock prediction system based on financial news. *Information Processing and Management, 45*(5), 571-583.

[31] Shah, V. H. (2007). Machine learning techniques for stock prediction. *Foundations of Machine Learning| Spring, 1*(1), 6-12.

[32] Shen, S., Jiang, H., and Zhang, T. (2012). *Stock market forecasting using machine learning algorithms*. Department of Electrical Engineering, Stanford University, Stanford, CA, pp. 1-5.

[33] Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C. (2018). Disan: Directional self-attention network for rnn/cnn-free language understanding. *In Thirty-Second AAAI Conference on Artificial Intelligence*.

[34] 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, 2*, 24-29.

[35] Song, Y. (2018). *Stock trend prediction: Based on machine learning methods*. Diss UCLA.

[36] Sundermeyer, M., Schlüter, R., and Ney, H. (2012). LSTM neural networks for language modeling. *In Thirteenth Annual Conference of the International Speech Communication Association*, 194-197.

[37] Teixeira, L. A., and De Oliveira, A. L. I. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications, 37*(10), 6885-6890.

[38] Vargas, M. R., Dos Anjos, C. E., Bichara, G. L., and Evsukoff, A. G. (2018). Deep learning for stock market prediction using technical indicators and financial news articles. *In 2018 International Joint Conference on Neural Networks(IJCNN)*, IEEE, 1-8.

[39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *In Advances in Neural Information Processing Systems*, 5998-6008.

[40] Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. *In Proceedings of the European Conference on Computer Vision(ECCV)*, 3-19.

[41] Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). *Comparative study of cnn and rnn for natural language processing*. arXiv preprint arXiv:1702.01923.

[42] Yoshihara, A., Fujikawa, K., Seki, K., and Uehara, K. (2014). Predicting stock market trends by recurrent deep neural networks. *In Pacific Rim International Conference on Artificial Intelligence*, Springer, Cham, 759-769.

[43] Yu, E., Kim, Y., Kim, N., and Jeong, S. R. (2013). Predicting the direction of the stock index by using a domain-specific sentiment dictionary. *Journal of Intelligence and Information Systems, 19*(1), 95-110.

[44] Yu, H., Chen, R., and Zhang, G. (2014). A SVM stock selection model within PCA. *Procedia Computer Science, 31*, 406-412.

[45] Zhai, Y., Hsu, A., and Halgamuge, S. K. (2007). Combining news and technical indicators in daily stock price trends prediction. *In International Symposium on Neural Networks*, Springer, Berlin, Heidelberg, 1087-1096.

[46] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *In Advances in Neural Information Processing Systems*, 649-657.

[47] Zhou, C., Sun, C., Liu, Z., and Lau, F. C. M. (2015). *A C-LSTM neural network for text classification*. arXiv preprint arXiv:1511.08630

<Appendix A.1> Experiments Results for Each Model (Precision)

MID Company	SVM	CNN_T	LSMT_T	CNN_TC	LSTM_TC	CNN_T/CNN_C	LSTM_T/LSTM_C	CNN_T/LSTM_C	LSTM_T/CNN_C
CJ LOGISTICS	43.52%	39.13%	48.75%	39.13%	34.78%	36.36%	34.41%	50.00%	48.00%
GS E&C	31.16%	53.13%	52.08%	53.13%	50.00%	54.10%	50.00%	50.00%	52.08%
Kakao	47.24%	43.21%	44.81%	56.76%	35.07%	34.84%	43.24%	56.76%	47.27%
KB Financial Group	43.47%	43.75%	52.86%	43.75%	34.62%	52.86%	45.03%	44.81%	44.33%
KT&G	45.64%	36.26%	52.05%	55.56%	41.03%	40.21%	40.00%	41.90%	55.56%
LGH&H	47.53%	48.05%	46.36%	48.05%	46.23%	48.10%	48.21%	48.41%	39.45%
LG ELECTRONICS	31.22%	28.57%	69.23%	28.57%	41.05%	34.78%	30.77%	37.50%	38.10%
LGCHEM	43.35%	41.92%	52.63%	40.89%	33.33%	46.22%	50.00%	41.92%	69.23%
NAVER	47.27%	62.50%	31.16%	62.50%	37.08%	38.24%	34.08%	39.81%	40.00%
POSCO	41.98%	41.56%	31.48%	41.56%	33.33%	37.93%	38.57%	40.00%	48.48%
S-Oil	49.29%	21.88%	50.00%	21.88%	47.17%	32.00%	41.75%	31.82%	43.21%
SKTelecom	39.66%	48.48%	62.83%	48.48%	46.69%	48.89%	43.61%	45.93%	46.39%
SK Innovation	38.64%	66.67%	57.63%	66.67%	54.55%	43.75%	46.15%	60.00%	58.93%
SK hynix	31.48%	39.80%	56.76%	46.28%	37.94%	40.83%	33.33%	36.36%	50.00%
KorZinc	43.08%	39.79%	37.93%	49.57%	38.46%	31.48%	48.09%	55.56%	66.67%
KiaMtr	48.59%	63.64%	47.11%	63.64%	30.22%	55.14%	46.58%	55.32%	62.83%
DaelimInd	31.89%	41.18%	55.43%	41.18%	46.58%	52.73%	53.23%	53.85%	49.25%
DSME	30.28%	46.61%	40.54%	46.61%	49.44%	48.03%	47.83%	47.83%	47.83%
SAMSUNG LIFE	46.23%	50.00%	48.98%	50.00%	56.76%	33.33%	41.94%	53.85%	45.45%
Samsung Elec	33.86%	48.56%	46.81%	48.56%	40.00%	48.56%	48.56%	41.61%	47.73%
Samsung HvyInd	34.55%	32.19%	49.13%	38.97%	34.79%	48.75%	47.46%	30.99%	52.05%
Celltrion	46.82%	39.76%	32.33%	39.76%	29.89%	35.31%	38.75%	42.09%	40.23%
ShinhanGroup	47.08%	57.63%	52.03%	57.63%	46.84%	55.45%	56.07%	55.84%	52.63%
S-1	36.94%	51.35%	53.13%	51.35%	52.38%	48.16%	42.22%	51.22%	54.55%
Yuhan	37.57%	43.67%	43.70%	42.86%	44.04%	33.83%	44.46%	34.41%	57.63%
KEPCO	34.05%	39.60%	37.09%	33.95%	50.00%	40.16%	22.22%	35.71%	44.44%
HanmiPharm	49.01%	37.50%	34.62%	37.50%	43.86%	31.58%	44.15%	66.67%	38.89%
HyundaiEng&Const	35.29%	41.22%	57.63%	41.22%	32.88%	45.53%	39.58%	38.64%	50.00%
HYUNDAIGLOVIS	36.06%	30.40%	48.33%	46.57%	36.36%	45.45%	34.15%	50.00%	48.98%
Mobis	46.89%	50.00%	38.85%	50.00%	48.84%	47.76%	47.03%	50.66%	48.33%
HYUNDAI STEEL	43.05%	54.55%	52.05%	54.55%	35.71%	39.71%	50.00%	46.67%	40.23%
KSOE	44.07%	39.54%	62.50%	55.43%	33.33%	41.38%	36.92%	33.33%	49.14%
HyundaiMtr	47.87%	40.23%	32.46%	40.23%	42.93%	38.18%	40.00%	43.24%	39.25%
Average per model	41.05%	44.31%	47.86%	46.75%	41.40%	42.72%	42.68%	45.84%	49.00%

<Appendix A.2> Experiments Results for Each Model (Recall)

MID Company	SVM	CNN_T	LSMT_T	CNN_TC	LSTM_TC	CNN_T/CNN_C	LSTM_T/LSTM_C	CNN_T/LSTM_C	LSTM_T/CNN_C
CJ LOGISTICS	21.50%	13.24%	1.47%	13.24%	11.76%	17.65%	4.41%	5.88%	17.65%
GS E&C	22.39%	22.67%	2.67%	22.67%	4.00%	44.00%	2.67%	6.67%	33.33%
Kakao	19.06%	39.16%	13.54%	33.70%	38.47%	42.33%	16.67%	8.33%	19.79%
KB Financial Group	30.25%	97.67%	93.02%	97.67%	20.93%	43.02%	38.09%	95.35%	50.00%
KT&G	33.86%	45.21%	52.05%	45.21%	43.84%	53.42%	49.32%	60.27%	53.42%
LGH&H	86.05%	86.05%	45.70%	86.05%	56.98%	88.37%	94.19%	88.37%	50.00%
LG ELECTRONICS	35.29%	1.98%	2.97%	1.98%	46.05%	7.92%	3.96%	5.94%	15.84%
LGCHEM	27.37%	39.00%	17.65%	41.13%	0.98%	53.92%	1.96%	8.82%	20.59%
NAVER	33.30%	5.21%	10.42%	5.21%	47.00%	13.54%	32.77%	36.77%	16.67%
POSCO	15.69%	29.63%	31.48%	29.63%	0.93%	30.56%	25.00%	11.11%	27.78%
S-Oil	22.75%	10.94%	50.00%	10.94%	39.06%	12.50%	67.19%	10.94%	54.69%
SKTelecom	30.02%	42.48%	62.83%	42.48%	33.72%	58.41%	51.33%	54.87%	68.14%
SK Innovation	62.83%	4.30%	5.38%	4.30%	25.81%	30.11%	6.45%	6.45%	35.48%
SK hynix	15.25%	36.36%	6.42%	48.71%	43.69%	30.68%	0.92%	40.92%	2.75%
KorZinc	35.66%	6.82%	11.36%	6.82%	34.09%	13.64%	9.09%	22.73%	34.36%
KiaMtr	27.03%	23.14%	47.11%	23.14%	48.10%	48.76%	28.10%	42.98%	49.38%
DaelimInd	20.25%	22.83%	55.43%	22.83%	36.96%	31.52%	35.87%	30.43%	35.87%
DSME	38.36%	62.50%	32.91%	62.50%	37.40%	82.95%	36.38%	37.39%	36.05%
SAMSUNG LIFE	15.81%	1.12%	13.48%	1.12%	23.60%	1.12%	29.21%	39.33%	39.33%
Samsung Elec	18.58%	31.24%	44.25%	45.54%	1.69%	30.29%	47.07%	37.76%	88.98%
Samsung HvyInd	33.39%	36.12%	46.78%	41.45%	43.04%	46.14%	37.55%	44.65%	33.59%
Celltrion	48.60%	32.61%	46.91%	44.32%	49.13%	48.60%	42.82%	48.13%	80.37%
ShinhanGroup	45.53%	55.28%	52.03%	55.28%	30.08%	45.53%	48.78%	34.96%	32.52%
S-1	37.71%	38.09%	15.79%	35.88%	77.19%	33.48%	33.33%	73.68%	31.58%
Yuhan	17.54%	32.78%	43.75%	41.63%	40.82%	42.85%	32.16%	48.90%	42.19%
KEPCO	35.25%	39.43%	34.67%	42.89%	9.41%	39.96%	2.35%	5.88%	42.35%
HanmiPharm	35.30%	8.00%	25.33%	8.00%	46.72%	16.00%	30.22%	2.67%	18.67%
HyundaiEng&Const	27.14%	59.80%	0.98%	59.80%	36.59%	54.90%	18.63%	16.67%	48.04%
HYUNDAIGLOVIS	16.20%	1.54%	12.31%	1.54%	6.15%	15.38%	21.54%	33.85%	36.92%
Mobis	54.90%	2.08%	45.26%	2.08%	21.88%	46.82%	90.63%	80.21%	60.42%
HYUNDAI STEEL	33.90%	7.59%	12.66%	7.59%	6.33%	34.18%	31.65%	8.86%	44.30%
KSOE	27.53%	44.17%	5.56%	36.09%	0.93%	40.33%	22.22%	3.70%	79.63%
HyundaiMtr	32.20%	30.70%	32.46%	30.70%	43.07%	36.84%	31.58%	28.07%	36.84%
Average per model	32.01%	30.60%	29.53%	31.88%	30.50%	37.45%	31.03%	32.77%	40.53%

◆ About the Authors ◆

Sang Hyung Jung

Sang Hyung Jung is an undergraduate student in the School of Business at Hanyang University, Seoul, Korea. His research interests include Financial engineering, natural language processing, data engineering and deep learning.

Gyo Jung Gu

Gyo Jung Gu is an undergraduate student in the Department of Finance and Department of Computer Science at Hanyang University, Seoul, Korea. His research interests are deep learning, computer vision and financial engineering.

Dongsung Kim

Dongsung Kim is a postdoctoral research at the School of Business, Hanyang University, Seoul, Korea. He received a Ph.D in Management Information Systems from School of Business, Hanyang University. His research interests are focused on data mining, business analytics, sentiment analysis, application of machine learning techniques, and social network analysis.

Jong Woo Kim

Jong Woo Kim is a professor at the School of Business, Hanyang University, Seoul, Korea. He received B.S. degree from the Department of Mathematics at Seoul National University, Seoul, Korea. He received his M.S. and Ph.D. degrees, respectively, from the Department of Management Science, and the Department of Industrial Management at Korea Institute of Science and Technology (KAIST), Korea. His current research interests include intelligent information systems, data mining applications, social network analysis, text mining application, collaborative systems, and e-commerce recommendation systems. His papers have been published in *Expert Systems with Applications*, *Cyberpsychology Behavior and Social Networking*, *Computers in Human Behavior*, *Information Systems Frontiers*, *International Journal of Electronic Commerce*, *Electronic Commerce Research*, *Mathematical and Computer Modeling*, *Journal of Intelligent Information Systems*, and other journals.

Submitted: March 10, 2020; 1st Revision: July 11, 2020; Accepted: August 26, 2020