Applied Soft Computing 130 (2022) 109673

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Factors affecting text mining based stock prediction: Text feature representations, machine learning models, and news platforms

Wei-Chao Lin [a,b,c], Chih-Fong Tsai [d,*], Hsuan Chen [d]

[a] Department of Information Management, Chang Gung University, Taoyuan, Taiwan
[b] Graduate Institute of Management, Chang Gung University, Taoyuan, Taiwan
[c] Department of Thoracic Surgery, Chang Gung Memorial Hospital at Linkou, Taoyuan, Taiwan
[d] Department of Information Management, National Central University, Taoyuan, Taiwan

ARTICLE INFO

Article history:
Received 9 October 2021
Received in revised form 18 August 2022
Accepted 18 September 2022
Available online 5 October 2022

Keywords:
Machine learning
Text mining
Feature representation
Stock prediction
Financial news

ABSTRACT

Text mining techniques have demonstrated their effectiveness for stock market prediction and different text feature representation approaches, (e.g., TF–IDF and word embedding), have been adapted to extract textual information from financial news sources. In addition, different machine learning techniques including deep learning have been employed to construct the prediction models. Various combinations of text feature representations and learning models have been applied for stock prediction, but it is unknown which performs the best or which ones can be regarded as the representative baselines for future research. Moreover, since the textual contents in the financial news articles published on different news platforms are somewhat different, the effect of using different news platforms may have an impact on prediction performance so this is also examined in the experiments comparing eight different combinations comprised of two context-free and two contextualized text feature representations, i.e. TF–IDF, Word2vec, ELMo, and BERT, and three learning techniques, i.e. SVM, CNN, and LSTM. The experimental results show that CNN+Word2vec and CNN+BERT perform the best. The textual material is taken from three public news platforms including Reuters, CNBC, and The Motley Fool. We found that the learning models constructed and the news platforms used can certainly affect the prediction of stock prices between different companies.

1. Introduction

Stock prediction has long been regarded as a very interesting and important research problem in finance, economic, information technology, etc. One major objective is to develop effective prediction models, which can correctly forecast whether future stock prices will rise or fall [1–3].

To achieve this objective, depending on the input features used to develop the prediction models, there are three types of methods. The first one, the earliest method, is based on the extraction of related indicators through fundamental and technical analysis with the former being concerned with the company that is selling the stock as opposed to the actual stock and the latter focusing on prediction of the future price from past and present stock prices [4]. The second type of method is based on the application of text mining techniques to extract textual features from financial news or tweets. Specifically, natural language processing techniques are used to analyze the semantic, sentiment, and/or event described in the texts [5,6]. The third type of method focuses on generating candlestick charts to extract (low-level or visual) image features to facilitate the forecasting of stock movements [7–9].

In this paper, we focus on the second type of method based on the use of financial news. In the literature, the extraction of textual features from financial news or tweets is usually accomplished through the term frequency–inverse document frequency (TF–IDF) technique. It calculates the number of times a term occurs in a document and distinguishes between the weights of terms that occur very frequently in the document set and the weights of terms that occur rarely [10]. Another approach is based on the bag-of-words (BOW) model, where a document is represented as the bag of words, in which the frequency of occurrence of each word is calculated for the bag [11]. The TF–IDF and BOW text feature representation approaches are widely used to construct stock price prediction models in combination with support vector machine (SVM) or with multiple kernel learning (MKL) [12–20].

Different from the 'traditional' machine learning techniques, such as SVM, deep learning techniques, such as convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory (LSTM), have also been used to predict stock prices in recent years (Ding et al. 2015; [13,15,16,21–29]).

* Corresponding author.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

https://doi.org/10.1016/j.asoc.2022.109673
1568-4946/

In particular, the text feature representations applied for constructing deep learning models are based on some text embedding methods, such as word, sentence, and event embedding, whose aim is to translate a high-dimensional vector into a low-dimensional space. That is, an embedding captures specific semantics of the input by placing semantically similar inputs close together in the embedding space [30].

Therefore, depending on the text feature representation methods used, the learning models constructed can be different. This raises several unanswered questions that have never been fully explored. First, since there are a number of different text feature representation methods which should be used to process the inputs, whether for traditional machine learning or deep learning techniques, the first question is: which specific text feature representation method combined with a learning model performs the best in terms of stock prediction?

Second, since the results of text feature representation are quite dependent on the textual content of news articles per se, and different news platforms may use different language to describe the same issue for the same company on the same day, (c.f. Fig. 10), the same combinations of text feature representations and learning models may perform differently when applied to news articles collected from different platforms. Therefore, the second question addressed in this study is: will the news articles collected from different platforms affect the performance of the stock prediction models?

To answer these two questions, four different text feature representations are compared, which are TF–IDF, Word2vec, ELMo, and BERT, the first two being context-free representation methods and the second two being contextualized representation methods (c.f. Section 2.1). For the learning models, one representative traditional machine learning technique, SVM, and two popular deep learning techniques, CNN and LSTM, are employed. Finally, the news articles are collected from Reuters, CNBC, and, The Motley Fool for performance comparison. In particular, we selected news articles related to four companies, i.e. Apple, Toyota, Chevron, and the Bank of America, belonging to the information technology, consumer discretionary, energy, and financial industries, respectively, are considered.

The rest of this paper is organized as follows. Section 2 gives an overview of the text feature representation methods and learning techniques used in this paper. In addition, a comparison of related works is provided in terms of the extracted textual features, the learning models constructed, the chosen news platforms, etc. Section 3 describes the experimental procedures applied to answer the two research questions. Section 4 presents the experimental results and Section 5 concludes the paper.

2. Literature review

2.1. Text feature representation

2.1.1. BOW

The BOW model is a widely used representation approach for document classification in which a text is represented as a bag of words. Various measures can be used to characterize and transform the text, the most common being based on the frequency of each word, i.e. the number of times a term appears in the text [31].

Since the BOW model does not consider the order of the terms, n-gram models are often used to represent the spatial information, with the bigram model commonly used for analyzing the sequence of two adjacent terms. The bigram model parses the text into two adjacent terms as a unit and stores the frequency of each unit [32].

2.1.2. TF–IDF

The term frequency–inverse document frequency (TF–IDF) statistic is applied to measure how important a word (or term) is within a document in a collection or corpus. In particular, the term frequency measures the number of times (i.e. frequency) each term occurs in each document, whereas the inverse document frequency focuses on determining the logarithmically scaled inverse fraction for documents that contain the term. Thus, the lower the number of documents containing a specific term, the higher the IDF and the more important the term is [31].

The TF–IDF feature representation can be obtained by

$$\text{TF–IDF}_{t,d} = tf_{t,d} \times \log(\frac{N}{df_t}) \quad (1)$$

where $tf_{t,d}$ means the frequency of term t in document d; df_t for the number of documents containing term t; and N is the total number of documents in the corpus.

As a result, terms that occur frequently in a document but are rather rare in all other documents have a high TF–IDF value. On the other hand, if a term occurs frequently in a document, but also appears very often in all other documents, its TF–IDF value is low. In other words, lower TF–IDF values reflect lower discriminative power to distinguish between different classes of documents.

2.1.3. Word embedding

Word embedding, such as Word2vec [33] and global vectors (GloVe) [34], has been introduced to overcome some of the limitations of the TF–IDF, such as the dependence of the extracted features on the dictionary of terms, the sparsity problem of high dimensionality, the lack of semantic relationships between the extracted terms, etc.

In Word2vec, a neural network model is used to learn word associations from a large corpus of text, which can detect synonymous words or suggest additional words for a partial sentence. Specifically, continuous BOW (CBOW) and skip-gram can be utilized. CBOW predicts the current word from a window of surrounding context words whereas skip-gram uses the current word to predict the surrounding window of context words. The computational complexity of CBOW is lower than that of skip-gram, but for infrequent words, skip-gram can produce more effective feature representation.

Unlike Word2vec, which is a prediction based technique, the global vector (GloVe) for word representation is a count-based technique for distributed word representation. It uses an unsupervised learning algorithm to map words into a meaningful space where the distance between words is related to semantic similarity. In particular, aggregated global word–word co-occurrence statistics from a corpus are used to represent linear substructures of the word vector space.

In addition to the 'static' feature representation like Word2vec or GloVe, contextualized word embedding techniques proposed recently can provide meaningful representations for words and their contexts. They are based on the bidirectional long short-term memory (BLSTM) model trained for producing context representation (context2vec) [35,36]. The embeddings from language models (ELMo) [37] and bidirectional encoder representations from transformer (BERT) [38] are two contextualized feature representation methods that were proposed recently.

In particular, ELMo models the complex characteristics of word use (e.g., syntax and semantics) and how these uses very across linguistic contexts (i.e. to model polysemy). Each word is assigned a representation that is a function of the entire input sentence. The vectors derived from a BLSTM are used for training with a coupled language model (LM) objective on a large text corpus. In other words, the ELMo representation is a function of all of the internal layers of the bidirectional LM.

The BERT method is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning both the left and right contexts in all layers. In particular, BERT is based on using a masked language model (MLM) pre-training objective, which randomly masks some of the words from the input in order to predict the original vocabulary id of the masked word based only on its context. Therefore, it enables the representation to fuse the left and the right context, which allows us to pre-train a deep bidirectional transformer.

2.2. Supervised learning techniques

2.2.1. SVM

The support vector machine (SVM) is one of most widely employed classifiers for various classification and regression problems, which is based on the statistical learning frameworks proposed by Vapnik [39]. Given a set of training examples, in which each example belongs to one of two pre-defined categories, i.e. (x_1, y_1), (x_2, y_2),..., (x_m, y_m), where $x_i \in R^d$ denotes the vectors in a d-dimensional feature space and $y_i \in \{-1, +1\}$ is a class label, the SVM algorithm builds a non-probabilistic binary linear classifier that assigns new (or unknown) examples to one category or the other.

More specifically, the training examples are mapped into a higher dimensional feature space than *d,* where a hyperplane is constructed to linearly separate the examples in the two predefined categories and maximize the width of the gap between these two categories. BOW and TF–IDF are the most widely used text feature representations applied in combination with SVM for stock prediction in related works (Ding et al. 2015; [13,14,16,17,19,20,24]).

2.2.2. CNN

In deep learning, a convolutional neural network (CNN) is a class of deep neural networks composed of an input layer, some hidden layers, and an output layer. In a CNN, the hidden layers include layers that perform convolutions, which typically include convolutional layers, pooling layers, and fully connected layers [40,41].

To perform the text classification task, the input layer takes the pre-trained word embedding representation of the text as the input which is represented by a matrix. That is, each row of the matrix corresponds to one token, typically a word (or a character), so that each row is a vector that represents a word. For example, a 10-word sentence using a 100-dimensional embedding produces a 10×100 matrix.

Then, a filter/kernel is applied for feature extraction of the input matrix, where the feature dimensionality is further reduced in order to reduce the complexity and computational complexity required by the pooling layer, to retain only the maximum value in a feature vector. Finally, in the fully connected layers, where each node of one layer is connected to each node of the other layer, the result of the pooling layer is taken as the input for performing the activation function on the class labels that will assign values for each class.

In the literature, word, sentence, or event embedding representations have been used in combination with CNN for stock prediction (Ding et al. 2015; [13,15,16,23,27,29]). While most studies did extract the Word2vec features, only Liu et al. [16] considered the ELMo feature representation for CNN.

2.2.3. RNN/LSTM

A recurrent neural network (RNN) is one type of artificial neural network, where the connections between nodes form a directed graph along a temporal sequence and the computation graph contains directed cycles. Differing from the regular feed-forward neural networks, such as CNN, which only consider the current input, information in the RNN travels in loops from layer to layer so that the state of the model is influenced by its previous states. In addition, it has a memory that allows the model to store information about its past computations. This sequential memory is preserved in the recurrent network's hidden state vector and represents the context based on the prior inputs and outputs. Therefore, the RNN demonstrates the dynamic temporal behavior and model sequences of input–output pairs [42].

However, the RNN models do have some disadvantages. For example, the computation time is slow due to their recurrent nature and they are inherently deficient at retaining information over long periods of time. To overcome these shortcomings, the long short-term memory (LSTM) architecture has been proposed, which can identify key information and retain it over long periods of time, and then use this information as necessary later in the sequence. A common LSTM is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [43].

In other studies, word, sentence, and event embedding representations have been used in combination with RNN for stock prediction [13,21,23,29] or LSTM [15,16,22–24,26–28]. In most studies, the Word2vec features are extracted. Only Ma et al. [24] considered the BERT representation with LSTM.

2.3. Comparison of related works

A summary comparison of recent related works focusing on extracting textual features, especially from financial news articles, for the purpose of stock prediction, appears in Table 1. In particular, the feature representation methods used, learning models constructed, news platforms considered, prediction tasks, and evaluation metrics are compared. Note that other studies employing sentiment analysis as one of the NLP techniques and/or collect tweets as the data source are not considered here.

As can be seen in Table 1, the most widely used classification techniques are SVM, CNN, RNN, and LSTM, in which SVMs are generally combined with TF–IDF and BOW feature representations, whereas CNN, RNN, and LSTM are used with word, sentence, or event embedding feature representations. However, very few studies go so far as to compare the prediction performance of these combinations, i.e. TF–IDF/BOW representations with SVM and embedding representations with CNN/RNN/LSTM. Specifically, although there have been some works focusing on a comparison of the performance between different feature representations and classifiers, see for example, [13,15,16], the different embedding representations, including the context-free representation, i.e. Word2vec, and contextualized representations, i.e. ELMo and BERT, have not been fully examined. In this paper, we consider the feature representation techniques including TF–IDF, Word2vec, ELMo, and BERT. Moreover, the classification techniques include SVM, CNN, and LSTM.

It should also be noted that most related works collect news articles from Reuters, with Bloomberg being the second most popular platform. Most experiments consider the collection of news articles from only one platform. In some cases, efforts have been made to crawl articles from two or more platforms, such as Reuters and Bloomberg (Ding et al. 2015; [13,16,22,27,44]), but the collected articles are usually collected together for stock prediction. Thus, it is not known whether using articles published on different platforms would result in differences in prediction performance. In this paper, three news platforms are considered for performance comparison, which are Reuters, CNBC, and The Motley Fool, whose financial news are publicly downloadable.

Table 1
Comparison of related works.

Work	Feature representations	Learning models	News platforms	Prediction tasks	Evaluation metrics
Seong and Nam [20]	TF–IDF[a]	SVM, MKL[b]	Naver (Korea)	Stock price rise and fall	Accuracy
Guo and Tuckfield [15]	TF–IDF, word embedding	NB[c], RF[d], GBM[e], CNN[f], LSTM[g]	Kaggle	Stock price rise and fall	Accuracy
Kilimci and Duvar [23]	Word embedding	CNN, RNN[h], LSTM	Twitter and Bigpara, Public Disclosure Platform, Mynet Finans (Turkey)	Stock price rise and fall	Accuracy
Liu et al. [16]	BOW, ELMo[i], event embedding	SVM, CNN, LSTM	Reuters, Bloomberg	Stock price rise and fall	Accuracy, correlation coefficient
Ma et al. [24]	BOW[j], word/event embedding, BERT[k]	LSTM	Various websites in China	Stock price rise and fall	Accuracy, correlation coefficient
Long et al. [17]	BOW	SVM	ifeng.com (China)	Stock price rise and fall	Accuracy
Nam and Seong [18]	TF–IDF	MKL	naver.com (Korea)	Stock price rise and fall	Accuracy, F1
Souma et al. [28]	Word embedding, sentiment	LSTM	Reuters	Return rate	Accuracy
Hu et al. [21]	Word embedding	RF[l], MLP[m], RNN	eastmoney.com, finance.sina.com.cn (China)	Stock price rise and fall	Accuracy
Minh et al. [44]	TF–IDF, sentiment	GRU[n], TGRU[o]	Reuters, Bloomberg	Stock price rise and fall	Accuracy, precision, recall
Oncharoen and Vateekul [25]	Event embedding, technical indicators	CNN+LSTM	Reuters, Reddit World News	Stock price rise and fall	Accuracy, annualized return
Sardelich and Manandhar [26]	Word/sentence embedding	LSTM, BiLSTM[p]	Reuters	Stock prices	MSE[q], MAE[r]
Shi et al. [27]	Word embedding	CNN, LSTM	Reuters, Bloomberg, Twitter	Stock price rise and fall	Accuracy
dos Santos Pinheiro and Dras [13]	character/event/word/sentence embedding, BOW	CNN, RNN, SVM	Reuters, Bloomberg	Stock price rise and fall	Accuracy
Huynh et al. [22]	Word embedding	LSTM, GRU, BGRU[s]	Reuters, Bloomberg	Stock price rise and fall	Accuracy
Vargas et al. [29]	Sentence embedding	CNN, RNN	Reuters	Stock price rise and fall	Accuracy
Ding et al. (2015)	BOW, event embedding	SVM, CNN	Reuters, Bloomberg	Stock price rise and fall	Accuracy, correlation coefficient
Nassirtoussi et al. [19]	TF–IDF, sentiment	SVM	MarketWatch.com	Stock price rise and fall	Accuracy, precision, recall
Fortuny et al. [14]	BOW, sentiment, technical indicators	SVM	Flemish newspapers (Belgium)	Stock price rise and fall	Accuracy
The proposed study	TF–IDF, Word2Vec, ELMo, BERT	SVM, CNN, LSTM	Reuters, CNBC, The Motley Fool	Stock price rise and fall	AUC[t]

[a]TF–IDF: term frequency–inverse document frequency.
[b]MKL: multiple kernel learning.
[c]NB: naïve Bayes.
[d]RF: random forest.
[e]GBM: gradient boosting machine.
[f]CNN: convolutional neural network.
[g]LSTM: long short-term memory.
[h]RNN: recurrent neural network.
[i]ELMo: embeddings from language models.
[j]BOW: bag-of-words.
[k]BERT: bidirectional encoder representations from transformers.
[l]RF: random forest.
[m]MLP: multi-layer perceptron.
[n]GRU: gated recurrent unit.
[o]TGRU: two-stream GRU.
[p]BiLSTM: bi-directional LSTM.
[q]MSE: mean-square error.
[r]MAE: mean absolute error.
[s]BGRU: bidirectional GRU.
[t]AUC: area under the ROC (receiver operating characteristic) curve.

For prediction tasks, most studies have investigated the rise and fall of stock prices as a classification problem for stock prediction. Different from those studies [26] and [28] focused on forecasting stock prices and return rates, respectively, as a regression problem. Since stock prediction models can be regarded as the classification or regression problem, in this paper, we follow the experimental setup of most studies to construct the prediction models to forecast stock price rise and fall.

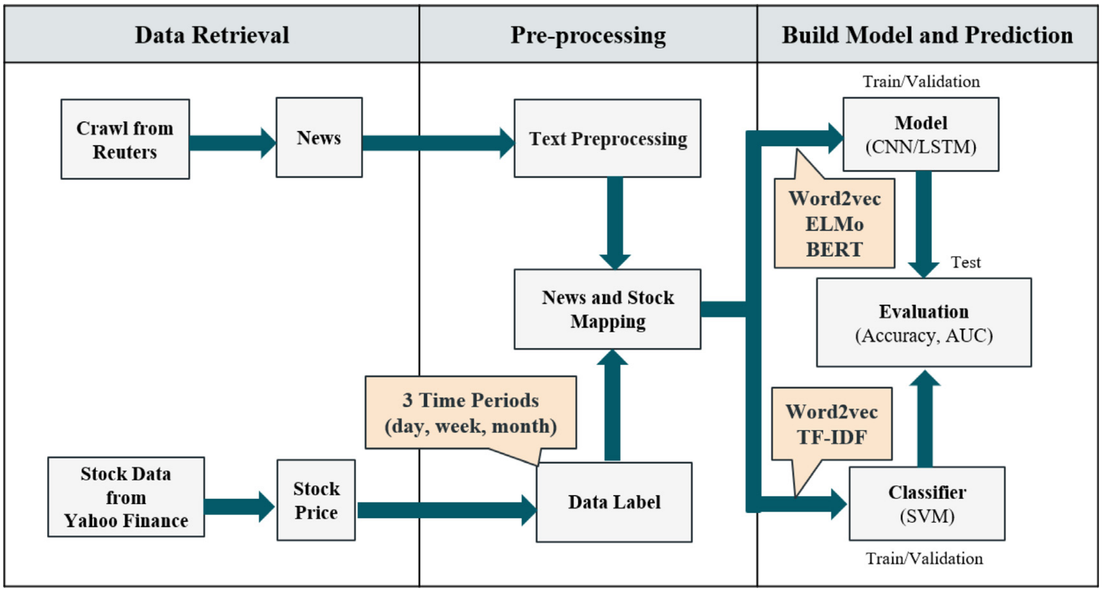


Fig. 1. The procedure of text mining based stock prediction.

Finally, the average prediction accuracy of the evaluation metrics is usually considered in related works. Only Minh et al. [44], Nam and Seong [18], and Nassirtoussi et al. [19] used additional metrics, such as F1, precision, and recall. Note that although the collected articles are likely associated with either the rise or fall of stock prices, making this a two-class classification problem, the amount of training data in the two classes is likely to be different. This can lead to the class imbalance problem which could affect the performance of the learning models. In this case, the average prediction accuracy is not a suitable metric to use for performance evaluation of different models. The F1 measure and AUC (i.e. area under the ROC (receiver operating characteristic) curve), are more reliable metrics [45]. Therefore, in this paper, the AUC rates of different prediction models are examined.

3. Research methodology

3.1. The procedure of text mining based stock prediction

In general, there are three steps to construct a text mining based stock prediction model. The first one is data retrieval and collection, which focuses on collecting the target financial news from specific news platforms as well as their corresponding stock prices. The second one is the data pre-processing step by related NLP techniques. That is, important textual information is extracted from the collected financial news for a specific text feature representation. The final component is to use the text feature representation to construct the prediction models by different machine and deep learning techniques. Fig. 1 shows the flow diagram composed of the three steps including the techniques used in this paper to construct the text mining based stock prediction models.

For the data retrieval and collection step, related financial news is crawled from Reuters, which is a widely used news platform for stock prediction (c.f. Table 1). Moreover, the news articles of two other platforms, i.e. CNBC,[1] and The Motley Fool[2] will also be crawled in order to examine the models' prediction performances by using these three news platforms (c.f. Section 4.2).

In the data pre-processing step, four different text feature representation methods are employed to extract related textual features from the crawled news articles, which are TF–IDF, Word2vec, ELMo, and BERT. Particularly, TF–IDF and Word2vec belong to the category of context-free representation methods and ELMo and BERT for the contextualized representation methods. Examining the models' prediction performances based on these text feature representations can identify the best one for stock prediction.

Finally, the model construction and prediction step is to employ different machine and deep learning techniques, including SVM, CNN, and LSTM, for performance comparison in order to find out the best combination of the text feature representation and prediction model. The following subsections describe these three steps in detail.

3.2. The dataset

Relevant news articles were crawled from the Reuters platform and the stock price information was collected from Yahoo Finance.[3] The time period for published news was between January 1, 2012 and December 31, 2019. News was collected related to four different companies belonging to four different industries based on the Global Industry Classification Standard (GICS) definitions, which are Apple (Information Technology), Toyota (Consumer Discretionary), Chevron (Energy), and the Bank of America (Financial). Note that these four companies are in the top 100 out of the world's 2000 largest public companies according to Forbes.[4] Table 2 lists the datasets of the four companies in terms of the total number of news articles, the total number of unique terms, and the average length (i.e. words) per article.

The class labels for stock price rise and fall cover short- and medium-term predictions, which are based on the prediction for the next day, week, and month. That is, the class label for a news article published on the ith date for the next day stock prediction is calculated by

$$result = closing\ price_{i+1} - closing\ price_i \tag{2}$$

1 https://www.cnbc.com/world/?region=world.
2 https://www.fool.com/.
3 https://drive.google.com/drive/folders/1-Xcj-1FK5TMZ--pfFUzWSw1DZpyu-9NK?usp=sharing.
4 https://www.forbes.com/global2000/#25835d08335d.

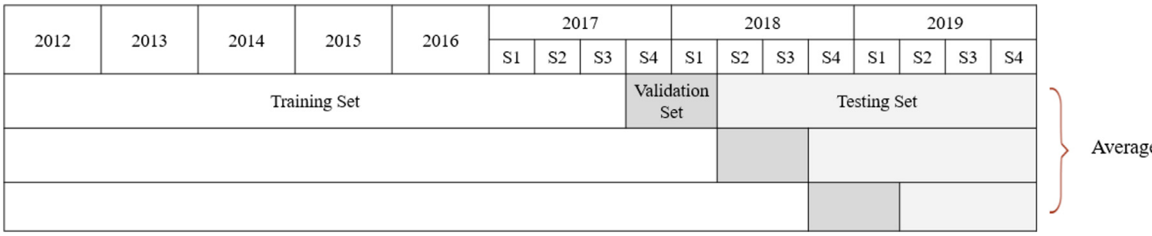


Fig. 2. The sliding window strategy.

Table 2
Dataset information.

	# of news articles	# of unique terms	Average length
Apple	5644	28,887	386
Toyota	4768	28,900	369
Chevron	5161	34,056	463
Bank of America	3431	24,353	421

Therefore, if the result, i.e. class label, is positive, the stock price will rise; otherwise, if the result is negative, the stock price will fall. Similarly, the class labels for predicting the stock prices for the next week and month of the ith date are based on the *closing price$_i$* subtracted from the closing prices of the next week and month of the ith date, respectively.

The dataset for each company is divided into training, validation, and testing subsets based on the sliding window strategy, as shown in Fig. 2. For example, for the first sliding window, the training subset is composed of news articles published from 2012 to the third season of 2017, the validation subset is from the fourth season of 2017 to the first season of 2018, and the testing subset is from the second season of 2018 to the end of 2019. The three sliding window strategies produce three prediction results, and the final predication performance of the learning model is based on the average of these three prediction results. The area under the ROC curve (AUC) is examined to demonstrate the prediction performance of the learning model.

3.3. Text feature representations

Following the collection of news articles for each of the four chosen companies, a data pre-processing step is performed. It includes removing related web links and punctuation (e.g. %, $, etc.), and reversing abbreviations (e.g. 'can't' and 'don't' to 'can not' and 'do not', respectively). Moreover, for the TF–IDF feature representation, stop word removal and lemmatization are executed.

In this study, four different text feature representation methods are compared, namely, TF–IDF, Word2vec, ELMo, and BERT. Specifically, TfidfVectorizer of Scikit-Learn[5] is used to extract the TF–IDF features; the Word2vec features are extracted based on the Google pre-trained model,[6] resulting in 300 feature dimensions per article.

The ELMo features are extracted based on the AllenNLP pre-trained models,[7] to produce 1024 feature dimensions per article, and BERT feature extraction is carried out based on Google's[8] Tensorflow implementation and its pre-trained models, resulting in 768 feature dimensions per article.

It should be noted that to extract textual features, news titles and contents can be extracted individually for different feature representations, i.e. title embedding and content embedding [46]. In order to find out the best feature representation for the later experiments (c.f. Section 4), a pilot study is conducted. Fig. 3 shows the average AUC rates of SVM, CNN, and LSTM based on the title and content based feature representation for prediction of the stock prices for next day, week, and month for Apple. Note that related parameters of the learning models and the computing environment are described in Section 3.4.

As we can see, in most cases using the news contents produces better prediction performance, except for SVM with Word2vec and CNN and LSTM with Word2vec for next month prediction. Therefore, the news contents will be used in subsequent experiments.

3.4. Prediction models

In addition to the deep learning techniques used in this paper, i.e. CNN and LSTM, as there are a number of different 'traditional' machine learning techniques, a pilot study is conducted to compare some well-known techniques including naïve Bayes (NB), k-nearest neighbor (KNN), and SVM.[9] The result of the pilot study allows us to find out the best traditional machine learning model for the later experiments (c.f. Section 4). Fig. 4 demonstrates the representative SVM in the category of traditional machine learning models, showing the prediction performance obtained with the NB, KNN, and SVM classifiers for predicting the stock prices for the next day, week, and month for Apple, Toyota, Chevorn, and the Bank of America, based on TF–IDF feature representation. Note that the SVM parameters are based on the default values of LinearSVC in the Scikit-Learn package.[10] As we can see, SVM performs better than the other two classifiers. Therefore, the SVM classifier will be used as the representative classifier for TF–IDF and word embedding feature representations in the subsequent experiments.

The CNN and LSTM models are implemented with Tensorflow,[11] using the Adam optimizer [47]. The learning rate and activation function are set to 0.0001 and Softmax, respectively. For CNN, four different kernel sizes, i.e. 3, 4, 5, and 7, are initially compared; the initial result showing the best one to be 7 is consistent with Kim's findings [48]. Next, the best kernel size is used with different hidden sizes including 16, 32, 64, and 128 for model training [24,29]. The activation function is based on rectified linear unit (ReLU). Moreover, the pooling layer is based on the max pooling operation where the pool size is set to 2, and the fully connected layer uses the Softmax activation function. For LSTM, 64, 128, and 256 hidden sizes are compared in order to find which the best parameter is.

5 https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

6 https://code.google.com/archive/p/word2vec/.

7 https://allennlp.org/elmo.

8 https://github.com/google-research/bert.

9 The computing environment is based on Intel(R) Core(TM) i7-9700 CPU @ 3.00 GHz with NVIDIA GeForce RTX 2080 Ti GPU, 64.0 GB memory, and the Window 10 operation system.

10 https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.

11 https://www.tensorflow.org/.

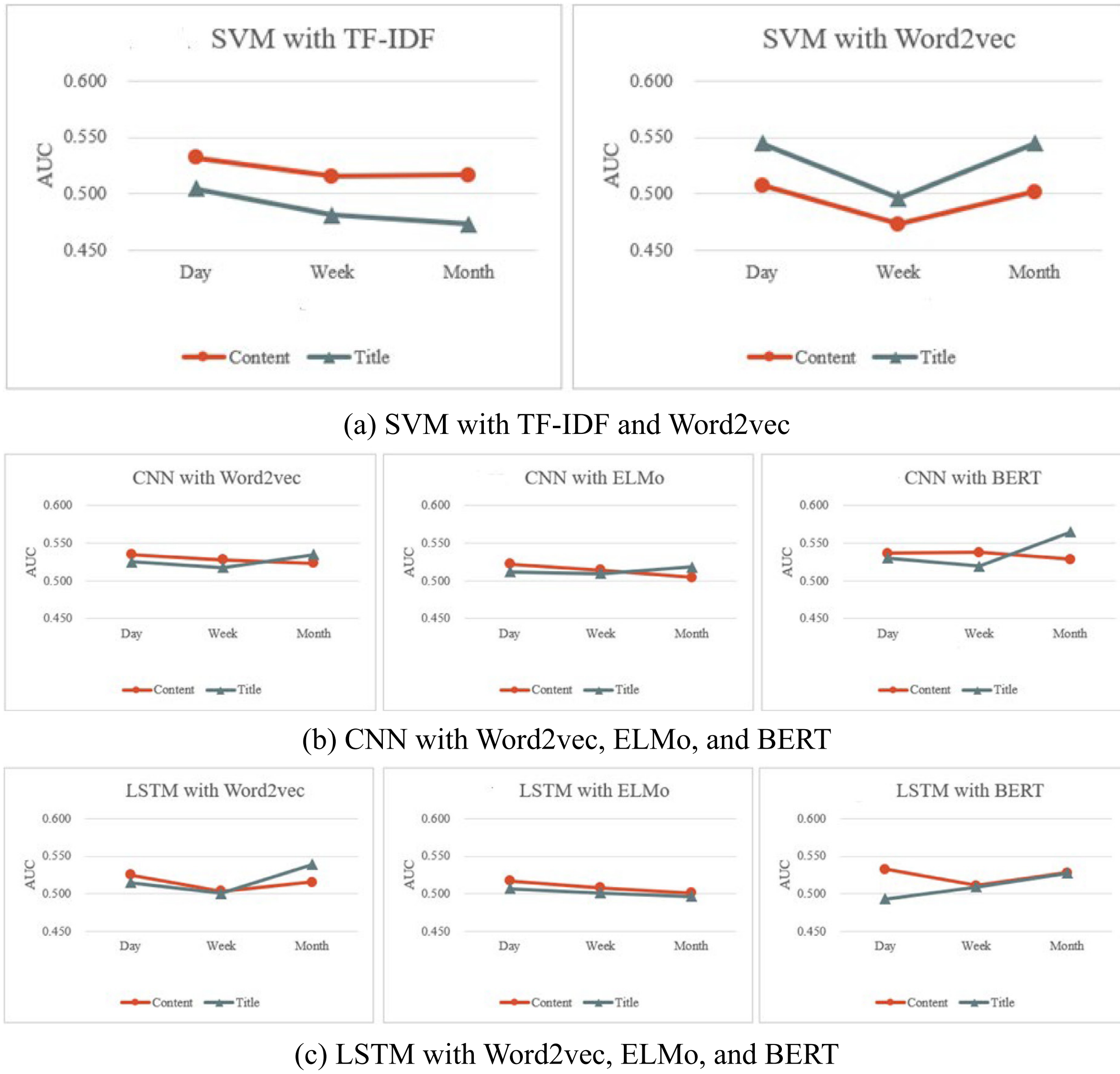


Fig. 3. Prediction performance of SVM based on news titles and contents for Apple.

4. Experimental results

4.1. Study I: The combinations of text feature representations and learning models

Figs. 5 to 8 show the AUC rates for eight different combinations of text feature representations and learning models, i.e. SVM+TF–IDF, SVM+Word2vec, CNN+Word2vec, CNN+ELMo, CNN+BERT, LSTM+Word2vec, LSTM+ELMo, and LSTM+BERT for Apple, Toyota, Chevron, and the Bank of America, respectively.

For SVM, TF–IDF is a better feature representation method than Word2vec. On the other hand, for the deep learning techniques, no matter which text feature representation method is used, CNN usually performs better than LSTM. In particular, CNN+Word2vec performs the best for the prediction of stock prices for different terms for Toyota and the Bank of America, whereas CNN+BERT performs the best for Apple and Chevron in most cases.

Moreover, the top two deep learning technique combinations, i.e. CNN+Word2vec and CNN+BERT, perform significantly better than the top traditional machine learning technique combination, i.e. SVM+TF–IDF ($p < 0.05$).[12] However, there is no significant difference in the level of performance between CNN+Word2vec and CNN+BERT. Therefore, these two combinations can be regarded as the representative baselines for text mining based stock prediction.

Table 3
Best combinations of text feature representation methods with learning models.

	Day	Week	Month
Apple	CNN+BERT	CNN+BERT	CNN+BERT
Toyota	CNN+Word2vec	CNN+Word2vec	CNN+Word2vec
Chevron	CNN+BERT	LSTM+ELMo	CNN+BERT
Bank of America	CNN+Word2vec	CNN+Word2vec	CNN+Word2vec

Table 3 lists the best combinations for predicting the stock prices for the next day, week, and month. In most cases, CNN+Word2vec and CNN+BERT are the better choices for text mining based stock prediction.

12 The statistical analysis is based on Wilcoxon signed ranks test [49].

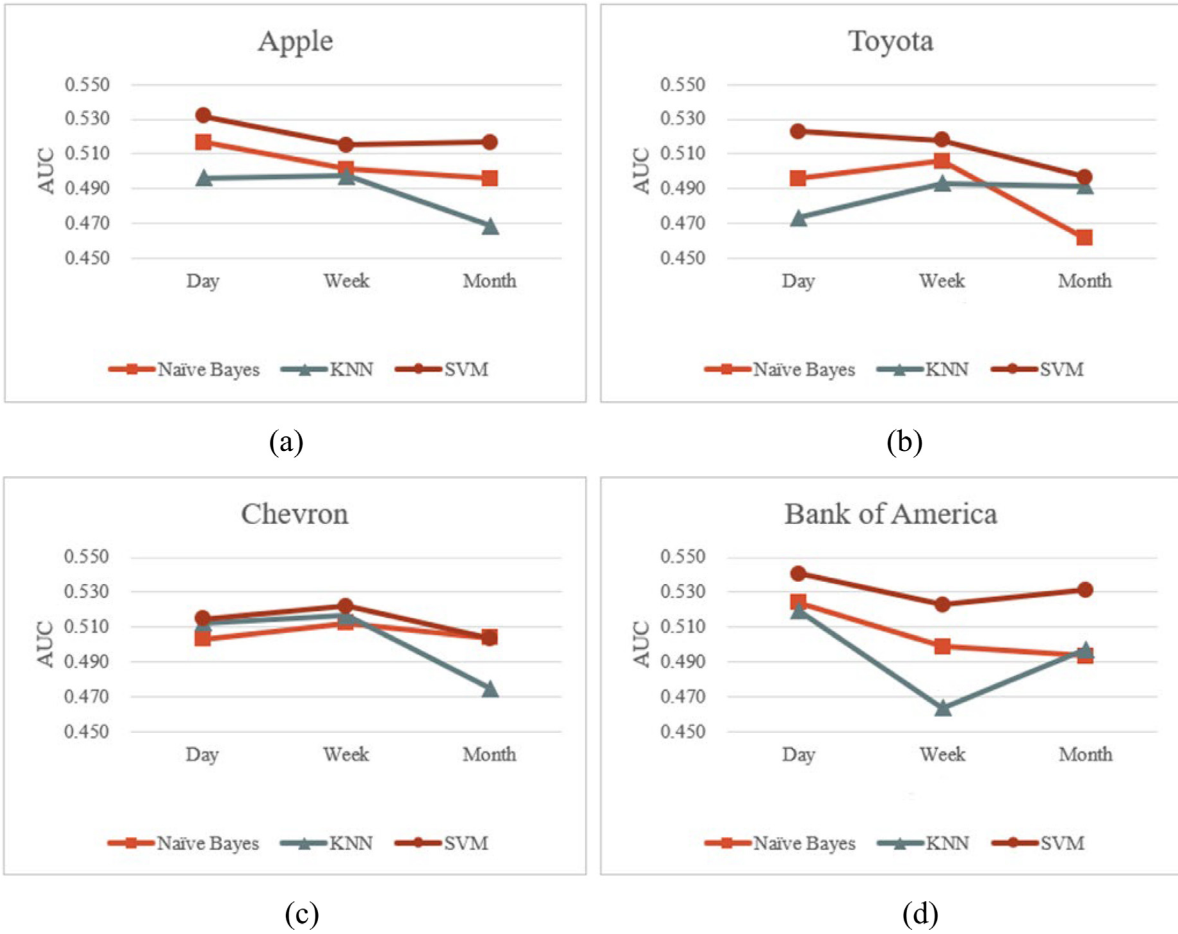


Fig. 4. Prediction performance of NB, LR, and SVM.

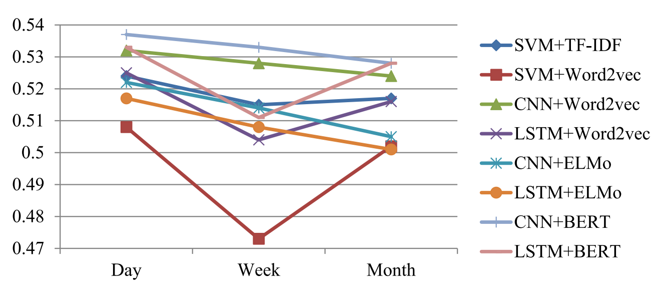


Fig. 5. AUC rates obtained with different combinations for Apple.

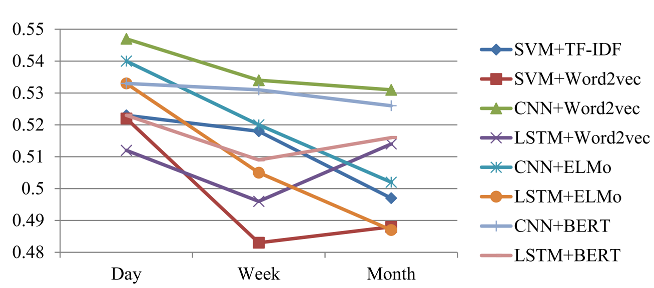


Fig. 6. AUC rates obtained with different combinations for Toyota.

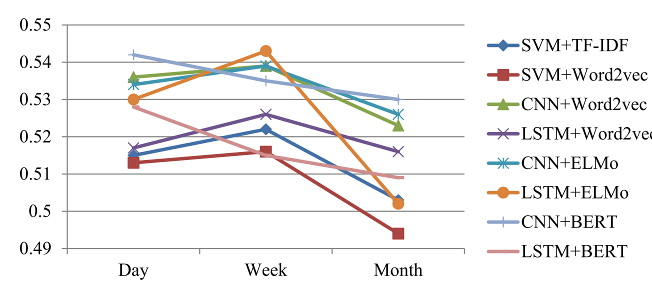


Fig. 7. AUC rates obtained with different combinations for Chevron.

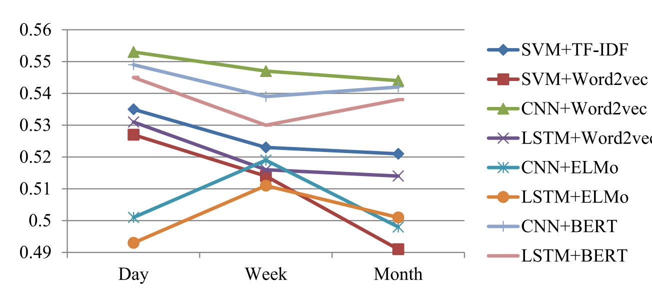


Fig. 8. AUC rates obtained different combinations for the Bank of America.

Table 4
Dataset information collected from CNBC and The Motley Fool.

	# of news articles	# of unique terms	Average length
CNBC			
Apple	14,499	64,763	556
Bank of America	4067	35,486	605
The Motley Fool			
Apple	8632	38,959	593
Bank of America	2882	24,163	571

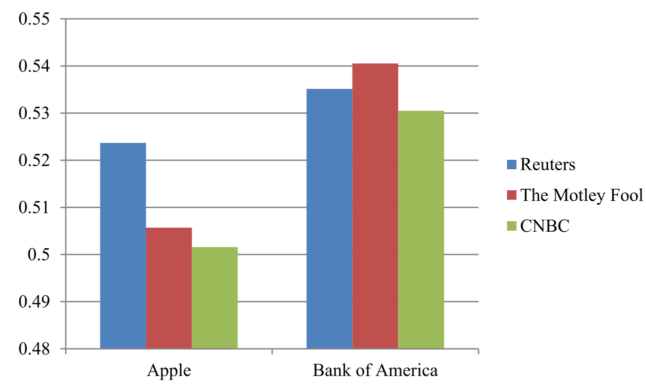


Fig. 9. The AUC rates obtained with SVM+TF–IDF.

4.2. Study II: The effect of different news platforms on prediction performance

In this study, in addition to Reuters, two other free public financial news websites are considered for performance comparison, namely, CNBC and The Motley Fool. The same as for Reuters, published news was crawled for the time between January 1, 2012 and December 31, 2019. Particularly, only news articles about Apple and the Bank of America were considered. This is because they present two different types of companies for the variation level of stock prices. That is, the stock prices of Apple widely varied between about 25 and 140, whereas the Bank of America only variated between about 10 and 35 within the five years. Therefore, Apple and the Bank of America are used to represent the four companies. Table 4 lists related dataset information for both CNBC and The Motley Fool.

Based on the results of study one, CNN+Word2vec and SVM+TF–IDF were chosen to represent the deep learning and traditional machine learning techniques, respectively, for performance comparisons. In addition, their next day stock prediction performance was also compared since their AUC rates were mostly higher than the predicted stock prices for the next week and month.

Figs. 9 and 10 show the AUC rates obtained with CNN+Word2vec and SVM+TF–IDF from different news platforms. As we can see, the techniques employed and the news platforms used can affect the prediction of stock prices for different companies. In particular, for SVM+TF–IDF, using the Reuters platform is the better choice for Apple, whereas The Motley Fool is a better choice for the Bank of America.

On the other hand, for CNN+Word2vec, using The Motley Fool platform can allow it to produce higher AUC rates than using the other news platforms for Apple, whereas Reuters is the better choice for the Bank of America.

Furthermore, we analyze the financial sentiments for news articles from different platforms based on the list of financial words created by Loughran and McDonald [50] and measure their

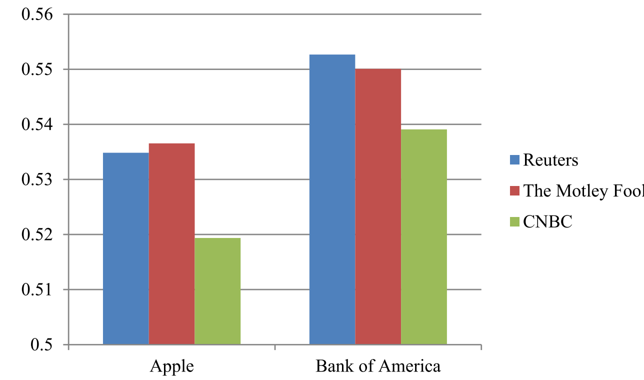


Fig. 10. The AUC rates obtained with CNN+Word2vec.

Table 5
Financial correlations for different news platforms (TF–IDF/Word2vec)..

	Reuters	The Motley Fool	CNBC
Apple	5.94%/2.87%	4.63%/2.58%	4.39%/2.58%
Bank of America	5.04%/3.17%	6.33%/3.51%	5.12%/3.09%

financial correlations [51] by

$$News\,Sentiment_i = \frac{\sum_i P + \sum_i N}{\sum_i T} \tag{3}$$

$$Financial\,Correlation = \frac{\sum News\,Sentiment}{\sum T\ News} \tag{4}$$

where $\sum_i P$ and $\sum_i N$ indicate the number of positive and negative sentiment words; and $\sum_i T$ is the total number of words in the ith news article; and $\sum T\ News$ indicates the total number of news articles on a specific platform.

Table 5 shows the financial correlations of different news platforms as determined by TF–IDF and Word2vet. As we can see, the Reuters platform provides the highest financial correlation for Apple, whereas The Motley Fool does so for the Bank of America.

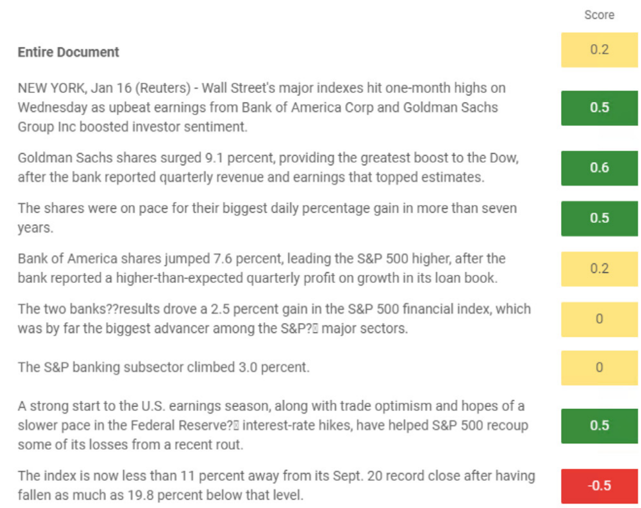
Fig. 11 shows an example of three news articles about the Bank of America, which were published on January 16, 2019 on the three platforms and their sentiment scores calculated by the Google Natural Language API are presented. Note that the sentiment score is between −1 (negative sentiment) and 1.0 (positive sentiment).

Wall Street's major indexes hit one-month highs on Wednesday as upbeat earnings from Bank of America Corp and Goldman Sachs Group Inc boosted investor sentiment. Goldman Sachs shares surged 9.1 percent, providing the greatest boost to the Dow, after the bank reported quarterly revenue and earnings that topped estimates. The shares were on pace for their biggest daily percentage gain in more than seven years. Bank of America shares jumped 7.6 percent, leading the S&P 500 higher, after the bank reported a higher-than-expected quarterly profit on growth in its loan book. The two bank's results drove a 2.5 percent gain in the S&P 500 financial index, which was by far the biggest advancer among the S&P's major sectors. The S&P banking subsector climbed 3.0 percent. A strong start to the U.S. earnings season, along with trade optimism and hopes of a slower pace in the Federal Reserve's interest-rate hikes, have helped S&P 500 recoup some of its losses from a recent rout. The index is now less than 11 percent away from its Sept. 20 record close after having fallen as much as 19.8 percent below that level.

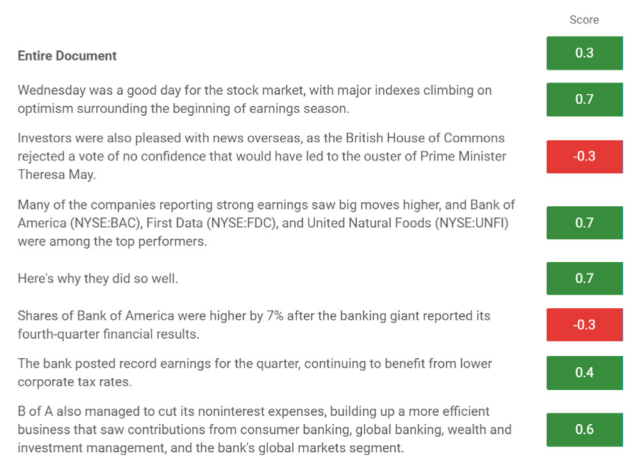
Wednesday was a good day for the stock market, with major indexes climbing on optimism surrounding the beginning of earnings season. Investors were also pleased with news overseas, as the British House of Commons rejected a vote of no confidence that would have led to the ouster of Prime Minister Theresa May. Many of the companies reporting strong earnings saw big moves higher, and Bank of America (NYSE:BAC), First Data (NYSE:FDC), and United Natural Foods (NYSE:UNFI) were among the top performers. Here's why they did so well. Shares of Bank of America were higher by 7% after the banking giant reported its fourth-quarter financial results. The bank posted record earnings for the quarter, continuing to benefit from lower corporate tax rates. B of A also managed to cut its noninterest expenses, building up a more efficient business that saw contributions from consumer banking, global banking, wealth and investment management, and the bank's global markets segment.

Bank of America posted the strongest earnings numbers of the big banks so far, CNBC's Jim Cramer contended Wednesday, while comparing BofA to e-commerce giant Amazon. "Bank of America is a growth stock after this", Cramer said on "Squawk Box" after the bank reported better-than-expected profit and revenue for the fourth quarter. "You know what they are? They're 'Amazon' but they make money", argued Cramer, apparently referring to how Amazon often plows money back into its businesses which can hurt near-term profitability. Bank of America shares jumped about 5 percent early Wednesday on quarterly results driven by a strong performance from its consumer-banking business and lower taxes. The BofA report followed J.P. Morgan's weaker-than-expected earnings Tuesday.

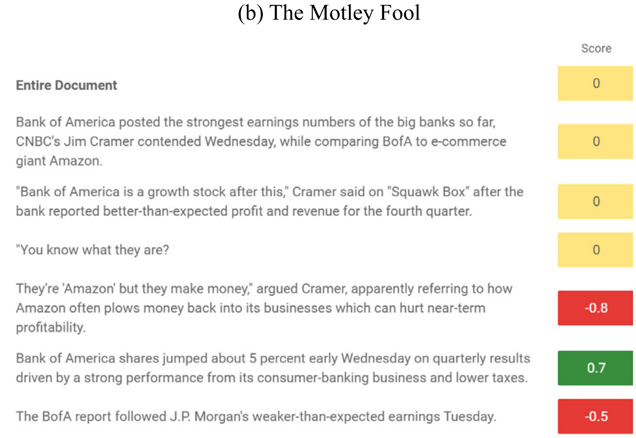
In addition, the news articles about the Bank of America collected from The Motley Fool contain more financial words than those collected from Reuters or CNBC, as shown in Table 4. Furthermore, there are more high-level positive and negative sentiment scores in the news articles from The Motley Fool. This means that related financial words as well as the sentiment scores residing within the news articles do, to some extent, reflect the future rise and fall of stock prices. Moreover, these results indicate that the same prediction models perform differently for financial news articles collected from different platforms.



(a) Reuters



(b) The Motley Fool



(c) CNBC

Fig. 11. Sentiment scores for the news articles from Reuters, The Motley Fool, and CNBC.

5. Conclusion

In this paper, combinations of different text feature representation methods with different learning models are compared in order to find the best combination for text mining based stock prediction. In particular, two context-free and two contextualized text feature representation methods, i.e. TF–IDF, Word2vec, ELMo, and BERT, are used to produce the input for traditional machine learning and deep learning models, i.e. SVM, CNN, and LSTM, which results in eight different combination approaches.

The experimental results show that the best combinations are CNN+Word2vec and CNN+BERT for the next day, week, and month stock predictions, significantly outperforming the traditional SVM machine learning approach. Although there is no significant difference in the prediction performance of CNN+Word2vec and CNN+BERT, the computational complexity of BERT feature extraction is much higher than that of Word2vec.

Moreover, the effect of using three different news platforms on prediction performance is also examined. The results show that the performance of the prediction models can be affected by the news platform selected. It turns out the number of financial words and the related positive and negative sentiment scores conveyed in the news articles are important factors for constructing effective stock prediction models.

There are several issues which could be considered in the future. First, before text feature extraction, some finance based sentiment word nets can be developed in order to filter out unrepresentative words from the article. Differences in the discriminative power of using the original news articles and preprocessed articles can be investigated. Second, since only four companies with different industries were chosen as the first attempt to investigate the three factors affecting text mining based stock prediction, it would be useful to examine the listed companies of different industries in specific markets. In other words, the best combination of techniques may not be the same for different industries and/or markets. Third, besides the text feature representation methods, image based feature representations, such as candlestick charts, can be used as the input for deep learning models. Therefore, text and image mining based approaches can be compared for short-, medium-, and long-term stock prediction, for different industries. Fourth, it would be interesting to conduct some human judgment for the perception of the news articles in relation to the companies. Fifth, there are several factors that can affect or improve the model performance. For example, the data with a certain range of 'unchanged' stock prices, such as rise and fall within 1.5%, can be removed. These data usually provide less discriminative power for the prediction models to distinguish between the rise and fall classes. Moreover, the collected dataset for each company is usually class imbalanced, where the rise and fall classes contain different numbers of data samples. Performing data re-sampling to balance the dataset during the model training stage and/or employing the cost sensitivity mechanism to avoid the model to produce bias results for the majority and minority classes are the common solutions to improve the final classification performance. Last but not least, since using different news platforms and news contents for the same companies can affect the performances of prediction models, it is worth investigating the ways of finding out the optimal news platforms and news contents to construct more effective prediction models.

CRediT authorship contribution statement

Wei-Chao Lin: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Chih-Fong Tsai:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Hsuan Chen:** Software, Validation, Resources, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This study was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 109-2410-H-182-012, and in part by Chang Gung Memorial Hospital at Linkou, under Grant BMRPH13 and CMRPG3J0732.

References

[1] O. Bustos, A. Pomares-Quimbaya, Stock market movement forecast: a systematic review, Expert Syst. Appl. 156 (2020) 113464.
[2] D.P. Gandhmal, K. Kumar, Systematic analysis and review of stock market prediction techniques, Comp. Sci. Rev. 34 (2019) 100190.
[3] D. Shah, H. Isah, F. Zulkernine, Stock market analysis: a review and taxonomy of prediction techniques, Int. J. Financ. Stud. 7 (2) (2019) 26.
[4] I. Nti, A.F. Adekoya, B.A. Weyori, A systematic review of fundamental and technical analysis of stock market predictions, Artif. Intell. Rev. 53 (2020) 3007–3057.
[5] B.S. Kumar, V. Ravi, A survey of the applications of text mining in financial domain, Knowl.-Based Syst. 114 (2016) 128–147.
[6] F.Z. Xing, E. Cambria, R.E. Welsch, Natural language based financial forecasting: a survey, Artif. Intell. Rev. 50 (2018) 49–73.
[7] S.-J. Guo, F.-C. Hsu, C.-C. Huang, Deep candlestick predictor: a framework toward forecasting the price movement from candlestick charts, in: International Symposium on Parallel Architectures, Algorithms and Programming, 1998, pp. 219–226.
[8] J. Lee, R. Kim, Y. Koh, J. Kang, Global stock market prediction based on stock chart images using deep q-network, IEEE Access 7 (2019) 167260–167277.
[9] C.-F. Tsai, Z.-Y. Quan, Stock prediction by searching for similarities in candlestick charts, ACM Trans. Manage. Inf. Syst. 5 (2) (2014) a9.
[10] H.C. Wu, R.W.P. Luk, K.F. Wong, K.L. Kwok, Interpreting TF-IDF term weights as making relevance decisions, ACM Trans. Inf. Syst. 26 (3) (2008) a13.
[11] Y. Zhang, J. Rong, Z.-H. Zhou, Understanding bag-of-words model: a statistical framework, Int. J. Mach. Learn. Cybern. 1 (2010) 43–52.
[12] Xiao Ding, Yue Zhang, Ting Liu, Junwen Duan, Deep learning for event-driven stock prediction, in: International Joint Conference on Artificial Intelligence, 2015, pp. 2327–2333.
[13] L. dos Santos Pinheiro, M. Dras, Stock market prediction with deep learning: a character-based neural language model for event-based trading, in: Proceedings of Australasian Language Technology Association Workshop, 2017, pp. 6–15.
[14] E.J. Fortuny, T. De Smedt, D. Martens, W. Daelemans, Evaluating and understanding text-based stock price prediction models, Inf. Process. Manage. 50 (2) (2014) 426–441.
[15] J. Guo, B. Tuckfield, News-based machine learning and deep learning methods for stock prediction, J. Phys. Conf. Ser. 1642 (2020) 012014.
[16] X. Liu, H. Huang, Y. Zhang, C. Yuan, News-driven stock prediction with attention-based noisy recurrent state transition, 2020, arXiv:2004.01878 [cs.CL].
[17] W. Long, L. Song, Y. Tian, A new graphic kernel method of stock price trend prediction based on financial news semantic and structural similarity, Expert Syst. Appl. 118 (2019) 411–424.
[18] K.H. Nam, N.Y. Seong, Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market, Decis. Support Syst. 117 (2019) 100–112.
[19] A.K. Nassirtoussi, S. Aghabozorgi, T.Y. Wah, D.C.L. Ngo, Text mining of news-headlines for FOREX market prediction: a multi-layer dimension reduction algorithm with semantics and sentiment, Expert Syst. Appl. 42 (1) (2015) 306–324.
[20] N. Seong, K. Nam, Predicting stock movements based on financial news with segmentation, Expert Syst. Appl. 164 (2021) 113988.
[21] Z. Hu, W. Liu, J. Bian, X. Liu, T.-Y. Liu, Listening to chaotic whispers: a deep learning framework for news-oriented stock trend prediction, in: ACM International Conference on Web Search and Data Mining, 2018, pp. 261–269.
[22] H.D. Huynh, L.M. Dang, D. Duong, A new model for stock price movements prediction using deep neural network, in: International Symposium on Information and Communication Technology, 2017, pp. 57–62.
[23] Z.H. Kilimci, R. Duvar, An efficient word embedding and deep learning based model to forecast the direction of stock exchange market using twitter and financial news sites: a case of Istanbul stock exchange (BIST 100), IEEE Access 8 (2020) 188186–188198.
[24] Y. Ma, L. Zong, P. Wang, A novel distributed representation of news (drnews) for stock market predictions, 2020, arXiv:2005.11706 [cs.CL].
[25] P. Oncharoen, P. Vateekul, Deep learning for stock market prediction using event embedding and technical indicators, in: International Conference on Advanced Informatics: Concept Theory and Applications, 2018, pp. 19–24.
[26] M. Sardelich, S. Manandhar, Multimodal deep learning for short-term stock volatility prediction, 2018, arXiv:1812.10479 [q-fin.ST].
[27] L. Shi, Z. Teng, L. Wang, Y. Zhang, A. Binder, DeepClue: visual interpretation of text-based deep stock prediction, IEEE Trans. Knowl. Data Eng. 31 (6) (2018) 1094–1108.
[28] W. Souma, I. Vodenska, H. Aoyama, Enhanced news sentiment analysis using deep learning, J. Comput. Soc. Sci. 2 (2019) 33–46.
[29] M.R. Vargas, B.S.L.P. de Lima, A.G. Evsukoff, Deep learning for stock market prediction from financial news articles, in: IEEE International Conference on Computational Intelligence and Virtual Environment for Measurement Systems and Applications, 2017, pp. 1–6.
[30] M.T. Pilehvar, J. Camacho-Collados, Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning, Morgan & Claypool, 2020.
[31] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1986.
[32] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: International Conference on Learning Representations, 2013, pp. 1–12.
[34] J. Pennington, R. Socher, C. Manning, GloVe: global vectors for word representation, in: International Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[35] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text classification algorithms: a survey, Information 10 (4) (2019) 150.
[36] O. Melamud, J. Goldberger, I. Dagan, Context2vec: learning generic context embedding with bidirectional LSTM, in: International Conference on Computational Natural Language Learning, 2016, pp. 51–61.

[37] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: International Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237.
[38] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BETT: pre-training of deep bidirectional transformers for language understanding, in: International Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.
[39] V.N. Vapnik, Statistical Learning Theory, first ed., Wiley-Interscience, 1998.
[40] L. Deng, Y. Liu, Deep Learning in Natural Language Processing, Springer, 2018.
[41] S. Raaijmakers, Deep Learning for Natural Language Processing, Manning Publications, 2020.
[42] D. Mandic, J. Chambers, Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, Wiley, 2001.
[43] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[44] D.L. Minh, A. Sadeghi-Niaraki, H.D. Huy, K. Min, H. Moon, Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network, IEEE Access 6 (2018) 55392–55404.
[45] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern. C 42 (4) (2012) 463–484.
[46] S.A. Farimani, M.V. Jahan, A.M. Fard, R.K. Tabbakh, Investigating the informativeness of technical indicators and news sentiment in financial market price prediction, Knowl.-Based Syst. 246 (2022) 108742.
[47] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2015, pp. 1–15.
[48] Y. Kim, Convolutional neural networks for sentence classification, in: International Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.
[49] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[50] T. Loughran, B. McDonald, When is a liability not a liability? Textual analysis, dictionaries, and 10-ks, J. Finance 66 (1) (2011) 35–65.
[51] P.K. Narayan, D. Bannigidadmath, Does financial news predict stock returns? New evidence from Islamic and non-Islamic stocks, Pac.-Basin Finance J. 42 (2017) 24–45.