

Predicting stock market using natural language processing

Natural
language
processing

Karlo Puh and Marina Bagić Babac

*Faculty of Electrical Engineering and Computing, University of Zagreb,
Zagreb, Croatia*

41

Abstract

Purpose – Predicting the stock market's prices has always been an interesting topic since its closely related to making money. Recently, the advances in natural language processing (NLP) have opened new perspectives for solving this task. The purpose of this paper is to show a state-of-the-art natural language approach to using language in predicting the stock market.

Design/methodology/approach – In this paper, the conventional statistical models for time-series prediction are implemented as a benchmark. Then, for methodological comparison, various state-of-the-art natural language models ranging from the baseline convolutional and recurrent neural network models to the most advanced transformer-based models are developed, implemented and tested.

Findings – Experimental results show that there is a correlation between the textual information in the news headlines and stock price prediction. The model based on the GRU (gated recurrent unit) cell with one linear layer, which takes pairs of the historical prices and the sentiment score calculated using transformer-based models, achieved the best result.

Originality/value – This study provides an insight into how to use NLP to improve stock price prediction and shows that there is a correlation between news headlines and stock price prediction.

Keywords Stock market prediction, Machine learning, Time-series analysis, Recurrent neural network, Natural language processing, BERT, GRU

Paper type Research paper

Received 2 August 2022
Revised 21 November 2022
28 January 2023
11 March 2023
Accepted 13 March 2023

1. Introduction

Predicting stock market prices has always been an interesting topic since it is closely related to making money. It gained some additional popularity in recent years due to the significant inflation rate which forced people to invest their money rather than save it. Predicting stock prices is not an easy task because of their volatile nature and a lot of different factors affecting their price. The most common way used to predict stock price movement is technical analysis, a method that uses historical market data to predict future prices. However, it turns out that technical analysis does not give very satisfying results, mostly due to a lack of additional information. Out of all the possible factors affecting the prices, it all comes down to the investors and their willingness to invest money. To extract the emotion of the investors, sentiment analysis is used. Existing studies have shown that there is a correlation between financial news headlines and stock market price movement. In the recent past, it is easily found a few examples of news headlines affecting the stock market and even cryptocurrency market prices.

In this paper, natural language processing (NLP) is used to explore possibilities to advance the traditional approaches to stock price prediction. NLP is a component of artificial intelligence that in general aims at understanding human (natural) language as it is spoken and written (Jurafsky and Martin, 2000). Thus, the goal of this research is to go beyond the numerical data of stock prices and use textual data as an additional resource of information about the stock market in making predictions. Moreover, various state-of-the-art NLP models ranging from baseline



© Karlo Puh and Marina Bagić Babac. Published by Emerald Publishing Limited. This article is published under the Creative Commons Attribution (CC BY 4.0) license. Anyone may reproduce, distribute, translate and create derivative works of this article (for both commercial and non-commercial purposes), subject to full attribution to the original publication and authors. The full terms of this license may be seen at <http://creativecommons.org/licenses/by/4.0/legalcode>

American Journal of Business
Vol. 38 No. 2, 2023
pp. 41-61
Emerald Publishing Limited
1935-5181
DOI 10.1108/AJB-08-2022-0124

models based on convolutional and recurrent neural networks to the most recent *Bidirectional Encoder Representations from Transformers* (BERT)-based models are designed, implemented and tested. Here, BERT is a transformer-based machine learning technique for NLP. Nevertheless, conventional statistical models for technical analysis are implemented as a benchmark. The dataset used for this paper contains the Dow Jones Industrial Average (DJIA) prices and Wall Street Journal news headlines in the period from January 2008 to December 2020.

The recent papers on stock price prediction differ from each other in terms of the used computational models (Ji *et al.*, 2021), the selected stock prices dataset (Kumari *et al.*, 2021), as well as the textual dataset ranging from financial online news to social media comments (Kameshwari *et al.*, 2021). Therefore, the optimal predictive model for specific research setting highly depends on specific objectives and available resources (Shilpaand and Shambhavi, 2021), leaving the door open for the enhancement of existing models. In addition, datasets differ according to the respective timelines and countries (Sidogi *et al.*, 2021), which are both highly important factors that influence a particular setting and initial conditions of the research, thus leaving room for studying from diverse data.

Our contribution lies in a systematic approach to exploring a variety of different neural network models and various parameters of these models in order to achieve the best performance, thus providing a theoretical framework to be used when choosing the appropriate model architecture depending on the particular purpose that goes beyond stock price prediction. Our approach is unique due to the chosen combination of the stock price dataset with the Wall Street Journal news and the proposed GRU-based (*gated recurrent unit*) neural network models, an improved variant of recurrent neural network with a gating mechanism. Moreover, only a few studies make use of the BERT-based architecture in a similar setting (Cheng and Chen, 2021), which is the state-of-the-art model in NLP, thus our study contributes to filling this gap by providing a comprehensive approach to extracting specific information from textual data and using it for financial predictions.

Our results achieved by different implemented neural network models indicate that using the information extracted from the news headlines alongside historical prices improves stock price predictions. The model that achieved the best result uses a fine-tuned version of BERT called FinBERT (Araci, 2019), a pre-trained NLP model to analyze the sentiment of the financial text, to extract sentiment from the news headlines and feed that information into the GRU cell alongside the historical prices.

2. Review of literature

In recent years, the interest in predicting stock market prices rose so has the number of published papers on that subject (Fazlija and Harder, 2022). One stream of research is based on traditional time series methodologies. Idrees *et al.* (2019) experimented with an efficient autoregressive integrated moving average (ARIMA) model to predict Indian stock market volatility. After comparing their results with the actual time series, they got a deviation of 5% error on average. In their paper, Wadi *et al.* (2018) use the ARIMA model to predict prices with data collected from Amman Stock Exchange (ASE) from January 2010 to January 2018. Their results have shown that the ARIMA model gives satisfying results for short-term prediction. To be specific, their best model, ARIMA (2,1,1) resulted in a root mean square error (RMSE) of 4.00. The only significant downside of their model is poor performance on long-term predictions.

Another stream of research uses evolving machine and deep learning models and techniques that perform well on time series tasks, such as convolutional models and recurrent neural networks. Zulqarnain *et al.* (2020) proposed a combined architecture that takes advantage of both convolutional and recurrent neural networks to predict trading signals. Their model is based on convolutional neural network (CNN) which processes signals and feeds them into GRU to capture long-term dependencies. GRU is used since it resolves the vanishing gradient problems efficiently, which is a problem for most recurrent neural

networks. They evaluated their model on three datasets for stock indexes of the Deutscher Aktienindex (DAX), the Hang Seng Index (HSI) and the S&P 500 Index in the period 2008 to 2016. As result, they achieved an accuracy of 56.2% on HSI, 56.1% on DAX and 56.3% on S&P 500 dataset. [Yadav et al. \(2020\)](#) used various configurations of long short-term memory (LSTM) hyperparameters to predict Indian stock market prices.

In order to predict stock market price movement more accurately, authors have recently started to use NLP to add some extra information or incorporate prevailing sentiments and expectations from textual data. [Mehtab et al. \(2019\)](#) compared several approaches to predict the NIFTY 50 index values of the National Stock Exchange of India in the period 2015–2017. They built several models based on machine learning but also deep learning-based LSTM models. Finally, they augmented the LSTM model with sentiment analysis on Twitter data. Specifically, they predicted stock price movement using the previous week's closing prices and Twitter sentiment. The mentioned model achieved the best results among all models in its ability to forecast the NIFTY 50 movement. In addition, [Wang and Wang \(2016\)](#) used data from Sina Weibo, China's largest and most widely used social media site and the SVM algorithm for stock price prediction and concluded that sentiment from social media contributed to improving prediction results. Likewise, [Kameshwari et al. \(2021\)](#) used sentiment analysis of news headlines from Reddit in addition to the DJIA prices to forecast the stock market movement using various machine learning algorithms. The best accuracy was achieved with a multi-layer perceptron, which is a simple neural network. Furthermore, [Ji et al. \(2021\)](#) used investors' comments and companies' news of the top 15 listed medical companies from the "Oriental Fortune website" to build long text feature vectors and then reduce the dimensions of the text feature vectors by stacked auto-encoder to balance the dimensions between text feature variables and stock financial index variables in predicting the stock price of the company "Meimian Health". They used a LSTM model for prediction, which is a variant of a recurrent neural network. In addition, [Mohan et al. \(2019\)](#) experimented with several different approaches using time series models, neural networks and several combinations of neural networks with financial news articles to predict S&P index prices. Their results suggest that there is a strong correlation between news articles and stock market prices.

Recently, [Sonkiya et al. \(2021\)](#) proposed a state-of-the-art method for stock market price prediction. In this paper, the authors use a version of the Googles BERT model pre-trained on financial corpus called fin-BERT to extract sentiment value from the news. Afterward, they use that sentiment value alongside technical indicators such as moving averages, Bollinger bands, RSI, etc. as input to generative adversarial network (GAN) which then predicts stock price. Experimental results have shown that the proposed GAN model achieves better results in comparison to traditional time series methods like LSTM, GRU or ARIMA. Furthermore, [Cheng and Chen \(2021\)](#) used a BiLSTM sentiment analysis model and FinBERT as a feature extractor to obtain the context information of the financial commentary dataset and combine BiLSTM along with multiple attention mechanisms to extract the sentiment of financial comments, and the results showed improved accuracy over the pure stock price dataset.

From this review of recent papers, it can be concluded that the contributions are highly dependent on the chosen dataset and its timeline, country of interest and many more different factors, which makes it difficult to make straightforward comparisons and conclusions. Our paper contributes to the specification, implementation and testing of several different neural network architectures and utilizes the Wall Street Journal news to investigate if and to what extent sentiment calculated from news contributes to the improvement of these models.

3. Theoretical framework

3.1 Autoregressive integrated moving average

ARIMA is a statistical model which uses time-series data for predicting future trends or better overall understanding of past data. The model's goal is to predict future moves by examining

the differences between values in the series instead of through actual values. For a better understanding of the ARIMA model, each of its components is described separately (Kotu and Deshpande, 2019):

- (1) AR (*Auto Regression*) - when a statistical model uses past data to predict future values, it is called autoregressive. Furthermore, autoregressive models assume that the future will resemble the past.
- (2) I (*Integrated*) indicates that data values have been replaced with the differenced values of d -order to obtain stationary data.
- (3) MA (*Moving Average*) means that the regression error is a linear combination of past errors.

The parameters are p , the number of lag observations in the model also known as the lag order; d , the number of times that the raw observations are differenced also known as the degree of differencing and q , the size of the moving average window also known as the order of the moving average. ARIMA model is expressed in the following equation (Kotu and Deshpande, 2019):

$$y_t = I + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \dots + \alpha_p y_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (1)$$

It can be seen in Eq. (1) that for the autoregressive part the predictors are lagged p data points and that they are lagged q errors for the moving average part. The final prediction is the differenced y_t in the d^{th} order. The described model is called the ARIMA (p, d, q) model. In this model, the data are differenced in order to make it stationary. A model that shows stationarity is one that shows there is constancy to the data over time. Most economic and market data show trends, so the purpose of differencing is to remove any trends or seasonal structures. Seasonality, or when data show regular and predictable patterns that repeat, could negatively affect the model (Matei et al., 2017). If a trend appears and stationarity is not evident, many of the computations throughout the process cannot be made with great efficacy (Bifet and Gavaldà, 2007).

The hardest part when working with the ARIMA model is obviously choosing optimized p , d and q parameters. Since it is our choice to set those parameters, we can end up with many different performing models. Statistical software can help identify the appropriate number of lags or amount of differencing to be applied to the data and check for stationarity.

3.2 Deep learning

3.2.1 CNN – convolutional neural network. A CNN is a type of neural network that has proven to be successful in processing data with lattice topology (Goodfellow et al., 2016). Although the first thing that comes to our mind when talking about convolutional neural networks are image-related tasks, they achieve satisfying results in NLP tasks as well. The main building blocks of convolutional architecture are convolutional layers and pooling layers. The main role of convolution is to obtain the most important features from the input. Convolutional layers include many kernels with weights that are learned through the training process (Bifet and Gavaldà, 2007). Those kernels are designed to generate an output by looking at the word and its surroundings (in the case of 1D convolution, i.e. text as input). That way, since similar words have similar vector representations, convolution will produce a similar value. Furthermore, because of the way neurons are connected in convolutional layers, they have significantly fewer parameters than fully connected layers which means that there are fewer parameters to learn. That gives us more efficiency and a smaller possibility to overfit the model (Goodfellow et al., 2016).

The output of the convolutional layer is called a feature map which is the result of the element-wise multiplication of input data representation and kernel. Since the feature map

output records the precise position of features, any movement results in a different map. To overcome that limitation of the feature map output, we use pooling layers. They reduce the size of feature maps using specific functions. The most popular choices are average pooling and maximum pooling (Matei *et al.*, 2017).

The pooling process enables us one of the biggest advantages of convolutional neural networks called translation invariance. That basically means that when some pattern is learned, CNN can recognize it later at any other different position. That is very useful when working with images but also in NLP tasks when working with text because it summarizes the presence of important features in input text or image.

3.2.2 RNN – recurrent neural network. An RNN (*recurrent neural network*) is a modification of a typical artificial neural network specialized for working with sequential and time-series data. The idea behind RNN is to be able to process arbitrary length data while keeping track of its order. The advantage of recurrent neural networks is their ability to memorize prior inputs and use that information alongside current input to generate meaningful output (Sherstinsky, 2020).

There are several different RNN architecture configurations depending on our task. Some of the most popular are one-to-one, one-to-many, many-to-one and many-to-many (Goodfellow *et al.*, 2016).

Although recurrent neural networks perform well and have advantages like processing data of any length, sharing weights and memorizing information they also have some disadvantages. Slow computation time and difficulty when accessing old information are some of the problems but they are not the biggest ones. The situation, when gradient values come close to zero and prevent a model from learning, is called a vanishing gradient problem. Besides that, there is also an exploding gradient problem, when gradients that update weights grow exponentially (Roy *et al.*, 2022). In order to overcome mentioned problems, new model architectures were proposed. LSTM and GRU are recurrent neural networks with some advantages over classical RNNs.

3.2.3 LSTM – long short-term memory. LSTM is a variation of a recurrent neural network that can handle long-term dependencies and also resolve vanishing gradient problems (Hochreiter and Schmidhuber, 1997). The reason why LSTMs work so well is their ability to add or remove information to the cell state. Structures called gates enable that kind of behavior. Gates are different neural networks that consist of a sigmoid layer and a pointwise multiplication operation. The core idea behind that is to forget or update data because the sigmoid layer squishes values between 0 and 1.

That way the network can learn which data is relevant or irrelevant and decide to keep or forget it. The first gate is called the forget gate and they decide which information to keep or discard. That step is demonstrated in Eq. 2, where h_{t-1} and x_t are inputs of LSTM, W_f is the weight, and b_f is the bias (Sherstinsky, 2020).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

Next, we want to update the cell state. The second gate, called the input gate, also using a sigmoid layer decides which values to update. Afterward, we combine the result of the input gate with the tanh layer to create the update on the cell state (Hochreiter and Schmidhuber, 1997).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

Specifically, to update the cell state, we multiply the old cell state by the forget gate, then add it with the input gate multiplied with \tilde{C}_t . The described process is shown in Eq. (5). As Eq. (6) shows, we first pass the current and the previous hidden state through the sigmoid. As a result of everything mentioned, we get the new hidden state shown in Eq. (7). In the end, the new hidden state and the cell state are carried over to the next cell (Hochreiter and Schmidhuber, 1997).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

Described LSTM model achieves much better results than traditional RNN but there is still a place for an upgrade. We have seen that LSTM uses information from the past, meaning that the current state depends on the information before that moment. In order to have more contextual information in every moment, i.e. increase the amount of network information, we use bidirectional LSTM. Bidirectional LSTM consists of two LSTMs, each one of them going in a different direction. The first one goes forward (from the past to the future) and the second one goes backward (from the future to the past). That kind of architecture enables us to understand the context much better.

3.2.4 GRU – gated recurrent unit. The GRU has a similar architecture as LSTM but uses only two gates, an *update* gate and a *reset* gate. The *update* gate replaces the role of the input gate and forget gate from LSTM architecture and decides which information to pass along to the next state (Goodfellow et al., 2016).

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{8}$$

The *reset* gate then decides how much of the past information to discard, i.e. forget.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{9}$$

Afterward, we multiply the previous hidden state with the reset gate which decides how much of the past information is relevant. The result represents the cell’s memory.

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{10}$$

Finally, we calculate the current hidden state h_t , which is passed down the network.

$$h_t = (1 - z) * h_{t-1} + z_t * \tilde{h}_t \tag{11}$$

We can see that GRU has simpler architecture than LSTM and fewer parameters and operations which results in faster execution time. It is not straightforward to conclude which model is better because it depends on the data. Some experiments show that LSTM performs slightly better on a large dataset (Roy et al., 2022).

3.2.5 BERT – bidirectional encoder representations from transformers. BERT is a state-of-the-art language model for NLP tasks (Devlin et al., 2019) that is based on the original Transformer architecture (Vaswani et al., 2017). Transformer architecture was designed to resolve sequence-to-sequence tasks while successfully dealing with long-range dependencies. Its architecture consists of the encoder which reads input text and the decoder which generates the output sequence.

Unlike recurrent neural networks, the Transformer model is based on an attention mechanism that tries to understand relations between words. In other words, the attention mechanism decides which parts of the sequence are important. The attention mechanism is described by equation (12).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (12)$$

where Q represents a set of queries packed into a matrix, K and V are keys and values. In reality, they are all vector representations of words. We scale the product with the square root of the key vector dimension to prevent pushing the *softmax* function into the area where it has small gradient values and avoid exploding gradient problems. The Transformer model performs self-attention multiple times in parallel which is called *multi-head attention* (Vaswani *et al.*, 2017).

Now that we have had a glimpse into how the Transformer model works, we can dive into BERT. We have seen that the Transformer model consists of two parts, the encoder and the decoder. Since BERT is a language representation model, it does not need the decoder part of the Transformer but uses only the encoder. One of the reasons BERT works so well is the encoder's ability to read the entire input sequence at once rather than reading it from left to right. That way it can take into consideration all of the word surroundings and have a better insight into the context. The fact that the BERT model was trained on a large corpus of text which consists of the entire English Wikipedia (2,500M words) and BooksCorpus (800M words) is also one of the reasons why it achieves excellent results but more important is its training strategy. BERT uses two different training strategies using unlabeled data:

- (1) Masked LM (*masked language modeling*) - we randomly replace 15% of the words in the input sequence with a [MASK] token and make the model predict the real value based on the provided context (other words in the input sequence).
- (2) NSP (*next sentence prediction*) – we make pairs of sentences wherein in 50% of the cases the second sentence is a random sentence from the corpus and in the other 50%, the second sentence is the actual next sentence. The model then needs to guess whether the sentences are connected or not.

During the training process, both mentioned strategies are trained together, and the combined loss function is minimized. After the training process, BERT can be easily finetuned using labeled data for specific tasks (Marijić and Bagić Babac, 2023).

The same architecture is used for training and fine-tuning BERT. The only difference is the output layer which is configured for a specific task. Experimental results have shown that BERT achieves state-of-the-art results on the eleven most common NLP tasks (Devlin *et al.*, 2019).

4. Data description

Since the main goal of this paper is to perform stock market price prediction, first we need to define what prices are we predicting. DJIA is a weighted stock market index that tracks the 30 largest blue-chip stocks in the market. DJIA is a benchmark index in the US and for that reason, it has been chosen for prediction in this task. Historical prices of DJIA have been downloaded from Yahoo Finance [1] ranging from January 2008 to December 2020. Downloaded historical data contains more different prices for every day, but in this paper, we are considering only the closing price. Finally, we normalize mentioned prices using *MinMax* scaler in the range from -1 to 1 to reduce data complexity. Normalization is performed using the following formula:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (13)$$

After we have dealt with the numerical part of our dataset, i.e. prices, the following paragraphs explain textual dataset preparation steps.

4.1 Data scraping

In order to perform sentiment analysis, the news headlines from the Wall Street Journal were scraped on daily basis during the specified period. The scraping was performed by using the *BeautifulSoup*, which is a Python library designed for extracting data from XML and HTML files. The processing of web scraping begins with importing the *requests* library. Then, the URL of the webpage of interest to scrape must be specified (in our case, that is https://www.wsj.com/?mod=wsjheader_logo). The HTTP *request* is sent to the specified URL and the response from the server is saved as an object and then parsed and prepared for analysis.

Just like the prices dataset, our headlines dataset contains every day's top 20 news headlines from January 2008 till December 2020.

4.2 Text preprocessing

NLP is a branch of artificial intelligence dealing with the interaction between humans and computers using a natural language. The aim of NLP is to read, understand and decode human words in a valuable manner. Some of the common NLP tasks and techniques are tokenization, part-of-speech tagging, dependency parsing, constituency parsing, lemmatization, stemming, stopwords removal, word sense disambiguation, named entity recognition, etc. (Jurafsky and Martin, 2000)

Preprocessing is one of the most important steps when performing any NLP task. Text preprocessing basically means bringing the text into a clean form and making it ready to be fed into the model. When it comes to data preprocessing, there are many useful techniques. Specifically in this paper, tokenization is the first step in preprocessing. Tokenization means splitting a sentence into a list of words. After tokenization, removing stop words comes as the next step. Stop words are words that are commonly used in any language. If we take for example English, stop words are "is", "the", "and", "a", etc. Those words are considered unimportant in NLP so they are being removed (Kostelev and Bagić Babac, 2022). Next comes the process of transforming a word into its root or lemma called lemmatization. An example of that would be "swimming" to "swim", "was" to "be" and "mice" to "mouse". Considering that machines treat the lower and upper case differently, all the text, i.e. words will be lowered for better interpretation. Finally, all punctuation is removed. That part of preprocessing also helps to remove noise and get rid of useless data (Musso and Bagić Babac, 2022).

To perform some of the previously mentioned preprocessing tasks, *spaCy* [2] was used, which is an open-source library for advanced and multilingual NLP in Python. After loading data for the English language, *spaCy* enables us to perform tokenization, lemmatization and stopwords removal. Examples of using *spaCy* with its preprocessed output are shown in Table 1.

4.3 Word representation

Since computers do not understand words or their context, it is necessary to convert text into the appropriate, machine-interpretable form. Word embeddings are mathematical representations of words that give similar representation to words that have a similar meaning (Mikolov *et al.*, 2013). In other words, those representations model the semantic meaning of words. Specifically, those representations are vectors that are positioned in space in such a way that vectors closer to each other have more similar semantic meanings.

One of the word representations used in this research is called GloVe, which stands for *Global Vectors for Word Representation* (Pennington *et al.*, 2014). It gained popularity due to

its good performance and simplicity. The GloVe is a *log-bilinear* model with a weighted least-squares objective function trained on a global word-word co-occurrence matrix that shows words' co-occurrence frequency with one another in a given corpus. The main idea behind GloVe is that ratios of word-word co-occurrence probabilities encode meaning.

Another way used to represent words with numbers is using SentiWordNet (Esuli and Sebastiani, 2006). SentiWordNet is a lexical resource that provides numerical scores to each WordNet's *synset*, a set of synonyms (Miller, 1995). Specifically, SentiWordNet gives each word objective, positive and negative scores. Each of these score values is limited to the range between zero to one and their sum is one. The final score, i.e. the sentiment of a word can be calculated using the positive and negative scores.

5. Models for predicting the stock market

5.1 Time series analysis models

This section describes implemented models that predict future prices using only past data. We could say that these models perform classical time series analysis using DJIA closing prices. The first model implemented is the conventional statistical analysis ARIMA model and it's used as a benchmark in this paper. Next, time series analysis was performed using GRU with one linear layer in the end. The proposed model architecture is illustrated in Figure 1.

The mentioned model consists of only one GRU cell and one linear layer. The input dimension in the GRU cell is 1 while the hidden size is 32 which is also the input size into the linear layer. Since the final output is the predicted price, i.e. one number, its dimension is also 1.

5.2 Natural language processing models

After models that focus only on time series analysis, in this section, several different models that use news headlines and their sentiment alongside past prices were implemented.

The first model to do that is based on a one-dimensional convolutional neural network whose job is to extract sentiment from the news headlines. Mentioned models' architecture consists of several convolutional and maximum pooling layers followed by four linear layers at the end. After each convolutional and linear layer, there is a *ReLU* activation function. Furthermore, dropout is applied several times during the forwarding pass in the network to prevent overfitting. The models' architecture is illustrated in Figure 2.

In order to represent news headlines, GloVe was used, and each word was represented with a 300-dimensional vector. This model has two inputs, a list of preprocessed words (vectors) taken from 20 news headlines and the previous day's price. Since CNN needs fixed-size input, the list was limited to 98 words, which is the average size of words in

Headline text	Preprocessed text
The rough transcript shows Trump pressed Ukraine on Biden	'rough', 'transcript', 'show', 'trump', 'press', 'ukraine', 'biden'
WeChat becomes a powerful surveillance tool everywhere in China	'wechat', 'powerful', 'surveillance', 'tool', 'china'
NATO pushes EU to work with allies for security	'nato', 'push', 'eu', 'work', 'allies', 'security'
Stocks plunge and traders panic: 'Did someone fat-finger this?'	'stock', 'plunge', 'trader', 'panic', 'fat-finger'

Source(s): Table created by author

Table 1.
Examples of news headlines preprocessing

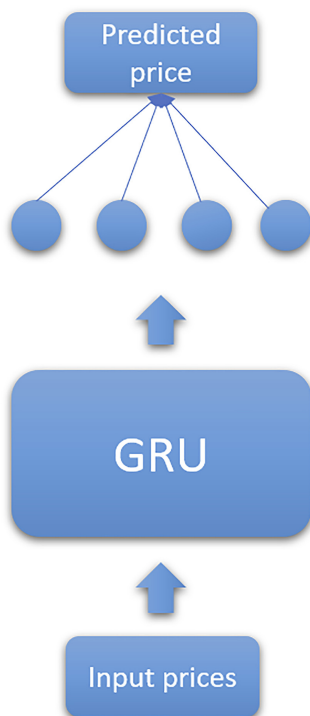


Figure 1.
GRU time series model
architecture

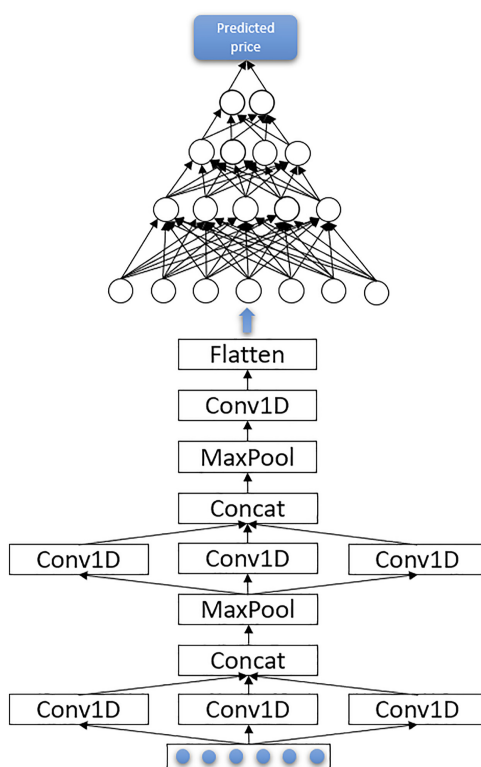
Source(s): Figure created by
author

20 news headlines. If the number of words in the news headlines overpass 98, they are discarded and if there are less than 98 words, padding is inserted to achieve the wanted size. While performing a forward pass with this model, in the penultimate linear layer the news sentiment is concatenated with the scaled previous day's price in order to predict the next day's price. The idea behind this approach is to look at the previous price and the next day's news in order to make a better prediction (Puh and Bagić Babac, 2022).

After a model that uses CNN to extract information from the news headlines, a more advanced architecture based on the LSTM is proposed. One of the advantages of LSTM over CNN is that there is no need to set a fixed-size input since LSTM can process arbitrary length sequences. This model also uses a price at time $t - 1$ alongside news sentiment at time t to predict the price at time t .

Figure 3 shows an illustration of the proposed model architecture that consists of an LSTM cell followed by two linear layers. Same as in the previous model, GloVe was used for word representation and the scaled price is concatenated with information extracted from the news headlines in the penultimate layer to make a prediction.

The next model combines recurrent neural networks and lexicon-based sentiment analysis for DJIA price prediction. To be more specific, words from the news headlines are converted into the sentiment score using SentiWordNet. Afterward, pairs of the historical prices and the news headlines polarities



Source(s): Figure created by author

Figure 2. CNN's model architecture

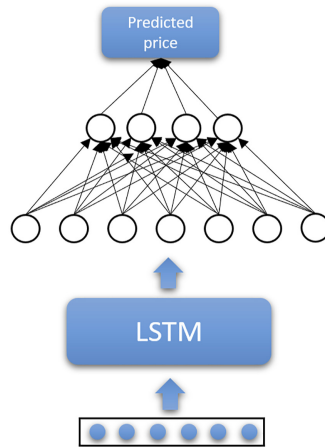
$$(Price_{t-1}, Score_t), (Price_{t-2}, Score_{t-1}), \dots, (Price_{t-m}, Score_{t-m+1})$$

are used for predicting the $Price_t$, where m represents the window size i.e. how many previous days are being considered for making the prediction. The architecture of the mentioned model consists of the recurrent neural network, for example, GRU or LSTM, and one linear layer in the end which outputs the predicted price. (Figure 4) During experiments, several architectures with different parameters were implemented and tested.

In the next approach, the same architecture based on the pairs of the last price and news sentiment score is used for price prediction. The only difference is in the way the sentiment score is calculated. Here the sentiment score is determined using a special version of BERT called FinBERT (Araci, 2019). It is built by fine-tuning the BERT model using the financial corpus for financial text sentiment classification. FinBERT takes text as input and returns one of three possible classes: *positive*, *neutral*, or *negative* alongside a number in the range between 0 and 1 that represents confidence. During the process of calculating the sentiment score (news polarity), we considered all 20 news headlines from each day and their confidence score.

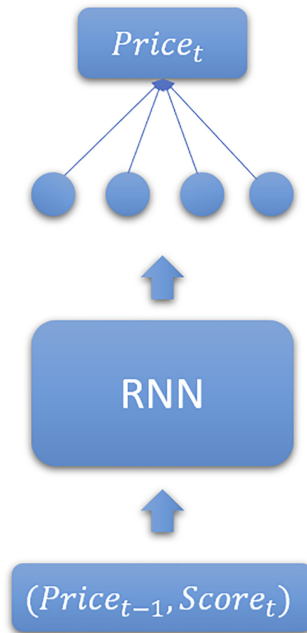
The last model proposed is constructed using the architecture from the previous model with one major difference. DJIA price at time t is predicted using pairs of the past price at time

Figure 3.
LSTM-based model
architecture



Source(s): Figure created by author

Figure 4.
Architecture of the
model based on the
news polarity



Source(s): Figure created by author

$t - 1$, sentiment score determined using FinBERT at time t and predicted price at time t . The predicted price is based on the time-series analysis GRU model with one linear layer which uses only the historical data to make a prediction.

6. Experimental results

The first thing we need to do, before the experiments, is to split the dataset to be able to test our models objectively. The original dataset is split into the training and testing dataset by an 80:20 ratio, and since this is a time-series task, data shuffle is not used. Since the US market operates only from Monday to Friday, only those days are used during the dataset’s creation and news headlines from the weekends are discarded.

Figure 5 shows DJIA prices in the period from January 2008 to December 2020. We can also see the dataset split which consists of 2,620 days (closing prices) for training and 656 days for testing. Next, we need some way to compare predicted prices with the actual prices, i.e. an error measure. The chosen evaluation methodology is called the RMSE, a standard deviation of the residuals.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (14)$$

RMSE is often the first choice when measuring the differences between numerical values, because it tells us how concentrated the values are around the line of the best fit.

The first model, which is also the benchmark in this paper, is the ARIMA time-series model. We experiment using different p , d and q parameters, but also with different window sizes. Window size is a crucial factor in the time-series analysis since it defines how many past values are considered for making a prediction. Table 2 shows experimental results using the ARIMA model for different hyperparameter combinations.

The best-achieved result using the ARIMA model is RMSE of 399.128 on the test dataset. In order to see the difference between the predicted and the actual prices, Figure 6 shows the comparison over 30 days.

After the ARIMA model, DJIA prices were predicted using GRU with one linear layer and only historical prices. The training process was conducted using different hyperparameter combinations and the results are shown in Table 3.

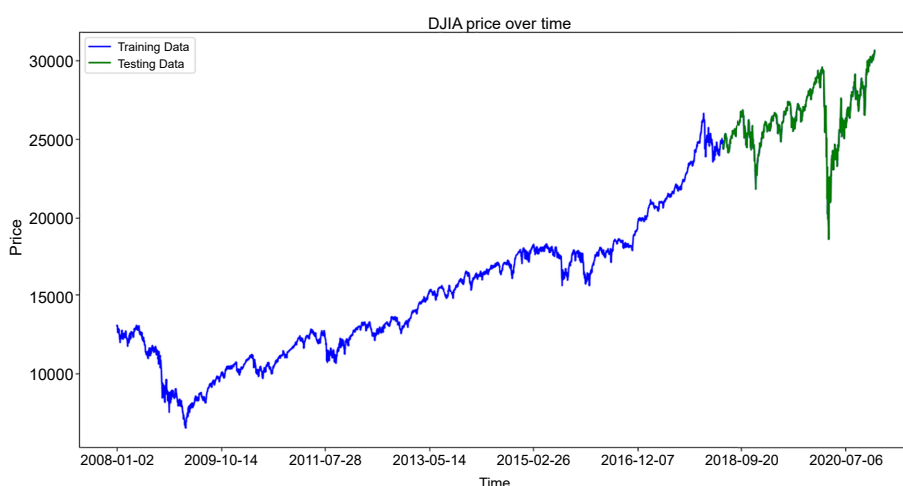


Figure 5.
Visualization of DJIA
dataset split

Source(s): Figure created by author

The results shown in Table 3 are achieved using the Adam optimizer and the *Mean Squared Error* (MSE) loss function (Goodfellow *et al.*, 2016).

The next model is based on CNN and it is the first one to use news headlines alongside historical prices as input. The results achieved by this model are summed up in Table 4.

Table 4 shows that the mentioned model outperforms previous approaches which do not use any information from the news headlines. As in the previous model, the MSE loss function and Adam optimizer were used during the training process. Figure 7 shows the difference between the real price and the price over 30 days predicted using the CNN model.

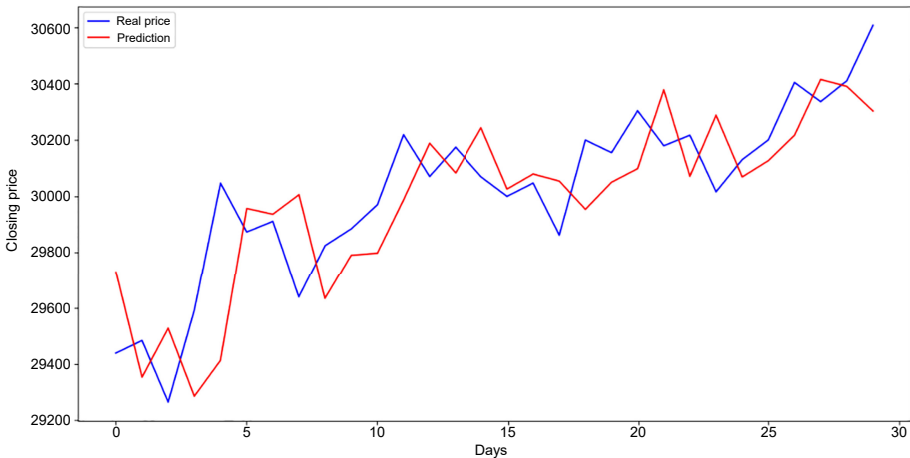
After the model that extracts information from the news headlines using CNN, this next model uses a more advanced LSTM architecture. Since the LSTM cell can have many different configurations like a various number of layers, dropout and it can also be easily

Table 2.
Results for different hyperparameters

Window size	p, d, q	RMSE
10	3, 1, 0	417.378
40	1, 1, 0	401.531
40	3, 1, 0	403.934
40	5, 1, 0	411.784
50	1, 1, 0	399.128
50	5, 1, 0	408.342

Source(s): Table created by author

Figure 6.
Comparison of the real and the ARIMA model predicted price



Source(s): Figure created by author

Table 3.
Results obtained using GRU with one linear layer

Window size	Learning rate	RMSE
10	0.01	394.903
20	0.02	391.426
20	0.01	392.554
30	0.02	393.215

Source(s): Table created by author

transformed into bidirectional LSTM, various combinations were tested during the experiments and the results are shown in Table 5.

Furthermore, we experiment with the model that uses pairs of historical price and sentiment scores calculated using SentiWordNet. Since the core of this model is a recurrent neural network, we experiment with different architectures. Table 6 shows hyperparameter combinations that achieved the best results.

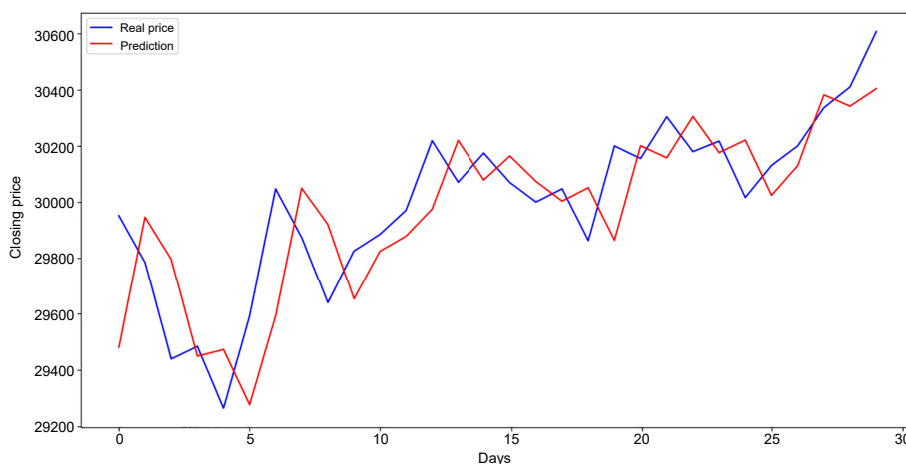
The following model uses the same architecture as the previous one and differs only in the process of calculating the news sentiment scores. In this approach, we use FinBERT for determining those scores. Table 7 shows the best results achieved by this model.

Finally, the last model uses the GRU models' price prediction alongside the pair of the historical price and the sentiment score provided by FinBERT. Table 8 shows the experimental results achieved by this configuration. Figure 8 shows the comparison of the best-performing models' prediction and the real price over 30 days.

Learning rate	Dropout probability	RMSE
0.002	0.2	388.423
0.001	0.2	386.126
0.001	0.6	385.422

Source(s): Table created by author

Table 4. CNN model performance overview



Source(s): Figure created by author

Figure 7. The difference between the real price and the price predicted using the CNN model

Bidirectional	Learning rate	No. of layers	Dropout	RMSE
False	0.001	1	0.1	385.782
False	0.0008	1	0.2	384.287
False	0.0008	2	0.2	386.024
True	0.001	1	0.3	386.580
False	0.01	1	0.2	391.873

Source(s): Table created by author

Table 5. Results achieved by the LSTM model

After experimenting with all the above models, the best results from each of them are shown in [Table 9](#).

56

Table 6.
Comparison of the results achieved using different RNN architectures

Architecture	Learning rate	Window size	RMSE
LSTM	0.01	3	378.447
LSTM	0.01	4	383.083
LSTM	0.02	3	377.522
GRU	0.01	3	376.323
GRU	0.001	3	386.155
GRU	0.02	4	378.864

Source(s): Table created by author

Table 7.
FinBERT models' RMSE score on the test set

Architecture	Window size	RMSE
GRU	3	374.103
GRU	4	375.215
GRU	5	386.996

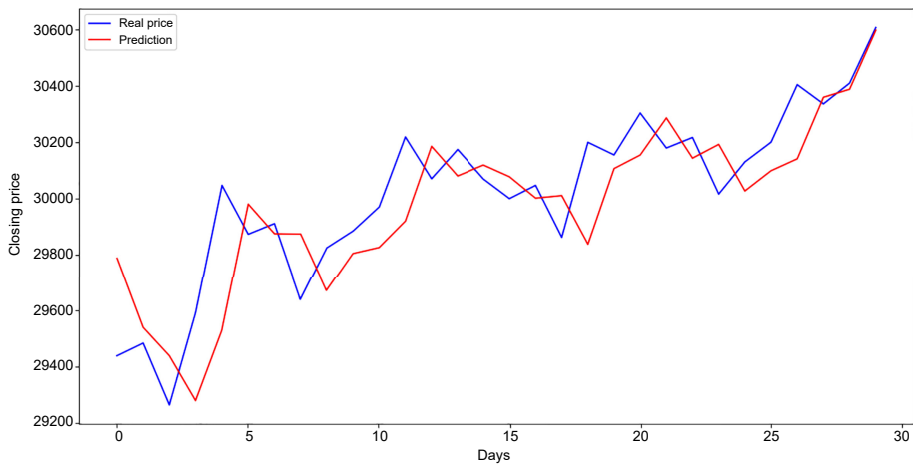
Source(s): Table created by author

Table 8.
The last model's performance on the test set

Architecture	Window size	RMSE
GRU	3	370.155
GRU	4	378.178
GRU	5	381.346

Source(s): Table created by author

Figure 8.
Comparison of the best-performing models' prediction and the real price



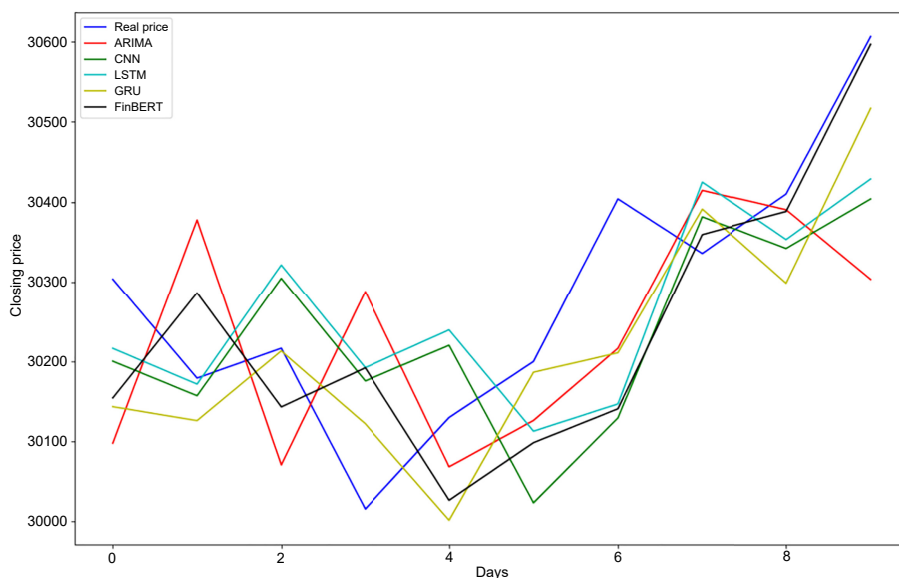
Source(s): Figure created by author

Model	RMSE
ARIMA	399.128
GRU (only prices)	391.426
CNN	385.422
LSTM	384.287
GRU (price and sentiment score pairs)	376.323
FinBERT	374.103
FinBERT (with prediction)	370.155

Source(s): Table created by author

Table 9.
RMSE for the tested models

We can conclude that the simplest ARIMA model achieved the worst result, i.e. has the largest RMSE on the testing data, followed by the GRU model which also uses only historical prices without any additional information for predictions. Although not huge, the difference is easily spotted in the results of the CNN-based model which extracts information from the news headlines. The fact that the LSTM architecture generally performs better than the CNN is not significantly manifested in this case since the difference in the RMSE is not big. The next noticeable difference between the results was achieved when feeding the pairs of the historical prices and the news sentiment to the recurrent neural network. In that case, the GRU-based model achieved slightly better results than the LSTM model. Furthermore, using the FinBERT model to calculate the news sentiment scores additionally improved the GRU models' performance. Finally, the model that used pairs of historical prices, sentiment scores, and other models' predictions managed to outperform all the previous models' results. Figure 9 shows the comparison of the real and the prices predicted using some of the implemented models. The prices are shown during the last ten days in the testing dataset.



Source(s): Figure created by author

Figure 9.
DJIA price prediction using different models

7. Conclusion

Given the rising interest in investments in the stock market, there is a need to improve the chance of making a good investment using tools that predict future prices. Successful stock price prediction is extremely hard because a lot of different factors affect its price. Besides the obvious factors such as economic and political, things like the oil price, interest rates or even some unplanned events can make the stock market price deviate. Most of the tools today rely only on historical prices when predicting future prices and ignore all of the above factors. Consequently, the results they achieve are not staggering. To improve stock price predictions' accuracy, we needed to find a way to take into consideration as many as possible of previously mentioned factors. One of the places where most of them are mentioned is in the news. For that reason, we developed a set of computational models which use information extracted from the news headlines alongside historical prices to make a better prediction.

This study provides a comprehensive framework for using NLP to improve stock price prediction and confirms the research hypothesis that there is a correlation between news headlines and stock price prediction. In addition, it confirmed that the FinBERT-based model outperforms all the other tested models achieving the lowest RMSE on the test set. For methodological comparison to related work, besides the benchmark ARIMA model, various state-of-the-art NLP models and architectures based on RNN and BERT were implemented and tested, which showed that the best result uses a FinBERT. Specifically, in the best-performing model, the sentiment score is calculated using FinBERT and then the sentiment score, price from the previous day and the prediction of the future price generated using a simple time-series analysis model are fed into a GRU cell with one linear layer in the end. As result, the model achieved an RMSE of 370.155 on the test set of 656 days.

Since NLP models for predicting stock prices have shown to have a marginal improvement over traditional techniques, our results can be interpreted in two ways. One way is to continue to support the traditional approach in a setting where a computationally less-intensive method is required that still achieves acceptable results. Another way is to use modern NLP techniques which improve the results to the extent depending on the different features of a specific dataset.

There are several limitations regarding using news headlines for stock market prediction. First, in this study, the top 20 news headlines for each day were scraped from the Wall Street Journal website. However, it is very common that many of those 20 headlines do not provide any useful information that can be used as an indicator of stock price movement. Furthermore, in this paper, we predicted the closing price of DJIA, which is not related to a single company. If that were the case, we could use the information from the news related only to the specific company and its internal politics. That approach would for sure significantly improve the models' accuracy, e.g. [Shilpaand and Shambhavi \(2021\)](#) obtained high accuracy using a stock dataset that includes two companies such as Reliance Communications and Relaxo Footwear).

Regarding the technical aspect of future work, possible improvements can be realized using the combination of GANs and FinBERT ([Sonkiya et al., 2021](#)). However, future avenues of this work may also involve analysis of filtered news, that is news that might possibly significantly improve the accuracy for predicting stock, i.e. generated by highly influential persons ([Metta et al., 2022](#)), organizations, or companies. Moreover, user comments, reactions and emotions to financial news may also make an avenue for future research ([Bagić Babac, 2022](#)).

Notes

1. <https://finance.yahoo.com/quote/%5EDJI/>
2. <https://spacy.io/>

References

- Araci, D. (2019), *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*, arXiv preprint arXiv:1908.10063. Master Thesis, Master information studies Data Science, Faculty of Science, University of Amsterdam.
- Bagić Babac, M. (2022), "Emotion analysis of user reactions to online news", *Information Discovery and Delivery*, Vol. ahead-of-print No. ahead-of-print, doi: [10.1108/IDD-04-2022-0027](https://doi.org/10.1108/IDD-04-2022-0027).
- Bifet, A. and Gavaldà, R. (2007), "Learning from time-changing data with adaptive windowing", *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, pp. 443-448.
- Cheng, W. and Chen, S. (2021), "Sentiment analysis of financial texts based on attention mechanism of FinBERT and BiLSTM", *2021 International Conference on Computer Engineering and Application (ICCEA)*, pp. 73-78, doi: [10.1109/ICCEA53728.2021.00022](https://doi.org/10.1109/ICCEA53728.2021.00022).
- Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019), "Bert: pre-training of deep bidirectional transformers for language understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, Vol. 1, Association for Computational Linguistics, pp. 4171-4186, (Long and Short Papers), doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423).
- Esuli, A. and Sebastiani, F. (2006), "SENTIWORDNET: a publicly available lexical resource for opinion mining", *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, European Language Resources Association (ELRA).
- Fazlija, B. and Harder, P. (2022), "Using financial news sentiment for stock price direction prediction", *Mathematics*, Vol. 10 No. 13, p. 2156, doi: [10.3390/math10132156](https://doi.org/10.3390/math10132156).
- Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, MA.
- Hochreiter, S. and Schmidhuber, J. (1997), "Long short-term memory", *Neural Computation*, Vol. 9 No. 8, pp. 1735-1780.
- Idrees, S.M., Alam, M.A. and Agarwal, P. (2019), "A prediction approach for stock market volatility based on time series data", *IEEE Access*, Vol. 7, pp. 17287-17298.
- Ji, X., Wang, J. and Yan, Z. (2021), "A stock price prediction method based on deep learning technology", *International Journal of Crowd Science*, Vol. 5 No. 1, pp. 55-72.
- Jurafsky, D. and Martin, J.H. (2000), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, Upper Saddle River, NJ.
- Kameshwari, S., Kaniskaa, S., Kaushika, S. and Anuradha, R. (2021), "Stock trend prediction using news headlines", *2021 IEEE India Council International Subsections Conference (INDISCON)*, pp. 1-5, doi: [10.1109/INDISCON53343.2021.9582219](https://doi.org/10.1109/INDISCON53343.2021.9582219).
- Kosteļej, M. and Bagić Babac, M. (2022), "Text analysis of the Harry Potter book series", *South Eastern European Journal of Communication*, Vol. 4, pp. 17-30.
- Kotu, V. and Deshpande, B. (2019), "Chapter 12 – Time Series Forecasting", in Kotu, V. and Deshpande, B. (Eds), *Data Science*, 2nd ed., Morgan Kaufmann, Cambridge, MA, pp. 395-445.
- Kumari, J., Sharma, V. and Chauhan, S. (2021), "Prediction of stock price using machine learning techniques: a survey", *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, 2021, pp. 281-284.
- Marijić, A. and Bagić Babac, M. (2023), "Predicting song genre with deep learning", *Global Knowledge, Memory and Communication*, Vol. ahead-of-print No. ahead-of-print, doi: [10.1108/GKMC-08-2022-0187](https://doi.org/10.1108/GKMC-08-2022-0187).
- Matei, O., Rusu, T., Petrovan, A. and Miḥuṭ, G. (2017), "A data mining system for real time soil moisture prediction", *Procedia Engineering*, Vol. 181, pp. 837-844.
- Mehtab, S. and Sen, J. (2019), "A robust predictive model for stock price prediction using deep learning and natural language processing", *Proceedings of the 7th International Conference on Business Analytics and Intelligence (BAICONF, 2019)*, Bangalore, India, Indian Institute of Management.

- Metta, S., Madhavan, N. and Narayanan, K.K. (2022), "Power of 280: measuring the Impact of elon musk's tweets on the stock market", *Ushus-Journal of Business Management*, Vol. 21 No. 1, pp. 17-43.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J. (2013), "Distributed representations of words and phrases and their compositionality", *Advances in Neural Information Processing Systems*, Curran Associates, Red Hook, NY, pp. 3111-3119.
- Miller, G.A. (1995), "WordNet: a lexical database for English", *Communications of the ACM*, Vol. 38 No. 11, pp. 39-41.
- Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P. and Anastasiu, D.C. (2019), "Stock price prediction using news sentiment analysis", *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, pp. 205-208.
- Musso, M.-B.I and Bagić Babac, M. (2022), "Opinion mining of online product reviews using a lexicon-based algorithm", *International Journal of Data Analysis Techniques and Strategies*, Vol. 14 No. 4, pp. 283-301.
- Pennington, J.S. and Manning, C.R. (2014), "Glove: global vectors for word representation. EMNLP", available at: <https://nlp.stanford.edu/projects/glove/>
- Puh, K. and Bagić Babac, M. (2022), "Predicting sentiment and rating of tourist reviews using machine learning", *Journal of Hospitality and Tourism Insights*, Vol. ahead-of-print No. ahead-of-print, doi: [10.1108/JHTI-02-2022-0078](https://doi.org/10.1108/JHTI-02-2022-0078).
- Roy, M., Seethi, V.D.R. and Bharti, P. (2022), "CovidAlert - a wristwatch-based system to alert users from face touching", in Lewy, H. and Barkan, R. (Eds), *Pervasive Computing Technologies for Healthcare, PH 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, Cham, Vol. 431.
- Sherstinsky, A. (2020), "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network", *Physica D: Nonlinear Phenomena*, Vol. 404, 132306.
- Shilpa, B.L. and Shambhavi, B.R. (2021), "Combined deep learning classifiers for stock market prediction: integrating stock price and news sentiments", *Kybernetes: The International Journal of Systems and Cybernetics*, Vol. 52 No. 3, pp. 748-773, doi: [10.1108/K-06-2021-0457](https://doi.org/10.1108/K-06-2021-0457).
- Sidogi, T., Mbuva, R. and Marwala, T. (2021), "Stock price prediction using sentiment analysis", *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021, pp. 46-51, doi: [10.1109/SMC52423.2021.9659283](https://doi.org/10.1109/SMC52423.2021.9659283).
- Sonkiya, P., Bajpai, V. and Bansal, A. (2021), "Stock price prediction using BERT and GAN", ArXiv, abs/2107.09055.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017), "Attention is all you need", *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, Red Hook, NY, USA, Curran Associates Inc., pp. 6000-6010.
- Wadi, S.A.L., Almasarweh, M., Alsarairh, A.A. and Aqaba, J. (2018), "Predicting closed price time series data using ARIMA Model", *Modern Applied Science*, Vol. 12 No. 11, pp. 181-185.
- Wang, Y. and Wang, Y. (2016), "Using social media mining technology to assist in price prediction of stock market", *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, 2016, pp. 1-4, doi: [10.1109/ICBDA.2016.7509794](https://doi.org/10.1109/ICBDA.2016.7509794).
- Yadav, A., Jha, C.K. and Sharan, A. (2020), "Optimizing LSTM for time series prediction in Indian stock market", *Procedia Computer Science*, Vol. 167, pp. 2091-2100.
- Zulqarnain, M., Ghazali, R., Ghouse, M.G., Hassim, Y.M.M. and Javid, I. (2020), "Predicting financial prices of stock market using recurrent convolutional neural networks", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol. 12 No. 6, pp. 21-32.

Further reading

- Bagić Babac, M. and Podobnik, V. (2016), "A sentiment analysis of who participates, how and why, at social media sports websites: how differently men and women write about football", *Online Information Review*, Vol. 40 No. 6, pp. 814-833, doi: [10.1108/OIR-02-2016-0050](https://doi.org/10.1108/OIR-02-2016-0050).

-
- Cvitanović, I. and Bagić Babac, M. (2022), “Deep Learning with self-attention mechanism for fake news detection”, in Lahby, M., Pathan, A.S.K., Maleh, Y. and Yafooz, W.M.S. (Eds), *Combating Fake News with Computational Intelligence Techniques*, Springer, Switzerland, pp. 205-229.
- Gao, T., Chai, Y. and Liu, Y. (2017), “Applying long short term memory neural networks for predicting stock closing price”, *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 575-578, doi: [10.1109/ICSESS.2017.8342981](https://doi.org/10.1109/ICSESS.2017.8342981).
- Gite, B., Sayed, K., Mutha, N., Marpadge, S. and Patil, K. (2017), “Surveying various genetic programming (GP) approaches to forecast real-time trends & prices in the stock market”, *2017 Computing Conference*, pp. 131-134.
- Islam, S.B., Hasan, M.M. and Khan, M.M. (2021), “Prediction of stock market using recurrent neural network”, *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0479-0483, doi: [10.1109/IEMCON53756.2021.9623206](https://doi.org/10.1109/IEMCON53756.2021.9623206).
- Kavinnilaa, J., Hemalatha, E. and Jacob, M.S. and Dhanalakshmi, R. (2021), “Stock price prediction based on LSTM deep learning model”, *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2021, pp. 1-4, doi: [10.1109/ICSCAN53069.2021.9526491](https://doi.org/10.1109/ICSCAN53069.2021.9526491).
- Mathanprasad, L. and Gunasekaran, M. (2022), “Analysing the trend of stock market and evaluate the performance of market prediction using machine learning approach”, *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2022, pp. 1-9.
- Poch Alonso, R. and Bagić Babac, M. (2022), “Machine learning approach to predicting a basketball game outcome”, *International Journal of Data Science*, Vol. 7 No. 1, pp. 60-77.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), “Dropout: a simple way to prevent neural networks from overfitting”, *The Journal of Machine Learning Research*, Vol. 15 No. 1, pp. 1929-1958.

Corresponding author

Marina Bagić Babac can be contacted at: marina.bagic@fer.hr

For instructions on how to order reprints of this article, please visit our website:

www.emeraldgrouppublishing.com/licensing/reprints.htm

Or contact us for further details: permissions@emeraldinsight.com