Forecasting Directional Movement of Stock Prices using Deep Learning

Deeksha Chandola[1] · **Akshit Mehta[2]** · **Shikha Singh[3]** ·
Vinay Anand Tikkiwal[3] · **Himanshu Agrawal[4]**

Abstract

Stock market's volatile and complex nature makes it difficult to predict the market situation. Deep Learning is capable of simulating and analyzing complex patterns in unstructured data. Deep learning models have applications in image recognition, speech recognition, natural language processing (NLP), and many more. Its application in stock market prediction is gaining attention because of its capacity to handle large datasets and data mapping with accurate prediction. However, most methods ignore the impact of mass media on the company's stock and investors' behaviours. This work proposes a hybrid deep learning model combining Word2Vec and long short-term memory (LSTM) algorithms. The main objective is to design an intelligent tool

Akshit Mehta, Shikha Singh, Vinay Anand Tikkiwal, Himanshu Agrawal these authors contributed equally to this work.

✉ Deeksha Chandola
deekshc@cse.yorku.ca

Akshit Mehta
akshit.9@gmail.com

Shikha Singh
shikhasingh10996@gmail.com

Vinay Anand Tikkiwal
vinay.anand@jiit.ac.in

Himanshu Agrawal
himanshu.agrawal@jiit.ac.in

[1] Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

[2] Hebert Werthim College of Engineering, University of Florida Gainesville, Gainesville, USA

[3] Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology, Noida, India

[4] Department of Computer Science & Engineering and Information Technology, Jaypee Institute of Information Technology, Noida, India

to forecast the directional movement of stock market prices based on financial time series and news headlines as inputs. The binary predicted output obtained using the proposed model would aid investors in making better decisions. The effectiveness of the proposed model is assessed in terms of accuracy of the prediction of directional movement of stock prices of five companies from different sectors of operation.

1 Introduction

Stock market behaviour is unpredictable and depends on various factors, including but not limited to the global economy, geopolitical conditions, company's performance, investor expectations, and financial reports. A company's profit plays a vital role in determining gain or loss in the stock price. Also, it is a challenge for an investor to predict market behaviour. In order to make a profitable investment, there is a need for advanced knowledge related to the market trend, which could help these investors minimize risk and maximize their profit. The advancement of the data science field that incorporates various algorithms, tools and machine learning principles to discover hidden information or patterns from the raw data as the primary goal [1]. Recent studies suggested an increasing demand for data science and forecasting methods to predict stock market behaviour and get insights related to a company's financial health. In this work, we have developed a forecasting model that seeks to predict future values based on past, and present data [2]. The forecasting algorithm is so crucial for companies because predicting future events or, in this case, future stock prices is a critical input to many types of planning and decision-making processes such as financial management and risk management [3].

In literature, three broad approaches have been proposed for predicting the stock price of an organization. The first methodology is technical analysis in which the historical price of stocks, like closing and opening price, the volume traded, and the relative values are used in predicting the future price of the stock [4]. The second is qualitative analysis, which is performed based on external factors like company profile, market situation, political and economic factors, textual information in the form of financial news articles, social media (Twitter) and even blogs by economic analyst [5]. The third approach is the hybrid analysis which includes predicting stock price market movement based on the company's financial data as well as financial news articles in order to improve the quality of prediction [6]. These days availability of information from various sources and easy accessibility to a company's financial data make it simpler to integrate for stock market forecasting.

Most of the previous work in this area has revealed stock market prediction as a challenging task [7]. Early work introduced the efficient market hypothesis (EMH) [8], which is based on the fact that an asset's price is unpredictable until complete past trading information is provided. Similarly, random walk theory (RWT) [9] has been proposed for predicting the stock price, which is based on the fact that stock prices do not depend on history; instead, it only reacts to new changes or information. These two hypotheses depend only on the financial dataset; they do not account for

other sources such as news articles, social media information, financial blogs and many more. With the multi-sourced information available online, these two hypotheses proved highly inaccurate in predicting the stock price. Some classical algorithms based on technical analysis are also used, which include moving average convergence divergence (MACD), and rate of change (ROC), which gives the ratio of the current stock price to the past stock price of n days earlier, where n days varies from 5 to 10 days stock price data [10] and relative strength index (RSI) which gives an analysis of latest upward price trend relative to the downward price trend, considering 9 to 14 days data approximately [11].

Recent work in the domain shows that stock market prediction can be enhanced using machine learning techniques. Standard algorithms like linear regression predicted directional movement of the stock price using technical data. Support vector machine (SVM) [12] forecast weekly stock market movement direction of NIKKEI 225 index using financial news articles as compared with other forecasting methods like linear discriminate analysis (LDA) and backpropagation neural networks. The main problems with these forecasting methods are that they do not consider the temporal nature of news data, which is essential for achieving greater accuracy.

In particular, for stock market analysis, the data size is huge, multi-sourced, and non-linear. To deal with this variety of data, we need highly efficient models to identify the hidden patterns and complex relations in a large corpus. Deep learning hierarchical models such as convolutional neural network (CNN) [13], recurrent neural network (RNN) [14, 15], long short term memory (LSTM) [16] and recurrent convolutional neural network (RCNN) [17] are capable for finding the hidden features through a self learning process. The performance of five neural network architectures such as multilayer perceptron (MLP) network, elman neural network (ENN), jordan neural network (JNN), radial basis functions neural networks (RBF), and multiple linear regression (MLR) to predict the six most traded stocks of the official Brazilian stock exchange during the Covid-19 period is compared in [18]. Their analysis concluded that these models provide reasonable predictions and thus can be used as support models for the companies. Not just stock market forecasting but the use of deep learning models are widely adapted for forecasting solar radiance [19, 20]. These models have certain advantages over the traditional techniques, such as taking a vast dataset as input in learning complex non-linearity in the dataset through various levels in the network, hence avoiding over-fitting.

From the above discussion, it is possible to understand some crucial points in constructing a deep learning model. The first point is related to the selection of period; with the unpredictable nature of stock prices, the authors suggested that intra-day prediction is highly suitable over weekly or monthly price prediction [21]. The second point is an extension of the first, which includes the collection of massive relevant data in order for accurate prediction [13]. The growing web information and easy accessibility of the company's data from their website contribute to gathering huge relevant information required for the training of the deep neural network models. With the vast availability of textual data, all the financial news articles are crawled from the web and can be used for further analysis. It is required as many recent studies have shown that sentiment and emotions are significant in the decision-making process of investors. Many natural language processing (NLP) techniques such as bags-of-words, named

entities, noun phrases and also embedding techniques are used for feature extraction [22]. A model using an artificial neural network (ANN) for predicting stock market movement was developed by [23]. They trained the model on the stock prices of Google (NASDAQ: Google) and also collected financial news articles from Dow Jones to predict the upward and downward movement of the stock. Another work [24] proposed a hybrid model that utilizes news articles for text mining and the corresponding stock prices of the selected companies. They extracted the textual psycholinguistic features using LIWC and TAALES software from the collected news articles related to Indian companies. They experimented with various regression models such as random forest (RF) and SVR. They concluded that models with LIWC features achieve the best prediction compared to TAALES features.

In this paper a hybrid deep learning model is proposed, which combines a deep learning model long short term memory (LSTM), with the Word2vec model to predict the intra-day directional movement of the company's stock price. The choice of LSTM as the model makes the prediction of stock price direction more efficient and accurate as it can hold past information. The model uses the closing price of financial time series data, and financial news headlines published the day before as input. The effectiveness of the model is tested on five different sectors of companies, such as in the technology sector: Apple, fast-moving consumer goods company: PepsiCo, integrated power sector: Nuclear electric power generation (NRG), consumer services sector: American Public Education (APEI) and communication services sector: American Telephone and Telegraph (AT&T).

The paper is organized as follows: Section 2 represents model implementation with an overview of the theoretical concepts of machine learning. Section 3 includes the detailed proposed model algorithm, and in Sect. 4, results are presented and discussed, with Sect. 5 concluding the paper.

2 Prediction using LSTM

Stock market prediction is a complex task due to its dependability on many factors such as market trends and financial news in the market [25]. In this section, the proposed Word2vec-LSTM model design is explained in detail to predict the directional movements of the stock market, using financial time series and news headlines as input. A brief structure of the model is shown in Fig. 1. This design consists of five stages: Input layer, Word Embedding, LSTM, Dense and Output Layer. These stages are explained below.

2.1 Input Layer

The model uses two types of input, the first one is the closing price of the financial time series, and the second one is the set of news headlines. These inputs are shown separately in the Fig. 1 for better understanding.

The financial times series database of Apple, PepsiCo, NRG, AT&T and APEI is selected. This database is obtained from [26] for the period of August 8, 2008, to July
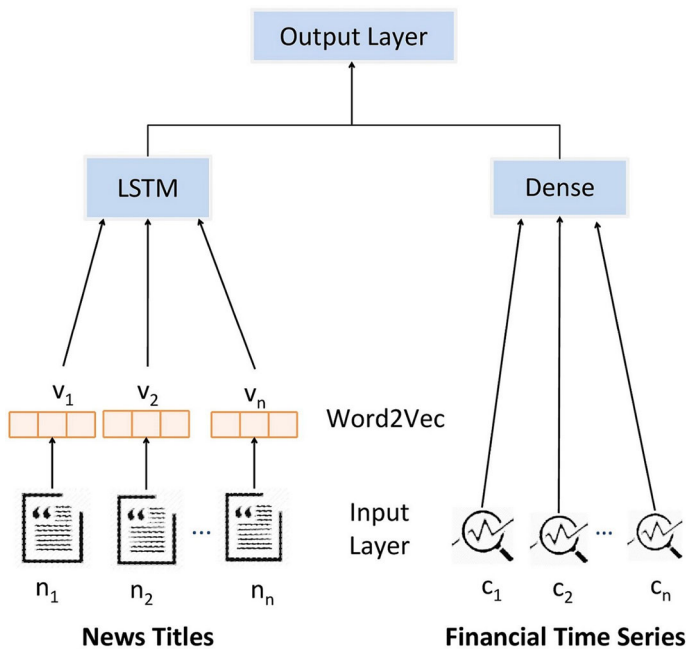
Fig. 1 Block diagram of the proposed Hybrid Model

Table 1 Statistics of the dataset

	Dataset	Training dataset	Validation dataset
Time	08/08/2008	08/08/2008	01/01/2015
Interval	−01/07/2016	−31/12/2014	−01/07/2016

1, 2016. The closing price of each of these databases is used to calculate the binary value of the target output. The calculated value forecast the directional movement by comparing the closing price (c_{t+1}) of t+1 with the closing price (c_t) of the t day.

In the case of news headlines database is obtained from the [27] for the same period. The database consists of the publishing date and the headlines of that day. In the work of [21] experiments were carried out, which shows that news titles are more efficient in forecasting than news content. There from, the proposed model uses only news titles. News of each and every day is shown in Fig. 1 as n_n. Finally, these two inputs are aligned according to their date. Table 1 shows a statistic of the dataset that is used for training and validation.

2.2 Word Embedding

In order to generate word-embedding of a large corpus of news headlines, a Word2vec model is used. Word embedding is a technique of feature learning and language modelling where words and sentences are transformed into vector representation [28, 29].

S No.	Label	News Headlines
0	0	['b"Georgia \'downs two Russian warplanes\' as...
1	1	["b'Why wont America and Nato help us? If they...
2	0	["b'Remember that adorable 9-year-old who sang...
3	0	["b' U.S. refuses Israel weapons to attack Ira...
4	1	["b'All the experts admit that we should legal...
5	1	["b"Mom of missing gay man: Too bad he\'s not ...
6	0	["b'In an Afghan prison, the majority of femal...
7	0	["b"Man arrested and locked up for five hours ...
8	1	["b'Two elderly Chinese women have been senten...
9	1	['b"British resident held in Guantanamo Bay wi...

Fig. 2 Sample News headlines

It allows capturing the relations in language that are very difficult to capture manually. With the ongoing research in the NLP area, many papers use word embedding models. Based on the results reported in these papers [22, 30–32] we have used Word2vec model in this work.

We used a large corpus of day-wise financial news headlines constituting various stocks as input in this model. It is responsible for affecting the stock market price trend. It is challenging to handle this vast data using traditional data processing tools. With the main objective of decreasing the enormous dataset dimension and eliminating the risk of overfitting, a highly efficient Word2vec learning algorithm is applied. The main idea behind this model is to have all the words with similar contexts occupy close positions. Gensim, a python library, is applied to implement Word2vec. In generating word vectors from the input news headlines, Word2vec starts with the first step of building a vocabulary from the text corpus and starts learning the vector representation of each word. In the second step, it starts calculating the cosine distance between each word, which in turn generates a cluster of similar words. This ability of Word2vec helps to reduce the feature dimension. Finally, an embedding matrix is created where each row represents a training example, and columns indicate generated word vectors. Figures 2 and 3 shows the news headlines before and after the Word2vec algorithm is applied.

2.3 Long Short-Term Memory (LSTM)

LSTMs are a particular type of recurrent neural network (RNN) [33]. They can capture context-specific information from large datasets and use them for future prediction. As its name suggests, each LSTM unit or cell remembers the information for a long or short duration of time. Predicting the output of its new cell state, it takes the information stored in the old cell state. This feature provides memory to the network, which helps improve future predictions. Thus, LSTM networks are best capable of finding out how

```
[array([-0.52349484,  0.2628807 ,  0.4286829 ,  0.200244  ,  0.45367444,
        -0.2915945 ,  0.5092316 , -0.21547839, -0.27057257,  0.22221328,
        -0.14690973,  0.07771457,  0.65829605, -0.38468844,  0.4692674 ,
         0.11054122, -0.3916847 ,  0.18724509, -0.02099981, -0.2582507 ,
        -0.1280313 ,  0.28013536,  0.29736   ,  0.0019753 ,  0.53897583,
        -0.04553477, -0.5350276 ,  0.18257523,  0.1083542 ,  0.1463223 ,
        -0.253804  ,  0.47878805, -0.17264816,  0.3341093 , -0.5871108 ,
        -0.2971948 ,  0.53448063, -0.13676217,  0.23416486, -0.22946547,
         0.06942674, -0.22940116,  0.5194889 , -0.00531626, -0.18557093,
         0.14532788, -0.11776464,  0.04379674, -0.59225804,  0.22479847,
         0.04332066,  0.16221142, -0.15797912,  0.4634164 , -0.29215774,
         0.55699193, -0.22516586, -0.22522834,  0.4085444 ,  0.14070989,
        -0.5172651 , -0.16827731, -0.05604618, -0.03134907,  0.2666442 ,
        -0.33521312, -0.16387203, -0.23454595, -0.00234719,  0.6385315 ,
        -0.5616836 , -0.0469989 ,  0.9302666 , -0.24199715, -0.23300712,
```

Fig. 3 News Headlines processed using Word2vec algorithm

financial news headlines and closing prices of stocks affect the price trends of several stocks over a more extended period. These networks also decide for how long the past information related to stock price needs to be stored to predict the new price trends accurately.

LSTM network can be seen as a gated cell. Gated means that the cell decides whether or not to store or delete information based on the importance or weights it assigns to the information. LSTM has three gates: input, forget and output gate. The forget gate, f_t, decides which state needs to be remembered and which should be forgotten. The input gate, i_t, decides which value is updated by the input signal. Finally, the output gate, o_t, decides whether the cell state will affect other neurons or not. Moreover, it consists of a logistic layer and the layer where the former layer generates numbers between zero and one, and the last layer generates a new vector that gets added up to the state. The LSTM equations are shown in equations (1)–(6) below. Where x_t, the output of the embedding layer, is used as an input to the recurrent neural network layer, y_t is the output, and W is the weight matrix. The key element of LSTM s_t acts as a memory cell and is controlled by three different gates. Figure 4 shows the detailed architecture of LSTM.

$$f_t = \sigma(W_{xf}X_t + W_{yf}Y_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma(W_{xi}X_t + W_{yi}Y_{t-1} + b_i) \tag{2}$$

$$g_t = \tanh(W_{xg}X_t + W_{yg}Y_{t-1} + b_g) \tag{3}$$

$$s_t = f_t * s_{t-1} + i_t * g_t \tag{4}$$

$$o_t = \sigma(W_{xo}X_t + W_{yo}Y_{t-1} + b_o) \tag{5}$$
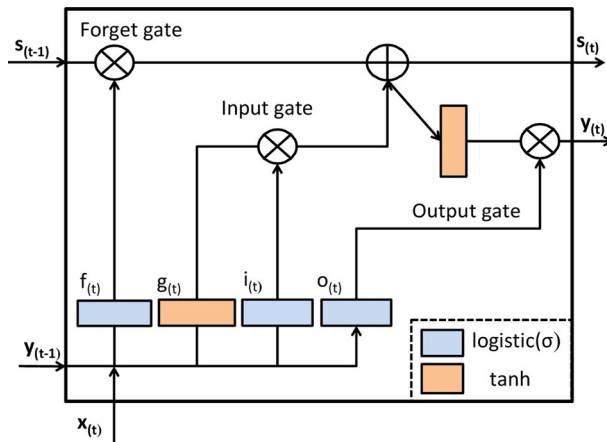
$$y_t = O_t * \tanh(s_t) \tag{6}$$

Fig. 4 LSTM Architecture

Table 2 List of abbreviations used in Sect. 3

Abbreviation	Description
N	News Headliness
T	Financial Time Series
W	Window Size
E	Embedding Size
F	Forecast, for this model F = 1
Fm	Final Model
t_r	Training Dataset
t_s	Testing Dataset
Nt_r, Nt_s	News titles of training and testing dataset
Tt_r, Tt_s	Financial time series of training and testing dataset
$Nt_r V, Nt_s V$	Vector value of news titles of training and testing dataset
Xt_r, Xt_s	X-value of time series of training and testing dataset
$Nt_r X, Nt_s X$	X-value of news titles of training and testing dataset
Yt_r, Yt_s	Binary output value of training and testing dataset

2.4 Output Layer

The last stage of this model is a fully connected layer with softmax as an activation function whose output is the probability distribution over labels. The direction is represented in the form of binary labels. The binary label [1, 0] represents that the stock price will increase, and the label [0, 1] represents that the stock price will decrease.

Fig. 5 Flowchart of processing of the News Headlines
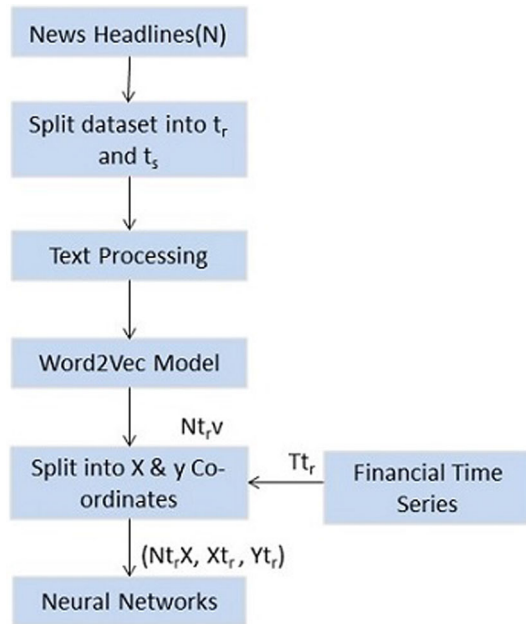
3 Methodology

The LSTM model framework is divided into qualitative and technical analyses, with news titles and financial time series as input. Each of these input data is divided into training and validation databases as shown in Table 1. These databases undergo word embedding and neural network algorithms to merge and give a binary output finally. Table 2 contains the list of acronyms that have been used in this section. All the explanation in this section has been done by taking the example of the training dataset (t_r). The detailed working of the model is shown from Algorithm 1-3. The frameworks are explained below.

3.1 Qualitative Analysis

The dataset of the news headlines obtained from the Reuters website consists of the publishing date along with the news titles. This data is further divided into training (Nt_r) and testing (Nt_s) datasets, as described in Table 1. The next step is to prepare the data for the Word2vec model. The process of data undergoes processing in which text is tokenized, and all the stop-word and punctuation are removed. The text further undergoes stemming. This tokenized text is converted into sentences and stored in a pickle file for further use. This processed text is used as an input of the Word2vec model. The description of this process is shown in Algorithm 2. The text vector ($Nt_r V$) obtained is used to find the sentence vector ($Nt_r X$). The average sentence vector and the x-train (Xt_r) and y-train (Yt_r) values are passed to the Neural Network model. Figure 5 shows a detailed scheme of this architecture.
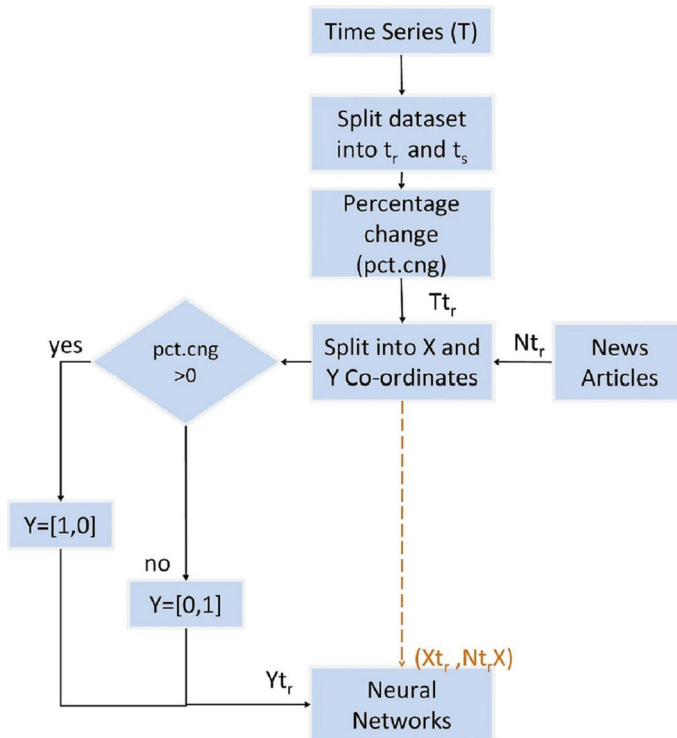
Fig. 6 Flowchart of processing of the Financial Time Series

3.2 Technical Analysis

The financial time series closing price (T) is used for the technical analysis. First, the data is split into training and testing datasets described in Table 1. Then this split data undergoes the change function, wherein the closing price of t+1 day is compared with the closing price of t day. The change in their prices is stored in Tt_r as a percentage change in value. In Split_into_XY() variable Xt_r stores the value of Tt_r averaged over the window (W). The input ($Nt_r V$) from the news headlines is used to calculate the sentence vector and then stored in $Nt_r X$. If the value stored in Tt_r of the day t+1 is greater than zero, then Y becomes equal to [1, 0], increase in price, else [0, 1], decrease in price. The value of Y after every iteration gets appended to Yt_r. Hence variable Yt_r contains the binary class output of the testing dataset. The working of this process is explained in detail in Algorithm 3. This model is explained in Fig. 6.

3.3 Model Architecture

The x-value of textual data $Nt_r X$ and the x-value of the time series dataset Xt_r are the inputs for the training of the model. These inputs undergo processing separately, and then the output is merged to give final binary values, as explained in Fig. 7.
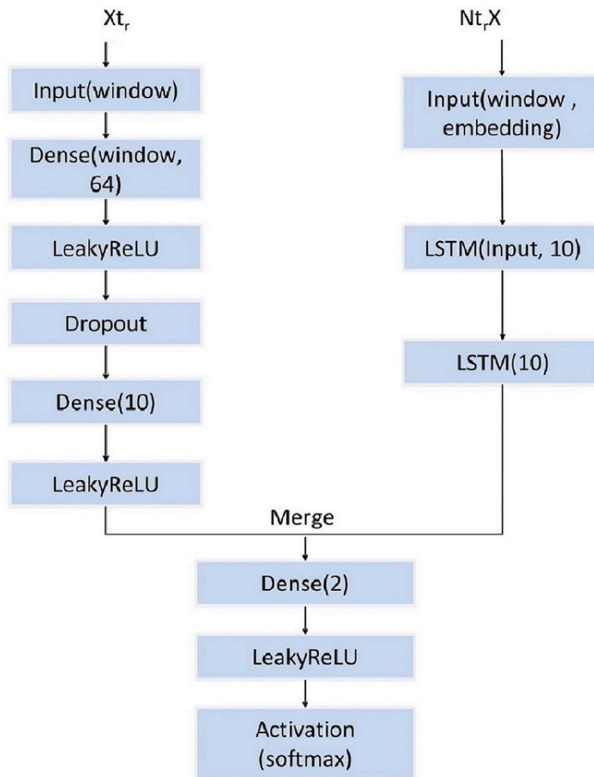
Fig. 7 A flowchart illustrating the working of the model architecture

The model should be aware of the input shape beforehand. For this reason, the first layer in Xt_r is the input of size 'window'. This data goes to the dense layer, a fully connected layer. It implies that all the inputs are connected to all the outputs. Thus, they form an input size of 'window' and an output size of 64. This output is passed through the activation functions. It is one of the most critical layers. It helps to produce non-linear decision boundaries through non-linear combinations of the inputs. In this model, LeakyReLU is used, which helps in deciding which neurons will be active. The next layer is dropout which is used for dropping out units in neural networks. Some of the neurons are chosen randomly and ignored during the training process. It again passes through the dense layer of size ten and then through the activation filter.

As discussed above, the first layer, $Nt_r X$, is an input of size (window, embedding). It then passes through the predefined LSTM function of Keras with the input of size (window, embedding) and output of size 10. In this step, the input and output dimensions are different. It is passed through the LSTM again and of size 10. The Working of the LSTM has already been explained in sect. 2.3. The two-layer LSTM model helps in extracting more abstract information. The output of this layer is merged with the final output of the Xt_r.

This combined data is then passed through the dense layer of size 2. The size of the output in this process is two because of the binary output. This output further undergoes two activation functions, LeakyReLU and softmax layer. The output of these functions is the final binary class output of the model.

Algorithm 1 Pseudo Code for the proposed model

Require: N, T
Ensure: Binary Class ([0,1] [1,0])
1: $Nt_r, Nt_s \leftarrow load\_text\_from\_dataset$ ▷ subroutine to load dataset
2: $Tt_r, Tt_s \leftarrow load\_closing Price\_from\_dataset$ ▷ subroutine to load dataset
3: $Nt_r V \leftarrow transform\_text\_into\_vector(Nt_r, E, W)$ ▷ call algorithm 2
4: $Nt_s V \leftarrow transform\_text\_into\_vector(Nt_s, E, W)$ ▷ call algorithm 2
5: $Xt_r, Nt_r X, Yt_r \leftarrow split\_into\_XY(Tt_r, Nt_r V, W, F)$ ▷ call algorithm 3
6: $Xt_s, Nt_s X, Yt_s \leftarrow split\_into\_XY(Tt_s, Nt_s V, W, F)$ ▷ call algorithm 3
7: $Fm \leftarrow neural\_network(W, E)$ ▷ call subroutine as defined in Fig 7

Algorithm 2 Transform_text_into_vector

Require: Nt_r, E, W
Ensure: $Nt_r V$
1: $model \leftarrow Word2Vec(Nt_r, E, W)$
2: $vector \leftarrow model$
3: **for** $x = 0 \rightarrow vector$ **do**
4:     $Nt_r V \leftarrow append(mean(x))$
5: **end for**

Algorithm 3 Split_into_XY

Require: $Tt_r, Nt_r V, W, F$
Ensure: $Xt_r, Nt_r X, Yt_r$
1: **for** $i = 0 \rightarrow vector$ **do**
2:     $X \leftarrow Tt_r[i : i + W]$
3:     $Y \leftarrow Tt_r[i + W + F]$
4:     **if** $Y > 0$ **then**
5:         $Y \leftarrow [1, 0]$
6:     **else**
7:         $Y \leftarrow [0, 1]$
8:     **end if**
9:     $Xt_r \leftarrow append(X)$
10:     $Nt_r X \leftarrow append(Nt_r V[i : i + W])$
11:     $Yt_r \leftarrow append(Y)$
12: **end for**

---

4 Results and Discussion

A comparison is made by implementing it on different datasets of companies from the different sectors to evaluate the effectiveness of this model using both news titles and

Actual Value	Predicted Value
[[0 1]	array([[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[1 0]	[0 1]
[0 1]	[1 0]
[1 0]	[1 0]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]

(a) Comparison of Predicted and Actual Value
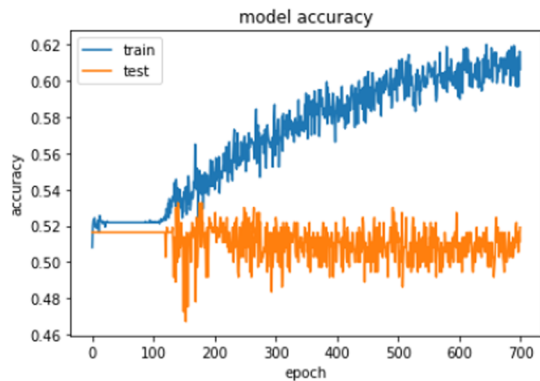
(b) Graph depicting the accuracy on training and validation dataset after each epoch.

Fig. 8 Directional stock price prediction for Apple

time series as input. Figure 8a shows the actual and predicted stock price direction of APPLE, a large-cap technology sector company, in terms of binary labels. Where [1,0] represents the stock price will increase. The label [0,1] represents that the stock price will decrease. The model was trained for 700 epochs, and the training and test accuracy can be observed in Fig. 8b.

Figure 9a shows the actual and predicted stock price direction of PepsiCo, a large-cap FMCG company, in terms of binary labels where [1,0] represents the stock price will increase. The label [0,1] represents that the stock price will decrease. The model was trained for 700 epochs, and the training and test accuracy can be observed in Fig. 9b.

Figure 10a shows the actual and predicted stock price direction of NRG, a large-cap integrated power company, in terms of binary labels. Where [1,0] represents the stock price will increase. The label [0,1] represents that the stock price will decrease. The model was trained for 700 epochs, and the training and test accuracy can be observed in Fig. 10b.

Figure 11a shows the actual and predicted stock price direction of APEI, a small-cap consumer services company, in terms of binary labels. Where [1,0] represents the stock price will increase. The label [0,1] represents that the stock price will decrease. The model was trained for 700 epochs, and the training and test accuracy can be observed in Fig. 11b.

Figure 12a shows the actual and predicted stock price direction of AT&T, a large-cap communication services company, in terms of binary labels. Where [1,0] represents the stock price will increase. The label [0,1] represents that the stock price will decrease. The model was trained for 700 epochs, and the training and test accuracy can be

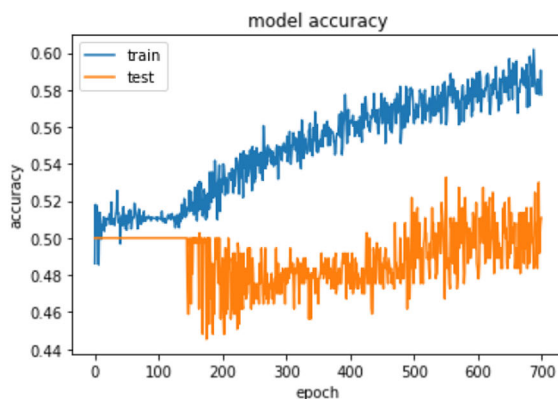| Actual Value	Predicted Value
[[0 1]	array([[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]

(a) Comparison of Predicted and Actual Value
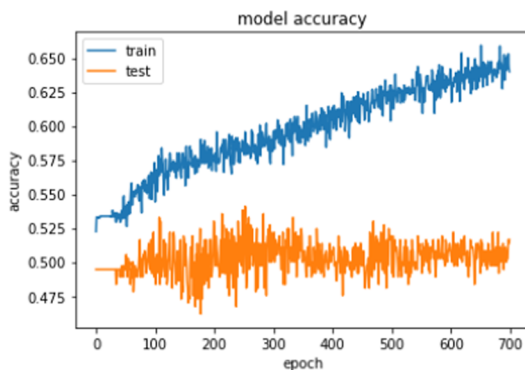
(b) Graph depicting the accuracy on training and validation dataset after each epoch.

Fig. 9 Directional stock price prediction for PepsiCo

Actual Value	Predicted Value
[[1 0]	array([[0 1]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[1 0]	[0 1]
[1 0]	[0 1]
[1 0]	[0 1]
[1 0]	[1 0]
[1 0]	[1 0]
[1 0]	[0 1]
[1 0]	[0 1]
[1 0]	[1 0]
[1 0]	[0 1]

(a) Comparison of Predicted and Actual Value

(b) Graph depicting the accuracy on training and validation dataset after each epoch.

Fig. 10 Directional stock price prediction for NRG

observed in fig. 12b. The efficiency of the directional forecasting on the training and validation dataset of these companies is summarized in Table 3. The result, when compared with the WB-NN (sum of each word in the news titles and the standard neural network) [13] concludes that Word2Vec-LSTM performed better in terms of accuracy. It could be explained by adding the LSTM layer to the model design.
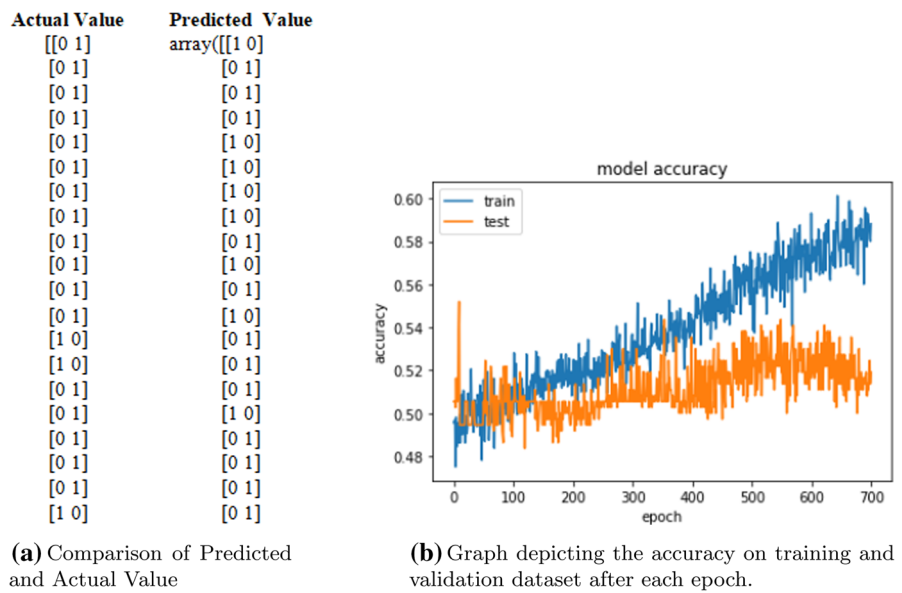
Actual Value	Predicted Value
[[0 1]	array([[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[1 0]	[0 1]
[1 0]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[1 0]	[0 1]

(a) Comparison of Predicted and Actual Value

(b) Graph depicting the accuracy on training and validation dataset after each epoch.

Fig. 11 Directional stock price prediction for APEI

Actual Value	Predicted Value
[[0 1]	array([[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[0 1]
[0 1]	[1 0]
[0 1]	[0 1]
[0 1]	[0 1]

(a) Comparison of Predicted and Actual Value

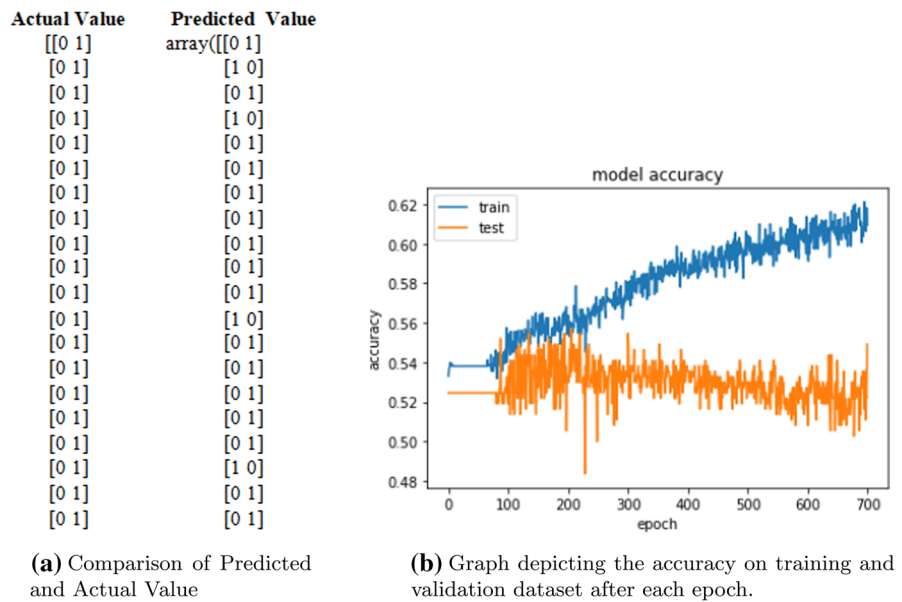**(b)** Graph depicting the accuracy on training and validation dataset after each epoch.

Fig. 12 Directional stock price prediction for AT &T

Table 3 Results of stock price directional forecasting

Dataset	Accuracy on training data (%)	Accuracy on validation data (%)
PepsiCo	62.1	53
Apple	65.4	51.6
APEI	59.8	51
NRG	60.1	51.9
AT&T	61.6	54.9

5 Conclusion and Future Scope

The paper presents a deep learning model that helps the investors comprehend the market's trading behaviour. The framework combines word embedding with recurrent neural network for predicting stock price directional movement. The model takes a combination of financial time series and news headlines as input. Compared to other relevant work, the use of hybrid input positively influences the output. Regarding the text representation, it is observed that both good news and bad news induce a change in the stock price. It helps in the macroeconomic analysis of the stock. Hence, the news titles from the day before are aligned in a unique instance and used for directional forecasting. This model outperforms previously proposed models that have used news of past weeks and months, with improved accuracy of 65.4%. The result supports the hypothesis that the information in the news headlines has a short temporal effect on the investors. The LSTM based architecture remembers context-specific temporal dependencies between news titles for a more extended period while performing forecasting as compared to other RNN models. The distributed representation of textual information and numerical data with the Word2Vec-LSTM model can capture the time-series influence of input data as compared to other models. The comparison of results between multiple companies in the different sectors shows the efficacy of the model.

For future work, Convolutional Neural Network (CNN), which displays superior ability to extract semantic information from the text compared to RNN, could be incorporated for better model performance. Also, reinforcement learning can be used to create improved trading strategies.

Declarations

References

1. Shi Y (2022) Advances in big data analytics: theory, algorithms and practices. Springer, Berlin
2. Tien JM (2017) Internet of things, real-time decision making, and artificial intelligence. Ann Data Sci 4(2):149–178. https://doi.org/10.1007/s40745-017-0112-5
3. Olson DL, Shi Y, Shi Y (2007) Introduction to business data mining. McGraw-Hill, New York
4. Pan Y, Xiao Z, Wang X, Yang D (2017) A multiple support vector machine approach to stock index forecasting with mixed frequency sampling. Knowledge-Based Syst 122:90–102. https://doi.org/10.1016/j.knosys.2017.01.033
5. Nguyen TH, Shirai K (2015) Topic modeling based sentiment analysis on social media for stock market prediction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1354–1364
6. Feldman R, Rosenfeld B, Bar-Haim R, Fresko M (2011) The stock sonar-sentiment analysis of stocks based on a hybrid approach. In: Twenty-Third IAAI conference. https://ojs.aaai.org/index.php/AAAI/article/view/18854
7. Kumar G, Jain S, Singh UP (2020) Stock market forecasting using computational intelligence: A survey. Archives of computational methods in engineering, 1–33. https://doi.org/10.1007/s11831-020-09413-5
8. Fama EF (1965) The behavior of stock-market prices. J Bus 38(1):34–105
9. Malkiel BG (2007) A random walk down Wall Street: the time-tested strategy for successful investing. WW Norton & Company
10. Khan AU, Gour B (2013) Stock market trends prediction using neural network based hybrid model. International J Comput Sci Eng Information Tech Res 3(1):11–18
11. Hari Y, Dewi LP (2018) Forecasting system approach for stock trading with relative strength index and moving average indicator. J Telecommun Electron Comput Eng (JTEC) 10(2–3):25–29
12. Huang W, Nakamori Y, Wang S-Y (2005) Forecasting stock market movement direction with support vector machine. Comput Oper Res 32(10):2513–2522. https://doi.org/10.1016/j.cor.2004.03.016
13. Ding X, Zhang Y, Liu T, Duan J (2015) Deep learning for event-driven stock prediction. In: Twenty-Fourth international joint conference on artificial intelligence
14. Chen W, Zhang Y, Yeo CK, Lau CT, Lee BS (2017) Stock market prediction using neural network through news on online social networks. In: 2017 International smart cities conference (ISC2), pp. 1–6. https://doi.org/10.1109/ISC2.2017.8090834. IEEE
15. Vijh M, Chandola D, Tikkiwal VA, Kumar A (2020) Stock closing price prediction using machine learning techniques. Procedia Comput Sci 167:599–606. https://doi.org/10.1016/j.procs.2020.03.326
16. Vargas MR, De Lima BS, Evsukoff AG (2017) Deep learning for stock market prediction from financial news articles. In: 2017 IEEE International conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA), pp. 60–65. https://doi.org/10.1109/CIVEMSA.2017.7995302. IEEE
17. Kim T, Kim HY (2019) Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. PloS one 14(2):0212320. https://doi.org/10.1371/journal.pone.0212320
18. Teixeira Zavadzki de Pauli S, Kleina M, Bonat WH (2020) Comparing artificial neural network architectures for brazilian stock market prediction. Ann Data Sci 7(4):613–628. https://doi.org/10.1007/s40745-020-00305-w
19. Tikkiwal VA, Vir Singh S, Gupta HO (2020) Day-ahead forecasting of solar irradiance using hybrid improved cuckoo search-lstm approach. In: 2020 2nd International conference on advances in com-

puting, communication control and networking (ICACCCN), pp. 84–88. https://doi.org/10.1109/ICACCCN51052.2020.9362839

20. Chandola D, Gupta H, Tikkiwal VA, Bohra MK (2020) Multi-step ahead forecasting of global solar radiation for arid zones using deep learning. Procedia Comput Sci 167:626–635. https://doi.org/10.1016/j.procs.2020.03.329

21. Ding X, Zhang Y, Liu T, Duan J (2014) Using structured events to predict stock price movement: An empirical investigation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1415–1425

22. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv:1301.3781

23. Ho K-Y, Wang WW (2016) Predicting stock price movements with news sentiment: An artificial neural network approach. In: Artificial neural network modelling, pp. 395–403. Springer, Berlin. https://doi.org/10.1007/978-3-319-28495-8_18

24. Kumar BS, Ravi V, Miglani R (2021) Predicting indian stock market using the psycho-linguistic features of financial news. Ann Data Sci 8(3):517–558. https://doi.org/10.1007/s40745-020-00272-2

25. Saini A, Sharma A (2019) Predicting the unpredictable: an application of machine learning algorithms in indian stock market. Annals of Data Science, 1–9. https://doi.org/10.1007/s40745-019-00230-7

26. "Yahoo Finance - Business Finance, Stock Market, Quotes, News." Yahoo! Finance, Yahoo! http://in.finance.yahoo.com/budget

27. "Reuters Business and Finance - Business News India, Latest Financial News, Finance Business Headlines, India.", Thomson Reuters. http://in.reuters.com/finance

28. Mikolov T, Chen K, Corrado GS, Dean JA (2015) Computing numeric representations of words in a high-dimensional space. Google Patents. US Patent 9,037,464

29. Al-Saqqa S, Awajan A (2019) The use of word2vec model in sentiment analysis: A survey. In: Proceedings of the 2019 international conference on artificial intelligence, robotics and control, pp. 39–43. https://doi.org/10.1145/3388218.3388229

30. Poženel M, Lavbič D (2019) Discovering language of the stocks. arXiv:1902.08684, https://doi.org/10.48550/arXiv.1902.08684

31. Kilimci ZH, Duvar R (2020) 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 8:188186–188198. https://doi.org/10.1109/ACCESS.2020.3029860

32. Muhammad PF, Kusumaningrum R, Wibowo A (2021) Sentiment analysis using word2vec and long short-term memory (lstm) for indonesian hotel reviews. Procedia Comput Sci 179:728–735. https://doi.org/10.1016/j.procs.2021.01.061

33. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735