Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

Stock market prediction based on deep hybrid RNN model and sentiment analysis

Ancy John & T. Latha

To cite this article: Ancy John & T. Latha (2023) Stock market prediction based on deep hybrid RNN model and sentiment analysis, Automatika, 64:4, 981-995, DOI: 10.1080/00051144.2023.2217602

To link to this article: https://doi.org/10.1080/00051144.2023.2217602

AUTOMATIKA
2023, VOL. 64, NO. 4, 981–995
https://doi.org/10.1080/00051144.2023.2217602

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Stock market prediction based on deep hybrid RNN model and sentiment analysis

Ancy John[a] and T. Latha[b]

[a]St. Xavier's Catholic College of Engineering, Nagercoil, India; [b]Department of ECE, St. Xavier's Catholic College of Engineering, Nagercoil, India

ABSTRACT
Stock market movements, stocks, and exchange rates are the primary subjects and active areas of research for analysts and researchers. The stock prices is being influenced by financial news, which has been demonstrated to be an important element in fluctuating stock prices. Furthermore, previous research mostly evaluated shallow characteristics and ignored functional relationships between words in a sentence. Many studies have attempted to analyse the sentiment of investors' reactions to corresponding news occurrences. In this paper, we proposed a unique methodology for predicting the stock prices trend by using both stock features and financial news. The proposed methodology is the hybrid Recurrent Neural-Network (HyRNN) architecture. This design includes Bidirectional Long Short-Term Memory (Bi-LSTM) on top of the Gated Recurrent Unit (GRU) and stacked Long Short-Term Memory (sLSTM). The performance of HyRNN for forecasting stock price can be considerably improved by mixing the sentiments of financial news with the features of stock as an input to the model. In comparison to earlier statistical models, the suggested model increases the analysing capability of GRU, LSTM, RNN, and proposed models independently. The findings of this study shows the deep learning (DL) approach has high potential for predicting stock price changes.

ARTICLE HISTORY
Received 17 March 2023
Accepted 19 May 2023

KEYWORDS
LSTM; neural network; sentiment analysis; stock market; intelligence stock market; sentiment detaining

1. Introduction

Forecasting the stock market is a crucial task in the global stock market. Typically, forecasting the stock market is involved accurately predicting both trend and price of a stock in order to increase trading profits. Because of the stock market's non-linear and volatile character, achieving a precise forecast of the stock trend has proven to be a challenging process. People who depend on the effective market hypothesis notion [1] believe that based on historical stock data, future stock prices can be predicted. Others who depend on random hypotheses believe that future prices of stock are unaffected by past stock data, henceforth no valid stock features can be detected in historical stock data to represent the stock features of future stock sequences [2].

The forecast of all stocks price is a difficult subject in finance, mathematics, and engineering. It has received a lot of interest from both academics and business people because of its financial gain. Most financial analysts and investors have long been fascinated by stock price forecasting. Nonetheless, since there are so many other factors that can influence stock prices, determining the ideal time to sell or buy has been a challenging task for investors.

A novel approach for predicting the sell or buy indication to investors is proposed in the stock market prediction analysis. Forecasting of the stock market is accomplished using the proposed method, which combines both stock-related data and news feeds. Investors buy or sell their products based on the combined outcome of Sensex points and sentence opinion. These analytical models relate linear correlations well but aren't viable in forecasting the stock market because of the non-linear nature of stock market transactions. Many researchers have developed more nonlinear models for forecasting stock market using artificial neural networks (ANNs), genetic algorithms, fuzzy–neural systems, evolutionary and/or particle swarm approaches since the emergence of artificial intelligence [3–9].

Researchers found that news related to finance is being influenced by gossip spread over social media platforms, has contributed to increased stock market volatility. Any system for forecasting the stock market must take the sentiment of such news into account. It is apparent that manually structuring such news emotion for vibrant forecasting of stock market is unfeasible. As a result, media sentiment analysis and natural language processing (NLP) algorithms are proposed

CONTACT Ancy John ✉ ancyjohn.rs@outlook.com St. Xavier's Catholic College of Engineering, Nagercoil, Tamilnadu 629003, India

to automatically align stock market indices and news sentiment to improve stock market prediction accuracy [10]. Deep learning (DL), has many ANN layers, which is shown some impressive outcomes in forecasting stock market, both with and without the use of news sentiment indicators [11–16]. The LSTM model or LSTM's hybridization appears to be the most prominent deep-learning model for forecasting the stock market [11,15]. The GRU model or its hybrid model [15–20] is another prominent deep-learning model for forecasting the stock market. In each new publication, both models appear to boast about their superior performance. However, since these studies used different designs, assumptions, and implementations – as well as performing with various stocks in differentiations – difficult to establish a quantitative comparison among the results of these two prominent models under similar circumstances. Furthermore, due to variances in study conditions, the results of several studies on whether integrating the sentiments of financial news in forecasting stock may result in improved performance than without financial news sentiments, even for the same model [21–24]. This research aims to do two things: first, conduct a normalized comparison of the RNN-LSTM and the GRU models' stock forecasting performances against similar conditions, and secondly, quantitatively assess the importance of integrating sentiments of financial news in forecasting the stock market versus just using stock features. To accomplish these objectives, we create a cooperative network model that treats RNN-LSTM and GRU models with similar inputs equally, which include relevant characteristics from previous news sentiment scores and stock data for the prediction of stock. The development of this collaborative deep-learning architecture was done utilizing existing tools or methods employed in research works published previously by other authors to confirm an objective quantification. This eliminates any potential bias induced by any unverified algorithm we propose. Given the high diversification in evaluating the level of stock market news sentiment demonstrated by various sources in advanced nations, it is convincible to use data on the stock market from a country where news sources are trustworthy and available for a long time to satisfy the goal.

The combination of several deep learning methods can efficiently raise the potential of hybrid models. A novel hybrid DL model is proposed in this study to estimate the trends of varying stock prices. The following is a list of the significant contributions of this work:

- To effectively forecast the varying prices of stock, we introduce HyRNN, a unique hybrid model that incorporates a GRU, LSTM, Bi-LSTM, along with the optimum settings of hyper-parameter. The capacity to effectively extract features and thereby learn the forecast pattern of stock using both previous and prospective data, together with combining GRU units and stacked LSTM, offers an efficient method of dealing with complicated data patterns.
- Sentiment lexicons based on VADER are used to classify news articles either as negative, positive, or neutral categories.
- For the prediction of stock, a new hybrid model has been proposed which integrates sentiments score of news and stock data. Based on this combination utilization, the performance of prediction improves.
- The proposed forecast deep learning model surpassed other existing deep learning-based techniques and it was efficient in predicting stock market automatically, according to the findings.

The proposed forecast deep learning model surpassed other existing deep learning-based techniques and it was efficient in predicting the stock market automatically, according to the findings.

2. Related work

The research on stock prediction techniques is briefly summarized in this section. It highlights stock prediction strategies that solely consider numerical financial stock data. The extraction of features from textual data is then discussed. It includes a summary of algorithms exclusively for the prediction that use both textual and numerical in combination as inputs to make predictions.

Stock market patterns are exceedingly volatile, which makes forecasting difficult. Researchers are attracted to examine sophisticated strategies for better prediction because of the volatile nature of the market. Stock market prediction trend with significantly high accuracy produces a lot of revenue. Technical analysis and fundamentals are the two most used methods for predicting stock trends. The fundamental analysis covers not just the statistics of stock but also political events, industry performance, and conditions of economic while the technical analysis examines historical data and stock price volumes, whereas [25,26]. Since it considers the entire market, fundamental analysis is more realistic. This survey emphasizes fundamental analytical research, in which textual material is analysed alongside historical stock price data to forecast stock trends.

For the prediction of stocks, input features are fed into a machine-learning (ML) system. For stock trend prediction, two types of ML approaches are been utilized in the literature. Deep-learning (DL) is a substitute for the ML approach in which only a few composite layers, such as Support Vector Machine (SVM) and Artificial Neural Network (ANN), are used. Whilst DL techniques such as Convolutional Neural Networks (CNN) have many hidden layers.

Many DL methods are used in the prediction of stock in various stock markets throughout the world in recent years. Chen et al. [11] employed the LSTM model to forecast the stock market of China. In this architecture, a single layer of input follows numerous LSTM layers and a dense-layer, and a single output-layer with many neurons. Six alternative strategies were used to predict stock prices using many stock variables such as low price, open price, closing price, and open price. The normalized features and SSE indices were found to improve forecasting accuracy in this investigation. In this analysis, the sentiment of financial news wasn't taken into account. Samarawickrama and Fernando [14] used a simple recurrent neural network (SRNN), multilayer perceptron (MLP), LSTM, and architectures of GRU to forecast stock prices on the Colombo Stock Exchange (CSE). Despite taking financial news sentiments into account, the research employed the closing, low and high, price of the previous two days as input variables. What's more fascinating about the study's findings is that the MLP model outperformed the others when it came to forecasting the closing price of the next day. This result could be explained by the fact that these models only analysed stock features from the previous two days, limiting the DL ability to monitor more possible indications.

For stock price prediction, Althelaya et al. [12] analyzed the stacked LSTM (SLSTM) and bidirectional LSTM (BLSTM) methods. In the learning process, BLSTM utilized all input by using succeeding and preceding input sequences. To conduct deep learning in SLSTM, numerous layers of LSTM were stacked. Consequently, the BLSTM method outperformed both short-term and long-term stock forecasts, while the SLSTM model outperformed for only short-term stock price predictions. Yet, no financial sentiment was taken into account in this study. Li et al. [15] used an LSTM model with four layers and 30 nodes to forecast the CSI300 index values using stock indicators and investor sentiments. A naive Bayes classifier was used to analyse investor sentiments. This model surpassed support vector machine (SVM) approaches in terms of prediction accuracy, according to this study. However, when compared to the effectiveness of the LSTM method, it ultimately failed when solely using stock indicators and ignoring the sentiments of investors. Jiawei and Murata [13] have used the LSTM method with a pre-processing technique to lower the features of stock dimensions and a sentiment analyzer to generate prediction trends in financial stock to find the elements that influence the prediction of stock market trends. The findings revealed that "market sentiment" was a significant influencing factor in the stock market, and that it may aid in enhancing the accuracy of prediction. However, if only the features of stock were utilized as input, it stopped looking into the level of such influence.

Li. et al. [25] introduced an LSTM method for stock trend prediction that included stock attributes with the polarity of sentiment from financial news. When compared to the baseline model, which consisted of multiple-kernel learning (MKL) and SVM, our LSTM model incorporating basic news sentiment performed better. A comparison of the LSTM model's performance when only stock indicators were utilized as input are being more beneficial in determining the impact of news-sentiments on stock predictions. GRU is a deep-learning architecture developed by Cho et al. [27] to solve the difficulties of explosion and vanishing of gradients in conventional RNNs when learning dependencies in long-term. As a result, GRU-related methods, including LSTM, are lately been employed for prediction in financial investment, such as the prediction of Bitcoin price [27]. The results revealed that the two GRU models outperformed other models in terms of prediction accuracy. It should be emphasized that the goal of this study was to forecast trading signals for stock indices utilizing the data of stock without taking financial news sentiments into account. Using stock information from Yahoo Finance, Rahman et al. [16] used the GRU method to forecast the prices of the future stock market. The authors stated that the proposed strategy accurately forecasted future prices. The sentiment of financial news was obviously ignored, and no comparative study for the performance of other deep learning models was done in this study. Shakya and Saud [17] used stock market variables taken from the Indian Stock Exchangeto examine the efficacy of two DL models: RNN-LSTM, and GRU in the prediction of stock price. Among the models examined, GRU proved to be the most efficient in the prediction of stock price. In this analysis of stocks, the sentiment of financial news was not taken into account. Dang et al. [18] introduced two-stream GRU methods that forecasted S&P 500 index patterns and prices using sentiments of financial news and stock attributes as inputs. The two-different GRU model outperforms other deep learning models, including LSTM and the original GRU, according to the results. Due to the intricacy of the extended GRU, scientists highlight that training the two-different GRU model is consuming lot of time and demands a lot of computer resources. The literature review reflects the goals of this study as previously stated sessions, namely, to conduct a normalized comparison of the RNN-LSTM and the GRU methods for forecasting the trends in the stock market under similar circumstances with similar attributes and to quantitatively assess the importance of joining the sentiments of financial news in forecasting stock market along with the stock data set.

3. Proposed prediction model

Figure 1 shows an outline of how the proposed idea works and the paper's approach. Stock features and financial news are both collected and concatenated before the time sequence generator, which now uses our

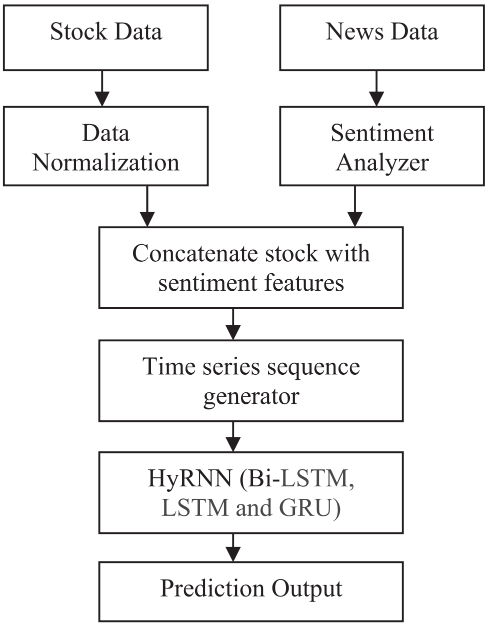

Stock Data
News Data
Data Normalization
Sentiment Analyzer
Concatenate stock with sentiment features
Time series sequence generator
HyRNN (Bi-LSTM, LSTM and GRU)
Prediction Output

Figure 1. Process flow diagram of the proposed model.

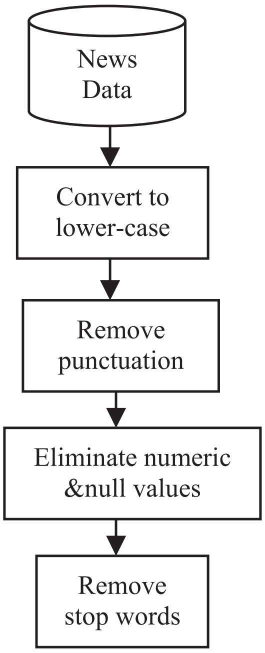


Figure 2. Pre-processing stages.

suggested model HyRNN (Bi-LSTM, LSTM, and GRU) to produce the final prediction output.

The proposed approach is divided into four stages, as shown in Figure 2. (A) The retrieved content is pre-processed in the pre-processing stage to eliminate unnecessary data such as punctuation and stop words. (B) The following stage is to label news articles depending on stock prices. (C) Using a lexicon-based method, annotate the sentiments of news into negative, positive, and neutral, categories in the third phase. Lexicon tokenizes the news items first, then assigns the scores of sentiments to each individual word. After that, based on the attribute settings, the sentiment score of each word in an article is summed, and the article is classified into either of three categories: neutral, positive, or negative. (D) Finally, combined GRU, LSTM, Bi-LSTM and RNN networks are applied to the dataset of financial news, and several scenarios are performed.

3.1. Data collection

Yahoo Finance was used to get the prices and volume data. Headlines of financial news are retrieved from the NASDAQ website for the textual data. For every stock, we gather the information that is specific to each stock. Based on the availability of financial news in the website's archive, each stock will contain two to three years of financial data. All data instances have a date associated with them. Every day, there is only one instance of price and volume, yet there are many news headlines [28].

This information includes the company's significant key events, as well as daily stock prices for the same period of time. Open, Low, High, Close, Modified Close and Volume are the six parameters that make up daily stock prices. We used Modified Close price as the everyday stock price throughout the project for integrity.

3.2. Pre-processing

3.2.1. News articles

One of the most major roles in transforming unstructured data into a structural format is pre-processing [29]. The role of this is to significantly improve the input data quality hence the proposed models can recognize better patterns in stokes and hence extract distinguishing features. The news articles have both lower-case and upper-case words, punctuation, and stop words, and these significantly influence the capacity of the learning model.

Several processes are included in the pre-processing for this study, including removal of punctuation, conversion of text case, elimination of numeric and null values, and removal of stop words. Figure 1 illustrates the pre-processing procedures, with each step explained briefly.

(1) ***Converting to Lowercase:*** The algorithm will consider the occurrences of "effect" and "Effect" as two (case sensitive) distinct words because probabilistic ML algorithms are case sensitive. As a result, the text of drug evaluations is transformed to lower-case as the first phase of pre-processing.

(2) ***Removing Punctuation*:** Punctuations such as $ % #)! # & (.,"" are eliminated from the evaluations in the second phase. Punctuation has been eliminated from the input data since it is irrelevant to the subject. Punctuation also makes it harder for an algorithm to tell the difference among other symbols and punctuation.

(3) ***Eliminating Numeric and Null Values:*** In this phase, both numeric and null values are eliminated from the stock reviews. Since the digits aren't important while dealing with reviews or textual data, the reviews are pre-processed to eliminate them. The same is true for null values, which do not improve the model's performance. Furthermore, null values, like numeric values, have no effect on the sentiment of the text under consideration.

(4) ***Removing Stop words:*** Stop-words are deleted from reviews after numeric and null values are eliminated. It is one among the main role in pre-processing since it eliminates the irrelevant data from text analysis. Stop-words are words that the model can't predict since they have no meaning.

3.2.2. Daily stock prices

Stock prices retrieved daily from Yahoo-Finance were aligned on the CSV-format, every row has six parameters open (opening-price), high (the highest price), include-date (trading-date), close (closing-price), low (lowest-price), adj. close (adjusted-closing), and volume. Unnecessary information was eliminated: "adj. closes" is no longer required, so it was eliminated to conserve system resources.

(1) Labelling of Document

The purpose of labelling is to categorize articles as positive, causing the price to rise, and negative, causing the price to fall. We use two approaches (positive and negative) to indicate the trend of stock prices in this analysis.

A close-to-close return and open-to-close return are called (overnight-return) and (daytime-return) respectively are two ways of identifying news articles using past stock values. The label of the article was determined using the open-to-close price return method. We selected open-to-close over close-to-close return since, as stated [30], the records of the open-to-close are more identical to the entire return of the similar stock, implying that the open-to-close return contributed more to the total return.

$$Article\_label = \{Positive\ if K_i\ \geq\ 0\ Negative\ if K_i\ <\ 0$$

If K_i is $>=$ to 0 then the article is considered as +tive on that day and hence the price of the stock rise while if K_i is <0 then the article label is −tive hence the price of the stock will go down.

The closing and opening stock prices are retrieved from the database for each article, and the Open-to-Close return is calculated using the opening and opening and closing values. The value of the return is then utilized to categorize the news article as either good or bad.

Table 1. Sentiment features.

Compound	Negative	Neutral	Positive
−0.9982	0.235	0.724	0.041
−0.9858	0.191	0.721	0.089
−0.9715	0.128	0.816	0.056
−0.9809	0.146	0.788	0.066
−0.9882	0.189	0.717	0.094
−0.9892	0.177	0.753	0.07
−0.9938	0.224	0.719	0.056
−0.9974	0.247	0.716	0.037
−0.9913	0.182	0.758	0.06
−0.9966	0.218	0.717	0.065
−0.976	0.145	0.778	0.078
−0.9979	0.274	0.701	0.024
−0.9913	0.222	0.712	0.066
−0.9921	0.181	0.771	0.048
−0.9153	0.15	0.737	0.113
−0.9575	0.134	0.789	0.076
−0.9971	0.241	0.706	0.052
−0.9866	0.162	0.779	0.059
−0.9893	0.187	0.724	0.089

3.3. Lexicon-based method

The lexicon technique contains sentiment-related lexicons, the objective of this technique is to classify words in an article as neutral, positive, or negative [31]. By analyzing the polarity of sentiment-bearing words in an article, the polarity of the text is determined, which is the fundamental principle of classification for lexicon-based sentiment analysis.

A lexicon with word polarity values is known as a sentiment lexicon. In the lexicon, words or groups of phrases are identified with their matching sentiment polarity score. (word, score of sentiment polarity) is how the sentiment lexical database's tuples are represented. In a lexicon-driven approach, every lexicon has a polarity value that ranges from negative, neutral, or positive. This range is being used to categorize the text that is being reviewed. The polarity range can be stated as follows:

$$K_p\ =\ [p_s,\ nt_s,\ n_s] \tag{1}$$

where, K_p = range of polarity-score, p_s = +tive score, n_s = −tive score, nt_s = neutral-score.

A value of threshold within the sentiment lexicon's polarity range is set for the classification of sentiment scores as negative, positive, or neutral. One of the lexicons used in this research is VADER (Table 1).

VADER is considered as a gold-standard for sentiment-lexicon in the English language that was introduced in 2014. It's a valence-based paradigm with a human-centred perspective. To enhance sentiment analysis, the combination of empirical validation by human raters with qualitative approaches [32] is employed. The authors in the reference [32] employed VADER in their analysis and concluded that its efficiency is comparable to human raters. VADER is determined by the valence score of the word in the dictionary. The valence score is in the range of $K_p = [+4.0, -4.0]$, where, −4 indicates negative sentiment while +4.0 indicates positive sentiment. The VADER library offers a big corpus with around 7,517 words bearing sentiment to analyse the sentiment polarity of a text. It includes slang words, as well as acronyms and their valence scores [30]. When utilizing the VADER sentiment lexicon to analyse a text, the result tuple looks like this: sentiment (compound, neu, neg, pos,). Negative, neutral, and positive rates are represented by neg, neu, and pos, correspondingly, whereas compound is a normalized aggregate score of all ratings. $K_p = [+1.0,\ 1.0]$, where $+1.0$ represents a most $+$tive sentiment and 1.0 represents the most $-$tive emotion.

The data is categorized based on the percent of the sentence that has a neutral, negative, or positive sentiment, and then a compound score is generated for the sentence based on its normalized aggregate. Every sentiment score of news headlines is tallied and condensed so that it is used to assess the market sentiment of the corresponding stock.

3.4. Deep learning process

The proposed technique is ready for modelling after determining sentiment ratings. Then, six distinct DL models, RNN + LSTM + GRU, RNN + GRU, RNN + LSTM, GRU, LSTM, RNN are modelled using sentiment scores and stock prices as input feed to the DL approach. The deep learning model with the highest classification results is selected to forecast the trends of the Stock Exchange based on all these steps.

3.4.1. RNN

RNN is a form of ANN architecture that is specifically tailored to process sequential data (time-series). RNNs have a feedback loop and can analyze the variable-length of input sequences by employing internal-state. In such a case, $n{-}1$ (the output of step) is given into the model to influence the result of step n, and so on for every succeeding step. Figure 2 shows an RNN model with a hidden-layer and a spread-out shape, where X_t is the input data, H_t is the hidden-state (HS), and Y_t represents the outcome of the RNN model [33].

Figure 3 shows how the RNN's hidden state H_t is affected by precedingHS values as well as theoutput of current–time step. The following formula can be used to get $\underline{H_t}$:

$$H_t = \propto_H(UX_t + UH_t + \ WH_t + b_H) \tag{2}$$

Based on the *Htatt*, the output-state *Yt* for the input *Xt* can be determined as follows:

$$Y_t = \propto_Y(VH_t + b_Y) \tag{3}$$

The non-linear activation function, such as the sigmoid function, tanh, or, ReLU is represented by αH and αY. The parameter matrices and vectors are U, V, W, and b, correspondingly. Unlike most deep neural networks, RNN uses the same set of attributes, vectors, and matrices throughout the whole process. It considerably reduces the number of parameters that RNN use to learn.

3.4.2. Long short-term memory

Due to the backpropagation approach over time, short-term memory and vanishing gradient issues are the most common in RNN. This problem can be solved by employing a special type of RNN termed LSTM. It also capable of learning long-term dependence. Figure 4 depicts an LSTM cell in its entirety. The forget gate, output gate, and input gate are the three gates of an LSTM cell. Each gate has a sigmoid-activation function that functions as a filter, determining which data to preserve and which to discard. Apart from the gates, the LSTM contains the following elements [33]:

- At time step t, Xt is the input data.
- Ht-1 (HS) from the earlier time-step or at time step t-1. In LSTM, it functions as short-term memory.
- The cell state at t-1 (time step) is Ct-1. In the LSTM, it functions as long-term memory. At LSTM, it is revised at two sites.
- σ = activation-function.
- The tanh produces output in the range of 1 to -1.
- The ft (sigmoid function outcome) determines whether it should forget or remember the preceding step memory *Ct*-1.
- The input sigmoid gate's output determines what new input data should be added to the cell state.
- The c $\sim$ t (cell state) as the outcome of applying tanh to the non-linear transformation of the external input data *Xt*.
- The sigmoid-activation of the vector formed by the point-wise addition of the previous HS Ht-1 and external input *Xt* yields ot. It regulates how much of the new cell state data is sent to the hidden state Ht and output Ht.
- Next cell-state = Ct.
- Ht as LSTM cell's output and
- Next HS is indicated as Ht.

A LSTM cell are updated in the internal-gate with each new external input data X_t. [34]:

$$f_t = \sigma(W_f[H_{t-1}, X_t] + b_f \tag{4}$$

$$i_t = \sigma(W_i[H_{t-1}, X_t] + b_i \tag{5}$$

$$o_t = \sigma(W_o[H_{t-1}, X_t] + b_o \tag{6}$$

$$\widetilde{c_t} = tanh(W_c[H_{t-1}, X_t] + b_c \tag{7}$$

Eventually, using the findings of cell state and internal-gates of candidate, LSTM modifies the output H_t and next cell state C_t as follows:

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes c_t \tag{8}$$

$$H_t = o_t \otimes tanh(C_t) \tag{9}$$

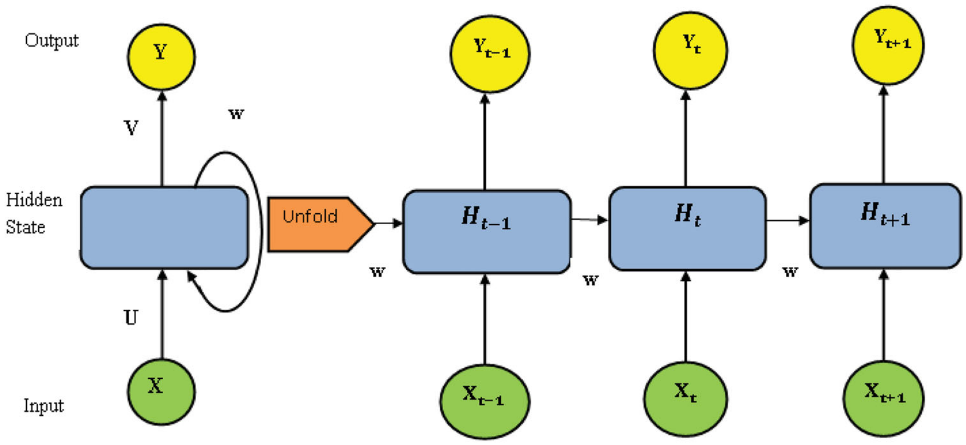


Figure 3. Unfold architecture of RNN [34].

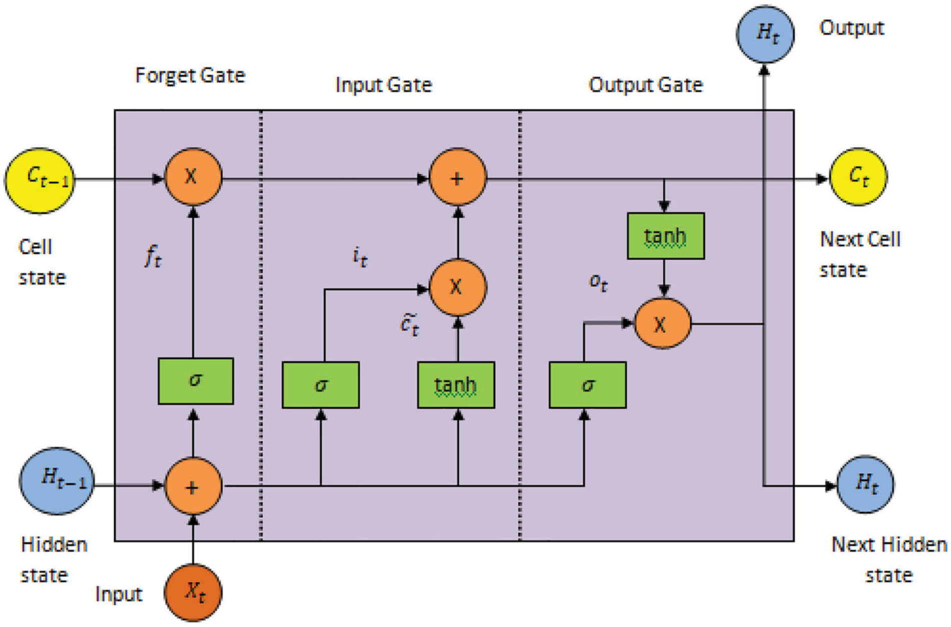


Figure 4. Architecture of an LSTM cell [35].

Three LSTM units were utilized in our proposed framework.64 memory units were chosen for two (left side of the GRU) and 50 for the last LSTM units (right side of the GRU) after considerable tuning.

3.4.3. Gated recurrent unit

GRU is a more refined form of RNN that includes a gating mechanism. It can avoid the problem of vanishing gradient that occurs with standard RNNs. It's similar to LSTM, however, it has fewer training attributes. Hence, it outperforms LSTM in terms of speed and memory usage. It reveals all of the hidden states with no control. GRU is a better form of RNN with a short series of input-data. The GRU unit's architecture is depicted in Figure 5.

In comparison to LSTM, GRU features two gates: a reset gate and an update gate. The forget-gate and the LSTM input-gate are combined in the update-gate. It aids the model in analyzing how much data from previous time steps may be carried forward. The reset-gate, on the other hand, determines how much previous data can be deleted. In addition, GRU eliminates the cell-state from the LSTM and transfers data across the concealed state. The GRU architectural unit's components are as follows:

- At *t* (time step), Xt is the external-input data. H_{t-1} is the HS at preceding time-step *t−1*.
- The reset-gate vector (rt), determines the amount of previous information to forget after point-wise multiplication with Ht-1.
- The output (zt) of the update gate sigmoid function, which decides how much information from the previous hidden-layer will be passed on to the next hidden level.
- h^t is an activation vector that employs the reset-gate vector rt to keep track of previous relevant data.
- At time step t, Yt is the output for the input data *Xt*.
- The next HS value is Ht, which is identical to *Yt*.

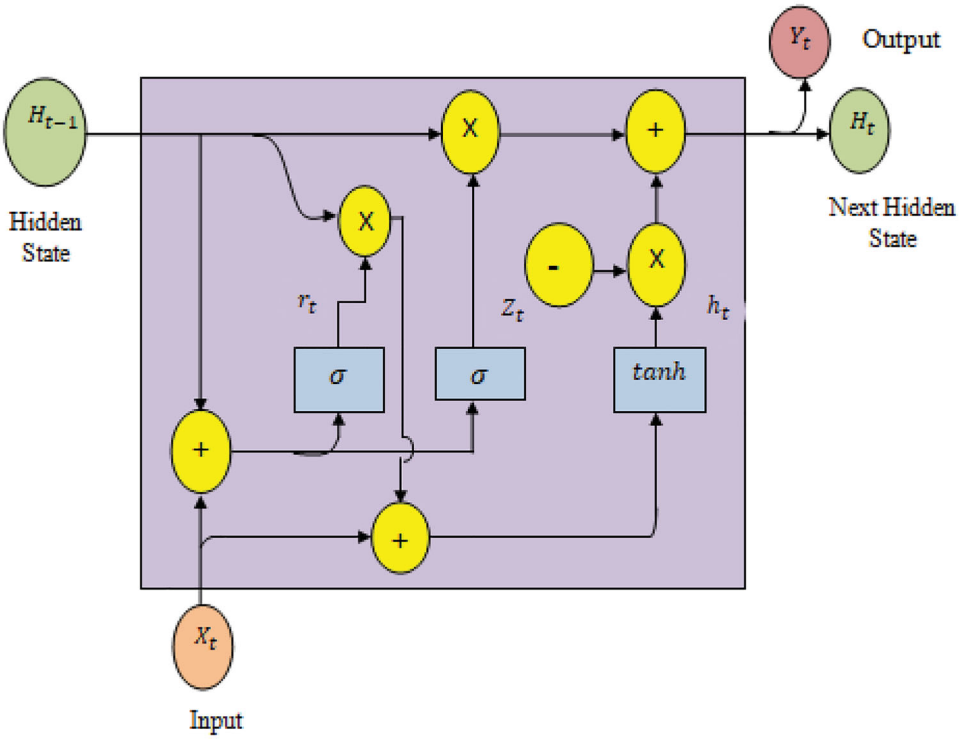


Figure 5. GRU cell architecture [29].

For every new input-data at time t, the subsequent updates are made to candidate activation vector and each internal-gate in a GRU cell [35]:

$$z_t = \sigma(W_z X_t + U_z H_{t-1} + b_z) \tag{10}$$

$$r_t = \sigma(W_r X_r + U_r H_{r-1} + b_r) \tag{11}$$

$$\hat{h}_t = tanh(W_h X_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{12}$$

Update of the update-gate vector yields the final output of the GRU cell unit as well as the following hidden-layer. This is the updated final output.

$$H_t = z_t \otimes \hat{h}_t + (1 - z_t \otimes H_{t-1}) \tag{13}$$

A GRU layer is integrated between LSTM layers in our proposed method, as shown in Figure 1. After adequate tuning, the output neuron layer of GRU unit is selected. The number of units is set at 64.

3.4.4. Bidirectional long short-term memory

Bi-LSTM is a distinctive RNN variation that combines two LSTM layers. In such a case, one LSTM unit analyzes inputs toward forwarding direction while the other unit analyzes inputs backward. It's an evolution of conventional LSTM, which keeps records of both previous and future data. Hence, the amount of information available to the network has increased significantly. The schematic diagram of the Bi-LSTM network is shown in Figure 6.

At t (time step), the forward-layer LSTM in Bi-LSTM consider $\vec{X}t$ as the input-data sequence and the previous HS value $\vec{h}_{t-1}$. The current HS value $\vec{h}_t$ is then calculated. The HS is updated using the internal equation as below:

$$\vec{f}_t = \sigma(\vec{W}_f[\vec{h}_{t-1}, \vec{X}_t] + \vec{b}_f) \tag{14}$$

$$\vec{i}_t = \sigma(\vec{W}_i[\vec{h}_{t-1}, \vec{X}_t] + \vec{b}_i) \tag{15}$$

$$\vec{o}_t = \sigma(\vec{W}_o[\vec{h}_{t-1}, \vec{X}_t] + \vec{b}_o) \tag{16}$$

$$\overrightarrow{\tilde{c}}_t = \sigma(\vec{W}_c[\vec{h}_{t-1}, \vec{X}_t] + \vec{b}_c) \tag{17}$$

$$\vec{C}_t = \vec{f}_t \otimes \vec{C}_{t-1} \oplus \overrightarrow{i}_t \otimes \overrightarrow{\tilde{c}}_t \tag{18}$$

$$\overrightarrow{h}_t = \vec{o}_t \otimes tanh(\vec{C}_t) \tag{19}$$

The internal operation of the forward LSTM unit in its current-state is $\vec{\square}LSTM$. In Bi-LSTM, the backward-layer modifies its value of HS as $h_{\leftarrow t}$ by considering the future HS value $h_{\leftarrow t+1}$ and the current input data sequence Xt. The following is a list of all internal modifications that occur within a backward-layer:

In a summary, the equation for the backward LSTM layer's hidden state output is as follows:

$$f_t^{\leftarrow} = \sigma\left(W_f^{\leftarrow}\left[h_{t+1}^{\leftarrow}, X_t^{\leftarrow}\right] + b_f^{\leftarrow}\right) \tag{20}$$

$$i_t^{\leftarrow} = \sigma\left(W_i^{\leftarrow}\left[h_{t+1}^{\leftarrow}, X_t^{\leftarrow}\right] + b_i^{\leftarrow}\right) \tag{21}$$

$$o_t^{\leftarrow} = \sigma\left(W_o^{\leftarrow}\left[\overrightarrow{h}^{\leftarrow}_{t+1}, X_t\right] + b_o^{\leftarrow}\right) \tag{22}$$

$$\tilde{c}_t^{\leftarrow} = \sigma\left(W_c^{\leftarrow}\left[h_{t+1}^{\leftarrow}, X_t^{\leftarrow}\right] + b_c^{\leftarrow}\right) \tag{23}$$

$$C_t^{\leftarrow} = f_t^{\leftarrow} \otimes C_{t+1}^{\leftarrow} \oplus i_t^{\leftarrow} \otimes \tilde{c}_t^{\leftarrow} \tag{24}$$

$$h_t^{\leftarrow} = o_t^{\leftarrow} \otimes tanh\left(C_t^{\leftarrow}\right) \tag{25}$$

Here, $\square^{\leftarrow}$ reflects the backward LSTM layer's entire internal functioning [33]. The hidden state's final

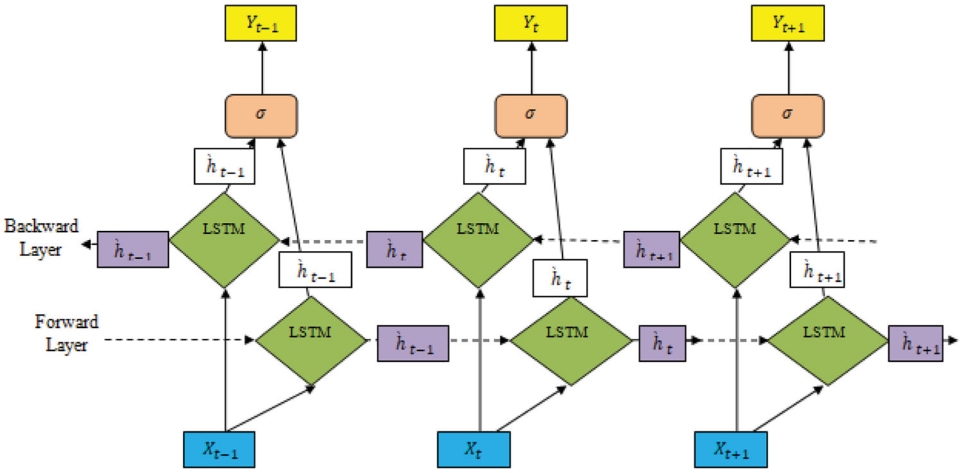


Figure 6. Architecture of bi-LSTM [34].

output incorporates the hidden state of a Bi-LSTM network's backward and forward layers, as well as an activation function. Hence the influence of current and succeeding data may be tracked back to a Bi-LSTM network's output.

4. Proposed hybrid RNN model

The output of the input-layer is then sent into a 128-hidden-memory-unit Bi-LSTM layer that combines backward and forward propagation. The Bi-LSTM output is then routed via two LSTM and a GRU layer. There are 64 hidden-memory units in each layer. The output is then be fed into a 50-hidden memory unit LSTM layer. Eventually, the output is transmitted through a single-layer fully-connected neural network. Bi-LSTM updates the HS in two directions using both backward and forward information. In comparison to Bi-LSTM, both GRU and LSTM are two distinct forms of RNN with fewer parameters.

LSTM outperforms other deep learning networks in larger datasets, while GRU outperforms other deep learning networks in minimum datasets; An optimum outcome is obtained by using a balanced combo of both RNN and finely tuned hyper-parameters (Figure 7).

4.1. Model assessment metrices

Various metrics are used to evaluate the daily stock price and trend projection. The prediction error is measured using the root mean square error (RMSE) and MAE, while the quality of matches between the actual and expected values is measured using the coefficient of determination (R2). With directional accuracy (DA), the accuracy of a prediction trend is evaluated. It uses formula to compare the predicted stock price to the actual price to determine the accuracy of stock trend predictions. The following is a list of all the measures.

$$RMSE = \sqrt{\left(\frac{1}{N}\sum_{d=1}^{N}(a_d - p_d)\right)} \tag{26}$$

$$MAE = \frac{1}{2}\sum_{d=1}^{N}|a_d - p_d| \tag{27}$$

$$DA = \frac{1}{N}\sum_{d=1}^{N} D_d, D_d = \{1, (a_{d+1} - a_d)$$

$$\times\ (p_{d+1} - a_d)0,\ otherwise \tag{28}$$

$$R^2 = 1 - \frac{\sum_{d=1}^{N}(a_d - p_d)^2}{\sum_{d=1}^{N}(a_d - \underline{a})^2} \tag{29}$$

where $\underline{a} = \frac{1}{N}\sum_{d=1}^{N} a_d; p_d$ is the predicted price and a_d is the actual price of stock at day d.

5. Experimental result

The analysis of results is classified into four subsections. They outline the training dataset, briefly describe the assessment criteria, and show the comprehensive experimental results of the proposed DL model, hence the performance of the proposed deep learning model is compared to that of other models. This also establishes the comparison of all the models' performance with only stock features, only sentimental features, and combined stock and sentimental features.

5.1. Dataset description

Two categories of data were used to predict stock price. The first category is the data set of stock price; the second category is the data set of news article. The stock price and new article data from March 2013 to November 2019 can be collected. The data's of Dow Jones Industrial Average is obtained from Yahoo Finance.

5.2. Hybrid-RNN description

Table 2 shows a summary of the proposed hybrid model. "WS" represents Window Size. In this study, we use a grid search strategy to take window sizes of 30

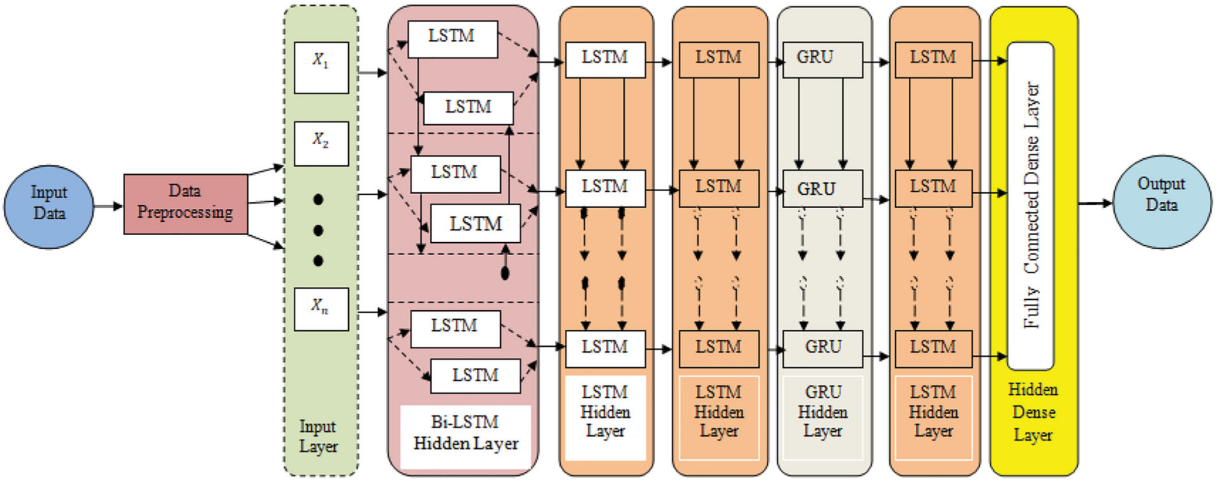


Figure 7. The proposed hybrid models.

Table 2. The proposed model is summarized below.

Layers	Output shape	Parameters
Bi-LSTM	(None,WS,128)	65668
LSTM	(None,WS,64)	49028
LSTM	(None,WS,64)	32824
GRU	(None,WS,64)	24580
LSTM	(None,50)	22620
Dense	(None,WS,1)	47
Total trainable parameters		194720

Table 3. Proposed hybrid model and its hyper-parameters.

Hyper-parameters	Values
Size of window	[30,60]
Size of batch	64
Hidden B-LSTM Layer neurons	128
1st layer neurons-hidden	64
2nd LSTM layers neurons-hidden	64
GRU Layer neurons	64
last LSTM layer neurons-hiddenrate of learning	46
Rate of learning	0.0001
Iteration	100
1st MEDR, β_1	0.9
2ND MEDR, β_2	0.999
Numerical stability ε	10^7

and 60 respectively. The fully-connected deep learning neural network is represented by the "Dense" layer. The proposed hybrid model has a total of 196,875 training parameters. The hyper-parameters employed in the proposed hybrid model are mentioned in Table 3. Each RNN unit in the proposed model has its batch size, kernel size, size convolution filter, and output neuron layer calibrated using the error method and a rigorous trial. The learning rate, Moment Exponential Decay Rate (MEDR) β_1, β_2, and ϵ of the Adam optimizer were adjusted based on [36]. Each model is iterated 100 times.

5.3. Experimental setup

To keep the comparisons consistent, both RNN-LSTM and GRU must have a similar activation function, no. layers, and inputs in the experimental. Every model consists of a single input layer, a GRU/ LSTM/ Bi-LSTM layer, a dropout layer, and ultimately a dense output layer. The no. of memory units in the input layer is the same as the no. of input features. There are 64 memory units in the RNN-LSTM/GRU layers and 128 memory units in the Bi-LSTM model. The hyperbolic tangent is the activation function utilized in each RNN-BiLSTM-LSTM/GRU layer. Table 2 contains the complete list of attributes used in these models. The earlier stopping strategy is used in these models to avoid overfitting and underfitting of training samples due to excessive or few epochs. Early stopping is a technique that permits defining a high no. training epochs and then end training when the performance of the model on the validating dataset stops advancing.

There are 3 subsets of the total data: training set, validation set, and test sets as demonstrated in Table 3. During the training phase, the validation data set is utilized to verify the loss function (LF) value for the respective epoch. The LF MAE is utilized in the method. At the end of each period, the MAE is calculated. After a specific number of epoch delays, the process of training is terminated if no further gain or change in the value of LF. The first indication of no further improvements isn't the ideal time to end the process of training since this model reaches a standstill or even deteriorates somewhat before improving. To cater this, a trigger delay in terms of the no. of epochs with no improvements is specified.

6. Results of experiment and discussion

Several typically used neural-networks were chosen as comparison models to test the stability and performance of the hybrid RNN model, including the GRU, LSTM, Bi-LSTM, and RNN models. The datasets for training were used to learn the attributes of the hybrid RNN, GRU, LSTM, and RNN models, and the datasets for testing were provided to the four prediction models. The loss value of the loss function is one approach

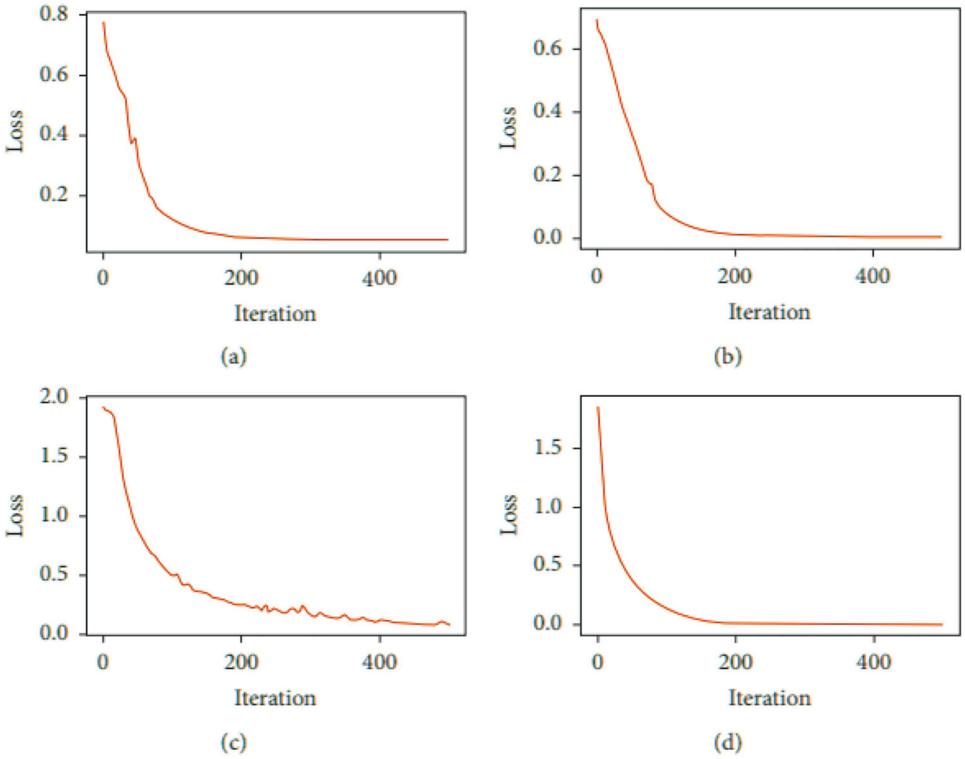


Figure 8. Losses of Neural networks on the training-set for 500 iterations. (a) RNN. (b) LSTM. (c) GRU. (d) Proposed Hybrid RNN.

to determine if a prediction model has been properly learned and trained. The cross-entropy function, which measures the distance between two probability distributions, was chosen as the loss function. The close the two probability distributions are, the lower the cross-entropy. The cross-entropy formula is:

$$Loss = -\sum_{i=1}^{m} z_i log z_i', \qquad (30)$$

z_i' is the predicted value and z_i is the actual value, correspondingly, and the size of the training samples is indicated by m. The forecasting model fits the training set better if the value of the loss is next to zero. Figure 8 depicts the loss value of the four-layer forecast methods.

Figure 8 shows that the RNN, GRU, and hybrid RNN models converge quickly, with loss values less than 0.1 after 200 iterations, whereas the LSTM model's loss value is around 0.2 after 200 iterations, with minor variations. After 500 iterations, all these techniques acquire a very minimal loss-value, indicating that they can effectively learn the training-set and the performance of the model was well on the testing set. The loss value fell relatively slowly after training the training set for more than 500 iterations. On the test set, all these models perform well, but they cannot forecast well on the training dataset if the iteration times are increased further. Since the models were clearly overfitted, the iteration times in this test were set at 500 times.

The proposed hybrid model and RNN-LSTM and GRU methods are run for many look-backs with epoch-100 and having a 32-batch size. There are two input sets in this experiment. Set I solely utilize historical attributes of stocks, whereas Set II includes attributes of historical stock as well as sentiment scores of news. As these inputs are separately utilized to train the proposed hybrid model and RNN-LSTM and the GRU. Various values of look-back are used in the studies. Table 4 contains the comprehensive statistical figures for SETI. Table 5 contains the comprehensive statistical figures for SETII (Table 6).

Table 4. Data distribution for training and testing.

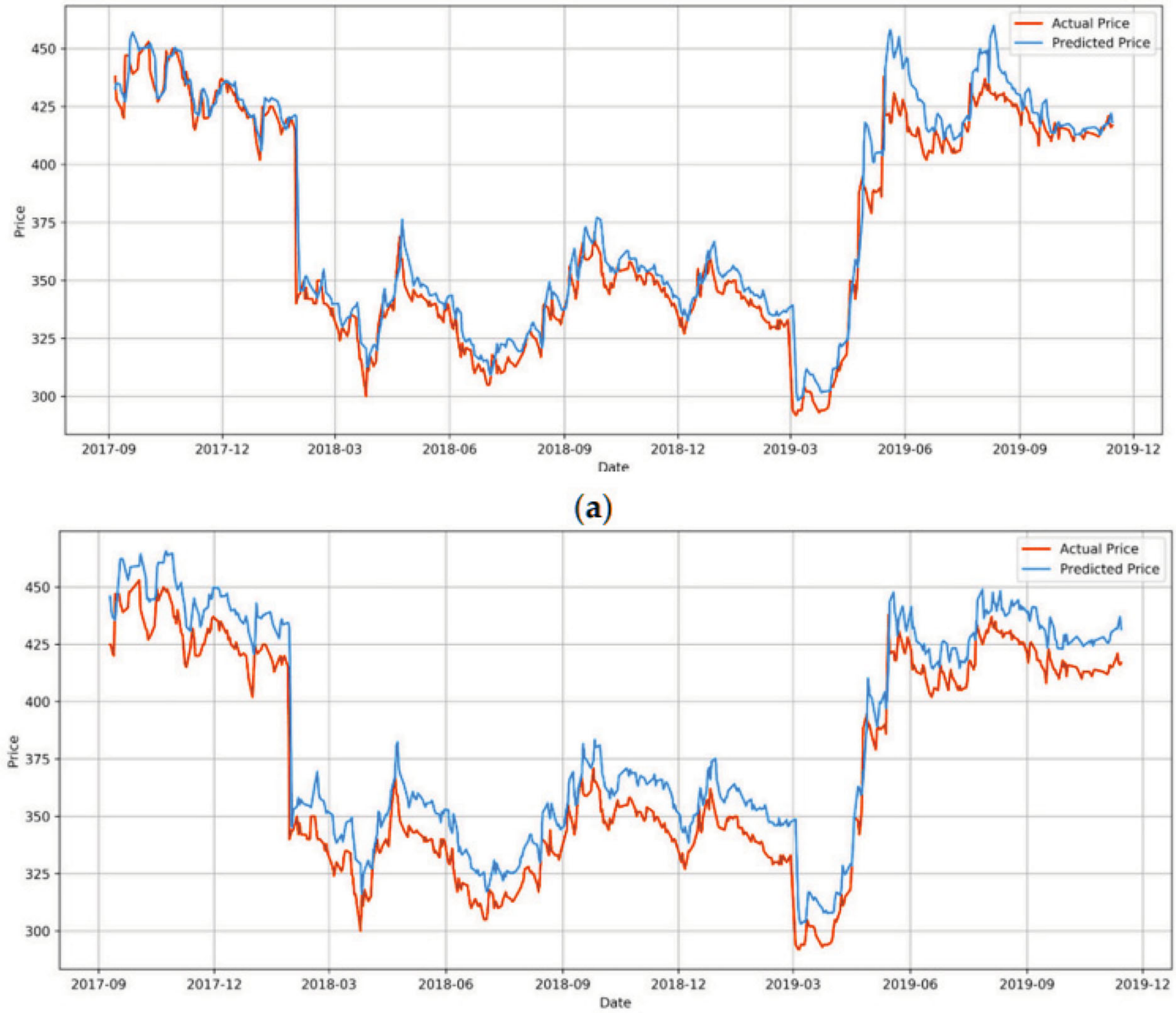
Data	Dates(mm/yyyy)	Samples
Total data	03/2012 to 11/2019	1991
Training	03/2012 to 05/2016	1186
Validation	06/2012 to 09/2016	292
Test data	10/2017 to 11/2019	513

Table 5. Summary of results of SETI.

Evaluation metrics	RNN (data)	RNN-LSTM (data)	RNN-GRU (data)	HyRNN (data)
MAE	48.47	17.689	15.726	14.622
RMSE	52.933	23.070	20.752	19.512
R^2	0.867	0.979	0.981	0.984
DA	0.58	0.60	0.65	0.68

Table 6. SETII results summary.

Evaluation metrics	RNN (data + news)	RNN-LSTM (data + news)	RNN-GRU (data + news)	HyRNN (data + news)
MAE	46.17	15.489	14.321	13.102
RMSE	49.824	21.870	19.152	18.142
R^2	0.871	0.981	0.983	0.987
DA	0.60	0.63	0.67	0.70



(a)

(b)

Figure 9. From 09/2017 to 11/2019, the following is the actual and expected price: (a) RNN-GRU-Stock; (b) Hybrid RNN-Stock.

The hybrid RNN model with combined data and news sentiment outperforms on all indicators. If the news emotion isn't taken into account, the order for hybrid RNN and RNN-LSTM and GRU is almost completely reversed. To put it another way, financial news perceptions have a significant impact on stock market forecasting. Since the distinction in DA between the models is smaller, all models can accurately forecast the trend in the stock.

6.1. Discussion

The goal of this work is to undertake a normalized comparison of the RNN-GRU and proposed hybrid RNN models' forecasting of stock market performances under the similar conditions, and to quantitativelyanalyse the importance of including sentiments of financial news in forecasting stock market versus using simply stock attributes. As a result, our findings are sufficient for us to meet the study's objectives. Our discussion is centred on the challenges that arise as a result of these planned objectives.

(A) Performances of RNN-GRU and proposed hybrid RNN Using Only Stock Features

The proposed hybrid RNN outperforms RNN-GRU in stock price prediction. Results indicate that the performance of RNN-GRU and proposed hybrid RNN are significantly diverse under identical settings using MAE as the metric, and the performance of the proposed hybrid RNN is better with 14.62 MAE compared to 15.7 for RNN-GRU (Table 5). Thus, appears to support the result given in [17]. However, after examining the value on the test plots, in Figure 8, it appears that RNN-GRU performed well than the proposed hybrid RNN for the majority of the time, with the exception of April to September 2019. It appears that the MAE's overall average mischaracterizes the complete forecasting scenario to some extent. With solely stock market features as input, it is more logical to conclude that both

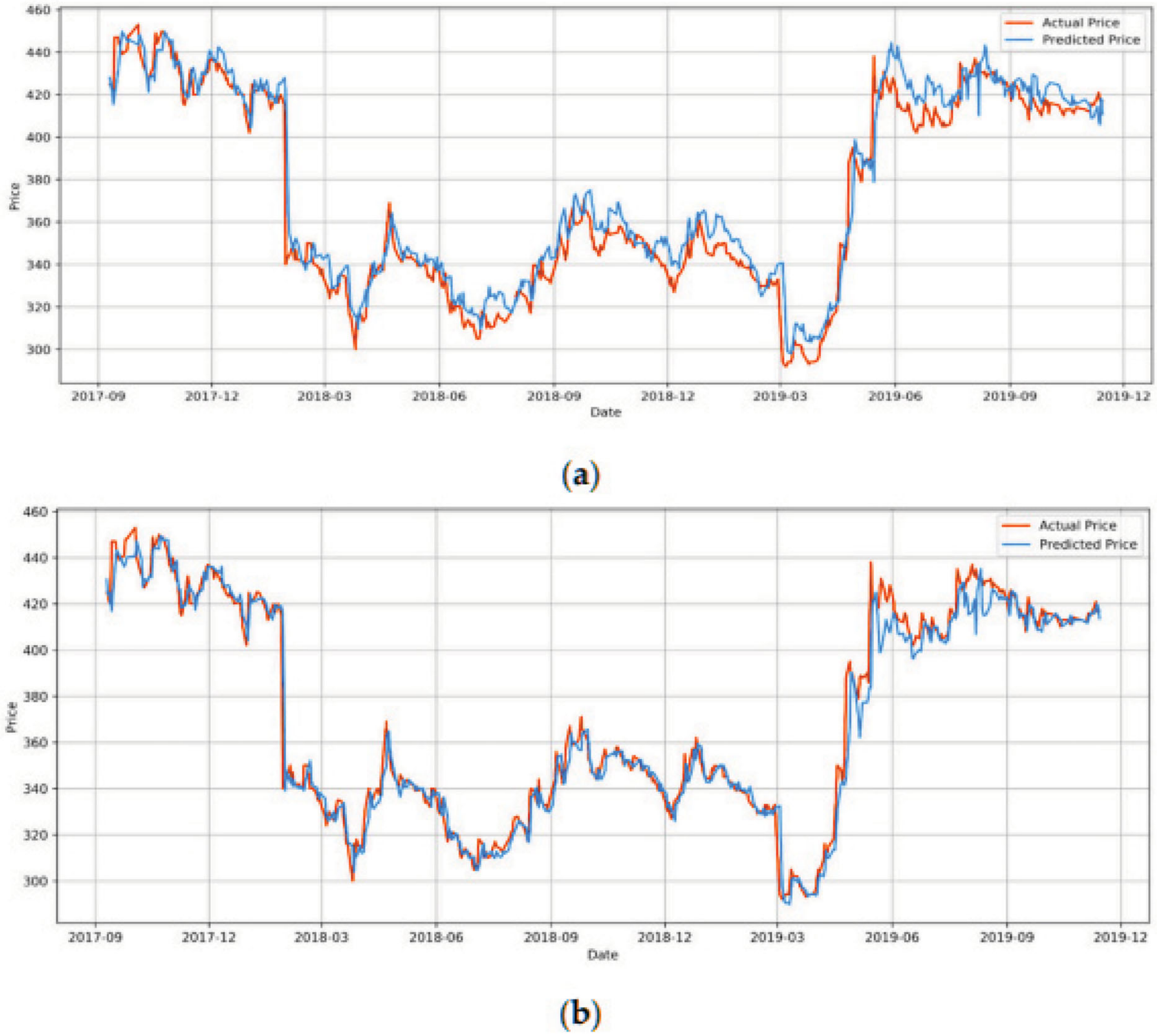


Figure 10. From 09/2017 to 11/2019, the following is the actual and expected price: (a) RNN-GRU- Stock + news; (b) Hybrid RNN-Stock + news

RNN-GRU and proposed hybrid RNN are contextual in forecasting stock. This finding appears to be in line with the findings of [8].

(B) Performances of RNN-GRU and proposed hybrid RNN with combined Stock Features and Financial News Sentiments

Our findings and prior studies show that combining sentiments of financial news with stock attributes as an input improves the performance of both RNN-GRU and proposed hybrid RNN in forecasting stock significantly Another viewpoint is that without taking financial news sentiments into account, both the RNN-GRU and the proposed hybrid RNN are situational in stock predictions. The following section will examine if both RNN-GRU and the proposed hybrid RNN are still contextual in forecasting stock when both stock market attributes and financial news sentiments are employed as inputs. The results demonstrate the performance of the RNN-GRU and the proposed hybrid RNN are significantly diverse ($p = 0.001$) under the same settings with MAE as the metric, and the proposed hybrid RNN method has improved results with an MAE of 13.1 compared to an MAE of 14.3 for the RNN-GRU model (Table 4). The coefficient of determination (R^2) among the hybrid RNN method (0.987) and the RNN-GRU (0.983).

When comparing the test plots of RNN-LSTM and proposed hybrid RNN methods in Figure 9, hybrid RNN seems to have outperformed the proposed RNN-GRU over the whole period. This means that the proposed cooperative deep-learning framework be developed as an efficient model to provide the best feasible forecasts.

7. Conclusion

Many investors and researchers are interested in the topic of the prediction of stock trends using deep learning since the accuracy of enhanced prediction will almost certainly result in huge profits. We proposed a stock prediction system in this study to assist investors in trading the stock market. To address these, RNN based hybrid model called HYRNN is proposed. This proposed model combines GRU units, LSTM, and Bi-LSTM to implement a deep learning method by combining the financial news sentiments data with the features of stock as the input. GRU, LSTM, and RNN DL models are also trained with a similar set of datasets to evaluate and compare the performance and efficiency of our model. The analysis of results significantly specifies that the proposed model fortified with the combination of stock features and sentiment of news surpassed the other compared deep learning models.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Malkiel BG, Fama EF. Efficient capital markets: a review of theory and empirical work. J Financ. 1970;25:383–417.

[2] Fama EF. Random walks in stock market prices. Financ Anal J. 1995;51:75–80.

[3] Shen G, Tan Q, Zhang H, et al. Deep learning with gated recurrent unit networks for financial sequence predictions. Proc Comput Sci. 2018;131:895–903.

[4] Araújo RDA. A quantum-inspired evolutionary hybrid intelligent approach for stock market prediction. Int J Intell Comput Cybern. 2010;3:24–54.

[5] Vui CS, Soon GK, On CK, et al. A review of stock market prediction with Artificial neural network (ANN). In Proceedings of the 2013 IEEE international conference on control system, computing and engineering, Mindeb, Malaysia, 29 November–1 December 2013; p. 477–482.

[6] Shi C, Zhuang X. A study concerning soft computing approaches for stock price forecasting. Axioms. 2019;8:116.

[7] Nti IK, Adekoya AF, Weyori BA. A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev. 2019;53:3007–3057.

[8] Lv D, Huang Z, Li M, et al. Selection of the optimal trading model for stock investment in different industries. PLoS One. 2019;14:e0212137.

[9] Atsalakis GS, Valavanis KP. Surveying stock market forecasting techniques–part II: soft computing methods. Expert Syst Appl. 2009;36:5932–5941.

[10] Seng J-L, Yang H-F. The association between stock price volatility and financial news – a sentiment analysis approach. Kybernetes. 2017;46:1341–1365.

[11] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: a case study of China stock market. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015; p. 2823–2824.

[12] Althelaya KA, El-Alfy E-SM, Mohammed S. Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In Proceedings of the 2018 9th international conference on information and communication systems (ICICS), Irbid, Jordan, 3–5 April 2018; p. 151–156.

[13] Jiawei X, Murata T. Stock market trend prediction with sentiment analysis based on LSTM neural network. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists (IMECS), Hong Kong, China, 13–15 March 2019; p. 13–15.

[14] Samarawickrama A, Fernando T. A recurrent neural network approach in predicting daily stock prices an application to the Sri Lankan stock market. In Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 15–16 December 2017; p. 1–6.

[15] Li J, Bu H, Wu J. Sentiment-aware stock market prediction: A deep learning method. In Proceedings of the 2017 International Conference on Service Systems and Service Management, Dalian, China, 16–18 June 2017; p. 1–6.

[16] Rahman MO, Hossain MS, Junaid T-S, et al. Predicting prices of stock market.

[17] Saud AS, Shakya S. Analysis of look back period for stock price prediction with RNN variants: a case study on banking sector of NEPSE. Proc Comput Sci. 2020;167:788–798.

[18] Dang ML, Sadeghi-Niaraki A, Huynh HD, et al. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access. 2018;6:55392–55404.

[19] Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv 2014, arXiv:1406.1078.

[20] Zhang M, Li X, Yue S, et al. An empirical study of TextRank for keyword extraction. IEEE Access. 2020;8:178849–178858.

[21] Janardhanan D, Barrett E. CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models, in Proc. 12th Int. Conf. Internet Technol. Secured Trans. (ICITST), Dec. 2017, p. 55–60.

[22] Wijnhoven F, Plant O. Sentiment analysis and Google trends data for predicting car sales, in Proc. 38th Int. Conf. Inf. Syst., Seoul, South Korea, 2017, p. 1–16.

[23] Thelwall M, Buckley K, Paltoglou G. Sentiment strength detection for the social web. J Assoc Inf Sci Tech. 2012;63(1):163–173.

[24] Mercer J. Functions of positive and negative type, and their connection the theory of integral equations. Phil Trans R Soc London A Math Phys Eng. Jan. 1909;209:415–446.

[25] Patel J, Shah S, Thakkar P, et al. (2015). Predicting stock and stock price index movement using trend deterministic data prep.

[26] Milosevic N. (2016). Equity forecast: predicting long term stock price movement using machine learning. arXiv. Available at https://arxiv.org/abs/1603.00751.

[27] Li X, Wu P, Wang W. Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong. Inf Process Manag. 2020;57:102212.

[28] Oliveira N, Cortez P, Areal N. The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. Expert Syst Appl. May 2017;73:125–144.

[29] Nguyen HM, Woo S, Im J, et al. A workload prediction approach using models stacking based on recurrent neural network and autoencoder, in Proc. IEEE 18th Int. Conf. High Perform. Comput. Commun., IEEE 14th Int. Conf. Smart City, IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS), Dec. 2016, p. 929–936.

[30] Amin A, Hossain I, Akther A, et al. Bengali VADER: a sentiment analysis approach using modified VADER,

in Proc. Int. Conf. Electr., Comput. Commun. Eng. (ECCE), Feb. 2019, p. 1–6.

[31] Cheng Y, Wang C, Yu H, et al. GRU-ES: Resource usage prediction of cloud workloads using a novel hybrid method, in Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun., IEEE 17th Int. Conf. Smart City, IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS), Aug. 2019, p. 1249–1256.

[32] Rustam F, Khalid M, Aslam W, et al. A performance comparison of supervised machine learning models for COVID-19 tweets sentiment analysis. PLoS One. Feb. 2021;16(2):e0245909.

[33] Kirlic A, Orhan Z. Measuring human and Vader performance on sentiment analysis, vol. 1, 2017, p. 42–46.

[34] Tymann K, Lutz M, Palsbröker P, et al. GerVADER – a German adaptation of the Vader sentiment analysis tool for social media texts, in Proc. LWDA, 2019, p. 178–189.

[35] Dutta A, Kumar S, Basu M. A gated recurrent unit approach to bitcoin price prediction. J Risk Financ Manag. 2020;13:23.

[36] Shen H, Hong X. Host load prediction with bi-directional long shortterm memory in cloud computing, 2020, arXiv:2007.15582. [Online]. Available: http://arxiv.org/abs/2007.15582.