

# A cooperative deep learning model for stock market prediction using deep autoencoder and sentiment analysis

KS Rekha<sup>1,2</sup> and MK Sabu<sup>1</sup>

<sup>1</sup> Department of Computer Applications, Cochin University of Science and Technology, Kochi, Kerala, India

<sup>2</sup> Department of Computer Science and Engineering, College of Engineering Kidangoor, Kottayam, Kerala, India

## ABSTRACT

Stock market prediction is a challenging and complex problem that has received the attention of researchers due to the high returns resulting from an improved prediction. Even though machine learning models are popular in this domain dynamic and the volatile nature of the stock markets limits the accuracy of stock prediction. Studies show that incorporating news sentiment in stock market predictions enhances performance compared to models using stock features alone. There is a need to develop an architecture that facilitates noise removal from stock data, captures market sentiments, and ensures prediction to a reasonable degree of accuracy. The proposed cooperative deep-learning architecture comprises a deep autoencoder, lexicon-based software for sentiment analysis of news headlines, and LSTM/GRU layers for prediction. The autoencoder is used to denoise the historical stock data, and the denoised data is transferred into the deep learning model along with news sentiments. The stock data is concatenated with the sentiment score and is fed to the LSTM/GRU model for output prediction. The model's performance is evaluated using the standard measures used in the literature. The results show that the combined model using deep autoencoder with news sentiments performs better than the standalone LSTM/GRU models. The performance of our model also compares favorably with state-of-the-art models in the literature.

Submitted 28 April 2022

Accepted 25 October 2022

Published 30 November 2022

Corresponding author

KS Rekha, rekhaswapsnesh@cusat.ac.in

Academic editor

Ghufran Ahmad

Additional Information and Declarations can be found on page 24

DOI 10.7717/peerj-cs.1158

© Copyright 2022 S and K et al.

Distributed under Creative Commons CC-BY 4.0

OPEN ACCESS

**Subjects** Data Mining and Machine Learning, Sentiment Analysis, Neural Networks

**Keywords** Stock market prediction, Deep learning, Sentiment analysis

## INTRODUCTION

Stock markets offer an opportunity to invest and gain from the growth of an industry or organization. Investment in the stock is perceived as risky, considering the various factors influencing the fluctuations of the stock prices and the instability of stock markets. Stock markets are complex, volatile, and chaotic, and stock price prediction is a non-trivial problem. The need for improved prediction in this domain is recognized, and researchers have been trying to evolve models that support the prediction of stock prices.

Accurate stock price prediction is a challenging problem due to the dynamic and volatile nature of the stock market. Stock markets are subject to fluctuations due to different causes, and determining the movement of stock price helps minimize investors' risk ([Mohanty, Parida & Khuntia, 2021](#)).

Due to stock data's dynamic, nonlinear, and non-parametric nature ([Li & Bastos, 2020](#)), statistical models were initially explored in stock market prediction ([Kumar, Sarangi & Verma, 2021](#)), but their applications were limited by weak performance of the models. Researchers started to explore the machine learning and deep learning techniques in stock prediction ([Deepak, Uday & Malathi, 2017](#)) to improve performance compared to that of statistical models. Machine learning algorithms, to a great extent, can give satisfactory results on stock market prediction ([Parmar et al., 2018](#)).

[Strader et al. \(2020\)](#) described the research direction of the stock market prediction domain, specifically using Artificial Neural Networks (ANN), meta-heuristics, Support Vector Machines (SVM), and Artificial Intelligence (AI) based techniques. We can find a non-linear relationship in a dataset without having apriori information of dependency of input and output in an ANN ([Kurani et al., 2021](#)) which is an advantage of ANN. In regression problems involving complex functions, due to overfitting and getting trapped in local minima, the ANNs cannot generally provide accurate results.

Researchers focused on using the Multilayer Perceptron in stock market prediction ([Pan, Tilakaratne & Yearwood, 2005](#); [Devadoss & Ligori, 2013](#)) decades ago. Later, different ANN models and hybrid algorithms were used. Stock market data is noisy, and ANN exhibits unpredictable and inconsistent behavior on this data ([Guresen, Kayakutlu & Daim, 2011](#)). Deep learning techniques are found to be helpful in stock performance prediction ([Singh & Srivastava, 2017](#)). Some of the drawbacks of ANN can be overcome by using Deep Neural Networks (DNN) as they give a better approximation to nonlinear models ([Yu & Yan, 2020](#); [Nabipour et al., 2020](#)). DNNs give better performance when applied to some time series forecasting ([Le Roux & Bengio, 2010](#); [Sutskever & Hinton, 2008](#)). [Takeuchi & Lee \(2013\)](#) proposed a study that uses autoencoders and restricted Boltzmann machines for feature extraction from the input variables for stock trading strategies.

Machine learning techniques enable the prediction of stock price by integrating technical indicators, financial news and social media posts ([Albahli et al., 2022](#)). Stock price prediction problem can be modeled as a time series forecasting problem involving past data and other attributes like news sentiments ([Bhardwaj et al., 2015](#); [Mehta, Pandya & Kotecha, 2021](#)). Some studies use sentiment analysis methods, stock price datasets, and some works integrating stock price datasets and sentiment analysis are discussed in the literature review in Section 2. With the rapid developments in soft computing and different computational techniques in the last decades, researchers have proposed many methods based on artificial neural networks, heuristics, and meta-heuristics for stock market prediction ([Araújo, 2010](#); [Vui et al., 2013](#); [Shi & Zhuang, 2019](#); [Lv et al., 2019](#); [Atsalakis & Valavanis, 2009](#); [Haider Bangyal et al., 2022](#); [Pervaiz et al., 2021](#)). News has a significant impact on stock prices. News articles are available publicly online, and we can extract valuable data and information from them [Khan et al. \(2019\)](#). The type of news data useful in stock market prediction can be categorized as general news, financial news, and political news.

Despite the dynamic growth of the real stock market, integrating news sentiment, historical stock data, and autoencoders is an exciting research domain. We propose a method incorporating deep learning models and a deep autoencoder combined with

sentiment analysis for improved stock prediction. The integrated architecture involving these three aspects facilitates an efficient prediction as the autoencoder minimizes the effect of noise in the data to complement the prediction capability of the deep learning model.

The remainder of the paper is structured as follows. Section 2 describes a detailed literature review on stock market prediction techniques and methods. Section 3 provides the article contribution of the research, and Section 4 gives methodology that includes the dataset, features, and proposed architecture. This section describes the working of the model using a schematic diagram. Besides, different aspects of deep learning used in modeling are also described. Section 5 presents the results and analysis obtained from this study, followed by discussions. The paper concludes with Section 6 which also includes future work.

## RELATED WORK

We discuss published literature in three different categories here. These categories are related to using deep learning models in stock prediction, autoencoders in financial time series prediction, and sentiment analysis, respectively. The state-of-the-art models pertaining to these domains is reviewed in detail and elaborated.

### Deep learning techniques in stock prediction

Deep learning, an advanced version of machine learning, has an excellent capability for information extraction from time-series data ([Nabipour et al., 2020](#)). Since stock price prediction is a sequence prediction time series problem, one of the most suited architectures for this is Recurrent Neural Networks (RNN). RNNs and variants of RNNs perform well on stock market prediction. [Roondiwala, Patel & Varma \(2017\)](#) have done a study using RNN and LSTM. [Nelson, Pereira & de Oliveira \(2017\)](#) also predicted whether a stock would move up or down using LSTM.

LSTM algorithms are suitable for prediction involving non-linear data ([Rajakumari, Kalyan & Bhaskar, 2020](#)). [Chen, Zhou & Dai \(2015\)](#) applied an LSTM model for stock price prediction problems concerned with the Chinese stock market. This model comprises of an input, LSTM, and an output layer with multiple neurons. This work showed that the normalized features could increase the forecast accuracy. In this work, they have not considered the news sentiments. [Shahi et al. \(2020\)](#) have done a study incorporating the sentiments using the news headlines. A comparison was made to evaluate the relative performance of LSTM and GRU models in [Rahman et al. \(2019\)](#).

A deep neural network with connections between the layers, composed of multiple hidden units but not between units within layers, is termed a deep-belief network. In work done by [Wang et al. \(2018\)](#), the backpropagation neural network model and the LSTM model are used to find the prediction accuracy. [Pawar, Jalem & Tiwari \(2019\)](#) used RNN along with LSTM for stock market prediction. They have compared the performance of the model with the common algorithms like random forest, feed forward neural network, Support Vector Machine, and backpropagation. Deep learning methods are useful in stock price prediction and in other domains of finance like portfolio management ([Soleymani & Paquet, 2020](#)). Increasing the expected return on investment by proper rationing of

the investment amount into the different components of the portfolio is the objective of portfolio management.

### Autoencoder

[Bao, Yue & Rao \(2017\)](#) used stacked autoencoders and LSTM for the financial time series prediction problem. In this study, the input financial time series data is passed through a multi-resolution discrete wavelet transformation, and the denoised financial time series is passed through a stacked autoencoder to get the one-step-ahead output using LSTM. [Song et al. \(2022\)](#) have shown that deep autoencoders are useful in prediction tasks. [Gunduz \(2021\)](#) obtained results with reasonable accuracy for the Istanbul stock market by performing stock market prediction using autoencoder and LSTM with attention.

The architecture of a deep autoencoder comprises two deep belief networks with more than three shallow layers for encoding and the next layer set for the decoding half. One highlight of our paper is that it uses a supervised model as well as an unsupervised model. Due to the dynamic nature of stock markets, even predicting the direction of the stock movement is a challenging task ([Li et al., 2017](#)). They have used a system of stacked denoising autoencoders to predict the movement of stock indexes. Deep learning architecture including autoencoders is efficient in complex learning problems with insufficient samples and uncertain information as demonstrated by [Zheng et al. \(2021\)](#).

Denoising of data is a crucial aspect of stock price prediction. To denoise data, [Liu et al. \(2019\)](#) used autoencoders, and LSTM is used for prediction. In short-term electric load forecasting, autoencoders have been successfully used by [Liu, Zheng & Chen \(2019\)](#). The analysis was done with data on Chinese cities and found that the prediction error decreases using autoencoders compared to the traditional backpropagation (BP) neural networks. Although LSTM-based methods provide better performance than conventional methods in time series prediction, it has inherent limitations in modeling multivariate data. LSTM and stacked autoencoders were used by [Sagheer & Kotb \(2019\)](#) to validate the model's performance.

### Sentiment analysis

Sentiment analysis involves the detailed analysis of text with opinions and emotions. Sentiment analysis seeks to find whether the subjective parts of the text contain positive or negative sentiments. Studies show that sentiments positively or negatively impact stock market forecasting. In this research, we are considering the news headlines. The positive news is assigned a value of 1, and the negative information is assigned a value of 0.

The use of deep learning models along with sentiment analysis is considered a relevant research domain due to the dynamic nature of the stock markets ([Bhardwaj et al., 2015; Yadav et al., 2019; Jiawei & Murata, 2019; Moghar & Hamiche, 2020; Shahi et al., 2020; Mehta, Pandya & Kotecha, 2021](#)). [Mehta & Pandya \(2020\)](#) presents a comprehensive review of machine learning approaches to identify sentiments from a context.

[Bhardwaj et al. \(2015\)](#) used news sentiments and historical stock prices to predict the stock market. In this work, news articles and stock values are preprocessed, the sentiments

are classified, and the analysis is performed using the random forest method. Thus prediction is made, and visualization of stock data is performed. *Carosia, Coelho & da Silva (2021)* proposes strategies for investment based on sentiment analysis in the context of the Brazilian stock market. *Nemes & Kiss (2021)* performed stock movement prediction using stock news headlines using different sentiment analysis tools. Stock prediction incorporating investor sentiments and stock data is effective, as demonstrated by *Wu et al. (2022)* based on Chinese stock market data.

To predict the stock market, *Yadav et al. (2019)* used a combination of machine learning algorithms and lexicon-based labeling of financial news. The ‘market emotion’ is a significant factor influencing the stock market forecasts, as indicated in *Jiawei & Murata (2019)*, and in their work, they used LSTM to predict stock trend prediction. Feature selection was used to select useful indices of stock and use a deep learning model to do financial news sentiment analysis to predict stock trends. *Deorukhkar et al. (2019)* used LSTM networks, ARIMA, and Sentiment Analysis to predict the stock price. Natural language processing methods are used to bridge the stock market indices and the news sentiments to improve the performance of stock market prediction (*Seng & Yang, 2017*).

In the problems involving sentiments, the lexicon-based approach is robust and provides reasonably good results (*Hailong, Wenyan & Bo, 2014*). We have adopted Valence Aware Dictionary and sEntiment Reasoner (VADER), a lexicon and rule-based sentiment analysis tool, in our model. Each word is linked with a sentiment polarity value, and a predefined dictionary of words is used in a dictionary-based method (*Bhonde et al., 2015*).

Business sentiment can be effectively captured from news text based on newspaper articles (*Seki, Ikuta & Matsubayashi, 2022*). *Mehta, Pandya & Kotecha (2021)* have proposed models for stock price prediction by integrating sentiment analysis and different machine learning techniques. In addition, the association between news sentiments and stock trends was also explored. *Gite et al. (2021)* have also shown that prediction accuracy can be enhanced by fusing sentiments along with technical analysis. Prediction of stock market movement during the COVID-19 pandemic was made by *Das et al. (2022)* using different sentiment analysis tools and LSTM. As evident from the above works, using market sentiments captured from news articles can improve stock price prediction accuracy.

## ARTICLE CONTRIBUTION

To the best of our knowledge, research incorporating news sentiment analysis, historical stock price data, and deep autoencoders for noise reduction in a single stock prediction system is not available in the literature. Since time series problems such as stock market prediction are sequence prediction problems, it is crucial to consider the sequential relationships among the data. To address these research gaps, we have tried to:

1. Develop an integrated architecture based on deep learning with features captured from previous stock price data and a deep autoencoder for noise removal.
2. Understand the impact of news sentiment on stock market prediction.
3. Find which model of deep-learning, GRU or LSTM, performs better in stock market prediction.

High stakes and incentives for better stock prediction demand more accurate models that can improve the prediction accuracy.

## METHODOLOGY

In this section, we first introduce the proposed architecture used in our model. Later we discuss the deep learning components of the model, deep autoencoder, LSTM, and GRU, and the preprocessing methods used in this study. We name the models DeepAutoEncoder-Long Short Term Memory Sentiment Analysis (DAE-LSTMSA) and DeepAutoEncoder-Gated Recurrent Unit Sentiment Analysis (DAE-GRUSA).

### Proposed architecture

The proposed architecture is shown in Fig. 1. The stock prediction is made through a multi-stage approach comprising denoising the data using an autoencoder, which is fed to an LSTM/GRU model that also receives information based on sentiment analysis. The noisy historical stock market data is fed into the autoencoder, where it undergoes encoding and decoding. The autoencoder used in this work is unsupervised learning or self-supervised learning since it does not require any human intervention, such as data labelling.

The encoder transforms the input into a lower-dimensional representation, and the decoder tries to reconstruct the original data from the lower-dimensional representation. Therefore, the objective is to restore the initial data with the minimum amount of information loss. Stock prices for each day and news related to the company for the same period are contained in this data. The top 25 news on a day related to the stock is taken into consideration for the study. The daily stock prices include six values termed Open, High, Low, Close, Adjusted Close, and Volume. We are considering the Open, High, Low, and Close (OHLC) values in this research collected from Yahoo finance.

The news dataset used in this study for sentiment analysis with news headlines is extracted from the Kaggle community. The top 25 news from the period 01/03/2000 to 01/07/2016 was used in this study. After separating and removing the unwanted features, the top 25 news was concatenated into a single string for a single day. This process was continued for all days. After the news string of a single day was obtained, it was then combined with the appropriate date (time series) and the stock market data of the particular company. This process is called the pickling of the file, in which we combine the news articles in string format along with stock data in integer format. In the sentiment analysis part with the news dataset, we collated the headlines of the articles and dates. Later, by extracting and cleaning the data, we performed text processing.

### Preprocessing

Data preprocessing means converting raw data into efficient and useful data. The stock data is noisy, and the noise is to be removed. To denoise, the data we pass through a deep autoencoder, and the output we obtain is a denoised historical stock dataset. We have done min-max data normalization using Eq. (1) to make all values within the range of 0 and 1. This improves the accuracy, increases the gradient descent rate, and generates the optimal

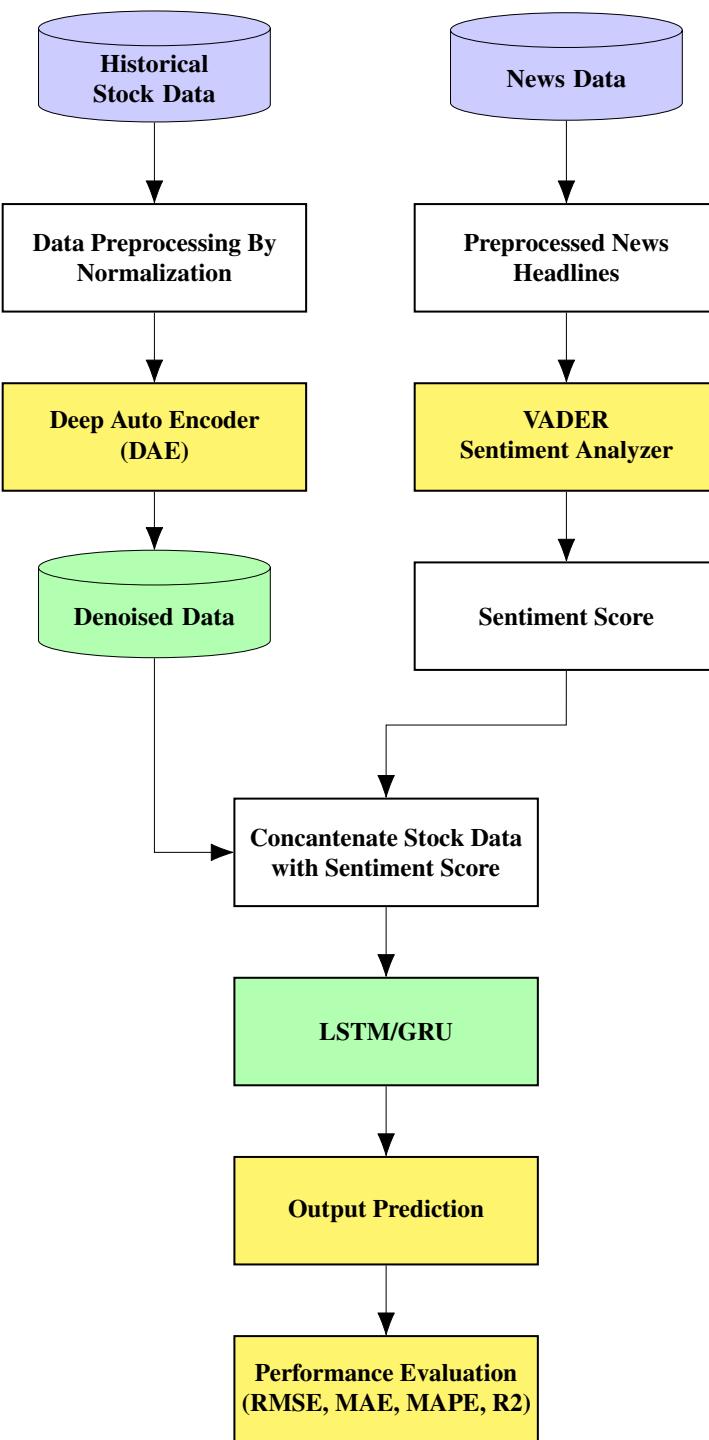


Figure 1 Proposed architecture.

Full-size DOI: 10.7717/peerjcs.1158/fig-1

solution ([Ding & Qin, 2020](#)).

$$y = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (1)$$

In [Eq. \(1\)](#) the minimum ( $x_{min}$ ) and maximum values ( $x_{max}$ ) relate to the value x being normalized. The normalized data is fed into the autoencoder for further processing. Data normalization aims to ensure quality data to the central parts of the model.

The text data is unstructured, and hence preprocessing is necessary before feeding it to the model. We create a list of the news section URL of the component companies and then extract the relevant news article's web links from the company's news section page. We remove the duplicate news articles based on news titles, and the main text from the selected news articles is extracted. By concatenating all the article headlines, a single string was formed for a single day after processing the relevant articles. After obtaining the single string for a day, it is merged with the appropriate date and stock index value. We compute the sentiment analysis score generated from textual data, which is added to the stock data and is used as the input of the prediction model.

### Deep autoencoder

The noisy stock market data is fed into the DAE, which undergoes encoding and decoding. The DAE used in this work is unsupervised learning or self-supervised learning since it does not require human intervention such as data labeling ([Skansi, 2018](#)). The encoder transforms the input to a lower dimensional representation, and a decoder reconstructs the original data from the lower dimensional representation. Therefore the objective is to restore the initial data with the minimum amount of information loss. A DAE with two symmetrical deep belief networks having an encoding half and the second with a decoding half is used here. In the encoder part, the input data is compressed to reduce the relevant information, which results in significant size reduction. The decoder produces the reverse operation of the encoder, *i.e.*, the uncompressed data is created as a reconstruction of the input as accurately as possible.

The DAE we have used in modeling can be diagrammatically represented as shown in [Fig. 2](#), which consists of an input layer, encoding deep belief network, compressed feature vector, decoding deep belief network, and an output layer.

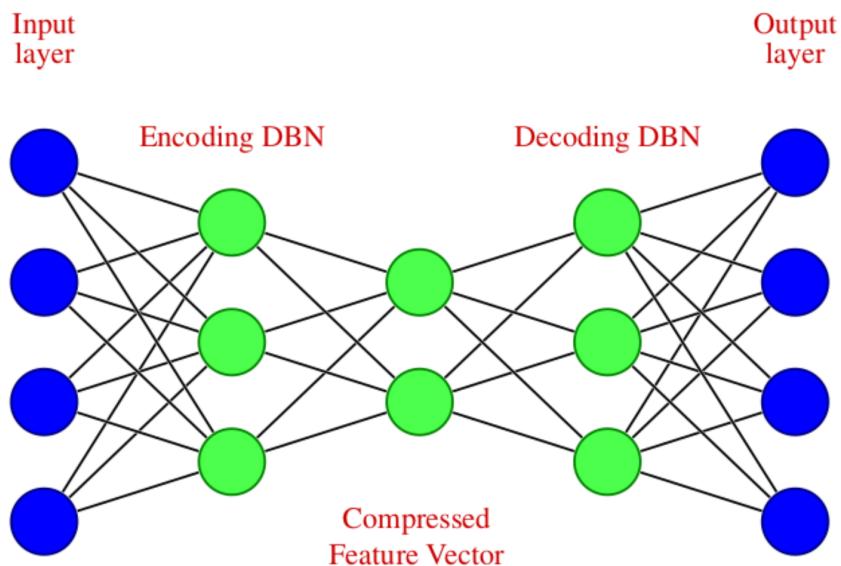
The DAE structure is shown in [Table 1](#) which consists of different layers such as input and dense layer, the shape of the output, and different parameters.

The notations used in the DAE are given in [Table 2](#). The process of encoding and decoding in the DAE follows [Eqs. \(2\)](#) and [\(3\)](#). The DAE is trained by error minimization as shown in [Eq. \(4\)](#).

$$t_i = f(w_t \cdot s_i + b_t) \quad (2)$$

$$d_i = g(w_d \cdot t_i + b_d) \quad (3)$$

$$E(S, D) = \frac{1}{2} \sum_{i=1}^n ||s_i - d_i||^2 \quad (4)$$

**Figure 2** Deep Autoencoder with four input and output features.

Full-size DOI: 10.7717/peerjcs.1158/fig-2

**Table 1** Deep autoencoder structure.

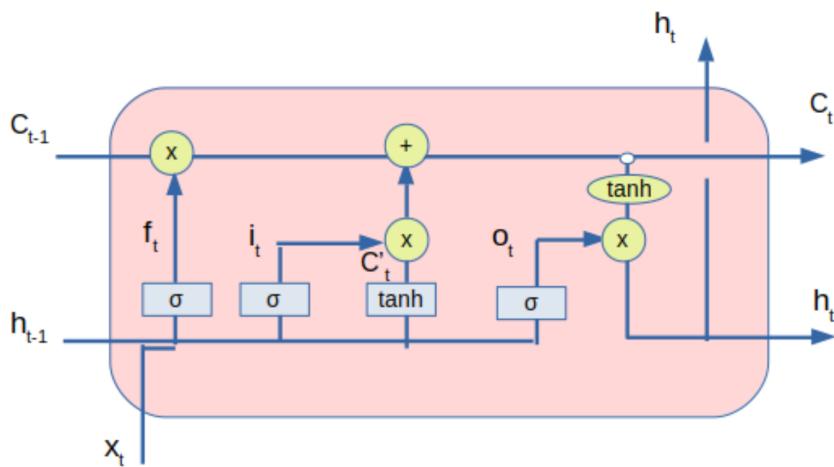
Layer (Type)	Output shape	Param #
input_1 (InputLayer)	(None, 4)	0
dense_1 (Dense)	(None, 80)	400
dense_2 (Dense)	(None, 50)	4050
dense_3 (Dense)	(None, 80)	4080
dense_4 (Dense)	(None, 4)	324

**Table 2** Deep autoencoder notations and descriptions.

Notation	Description
$S$	Input data to encoder, $S = \{s_1, s_2, \dots, s_n\}$
$T$	Original data characteristic sequence, $T = \{t_1, t_2, \dots, t_n\}$
$D$	Output data from decoder $D = \{d_1, d_2, \dots, d_n\}$
$f, g$	Sigmoid functions
$w$	Weights
$b$	Biases

## LSTM model

The processed data from the DAE, along with sentiment value, is provided to the LSTM network. LSTM is used as it can capture long-term dependencies, thereby eliminating the problem of vanishing gradient (Hochreiter, 1998). LSTM helps in feature extraction from data using its multi-layered architecture. The number of neurons in different layers and the

**Figure 3** LSTM with inputs and outputs.

Full-size DOI: 10.7717/peerjcs.1158/fig-3

**Table 3** LSTM notations and descriptions.

Notation	Description
$f_t$	Forget gate
$i_t$	Input gate
$o_t$	Output gate
$C_t$	Cell state
$h_t$	Hidden state
$\sigma$	sigmoid
$\tanh$	tanh
$W, U$	Weights
$b$	Bias
$\odot$	Element wise multiplication

number of layers impact the performance. The LSTM can be diagrammatically represented as shown in Fig. 3.

Table 3 represents the notations and descriptions used in LSTM.

The Eqs. (5), (6) and (7) represents the forget gate, input gate and output gate respectively. The candidate state and state of the memory cells at time  $t$  are given by the Eqs. (8) and (9) respectively whereas the Eq. (10) represents the equation for the hidden state.

The inputs provided to the LSTM are cell states and hidden states of time-step  $t - 1$ . The outputs of the LSTM are cell states and hidden states of time-step  $t$ . Forget gate dictates the extent of information that can be input into the LSTM unit. The input gate controls the extent of information that can be learned. Tanh is the activation layer that maps inputs into cell states. Output gate controls the information that can be output as a hidden state for the next time-step. Tanh activation layer activates the information of the cell state.

$$f_t = \sigma(W_f \times x_t + U_f \times h_{t-1} + b_f) \quad (5)$$

**Table 4** LSTM structure.

Layer (type)	Output shape	Param #
lstm_5 (LSTM)	(None, 2, 100)	42000
lstm_6 (LSTM)	(None, 256)	365568
dropout_4 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 4)	1028
Total params: 408,596		
Trainable params: 408,596		
Non-trainable params: 0		

$$i_t = \sigma(W_i \times x_t + U_i \times h_{t-1} + b_i) \quad (6)$$

$$o_t = \sigma(W_o \times x_t + U_o \times h_{t-1} + b_o) \quad (7)$$

$$C'_t = \tanh(W_c \times x_t + U_c \times h_{t-1} + b_c) \quad (8)$$

$$C_t = \sigma(f_t \odot C_{t-1} + i_t \odot C'_t) \quad (9)$$

$$h_t = \tanh(C_t) \odot o_t \quad (10)$$

The processed data from the autoencoder is provided to the LSTM network. LSTM structure is shown above in [Table 4](#).

LSTM helps in extracting features from data using its multi-layered architecture. The architecture has two connected layers, and in the output layer, we have used the sigmoid activation function. In between the second layer of LSTM and fully connected layers, we have a dropout rate of 0.2. The dataset is iterated over 100 epochs with a batch size of 32. The model is further analyzed for performance metrics described in the literature review.

### GRU model

The processed data from the autoencoder is also fed to the gated recurrent unit (GRU) network. GRU is also a type of RNN used in deep learning models. The architecture of the GRU is shown in [Fig. 4](#). The GRU comprises of update gate ( $z_t$ ), reset gate ( $r_t$ ) and a current memory content ( $\hat{h}_t$ ). The output of the GRU shown as ( $h_t$ ) is stored in the final memory. The update gate controls the input ( $x_t$ ) and previous output ( $h_{t-1}$ ) to be transferred to the next cell, which is regulated by ( $W_z$ ). The reset gate has deployed the extent of information to the forget gate. Only the relevant information is transferred to the next iteration based on the weight W.

The main operations by the GRU are given by the following equations:

$$z_t = \sigma(W_z \odot [h_{t-1}, x_t]) \quad (11)$$

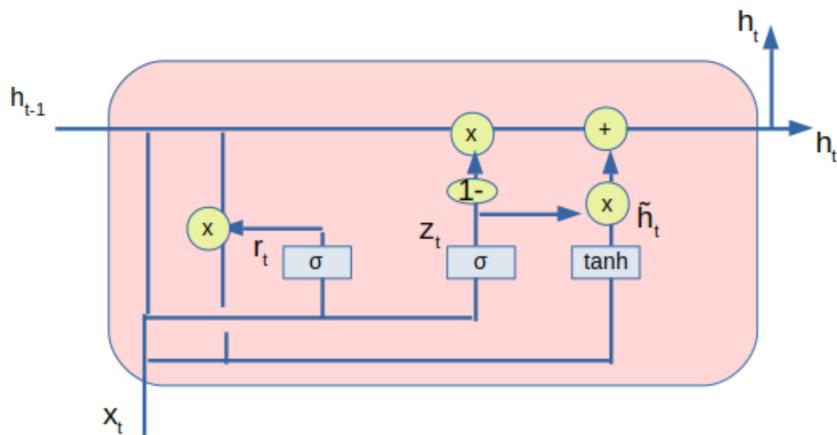


Figure 4 GRU with inputs and outputs.

Full-size DOI: 10.7717/peerjcs.1158/fig-4

Table 5 GRU notations and descriptions.

Notation	Description
$z_t$	Update gate
$r_t$	Reset gate
$o_t$	Output gate
$\tilde{h}_t$	Current memory content
$h_t$	Output state
$h_{t-1}$	Previous output state
$\sigma$	sigmoid
tanh	tanh
$W, W_z$	Weights
$\times$	Multiplication
$\odot$	Element wise multiplication

$$r_t = \sigma(W_z \odot [h_{t-1}, x_t]) \quad (12)$$

$$\tilde{h}_t = \tanh(W[r_t \times h_{t-1}, x_t]) \quad (13)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (14)$$

where ( $z_t$ ) and ( $r_t$ ) are intermediate values available from the update and reset gates and tanh is the hyperbolic tangent function and  $\sigma$  is the sigmoid function.

The notations and descriptions used in GRU is given in Table 5. The Eqs. (11) and (12) represents the values of the update gate and reset gate respectively. The current state of the memory cells at time  $t$  are given by the Eqs. (13) and (14) represents the output state.

**Table 6** GRU structure.

Layer (type)	Output shape	Param #
gru_5 (GRU)	(None, 2, 100)	31800
gru_2 (GRU)	(None, 256)	274176
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 5)	1285
Total params: 307, 261		
Trainable params: 307, 261		
Non-trainable params: 0		

**Table 7** Summary of results of evaluation metrics without deep autoencoder.

Evaluation metrics	LSTM	LSTMSA	GRU	GRUSA
RMSE	1.09	1.08	1.05	1.01
MAE	0.7638	0.7401	0.7285	0.6894
MAPE	1.6356	1.3824	1.4794	1.2502
R2	0.9768	0.9771	0.9803	0.9830

The structure of the GRU is shown in [Table 6](#). The model hyperparameters in GRU are optimized using Grid Search.

GRU overcomes the disadvantages of RNN and also eliminates the vanishing gradient problem. It helps in extracting features from data using its multi-layered architecture.

The architecture has two connected layers, and we use an update gate and a reset gate here. We have used the sigmoid function in the output layer. In between the second layer of GRU and fully connected layers, we have accommodated a dropout rate of 0.2. The dataset is iterated over 100 epochs with a batch size of 32. The model is further analyzed for performance metrics described in the literature review.

## VADER

The sentiment analysis is performed using VADER (Valence Aware Dictionary for sEntiment Reasoning) and creates a score index. Introduced by [Hutto & Gilbert \(2014\)](#), VADER is an integrated version of rule-based and lexicon-based software. It can detect the polarities and the sentiment intensity in the news. Researchers commonly use VADER as it is open-source, free to use, and good in speed performance tradeoffs. This software uses fewer resources than other machine learning-based models and needs a lesser amount of training data. VADER is part of the natural language processing package NLTK. We have analyzed the performance of VADER and Textblob and got better results for VADER. The sequence of steps in sentiment classification is shown in [Fig. 5](#).

To perform sentiment analysis using VADER, we read the news headlines and pre-processed the news headlines by cleaning them. We calculate the polarity compound, positive, negative, and neutral using the SentimentIntensityAnalyzer. The polarity score is taken, and the sentiment is classified as positive or negative. The positive news is given a value of 1, and the negative news is given a value of 0.

The experimental results are discussed in the next section.

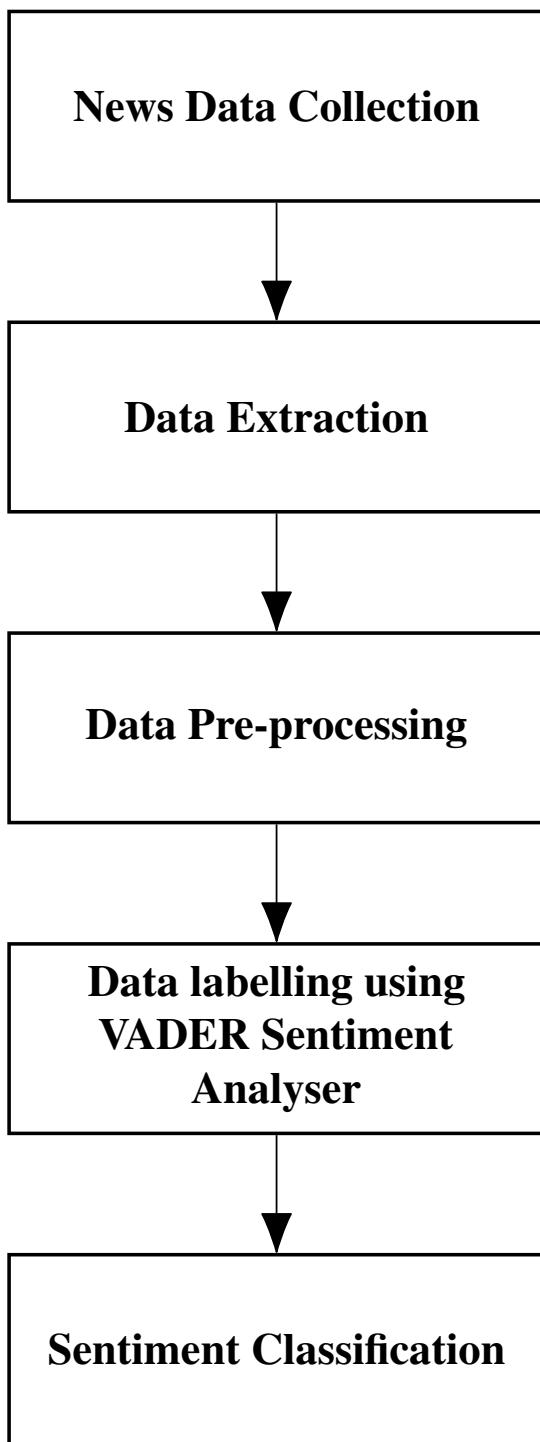


Figure 5 Sentiment classification of news headlines.

[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-5

## RESULTS AND DISCUSSIONS

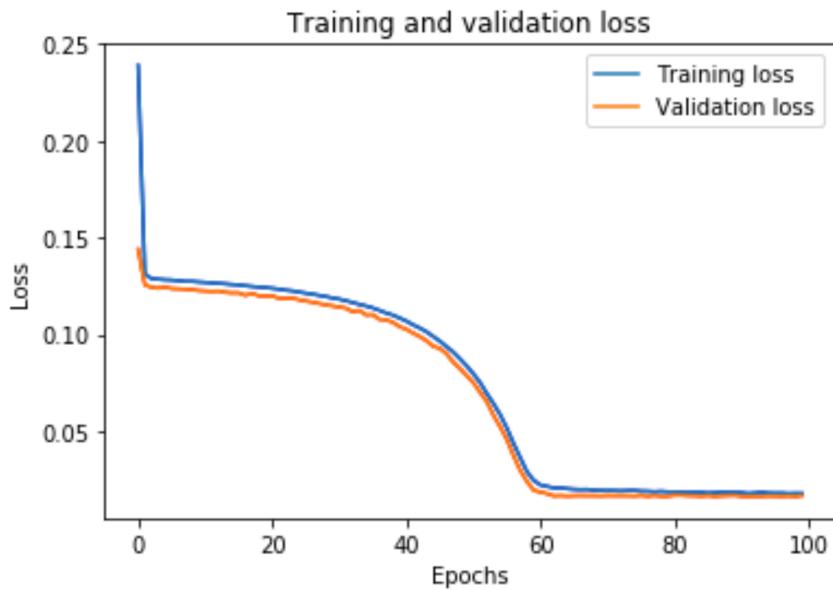
In order to check the performance of the proposed model, we have considered the stock of a reputed company and validated the results in this section. We have also compared the result with other state-of-the-art methods in literature ([Dey et al., 2021](#)). The experiment was performed on an i5, 2.7 GHz processor with 8 GB RAM, and we used the Linux operating system. The computations are done in Python open source programming language. We have done work with DAE and LSTM using stock dataset, which is accepted for publication, and have obtained reasonable results with this model. This work is adopted in this model, and sentimental analysis is combined with it to enhance the performance of the model. The experimental analysis is performed with our accepted DAE-LSTM model for the stock dataset we considered here for our study. The results show that the DAE-LSTM model works better than considering the LSTM model alone. The final results show that passing historical data through DAE and combining it with LSTM/GRU, and concatenating with sentimental score gives better results than the stand-alone models.

### Dataset

The experimental dataset is divided into two parts: the historical dataset and news headlines to obtain the sentiment index. We consider the period from 01/03/2000 to 01/07/2016 for daily stock prices and news articles of the company in this study. The daily stock prices contain six values Open, High, Low, Close, Adjusted Close, and Volume. We are considering the Open, High, Low, and Close (OHLC) values for historical data in this is collected from Yahoo finance. The top 25 news on a particular day about the stock was considered for the study. The news dataset used in this study for sentiment analysis with news headlines was extracted from the Kaggle community. We conducted experiments to evaluate the performance of the proposed method. To compare the performance of the model with state-of-the-art methods, we have taken three types of datasets, namely Honda Motor Company (HMC), Oracle Corporation (ORCL), and Intuit Inc. (INTU) data. The historical data was collected from Yahoo finance and evaluated under the same conditions. The detailed analysis and results are given in the following sections.

### Training and validation loss

A graph of the learning performance of the model with time is given by the learning curve. In machine learning, especially in deep learning neural networks, we can use learning curves as a diagnostic tool for problems that learn from training datasets incrementally. From the learning curve, we can evaluate the performance of the model on train and validation sets to know whether the model is underfit, overfit, or goodfit. The  $x$ -axis indicates the time or epochs, whereas the  $y$ -axis denotes the line plot of learning. The training loss shows how accurately the model fits the training data, and the validation loss shows how well the model fits new data. Loss measures the goodness of the model and the smaller the loss, the better the relationship of the input data and output and hence better our model. Training loss is measured during each epoch, while validation loss is measured after each epoch. If there is a minimal gap between the training loss and validation loss that decreases to the point of stability, we can say that our model is a good fit.

**Figure 6** Training and validation loss for DAE-LSTM model.

Full-size DOI: 10.7717/peerjcs.1158/fig-6

The training loss and validation loss are obtained after training the model for 100 epochs. [Figure 6](#) shows the training and validation loss for the DAE-LSTM model, and [Fig. 7](#) shows the training and validation loss for the DAE-GRU model. From the graphs, we can infer a minimal gap between the two final loss values, which shows that the model is a good fit.

[Figure 8](#) shows the training and validation loss for the DAE-LSTMSA model, and [Fig. 9](#) shows the training and validation loss for the DAE-GRUSA model. It is evident from the graphs that the loss is converging, which indicates that the model is a good fit.

## Evaluation metrics

We used different performance metrics described in the literature to assess the correctness of the stock price prediction ([Ding & Qin, 2020](#)). The metrics used in our research are the root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R-squared ( $R^2$ ) to measure the goodness of fit between the actual and predicted values. The metrics can be expressed as in Eqs. (15), (16), (17) and (18).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)| \quad (16)$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{(y_i - \hat{y}_i)}{y_i} \right| \quad (17)$$

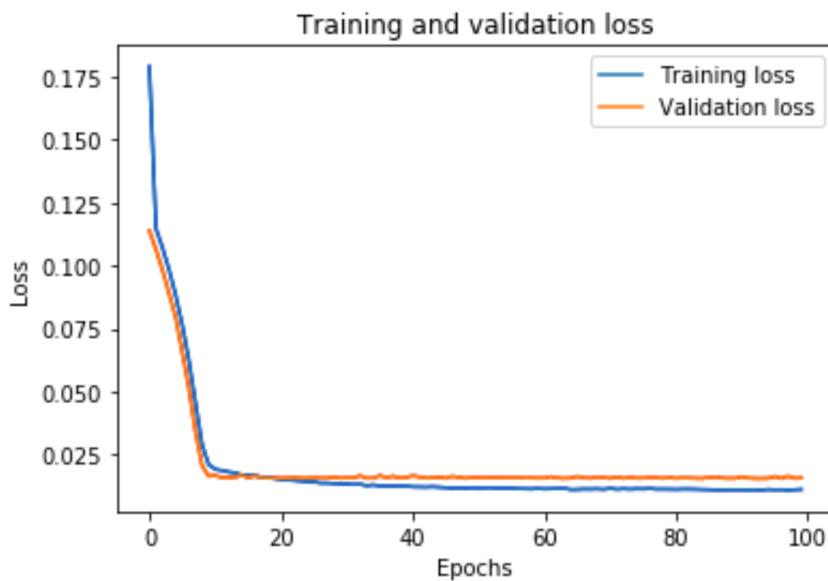


Figure 7 Training and validation loss for DAE-GRU model.

Full-size DOI: 10.7717/peerjcs.1158/fig-7

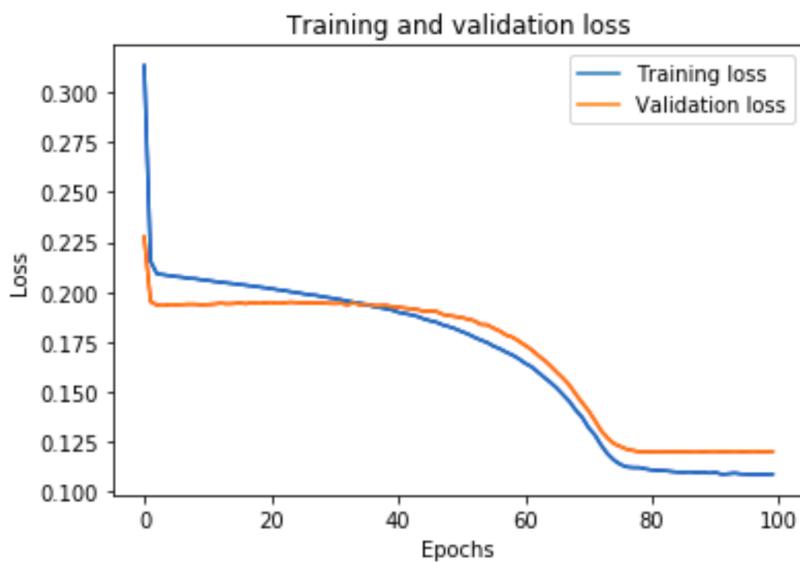
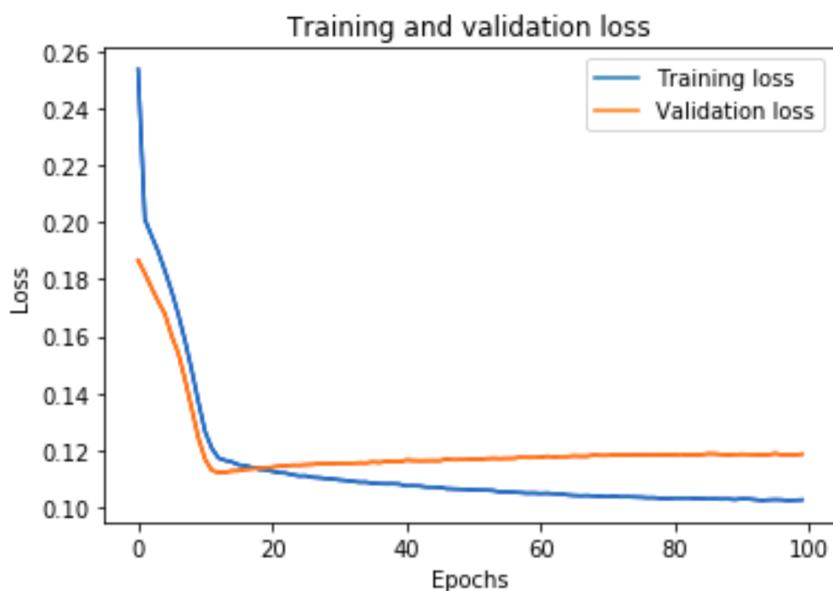


Figure 8 Training and validation loss for DAE-LSTMSA.

Full-size DOI: 10.7717/peerjcs.1158/fig-8

$$R^2 = 1 - \frac{(\sum_{i=1}^n (y_i - \hat{y}_i)^2)/n}{(\sum_{i=1}^n (\bar{y}_i - \hat{y}_i)^2)/n}. \quad (18)$$

Here n is the total number of samples,  $y_i$  and  $\hat{y}_i$ , indicates the actual value and predicted value of the test set; and the mean of real values of the test set is represented by  $\bar{y}_i$ . The lower the value of RMSE, MAE, and MAPE better the model. The range of  $R^2$  is normally

**Figure 9** Training and validation loss for DAE-GRUSA.

[Full-size](#) DOI: [10.7717/peerjcs.1158/fig-9](https://doi.org/10.7717/peerjcs.1158/fig-9)

between 0 and 1. The closer the value of  $R^2$  to 1 better the model's ability to fit the data. The evaluation metrics are represented in the bar graphs in Figs. 10, 11, 12 and 13.

### Performance comparison

In this section, we compare the LSTM and GRU models with and without deep autoencoder. To make the cooperative model consistent in this research, both LSTM and GRU have the same number of layers. The models are executed with an epoch value of 100 and batch size of 32 to evaluate the performance. Two input layers, followed by an LSTM/GRU layer, succeeded by a dropout layer, and finally, a dense output layer, constitutes the model structure. The hyperbolic tangent is the activation function used in each layer of LSTM/GRU. We are dividing the total data set comprising 4,101 samples into 2,748 training samples and 1,353 test samples. The two input sets used in the experiment are historic stock values and the news sentiment labels. We used these inputs to train the LSTM/GRU models separately.

Our study aims to measure the LSTM and GRU model's performance under different conditions, as shown in Tables 7 and 8. After passing the historical dataset through a deep autoencoder, it is observed that the GRU model with news sentiments performs better than the LSTM counterpart under the same cooperative environment. We can see that news sentiment has an influential impact on stock market prediction.

We have plotted the trading days on the X-axis and the close price on the Y-axis. The original test close price and predicted test close price are shown in the plots. By observing the test plots of the two models in Figs. 14 and 15, it is evident that GRU performs better than LSTM in the models DAE-LSTM and DAE-GRU. The lower the value of the evaluation metrics RMSE, MAPE, and MAE better is the model. An R2 value close to 1 indicates a better model.

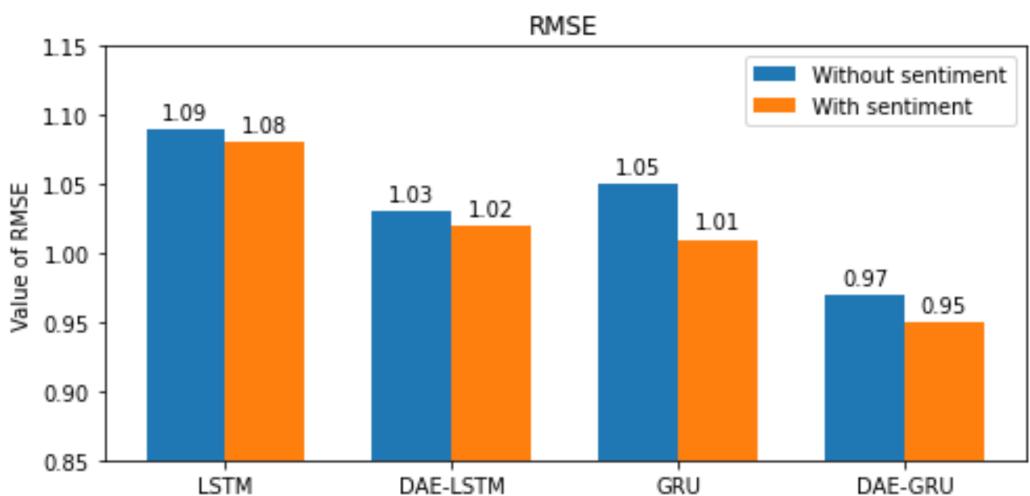


Figure 10 RMSE for each model.

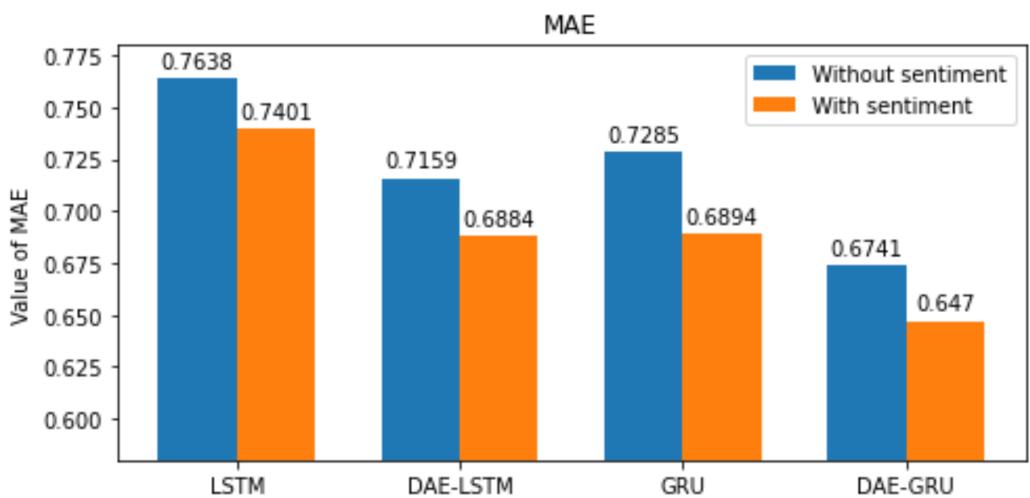
[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-10

Figure 11 MAE for each model.

[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-11

Under the same cooperative environment, the DAE-LSTMSA model performs better than the DAE-LSTM model, and the DAE-GRUSA model performs better than the DAE-GRU model. The performance is improved by using sentiment analysis with news headlines and historical stock datasets than considering historical stock prices alone while using the GRU model. DAE-LSTMSA and DAE-GRUSA models perform better than the DAE-LSTM and DAE-GRU models in performance metrics RMSE, MAPE, MAE, and R2. The test plots of Figs. 16 and 17 have a trading price on the X-axis and a close price on the Y-axis. For the models DAE-LSTMSA and DAE-GRUSA, we have plotted the original test close price and predicted test close price. From the test plots of the two models in Figs. 16

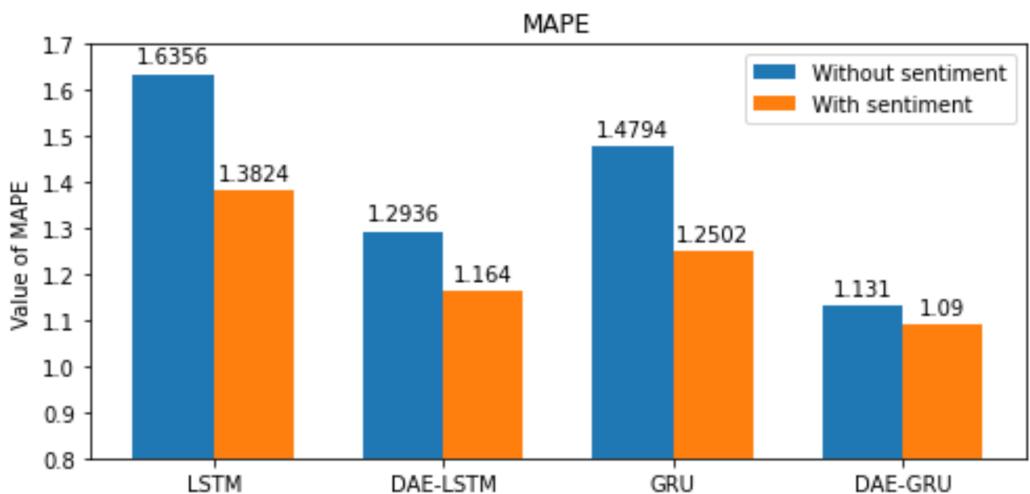


Figure 12 MAPE for each model.

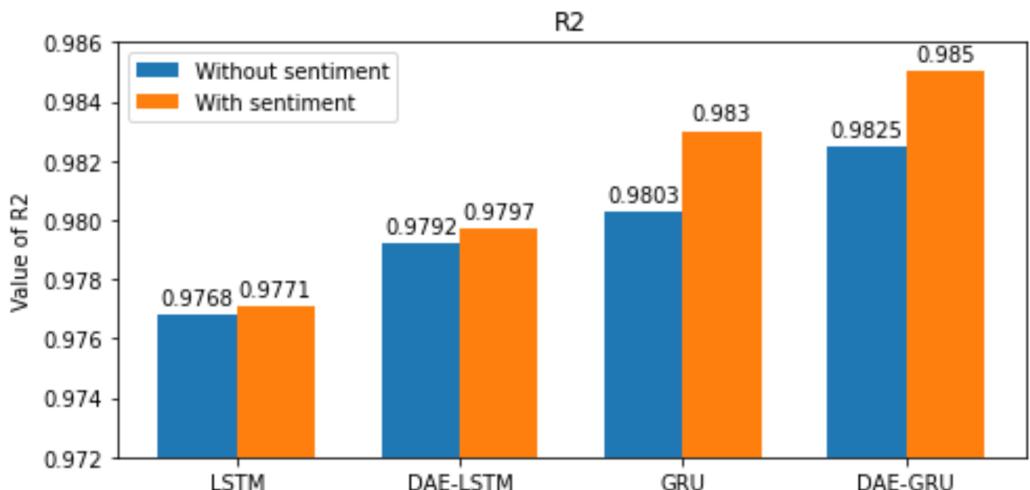
[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-12

Figure 13 R-squared for each model.

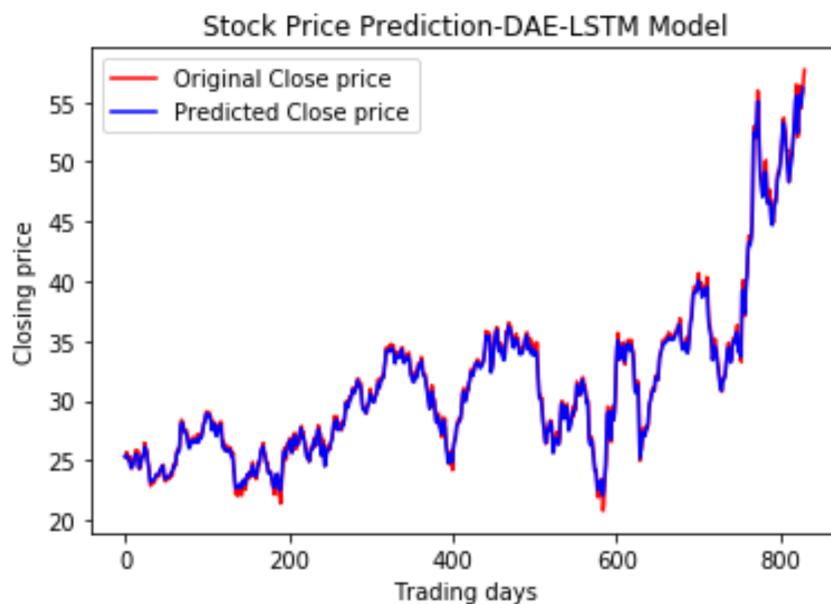
[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-13

and 17 and the data shown in the tables, it is clear that the DAE-GRUSA model provides better results than all the other models.

There are relatively few studies in the literature indicating the relationship between the stock price and news headlines. Our model introduces a new method by incorporating a deep autoencoder to denoise the historical stock data and then combine it with labeled sentiments. The results reveal that using news headlines significantly improves the performance of stock price prediction. The results indicate that deep learning models integrated with autoencoder and sentiment analysis are better equipped to predict stock prices. Financial markets are heavily affected by other market sentiments such as twitter

**Table 8** Summary of results of evaluation metrics with deep autoencoder.

Evaluation metrics	DAE-LSTM	DAE-LSTMSA	DAE-GRU	DAE-GRUSA
RMSE	1.03	1.02	0.97	0.95
MAE	0.7159	0.6884	0.6741	0.6470
MAPE	1.2936	1.1640	1.1310	1.09
R2	0.9792	0.9797	0.9825	0.9850

**Figure 14** Stock close price prediction DAE-LSTM model.

[Full-size](#) DOI: [10.7717/peerjcs.1158/fig-14](https://doi.org/10.7717/peerjcs.1158/fig-14)

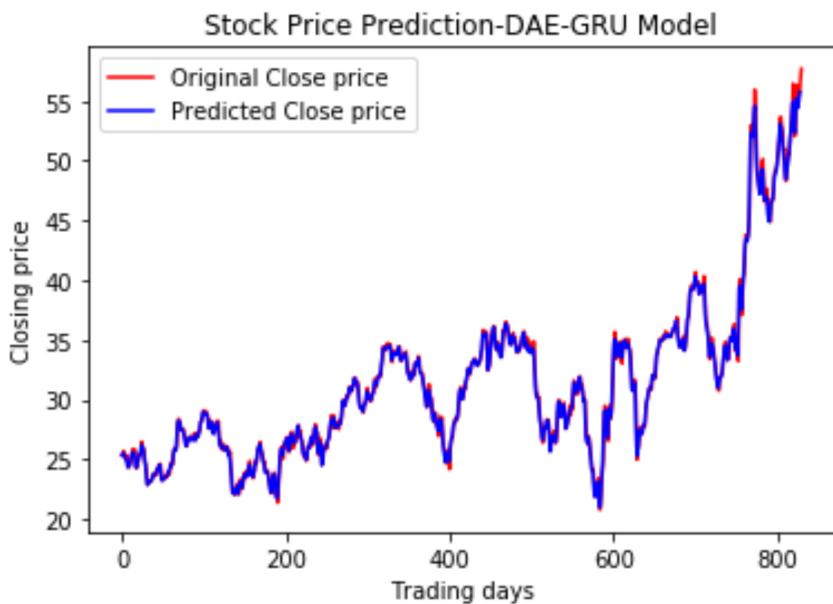
data and other external factors, which are not considered in this algorithm which is a shortcoming of this model.

### Comparison with state-of-the-art models

We have also compared the performance of our model with similar models in recent literature given in [Dey et al. \(2021\)](#) using the same data sets. The performance comparison is done based on three different stock data sets of HMC, ORCL, and INTU. The comparison is shown in [Tables 9, 10](#) and [11](#). It is observed that the proposed models perform better than the benchmark LSTM and GRU models.

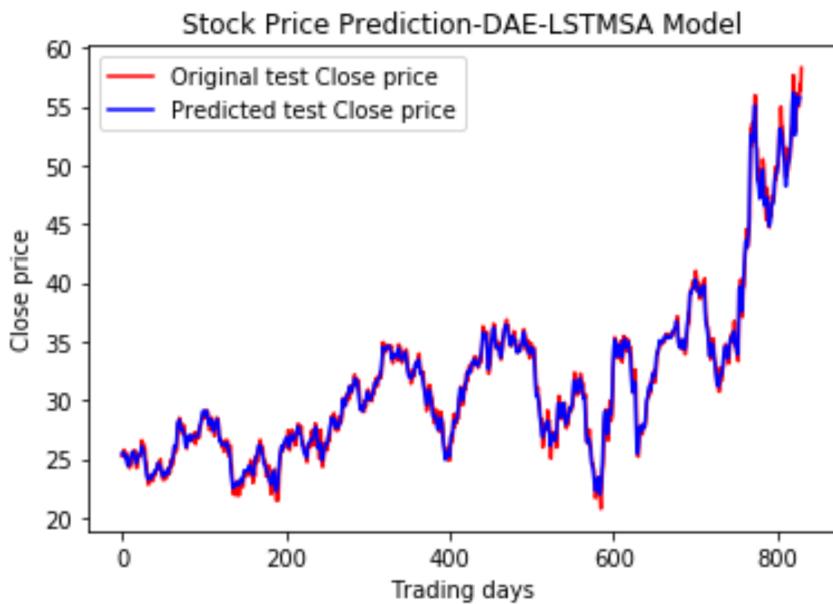
## CONCLUSION AND FUTURE SCOPE

A cooperative deep-learning architecture for stock market prediction using the deep autoencoder, LSTM, GRU, and sentiment analysis with news headlines was proposed in this work. The performance of our DAE-LSTMSA and DAE-GRUSA models was evaluated under the same conditions and compared with the benchmark DAE-LSTM and DAE-GRU models. Both models show lower error rates indicating better performance while using



**Figure 15** Stock close price prediction DAE-GRU model.

[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-15

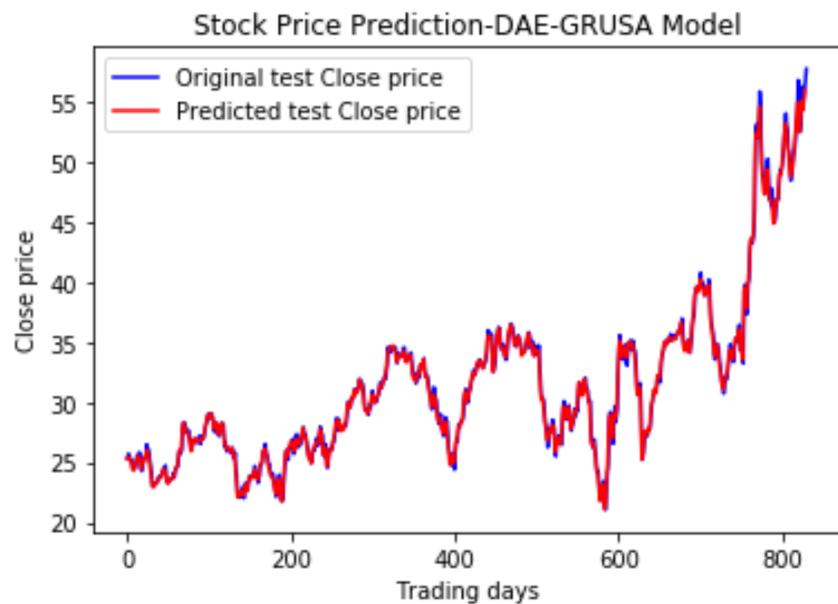


**Figure 16** Stock close price prediction DAE-LSTMSA model.

[Full-size](#) DOI: 10.7717/peerjcs.1158/fig-16

news headlines and the historical stock dataset. Results indicate that our models perform better than the state-of-the-art models in the literature.

We demonstrated that the DAE-GRUSA model provides better results than other models. We conclude that sentiment analysis with news significantly influences stock market prediction. This work is significant to the community because prediction problems

**Figure 17** Stock close price prediction DAE-GRUSA model.

[Full-size](#) DOI: [10.7717/peerjcs.1158/fig-17](https://doi.org/10.7717/peerjcs.1158/fig-17)

**Table 9** Comparison with state-of-the-art models: HMC stock.

Evaluation metrics	LSTM (State of the art)	DAE-LSTM (Proposed model)	GRU (State of the art)	DAE-GRU (Proposed model)
RMSE	0.7970	0.52	0.7811	0.37
MAE	0.5807	0.407	0.5695	0.2696
MAPE	2.1027	1.43	2.1001	0.9601
R2	0.9409	0.97	0.9606	0.9854

**Table 10** Comparison with state-of-the-art models: ORCL stock.

Evaluation metrics	LSTM (State of the art)	DAE-LSTM (Proposed model)	GRU (State of the art)	DAE-GRU (Proposed model)
RMSE	1.3969	1.13	1.4077	1.02
MAE	0.9862	0.9599	0.9996	0.8156
MAPE	1.9787	1.9605	2.0192	1.6297
R2	0.9015	0.9514	0.90	0.9604

involving dynamic stock data are challenging and have yet to reach maturity. Our proposed model performs reasonably well, but there is further scope for improvement by incorporating sentiment analysis using Twitter data and historical datasets. As a future scope, we can include Twitter data instead of using news sentiments alone. In addition, an advanced hyper-parameter selection may be used to optimize the proposed deep learning model. There is also scope for attempting the model on time series problems in other domains involving sequential data and sentiments.

**Table 11** Comparison with state-of-the-art models: INTU stock.

Evaluation metrics	LSTM (State of the art)	DAE-LSTM (Proposed model)	GRU (State of the art)	DAE-GRU (Proposed model)
RMSE	10.55	10.44	7.6649	7.46
MAE	7.8352	6.4967	4.9903	3.90
MAPE	3.4130	2.1205	2.3146	1.9945
R2	0.9664	0.9849	0.9822	0.9875

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

The authors received no funding for this work.

### Competing Interests

The authors declare there are no competing interests.

### Author Contributions

- KS Rekha conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- MK Sabu conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.

### Data Availability

The following information was supplied regarding data availability:

The data is available at Zenodo: Rekha KS. (2022). A cooperative deep learning model for stock market prediction using deep autoencoder and sentiment analysis (Version 1) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.7044026>.

## REFERENCES

- Albahli S, Nazir T, Mehmood A, Irtaza A, Alkhailah A, Albattah W. 2022.** AEI-DNET: a novel densenet model with an autoencoder for the stock market predictions using stock technical indicators. *Electronics* **11**(4):611 DOI [10.3390/electronics11040611](https://doi.org/10.3390/electronics11040611).
- Araújo RdA. 2010.** A quantum-inspired evolutionary hybrid intelligent approach for stock market prediction. *International Journal of Intelligent Computing and Cybernetics*.
- Atsalakis GS, Valavanis KP. 2009.** Surveying stock market forecasting techniques—Part II: soft computing methods. *Expert Systems with Applications* **36**(3):5932–5941 DOI [10.1016/j.eswa.2008.07.006](https://doi.org/10.1016/j.eswa.2008.07.006).
- Bao W, Yue J, Rao Y. 2017.** A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE* **12**(7):e0180944 DOI [10.1371/journal.pone.0180944](https://doi.org/10.1371/journal.pone.0180944).

- Bhardwaj A, Narayan Y, Dutta M.** 2015. Sentiment analysis for Indian stock market prediction using sensex and nifty. *Procedia Computer Science* **70**:85–91 DOI [10.1016/j.procs.2015.10.043](https://doi.org/10.1016/j.procs.2015.10.043).
- Bhonde R, Bhagwat B, Ingulkar S, Pande A.** 2015. Sentiment analysis based on dictionary approach. *International Journal of Emerging Engineering Research and Technology* **3**(1):51–55.
- Carosia , De Oliveira AE, Coelho GP, da Silva AEA.** 2021. Investment strategies applied to the Brazilian stock market: a methodology based on sentiment analysis with deep learning. *Expert Systems with Applications* **184**:115470 DOI [10.1016/j.eswa.2021.115470](https://doi.org/10.1016/j.eswa.2021.115470).
- Chen K, Zhou Y, Dai F.** 2015. A LSTM-based method for stock returns prediction: a case study of China stock market. In: *2015 IEEE international conference on big data (big data)*. Piscataway: IEEE, 2823–2824.
- Das N, Sadhukhan B, Chatterjee T, Chakrabarti S.** 2022. Effect of public sentiment on stock market movement prediction during the COVID-19 outbreak. *Social Network Analysis and Mining* **12**(1):1–22 DOI [10.1007/s13278-021-00834-z](https://doi.org/10.1007/s13278-021-00834-z).
- Deepak RS, Uday SI, Malathi D.** 2017. Machine learning approach in stock market prediction. *International Journal of Pure and Applied Mathematics* **115**(8):71–77.
- Deorukhkar OS, Lokhande SH, Nayak VR, Chougule AA.** 2019. Stock price prediction using combination of LSTM neural networks, ARIMA and sentiment analysis. *International Research Journal of Engineering and Technology* **6**(3):3497–3503.
- Devadoss AV, Ligori TAA.** 2013. Forecasting of stock prices using multi layer perceptron. *International Journal of Computing Algorithm* **2**:440–449.
- Dey P, Hossain E, Hossain MI, Chowdhury MA, Alam MS, Hossain MS, Andersson K.** 2021. Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains. *Algorithms* **14**(8):251 DOI [10.3390/a14080251](https://doi.org/10.3390/a14080251).
- Ding G, Qin L.** 2020. Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics* **11**(6):1307–1317 DOI [10.1007/s13042-019-01041-1](https://doi.org/10.1007/s13042-019-01041-1).
- Gite S, Khatavkar H, Kotecha K, Srivastava S, Maheshwari P, Pandey N.** 2021. Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Computer Science* **7**:e340 DOI [10.7717/peerj-cs.340](https://doi.org/10.7717/peerj-cs.340).
- Gunduz H.** 2021. An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination. *Financial Innovation* **7**(1):1–24 DOI [10.1186/s40854-020-00220-2](https://doi.org/10.1186/s40854-020-00220-2).
- Guresen E, Kayakutlu G, Daim TU.** 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications* **38**(8):10389–10397 DOI [10.1016/j.eswa.2011.02.068](https://doi.org/10.1016/j.eswa.2011.02.068).
- Haider Bangyal W, Hameed A, Ahmad J, Nisar K, Haque MR, Ibrahim A, Asri A, Rodrigues JJ, Khan MA, B Rawat D.** 2022. New modified controlled bat algorithm for numerical optimization problem. *Computers, Materials & Continua* **70**(2):2241–2259 DOI [10.32604/cmc.2022.017789](https://doi.org/10.32604/cmc.2022.017789).

- Hailong Z, Wenyan G, Bo J.** 2014. Machine learning and lexicon based methods for sentiment classification: a survey. In: *2014 11th web information system and application conference*. Piscataway: IEEE, 262–265.
- Hochreiter S.** 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(02):107–116 DOI [10.1142/S0218488598000094](https://doi.org/10.1142/S0218488598000094).
- Hutto C, Gilbert E.** 2014. Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: *Proceedings of the international AAAI conference on web and social media, volume 8*.
- Jiawei X, Murata T.** 2019. Stock market trend prediction with sentiment analysis based on LSTM neural network. In: *International multiconference of engineers and computer scientists*. 475–479.
- Khan W, Malik U, Ghazanfar MA, Azam MA, Alyoubi KH, Alfakeeh AS.** 2019. Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing* 1–25.
- Kumar D, Sarangi PK, Verma R.** 2021. A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings*.
- Kurani A, Doshi P, Vakharia A, Shah M.** 2021. A comprehensive comparative study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on stock forecasting. *Annals of Data Science* 1–26.
- Le Roux N, Bengio Y.** 2010. Deep belief networks are compact universal approximators. *Neural Computation* **22**(8):2192–2207 DOI [10.1162/neco.2010.08-09-1081](https://doi.org/10.1162/neco.2010.08-09-1081).
- Li AW, Bastos GS.** 2020. Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE Access* **8**:185232–185242 DOI [10.1109/ACCESS.2020.3030226](https://doi.org/10.1109/ACCESS.2020.3030226).
- Li J, Liu G, Yeung HWF, Yin J, Chung YY, Chen X.** 2017. A novel stacked denoising autoencoder with swarm intelligence optimization for stock index prediction. In: *2017 International joint conference on neural networks (IJCNN)*. Piscataway: IEEE, 1956–1961.
- Liu J, Chao F, Lin Y-C, Lin C-M.** 2019. Stock prices prediction using deep learning models. ArXiv preprint. [arXiv:1909.12227](https://arxiv.org/abs/1909.12227).
- Liu P, Zheng P, Chen Z.** 2019. Deep learning with stacked denoising auto-encoder for short-term electric load forecasting. *Energies* **12**(12):2445 DOI [10.3390/en12122445](https://doi.org/10.3390/en12122445).
- Lv D, Huang Z, Li M, Xiang Y.** 2019. Selection of the optimal trading model for stock investment in different industries. *PLOS ONE* **14**(2):e0212137 DOI [10.1371/journal.pone.0212137](https://doi.org/10.1371/journal.pone.0212137).
- Mehta P, Pandya S.** 2020. A review on sentiment analysis methodologies, practices and applications. *International Journal of Scientific and Technology Research* **9**(2):601–609.
- Mehta P, Pandya S, Kotecha K.** 2021. Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science* **7**:e476 DOI [10.7717/peerj-cs.476](https://doi.org/10.7717/peerj-cs.476).
- Moghar A, Hamiche M.** 2020. Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science* **170**:1168–1173 DOI [10.1016/j.procs.2020.03.049](https://doi.org/10.1016/j.procs.2020.03.049).

- Mohanty D, Parida AK, Khuntia SS.** 2021. Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine. *Applied Soft Computing* **99**:106898 DOI [10.1016/j.asoc.2020.106898](https://doi.org/10.1016/j.asoc.2020.106898).
- Nabipour M, Nayyeri P, Jabani H, Mosavi A, Salwana E.** 2020. Deep learning for stock market prediction. *Entropy* **22**(8):840 DOI [10.3390/e22080840](https://doi.org/10.3390/e22080840).
- Nelson DM, Pereira AC, de Oliveira RA.** 2017. Stock market's price movement prediction with LSTM neural networks. In: *2017 International joint conference on neural networks (IJCNN)*. Piscataway: IEEE, 1419–1426.
- Nemes L, Kiss A.** 2021. Prediction of stock values changes using sentiment analysis of stock news headlines. *Journal of Information and Telecommunication* **5**(3):375–394 DOI [10.1080/24751839.2021.1874252](https://doi.org/10.1080/24751839.2021.1874252).
- Pan H, Tilakaratne C, Yearwood J.** 2005. Predicting Australian stock market index using neural networks exploiting dynamical swings and intermarket influences. *Journal of Research and Practice in Information Technology* **37**(1):43–55.
- Parmar I, Agarwal N, Saxena S, Arora R, Gupta S, Dhiman H, Chouhan L.** 2018. Stock market prediction using Machine Learning. In: *2018 First international conference on secure cyber computing and communication (ICSCCC)*. Piscataway: IEEE, 574–576.
- Pawar K, Jalem RS, Tiwari V.** 2019. Stock market price prediction using LSTM RNN. In: *Emerging trends in expert applications and security*. Cham: Springer, 493–503.
- Pervaiz S, Ul-Qayyum Z, Bangyal WH, Gao L, Ahmad J.** 2021. A systematic literature review on particle swarm optimization techniques for medical diseases detection. *Computational and Mathematical Methods in Medicine* **2021**.
- Rahman MO, Hossain MS, Junaid T-S, Forhad MSA, Hossen MK.** 2019. Predicting prices of stock market using gated recurrent units (GRUs) neural networks. *International Journal of Computer Science and Network Security* **19**(1):213–222.
- Rajakumari KE, Kalyan MS, Bhaskar MV.** 2020. Forward forecast of stock price using LSTM machine learning algorithm. *International Journal of Computer Theory and Engineering* **12**(3).
- Roondiwala M, Patel H, Varma S.** 2017. Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)* **6**(4):1754–1756.
- Sagheer A, Kotb M.** 2019. Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Scientific Reports* **9**(1):1–16 DOI [10.1038/s41598-018-37186-2](https://doi.org/10.1038/s41598-018-37186-2).
- Seki K, Ikuta Y, Matsubayashi Y.** 2022. News-based business sentiment and its properties as an economic index. *Information Processing & Management* **59**(2):102795 DOI [10.1016/j.ipm.2021.102795](https://doi.org/10.1016/j.ipm.2021.102795).
- Seng J-L, Yang H-F.** 2017. The association between stock price volatility and financial news—a sentiment analysis approach. *Kybernetes*.
- Shahi TB, Shrestha A, Neupane A, Guo W.** 2020. Stock price forecasting with deep learning: a comparative study. *Mathematics* **8**(9):1441 DOI [10.3390/math8091441](https://doi.org/10.3390/math8091441).
- Shi C, Zhuang X.** 2019. A study concerning soft computing approaches for stock price forecasting. *Axioms* **8**(4):116 DOI [10.3390/axioms8040116](https://doi.org/10.3390/axioms8040116).

- Singh R, Srivastava S.** 2017. Stock prediction using deep learning. *Multimedia Tools and Applications* **76**(18):18569–18584 DOI [10.1007/s11042-016-4159-7](https://doi.org/10.1007/s11042-016-4159-7).
- Skansi S.** 2018. *Introduction to deep learning: from logical calculus to artificial intelligence*. Springer.
- Soleymani F, Paquet E.** 2020. Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoderDeepBreath. *Expert Systems with Applications* **156**:113456 DOI [10.1016/j.eswa.2020.113456](https://doi.org/10.1016/j.eswa.2020.113456).
- Song Q, Zheng Y-J, Yang J, Huang Y-J, Sheng W-G, Chen S-Y.** 2022. Predicting demands of COVID-19 prevention and control materials via co-evolutionary transfer learning. *IEEE Transactions on Cybernetics*.
- Strader TJ, Rozicki JJ, Root TH, Huang Y-HJ.** 2020. Machine learning stock market prediction studies: review and research directions. *Journal of International Technology and Information Management* **28**(4):63–83.
- Sutskever I, Hinton GE.** 2008. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation* **20**(11):2629–2636 DOI [10.1162/neco.2008.12-07-661](https://doi.org/10.1162/neco.2008.12-07-661).
- Takeuchi L, Lee Y-YA.** 2013. Applying deep learning to enhance momentum trading strategies in stocks. Technical Report. Stanford University.
- Vui CS, Soon GK, On CK, Alfred R, Anthony P.** 2013. A review of stock market prediction with Artificial neural network (ANN). In: *2013 IEEE international conference on control system, computing and engineering*. Piscataway: IEEE, 477–482.
- Wang Y, Liu Y, Wang M, Liu R.** 2018. Lstm model optimization on stock price forecasting. In: *2018 17th international symposium on distributed computing and applications for business engineering and science (dcabes)*. 173–177.
- Wu S, Liu Y, Zou Z, Weng T-H.** 2022. S\_I\_LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science* **34**(1):44–62 DOI [10.1080/09540091.2021.1940101](https://doi.org/10.1080/09540091.2021.1940101).
- Yadav A, Jha C, Sharan A, Vaish V.** 2019. Sentiment analysis of financial news using unsupervised and supervised approach. In: *International conference on pattern recognition and machine intelligence*. Cham: Springer, 311–319.
- Yu P, Yan X.** 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications* **32**(6):1609–1628 DOI [10.1007/s00521-019-04212-x](https://doi.org/10.1007/s00521-019-04212-x).
- Zheng Y-J, Yu S-L, Song Q, Huang Y-J, Sheng W-G, Chen S-Y.** 2021. Co-evolutionary fuzzy deep transfer learning for disaster relief demand forecasting. *IEEE Transactions on Emerging Topics in Computing* **10**(3):1361–1373.