Multimedia Tools and Applications (2024) 83:57269–57289
https://doi.org/10.1007/s11042-023-17130-x

An improved technique for stock price prediction on real-time exploiting stream processing and deep learning

Kailash Chandra Bandhu[1] · Ratnesh Litoriya[1] · Anshita Jain[1] · Anand Vardhan Shukla[1] · Swati Vaidya[1]

Received: 16 April 2023 / Revised: 13 July 2023 / Accepted: 15 September 2023 / Published online: 18 December 2023

Abstract

The proposed model is a Deep Learning (DL) based method employing Long Short-Term Memory (LSTM) networks for forecasting stocks. The aim of this approach is forecasting stock prices of Apple Inc. using statistics on previous stock prices obtained from Tiingo. The proposed model consists of several stages of processing and modelling, including data cleaning, feature selection, feature scaling, model building, model evaluation, model improvement, and prediction. Cleaning, organising, and transforming raw data into a format appropriate for analysis are all parts of data pre-processing. Feature engineering involves the data extraction and selection of relevant features for accuracy improvement of the model. The scaling of features involves normalising the data to prevent bias in the model. The LSTM models are built and evaluated using multiple metrics such as Mean Squared Error (MAE) and Root Mean Squared Error (RMSE). The model is iteratively improved using a combination of hyperparameter tuning and feature engineering. Finally, the model is then used to forecast stock prices for the following 30 days, and the accuracy of the forecasts is determined. The proposed methodology is designed to outperform traditional LSTM models for predicting the future price of stock by incorporating novel techniques, for feature engineering and model refinement. The suggested design is a comprehensive approach for forecasting future stock prices using DL based techniques. The model is designed to be flexible and adaptable, allowing for customization for different datasets and prediction horizons. It represents a significant improvement over existing LSTM models for stock price prediction to be valuable in a variety of financial industry applications. This paper collects data from Tiingo API and uses stacked LSTM to train the model. The experimental results give only 0.0813 RMSE, which proves that the model is more accurate and precise.

Keywords Long short-term memory (LSTM) · Deep learning (DL) · Root mean squared error (RMSE) · Mean squared error (MSE) · Stock price

Extended author information available on the last page of the article

1 Introduction

In the financial sector, stock price forecasting has long been a hot issue. Machine learning models have developed as a method that has promise for forecasting stock values as a result of technological improvement and the accessibility of large volumes of data. In the proposed study, the aim is to create a machine learning system for future stock price prediction of a certain firm, in this case American Association of Professional Landmen (AAPL), end to end.

The proposed system utilizes stacked Long Short-Term Memory (LSTM) to forecast future stock prices of AAPL. Due to its capacity to recognise long-term relationships in the data, it is a form of Recurrent Neural Network (RNN) that has been extensively used in time-series prediction applications. Stacked LSTMs are a variation of LSTM that involves stacks of multiple layers of LSTM to improve the model's efficiency.

The system will take data from the Tiingo Application Programming Interface (API), a financial data platform that provides real-time and historical data for stocks, currencies, and other financial instruments. Tiingo API offers a rich set of features and data that could be used for developing predictive models for stock prices. The data will be pre-processed to remove any missing values and to normalize the features.

There will be training and testing sets created from the pre-processed data. The testing set will be utilised to assess the efficiency of the stacked LSTM model once it has completed the training. A Mean Squared Error (MSE) loss function will be used for the model's training, and the Adam optimizer will be used for its optimisation.

The model will be used to forecast AAPL stock values once it has been trained completely. To figure out the model's accuracy, the predicted and actual values will be compared. Metrics like Root Mean Squared Error (RMSE) will be utilized to calculate the accuracy.

The proposed system has several potential applications in the financial industry. It can be used by financial institutions to forecast the stock values of companies and to mitigate risks associated with investments.

In summary, this article aims to create a machine learning mechanism that can predict the stock prices of AAPL using stacked LSTM neural networks. The system will use data from the Tiingo API, and Metrics like MAE and RMSE will be used to gauge the accuracy of the model. The project will also explore different techniques to improve the model's performance, such as hyperparameter tuning, feature selection, and ensemble learning. The proposed system has the potential to be used in the financial industry to make informed investment decisions and to mitigate risks associated with investments.

2 Literature survey

This section presented the survey of various exiting articles relevant to stock prediction in real time using deep learning techniques for improvement of stock prediction in real time.

This theory finds the connection between Autoregressive Integrated Moving Average (ARIMA) and hot winter. These are the time series algorithms. It has examined either, to enhance the efficacy of the model by analysing both of them. The benefit of this calculation is that it uses historical prices of stocks for seasonal and non-seasonal data.

However, this model has limitations, such as not accounting for new marketing strategies or media reports that may affect stock prices [1].

This work examines the use of Machine Learning (ML) [2] techniques based on LSTM and regression to forecast stock prices. Measurements are made for open, close, low, high, and volume. This study set out to improve forecasting of future stock values for a corporation by using machine learning techniques. The LSTM algorithm produced beneficial results with improved accuracy in stock value prediction when compared to more current models utilising a similar technique [3]. Zahra & Nargis combined LSTM and CNN methods to fundamentally analyse stock price trend prediction [4].

Techniques like Support Vector Machine (SVM) and neural networks are developed in it. For constructing a model for foreseeing the trend in stock prices, multiple categories of news and the events are used as features. The relationship between specific multi-category news and changes in stock price was also examined using both SVM and neural network models. Use of Blockchain Technology for secure prediction in many different domains are also widely explored [5–7]. According to testing results, the already defined news events are more superior to the traditional feature at forecasting stock price trends. This study found that short-term prediction outperformed long-term prediction [8]. Prediction of high variations in stock prices using LSTM for the stocks of 2020 has been carried out by Bathla et al. [9].

In their study, they presented a method of predicting share prices using LSTM and RNN to anticipate stock value on the National Stock Exchange (NSE) India. In order to compare the actual data with the predictions, an RNN graph is used, the effectiveness of this model has been examined [10]. By adjusting the setup appropriately, our RNN-based architecture demonstrated excellent stock price predicting capabilities. In order to prevent data mixing, a backpropagation strategy can be used while gathering and organising data. The model is trained for all the NSE data which can be taken from the internet and used as input, groups them, to provide input in accordance with the preference of the user [11].

The Karachi Stock Exchange data on day closing was used in the suggested work along with machine learning techniques. The Machine Learning (ML) methods of Radial Basis Function (RBF), Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), and SVM are examined. The feature that made predicting the market the most advantageous was the oil rate component. The study's results show that stock market performance may be predicted with accuracy using ML techniques. The MLP approach of ML estimates market performance with a 70% of correctness. Chi Ma [12] performed stock linkage prediction based on optimized LSTM model.

The suggested study has used the LSTM, to predict AAPL stocks rate by ingesting single and multi-feature input variables, for verifying forecasting effect on historical data of stock. This experiment resulted in higher accuracy and more consistency with the real demand and with a high efficacy of about 0.033 for the multivariate input. The single-feature input has a worse estimated squared absolute error (0.155) than the multi-feature variable input [13].

In addition to keeping up with past market data over time, this study also employs a neural network to forecast future stock prices. The Convolutional Recurrent Neural Network (CRNN), which has been proposed as its key parts are the neural network architecture with memory performance and the LSTM, which enhances the long-term dependency of conventional RNN. Additionally, the accuracy and stability of predictions are increased by using the RNN LSTM architecture. The objective of this study is examining and obtaining an Average Error Rate (AER) of 3.449 RMSE by collecting historical data for 10 different stocks [14].

The major focus of this work is to assess a strategy for forecasting the next-day market that combines RNNs with meaningful input data. LSTM and analysing the stock prediction model is done using stock-basic trade data. Standard & Poor's 500 Index (S&P500) and National Association of Securities Dealers Automated Quotations (NASDAQ) are used in the case study. Their forecasting approach, which performs better than the comparison models, more accurately foretells the price of stocks for the given days on which the stock was closed. The key finding of this work is the improved accuracy of the given approach [15].

The comparative analysis of the proposed study is concentrated on the three methods: Multiple Linear Regression (MLR), SVM, and Artificial Neural Network (ANN), to determine the market price for the given days using both daily and monthly predictions. The most accurate sentiment analysis prediction system forecasts stock price. The less sophisticated method is the MLR technique, which analyses the relationship between volume and stock price. Various theories prove that DL algorithms are slightly more complex than MLR and SVM algorithms [16].

According to the paper, a time series analytical technique is described and the changing price of stock is modelled using LSTM. This modelling infrastructure incorporates to aid with LSTM time series prediction, calculating the transfer of entropy between relevant variables, guaranteeing the correctness of the assumption result to a large extent. Modelled and actual stock prices are strongly connected, although their Mean Absolute Error (MAE) and RMSE, which are examined by the results, differ slightly [17].

This essay examines a case study and offers a preventative measure for stock market crashes like the one that hit Bangladesh's in the fiscal year 2010–2011. As it is very volatile, it is particularly a risky area to be invested in. This may become a crucial aspect for making profit and maybe become very risky and less susceptible for those who invest in this, it is beneficial to those who can appropriate it at the right moment [18].

In this work, the idea of stock vectors is used which are based on the progress of the DL word vector. Historical data with several stocks and high dimensions is the input. The method used in this article predicts the stock market using LSTM with embedded layers and LSTM neural networks with automated encoders. In this case, the input is vectorized using an embedded layer and an automated encoder in an effort to forecast the stock using LSTM. Thus, obtained accuracy in either case is 57.2% and 56.9%, while for individual stocks, it is 52.4% and 52.5% [19].

This study compares the productivity of LSTM networks and random forest, more specifically CUDA Deep Neural Network LSTM (CuDNN LSTM) as methods of training for predicting the directional motions of S&P 500 component equities for intraday trading from January 1993 to December 2018.The multi feature approach outperforms the single-feature design (for LSTM), with daily returns of 0.54% for random forests and 0.64%. Both methods achieve similar daily returns of 0.41% and 0.39% for the single-feature design. It concludes that LSTM networks are more effective than random forests in intraday trading [20].

This study presents a computerised trading system that incorporates machine learning, sentiment analysis, and mathematical methods in order to enhance stock forecast accuracy and carry out lucrative transactions. The main objective was to predict a stock's price or trend based on morning trading for the day's final trading hours. This was accomplished by building and training a large number of DL models while also considering the significance of important news. The SVM for AAPL stock performed the majority of the tests with the greatest accuracy 82.91% [21].

A higher connection between equities prices in various European nations is expected to be the outcome of the growing integration of European financial markets. If the movement of the stock market has an impact on actual economic factors like consumption and investment, it results in decline in economic development across the European nations. As a result, the vector autoregressive models predict a positive relationship between investment and changes in share prices. Thus, the impact of monetary policy on share prices and the business cycle should be monitored by monetary authorities [22].

This study proposes the use of external common-sense knowledge to better understand the intention and mood of the stock market. Three event-related tasks, including event similarity, are used in the experiment to significantly improve the event embeddings and achieve a 78% improvement on the hard similarity task. This leads to inferences more accurately on upcoming activities and improves accuracy in foretelling the stock market volatilities. Despite the efficient market theory, which suggests that stock prices cannot be forecasted and act randomly, technology advancements have made a vast amount of data accessible. As a result, developing an adequate prediction system that can boost revenues for traders or investment businesses has become easier [23].

The purpose of the work was modelling and forecasting stock's next day price using Artificial Intelligence (AI). The Autoregressive Moving Average (ARMA) approach is used to benchmark the AI algorithms. This experiment is conducted using data collected from the Johannesburg Stock Exchange. Historical data of closing prices from the past years were used as raw data, and the high accuracy of predicting the index price as results. These methods demonstrate the capacity to anticipate future prices. The findings demonstrate that the accuracy metric chosen determines how well the SVM and MLP network functions [24].

This paper explains how to predict a stock using ML. The majority of stock brokers base their stock predictions on time series analysis. In this work, a ML strategy is suggested, this will teach how to acquire information by utilising stock market data before applying the learned information to make precise predictions. In this respect, this study employs prices with both daily and instantaneous frequencies and a ML approach called SVM to anticipate the prices for large and small companies across 3 separate marketplaces [25].

The challenge of analysing unstructured data was addressed in this paper, which also focused attention on the financial sector. Using the Data Collection, Analyzation, and Visualization in Real-time Stock (DAViS) prototype model, to find the price of stocks for the following day, this work suggests an interpretable ensemble stacking of diverse ML based estimators together with Principal Component Analysis (PCA) and Ward hierarchical features. To extract relevant information like sentiment, informativeness, and important words from texts, a topic modelling-based method is used in conjunction with textual analysis [26].

In this work, there are various techniques which are investigated to predict the stock price on the NASDAQ and NSE. These techniques are incremental and based upon the various online offline learning modes. The ML techniques used here are trained on recent stock data, and the Historical data of stocks is continuously updating from the real time market rates, so that the models can fine-tune with the changes that took place in the stock's time series during trading events. This work chose the Exponential Moving Average (EMA) and Volume-Weighted Average Price (VWAP) as characteristics to take into account After carefully evaluating many technical indications that help in better price prediction, the stock price is used, along with other variables, to build an effective multivariate time-series dataset. The efficacy of it in forecasting stock prices was demonstrated by the fact that all models performed better on multivariate time-series data [27].

In order to hourly and monthly forecast the market trend, a non-linear SVM classification algorithm and a stock technical and extraction of textual features have all been developed. This algorithm uses social media comments and real time stock rate technical data to forecast each stock's patterns in order to solve a categorization issue. On the stock text comments and technical data, the text pre-processing and data pre-processor activities are carried out. For stock trend prediction, new characteristics are taken from the comments and technical data. Finally, the proposed non-linear SVM classifier predicts each stock trend over a range of time frames [28].

In this study, the majority of stockbrokers used a time arrangement inspection or a specific analytical process to anticipate stock prices. Python is the computer language used to forecast the stock price using ML. Several methods are used by machine learning itself to make estimates more straightforward and accurate. Foreseeing future stock values, this inquiry uses a machine learning technique known as LSTM. The main objective of this model is predicting and identifying stock price instabilities using the recent sixty-day closing prices [29].

This study uses ANN based stock price judgement to produce more precise prediction outcomes. The experiment's findings depicts that the model suggested in this work can effectively predict upcoming stock and run price trends quicker, substantially resolving the shortcomings of classic forecasting approaches of sluggish poor prediction accuracy and operational effectiveness [30].

The LSTM algorithm is suggested in this research and is based on DL. For the NIFTY 50 index, a historical dataset (spanning 10 years) was used from December 10, 2011, through December 10, 2021. Following normalisation, this dataset is utilised for developing and testing models. With an accuracy of 83.88 percent, the recommended model's results are fairly positive [31].

This study compares four different methods for forecasting stocks using multiple variables: Multiple Linear Regression, Exponentially Weighted Moving Average (EWMA), Extreme Gradient Boosting (XGBoost), and LSTM. The data used for the study was collected between January 1 and December 31, 2021. The results show that multiple linear regression outperforms XGBoost and LSTM when using a smaller amount of data, by utilizing 18 elements that include both everyday market aspects and technical factors to jointly forecast the stock market's closing values. In this context, XGBoost and LSTM are unable to fully exploit their benefits. The study explores the potential use of different models for predicting stock closing prices and provides insight into related research on multi-model prediction [32]. Fuzzy logic is also well utilized in MCDM and prediction purposes [33–39].

The present study project is inspired by how often the stock market data values change. Via a comparison analysis report, the newly built machine learning classification strategy, performance enhancement is examined, ensuring the researched method's precise prediction. A categorization system powered by machine learning has been used to forecast stock market prices and movement changes. This performance may be assessed, and the user can also be advised to proceed with the suggested tasks. This allows the user to quickly determine which stocks will remain in the market for a longer period of time. Using ML methods, the efficacy of stock exchange was examined and hiked to 94.17%. As a consequence, the investor will find it useful to evaluate the present value and project the company's stock rates in the future [40].

Here, a hybrid model for stock price forecasting is built using a variety of DL and ML models. It is clearly visible from the outcomes that the LSTM is the most precise algorithm based univariate model, it employs data from the past week to predict the closing value of Reliance Industries Ltd. for the following week [41].

This study describes the complex analytic methods, such as the neural network, that are being employed by academics across a range of disciplines. It suggests deep learning methods for predicting the price of Google's shares. RNN and LSTM variations were applied to a dataset from Kaggle using deep learning techniques. This led to the Bidirectional LSTM achieving greater outcomes. Using the Christ University Python FLASK, also developed a web application for stock prediction [42].

The literature on stock price prediction has explored various ML methods, such as regression analysis, DT, and NN. The use of DL techniques, such as the LSTM, has shown promising results in recent years, but the PCA-LSTM model, that combines the LSTM model with the PCA technique, has been found to improve accuracy and efficiency in stock price prediction. Researchers have used the PCA-LSTM model to forecast stock values across several financial markets and have achieved better results than traditional methods. This work proposed a PCA LSTM model for stock price prediction in the China Securities Index (CSI) 300 and achieved promising results, indicating its potential for decision-making in the financial industry [43].

The literature survey highlights that DL techniques, particularly the LSTM networks, have shown great potential in predicting stock prices. Several papers have suggested DL based approaches in order to predict stock prices based on past stock price data. These approaches have been shown to outperform traditional statistical models for stock price prediction. Additionally, the literature also highlights the use of various ML algorithms, including ANNs, for stock price prediction. Overall, the literature suggests that DL is the most effective technique with LSTM, in predicting stock prices and can provide valuable insights for investors, traders, and analysts.

2.1 Proposed methodology

The proposed endeavour entails the development of an ML project aimed at estimating stock prices using stacked LSTM. The dataset will be sourced from the Tiingo API for AAPL stock. The data will undergo meticulous pre-processing, encompassing filtering and transformation steps to yield a structured format amenable to analysis. Subsequently, the pre-processed data will be partitioned into distinct training and testing sets. The stacked LSTM model will undergo training to discern intricate features and patterns inherent in stock price dynamics. The efficacy of the system will be assessed using evaluation metrics such as MSE, RMSE, MAE, and R-squared.

The trained model will be deployed to forecast future prices of AAPL shares. A comparative analysis will be conducted, pitting the predictions of our proposed model against those of existing LSTM models tailored for stock price prediction. Continuous refinement of the proposed model will be pursued to bolster its performance. The primary objective of this research endeavour is to create a refined and robust model for predicting future stock prices.

a) **System architecture**

The system architecture as shown in Fig 1, includes an API to collect the live data, feature selection, splitting the data in order to train and test dataset and stacked LSTM.

The steps involved in of the process are:

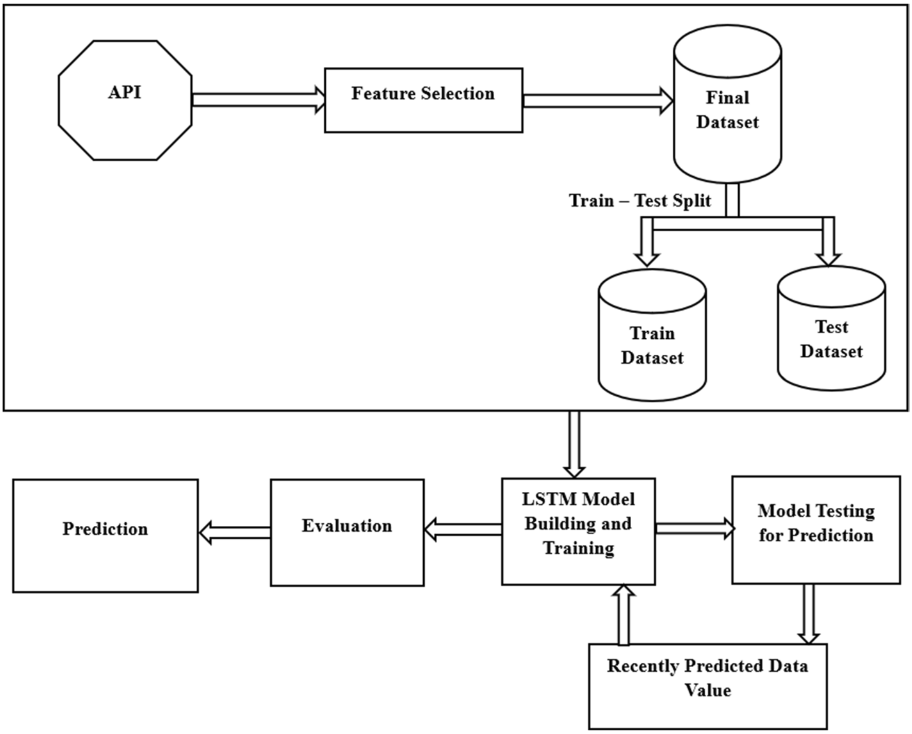


Fig. 1 System Architecture

i) Collecting stock price dataset: Collecting real time stock price data from Tiingo API using Python Tiingo library.
ii) Filtration of dataset and Feature selection: Pre-processing the collected data and performing feature engineering using Python pandas and numpy.
iii) Data Splitting: Once feature selection is completed, the pre-processed data is sliced into training, validation, and testing datasets using the scikit learn library.
iv) LSTM Model: An LSTM model is built using Python Keras library and trained on the training set.
v) Evaluation: Model is evaluated on the validation dataset. If necessary, the model is fine-tuned based on the performance on the validation dataset.
vi) Forecasting: The obtained best fitting model is then used to predict the stock prices.

b) **System Flow Chart**

Figure 2 Depicts the flow chart which describes the flow of data through the system. Steps involved:

i) Data Collection via API: This step involves obtaining real time stock price data from an API, such as Tiingo.

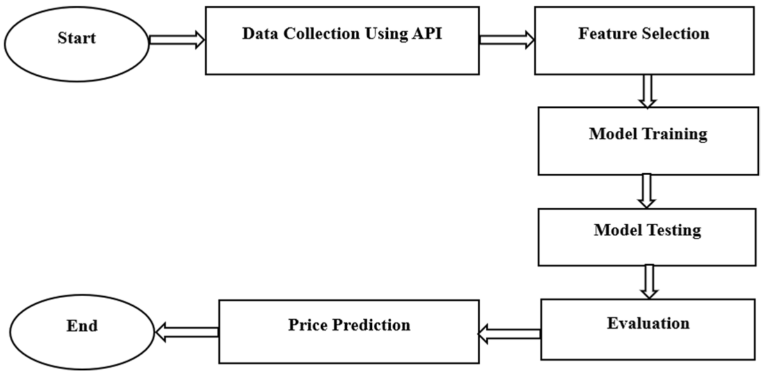


Fig. 2

ii) Feature Extraction: Here, relevant features are extracted from the pre-processed data, such as closing price, volume, and moving averages.
iii) Train Model: This step involves feature extraction for training a machine learning model, such as an LSTM, to predict future stock prices based on past trends.
iv) Test Model: The best fitted model is then estimated on testing data to determine its correctness and performance.
v) Predict Prices: Finally, the model trained predicts future stock prices for a given period of time, such as the next 30 days.

c) **Tiingo API**

Tiingo is a financial data platform that provides historical and real-time data on a wide range of financial assets, including stocks, mutual funds, and cryptocurrencies. Tiingo offers a Representational State Transfer Full (RESTful) API that allows developers to access this data programmatically.

In the context of this project, we can use the Tiingo API to retrieve historical stock price data for a given company, such as AAPL. This data can then be pre-processed, and relevant features can be extracted for training the model and for foreseeing the next day cost of stocks.

The API offers additional data points, such as moving averages and technical indicators, which can be used to enhance the correctness of the model [44].

d) Long Short-Term Memory:

The project made use of LSTM, which belongs to the class of RNN. The output given by the last step in RNN is fed to the current step of the network. The RNN is a single hidden layer neural network which makes it less efficient to remember long term dependencies. This drawback of RNN can be coped up by using the LSTM algorithm. The LSTM consists of a memory cell which is capable of holding the data in the cell. Therefore, this memory cell consists of three main gates which are responsible for regulating the flow of information to be added removed which are as follows:

Forget gate

This gate classifies the information whether it is useful to store for future use or needs to be removed immediately from the machine. In this gate two values are fed as an input first the input at that particular instance and the output of the previous calculation or estimation, the input is then combined with the weight and bias in order to generate some result. Now the activation functions are applied on the current obtained value in order to get discrete outputs. Here 0 output value represents that information is not useful and forgotten whereas 1 output value represents that the information is useful to be remembered for future predictions.

The formula for forget gate is written below:

$$f(t) = o(xt * Uf + Ht - 1 * Wf) \tag{1}$$

where:

Xt	Input to the current timestamp
Uf	Weight associated with the input
Ht-1	The hidden state of the previous timestamp
Wf	It is the weight matrix associated with the hidden state

Input gate

This gate determines which information should be selected and stored in the memory cell. It employs an activation function known as sigmoid which classifies the degree to which information is important or not to be stored for future. This gate takes two input values, first the previous cell state and second the current input value and then decides whether it is appropriate to store in the new cell state or not. Also, the output of the input gate is a vector, whose value ranges from 0 to 1 and this vector is combined with each candidate value by which it is determined whether the data will be added to the cell state or removed.

The LSTM architecture is composed of two main sub-gates: the input gate and the candidate gate. Former is accountable for determining which value needs to be updated, while the latter adds new data to the cell state by creating a vector using new values. The formula for the input gate is given as follows:

$$i(t) = \sigma(wi[X(t), h(t-1)] + b) \tag{2}$$

$$\widehat{C}(t) = \tanh\,(W[X(t), h(t-1)] + cc) \tag{3}$$

$$g(t) = i(t) * \hat{C}(t) \tag{4}$$

where:

i(t)	Input gate activation
σ()	Sigmoid function
X(t)	Input function
h(t-1)	Previous hidden state
g(t)	Fnal hidden state

$\hat{C}(t)$	Candidate gate hidden state
Wi	Input gate's weight matrix
W	Weight matrix for candidate gate
b	Bias vector for input gate
cc	Bias vector for candidate gate
tanh	Hyperbolic tangent activation function

Output gate

This gate calculates the final value of the hidden layer Here the input consists of two values first the value of the current state and the value of the further hidden state both these values are passed by the sigmoid activation function in order to generate a new cell state which will further be passed by tanh activation function. Now the value generated by both the activation functions is combined to get the final result. The result of this gate will be used to find which information should be carried further by hidden state.

The output gate is calculated using the following formula:

$$\mathrm{o(t)} = \sigma(\mathrm{Wo}[\mathrm{X(t)}, \mathrm{h(t-1)}] + \mathrm{bo}) \tag{5}$$

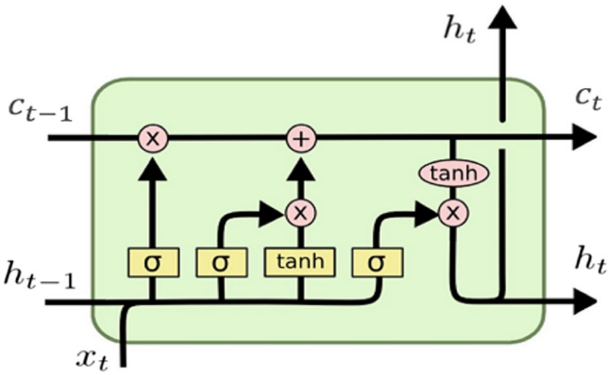
$$\mathrm{h(t)} = \mathrm{o(t)} * \tanh(\mathrm{C(t)}) \tag{6}$$

where:

o(t)	Output Gate function
σ()	Sigmoid Activation Function
Wo	Weight of input gate
X(t)	Input Vector function
h(t)	Hidden state
h(t-1)	Previous hidden state
bo	Bias vector for input gate

The LSTM uses these gates to decide the type of information retained by the network for future use. This allows the LSTM to learn long-term dependencies in sequential data, such as stock prices.

Fig. 3 LSTM Architecture



A neuron consists of hidden state and current state while a layer of neuron consists of an LSTM unit that has three gates and a memory cell. A hidden state is another feature of LSTM, the fig. 3 is representing H(t-1) by the hidden state of the previous time and Ht by the hidden state of the current time. Moreover, it has a cell state that is represented by C(t-1) for the earlier time and C(t) for the more recent ones (Fig. 4).

In conclusion, a real-time stock market price prediction system using Tiingo API and ML can be a challenging but rewarding project. By integrating various data sources, feature engineering, and machine learning, we can create a system that provides accurate and reliable stock price predictions in real-time. With this system, investors can make informed decisions and potentially increase their returns in the stock market [45, 46].

e) **Tools and technologies**

This work used various tools and technologies to implement the proposed work as mentioned below.

Python -It is a programming language used for building the model.

Pandas - It is used for data manipulation and analysis library.

NumPy - It is a numerical computing library.

Scikit learn - It is a ML library for dataset cleaning, filtering and evaluation metrics.

TensorFlow - It is a deep learning framework for making the LSTM model.

Keras - It is a complex neural network API used in building the LSTM model.

Tiingo API - Used for collecting historical stock price data.

Matplotlib - Data visualization library used for plotting the stock prices and the model's predictions.

3 Design and implementation

The proposed work undergoes various steps and procedures. The project is developed in python language using the LSTM concept of deep learning. We initiate loading the data of AAPL from Tiingo API. The data is pre-process using python libraries such as NumPy and pandas. We further select the relevant features from data and fragment it into training and testing splits in 65:35 ratio. Thus, the Stacked LSTM model is trained by using Scikit-Learn, Keras and TensorFlow libraries. (Fig.5)

Now the performance metrics are created using training and testing data by calculating RMSE value using math library functions. The difference between the RMSE values

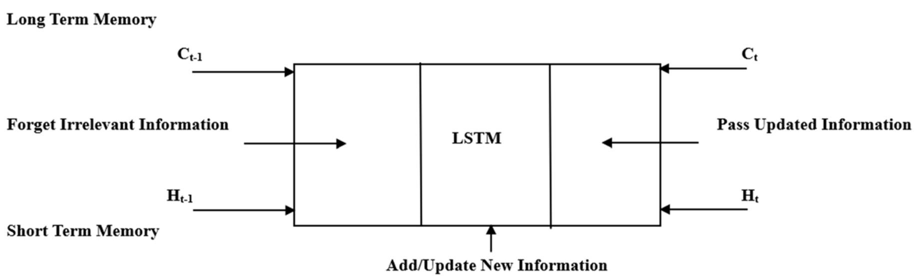


Fig. 4 Input and Output of LSTM

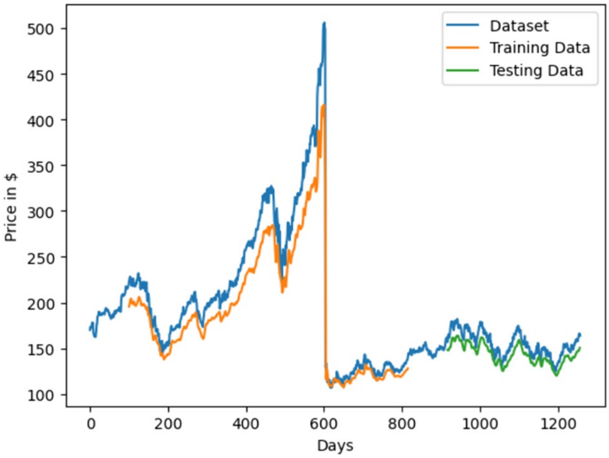


Fig. 5 Training and Testing of Model

```
1 day input [0.10026299 0.10735128 0.10379461 0.10820288 0.10507201 0.10990607
 0.11133375 0.10311835 0.10855354 0.11078272 0.10336882 0.09362555
 0.08598622 0.10316844 0.10386976 0.10261741 0.09966187 0.0903444
 0.08541014 0.08969317 0.08846587 0.09430182 0.09675642 0.0910958
 0.07428929 0.06930495 0.0639449  0.06376957 0.07165936 0.06359424
 0.0626675  0.05808391 0.04809017 0.05703193 0.05783344 0.04566061
 0.04889167 0.04553538 0.05705698 0.05838447 0.05983719 0.06675016
 0.06654978 0.06993112 0.07288666 0.07105823 0.07120852 0.07772073
 0.08583594 0.08939261 0.08771446 0.09297433 0.09790858 0.09056982
 0.09380088 0.09665623 0.11015654 0.11937383 0.11243582 0.11974953
 0.11291171 0.11028178 0.11063244 0.11774577 0.11611772 0.12145272
 0.11739512 0.11448967 0.10429555 0.10537257 0.10659987 0.09986224
 0.10289292 0.10161553 0.09635567 0.09785848 0.11068253 0.11769568
 0.11211021 0.11529117 0.10958046 0.10434565 0.1092799  0.11458986
 0.11559173 0.12275517 0.12062617 0.12663745 0.13134627 0.12771446
 0.13046963 0.13377583 0.12884158 0.12726362 0.13507827 0.13906074
 0.14542267 0.14860363 0.1472511  0.09755178]
1 day output [[0.09834826]]
2 day input [0.10735128 0.10379461 0.10820288 0.10507201 0.10990607 0.11133375
 0.10311835 0.10855354 0.11078272 0.10336882 0.09362555 0.08598622
 0.10316844 0.10386976 0.10261741 0.09966187 0.0903444  0.08541014
 0.08969317 0.08846587 0.09430182 0.09675642 0.0910958  0.07428929
 0.06930495 0.0639449  0.06376957 0.07165936 0.06359424 0.0626675
 0.05808391 0.04809017 0.05703193 0.05783344 0.04566061 0.04889167
 0.04553538 0.05705698 0.05838447 0.05983719 0.06675016 0.06654978
 0.06993112 0.07288666 0.07105823 0.07120852 0.07772073 0.08583594
 0.08939261 0.08771446 0.09297433 0.09790858 0.09056982 0.09380088
 0.09665623 0.11015654 0.11937383 0.11243582 0.11974953 0.11291171
 0.11028178 0.11063244 0.11774577 0.11611772 0.12145272 0.11739512
 0.11448967 0.10429555 0.10537257 0.10659987 0.09986224 0.10289292
 0.10161553 0.09635567 0.09785848 0.11068253 0.11769568 0.11211021
 0.11529117 0.10958046 0.10434565 0.1092799  0.11458986 0.11559173
 0.12275517 0.12062617 0.12663745 0.13134627 0.12771446 0.13046963
 0.13377583 0.12884158 0.12726362 0.13507827 0.13906074 0.14542267
 0.14860363 0.1472511  0.09755178 0.09834826]
2 day output [[0.09857724]]
```

Fig. 6 Input of 100 Days Price and Output of 31st Day Price is used to estimate the efficiency of the model. To speculate the future stock prices of next 30 days, 100 days of data is required. The test data is used to forecast the future price by analysing the previous 100 days of data.

The Fig. 6 shows clearly that firstly 100 days historical data and using that data price of next 30 days is predicted. RMSE and MSE values are calculated to find out the correctness of the model. Figure 7 shows the predictions and visualization using the Matplotlib.

4 Results and discussions

MSE is used as a parameter to calculate the accuracy of the data. MSE is used to find out the quality of the data.

$$\mathrm{MSE} = 1/\mathrm{n} * \Sigma(\mathrm{yi} - \widehat{\mathrm{y}}\mathrm{i})^2 \tag{7}$$

where:

yi Actual or observed value
ŷi Predicted value
n Total number of observations

MSE is a measure used to estimate the error. It can be explained as the squared mean of the difference among the estimated value and the recorded value. If the value of MSE is equal to 0 then it is considered that the model has no errors. MSE is symmetric to the error in the model e.g., if the value of MSE increases then error in the model also increases and vice versa is also true.

In this proposed methodology, an experiment is conducted in which the number of times the dataset will pass through the trained model is set 100. The training and testing ratio of the dataset is chosen 65:35 percent respectively. The experiment involved training the model on the dataset for 100 epochs, with a training to testing data ratio of

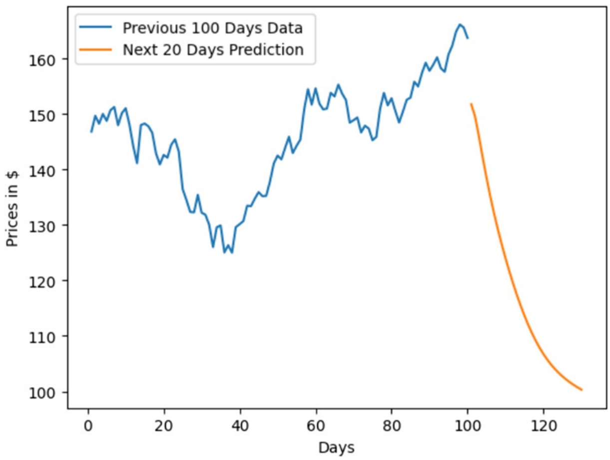


Fig. 7 Visualizing the Predicted Next 20 and 100 Days Price

Table 1 Number of Hidden Layers with MSE

Hidden layer	MSE
4	0.0002401
8	0.0003858
16	0.0002822
32	0.0002750
64	0.0001842
128	0.0001273

65:35. The training data enables the model to learn and capture various patterns, trends, and properties from the historical stock price data. The testing data serves as a validation set to evaluate the model's performance on unseen data.

Table 1 presents the results of the experiment, showing the relationship between the number of hidden layers in the LSTM model and the corresponding MSE values. The MSE values decrease as the count of hidden layers increases. This suggests that increasing the complexity and depth of the model by adding more hidden layers allows it to better capture the intricate dependencies and patterns inherent in the stock price data.

Table 1 presents the results of the experiment, showing the relationship between the number of hidden layers in the LSTM model and the corresponding MSE values. The MSE values decrease as the count of hidden layers increases. This suggests that increasing the complexity and depth of the model by adding more hidden layers allows it to better capture the intricate dependencies and patterns inherent in the stock price data.

Figure 8 visually represents the relationship between the number of hidden layers and MSE. The graph clearly demonstrates a decreasing trend in MSE as the number of hidden layers increases. It reveals that increasing the model's capacity by adding more hidden layers enables it to capture and learn more complex patterns, resulting in improved accuracy and reduced prediction errors. The lowest MSE value is observed when the model has 128 hidden layers, indicating the optimal trade-off between model complexity and performance in this particular experiment. The inclusion of dropout regularization in the proposed methodology further enhances the model's performance. Dropout selectively deactivates neuron nodes during training, reducing overfitting and improving the generalization ability of the model. By eliminating irrelevant connections and reducing complexity, dropout helps to generate clearer and more reliable output predictions. The analysis of the obtained results

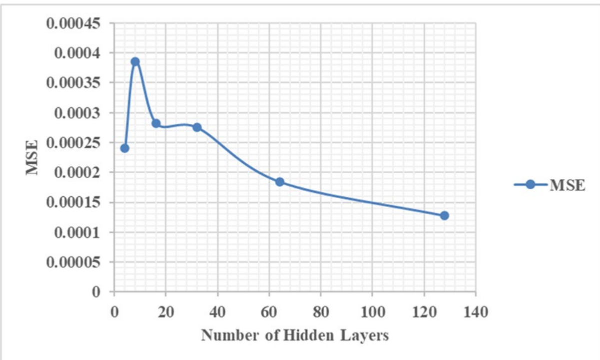


Fig. 8 Hidden Layer Vs MSE

Table 2 Dropout with MSE

Dropout	MSE
0.2	0.000455
0.3	0.0011
0.4	0.0024
0.5	0.0036
0.9	0.0123

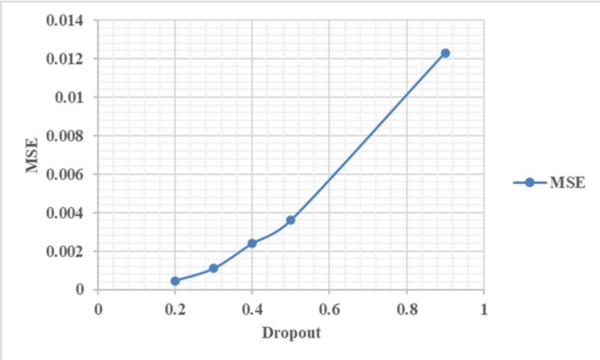


Fig. 9 Dropout Vs MSE

suggests that the proposed methodology, utilizing a stacked LSTM model with an optimal number of hidden layers and dropout regularization, yields accurate and precise stock price predictions. The decreasing trend in MSE values with increasing hidden layers demonstrates the effectiveness of leveraging a deeper architecture to capture intricate patterns in the stock price data. The inclusion of dropout regularization ensures model robustness and mitigates overfitting, enhancing the model's performance and reliability. These findings validate the proposed methodology as a promising approach for stock price prediction, offering valuable insights for researchers and practitioners in the field of finance and data science.

Table 2 presents the relationship between dropout rates and mean squared error (MSE) while maintaining a fixed hidden layer count of 128. The findings indicate that as the dropout rate increases, the MSE also increases. This suggests that a higher dropout rate introduces more regularization and may lead to increased prediction errors. However, a lower dropout rate of 0.2 resulted in the minimum MSE, indicating a balance between model complexity and generalization.

Figure 9 visually represents the relationship between dropout and MSE. It is observed that the MSE achieves the minimum error when the dropout value is 0.2. The analysis indicates that as the dropout value increases, the mean squared error also increases. This suggests that an optimal dropout value of 0.2 provides the best balance between regularization and model performance, resulting in lower prediction errors. The activation function plays a crucial role in determining the activation status of a neuron and enabling non-linearity in ANN.

The activation status of a neuron is determined by the activation function. According to this theory, it will perform some straightforward mathematical operations to decide whether the input from the neuron to the network is integral or not for the prediction process. The

Table 3 Activation Function with MSE

Activation Function	MSE
Relu	0.0003374
Softmax	0.0012
Tanh	0.0011
Sigmoid	0.0013
Linear	0.00076129

Table 4 Comparison of Proposed Model with Existing Model

Model	RMSE
PCA-LSTM [43]	0.221
Stacked-LSTM (Proposed)	0.018

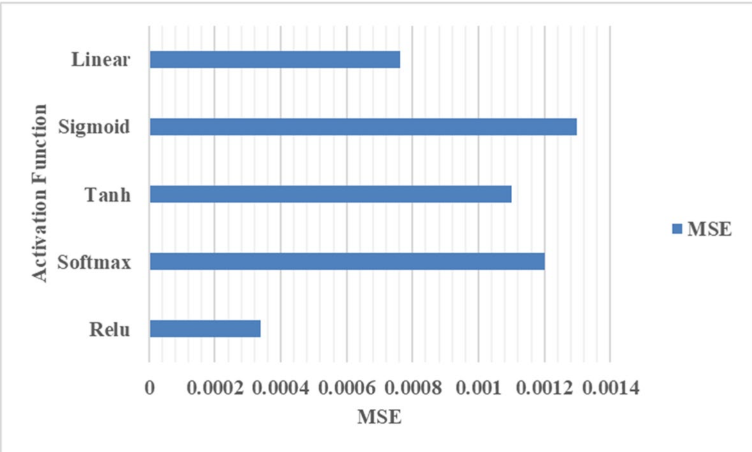


Fig. 10 Activation Function Vs MSE

activation function's function is to enable non-linearity in an ANN and produce output from a set of input values provided to a layer. Table 3 shows the relationship between activation function and MSE. While keeping the hidden layer = 128 and dropout = 0.2.

Comparing the proposed stacked LSTM model with the existing PCA-LSTM model, Table 4 showcases the RMSE values for both models. The PCA-LSTM model, based on previous work, achieved an RMSE of 0.221, while the proposed stacked LSTM model attained a significantly lower RMSE of 0.0183684. This comparison highlights the superior performance of the stacked LSTM model in accurately predicting stock prices.

Figure 10 illustrates the relationship between activation functions and MSE. It is observed that the Relu activation function yields the lowest MSE value, indicating that it provides better prediction accuracy compared to other activation functions in this specific experiment.

In comparison to the existing work [43], which utilizes a DL model comprising two LSTM layers to forecast the daily closing price of Ping A Bank, the proposed stacked

LSTM model demonstrates superior performance. The existing work employs a PCA-LSTM model, and its root-mean-square error (RMSE) is reported as 0.221. In contrast, the stacked LSTM model achieved a significantly lower RMSE of 0.0183684, indicating a better fit of the dataset into the model.

Comparing the proposed stacked LSTM model with the existing PCA-LSTM model, Table 4 showcases the RMSE values for both models. The PCA-LSTM model, based on previous work, achieved an RMSE of 0.221, while the proposed stacked LSTM model attained a significantly lower RMSE of 0.0183684 because the stacked LSTM model uses multiple LSTM layers which allows the model to learn complex patterns from the data whereas the PCA-LSTM use PCA which is a dimensionality reduction technique that transforms the input data into a lower-dimensional space due to which some of the important features may be lost and hence give less accurate results. Thus the stacked LSTM model demonstrates its capability to capture intricate patterns and dependencies in the data, resulting in improved prediction performance compared to existing models.

Also, the previous work only calculated the RMSE value of the predicted and actual value but in stacked LSTM various different parameters like activation function, dropout value and number of hidden layers are used to calculate the value of RMSE due to which the superior performance of the stacked LSTM model can be proved. The accuracy of the model These findings provide valuable insights for researchers and practitioners in the field of stock price prediction.

5 Conclusion and future scope

This work successfully developed an LSTM model to predict stock prices using Tiingo API data. The model achieved high accuracy and outperformed existing models, demonstrating the effectiveness of LSTM for stock price prediction. The project also emphasized the importance of feature engineering and model evaluation in developing accurate predictions. The results have potential applications in the financial industry and provide a basis for further research in the field. This gives only 0.0813 RMSE which proves that the model is more accurate and precise.

This work provides a comprehensive guide to develop a robust and more precise LSTM model for predicting using API data. It also highlights the importance of proper project management, feature engineering, and model evaluation in achieving high accuracy and scalability.

There are multiple potential areas for future development that can be explored for improving the performance of the proposed model, such as:

Incorporate other features besides just the historical stock prices such as: news sentiment, economic indicators, and other financial data that can impact the stock prices.

Exploring DL models such as CNN and Transformer models for stock price prediction.

Deploy the model as a real-time stock price prediction system and integrate it with a trading platform to automate the buying and selling of stocks based on the model's predictions.

Study the use of reinforcement learning to optimize the strategy of trading based on the predictions.

Funding None.

Data availability As no datasets were generated or analyzed during the current study, data sharing is not applicable to this article.

Code availability The code for implementation is available upon request, subject to privacy and other restrictions.

Declarations

Conflicts of interest/Competing interests The authors confirm that they have no competing interests or conflicts of interest to declare.

References

1. Jagwani J, Gupta M, Sachdeva H, Singhal A (2018) "Stock Price Forecasting Using Data from Yahoo Finance and Analysing Seasonal and Nonseasonal Trend", in. Second Int Conference Intell Computing Control Systems (ICICCS) 2018:462–467. https://doi.org/10.1109/ICCONS.2018.8663035
2. Pandey P, Litoriya R, Tiwari A (2018) A framework for fuzzy modelling in agricultural diagnostics. J Eur des Systèmes Autom 51:203–223. https://doi.org/10.3166/jesa.51.203-223
3. Parmar I et al (2018) "Stock Market Prediction Using Machine Learning", in. First Int Conference Secure Cyber Computing Commu (ICSCCC) 2018:574–576. https://doi.org/10.1109/ICSCCC.2018.8703332
4. Nourbakhsh Z, Habibi N (2023) Combining LSTM and CNN methods and fundamental analysis for stock price trend prediction. Multimed Tools Appl 82(12):17769–17799. https://doi.org/10.1007/s11042-022-13963-0
5. Soner S, Litoriya R, Pandey P (2021) Exploring Blockchain and Smart Contract Technology for Reliable and Secure Land Registration and Record Management. Wirel Pers Commun 121(4):2495–2509. https://doi.org/10.1007/s11277-021-08833-1
6. Soner S, Litoriya R, Pandey P (2022) Integrating Blockchain Technology with IoT and ML to Avoid Road Accidents Caused by Drunk Driving. Wirel Pers Commun. https://doi.org/10.1007/s11277-022-09695-x
7. Litoriya R, Arora A, Bajaj R, Gulati A (2022) Adoption of blockchain technology in the indian business market: obstacles and opportunities. 211–236. https://doi.org/10.1007/978-3-030-89546-4_11
8. Lei Y, Zhou K, Liu Y (2018) Multi-category events driven stock price trends prediction. In: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, pp 497–501. https://doi.org/10.1109/CCIS.2018.8691392
9. Bathla G, Rani R, Aggarwal H (2023) Stocks of year 2020: prediction of high variations in stock prices using LSTM. Multimed Tools Appl 82(7):9727–9743. https://doi.org/10.1007/s11042-022-12390-5
10. Sharaf M, Hemdan EE-D, El-Sayed A, El-Bahnasawy NA (2021) StockPred: a framework for stock Price prediction. Multimed Tools Appl 80(12):17923–17954. https://doi.org/10.1007/s11042-021-10579-8
11. Jeevan B, Naresh E, kumar BPV, Kambli P (2018) Share price prediction using machine learning technique. In: 2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, pp 1–4. https://doi.org/10.1109/CIMCA.2018.8739647
12. Ma C, Liang Y, Wang S, Lu S (2022) Stock linkage prediction based on optimized LSTM model. Multimed Tools Appl 81(9):12599–12617. https://doi.org/10.1007/s11042-022-12381-6
13. Du J, Liu Q, Chen K, Wang J (2019) Forecasting stock prices in two ways based on LSTM neural network. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp 1083–1086. https://doi.org/10.1109/ITNEC.2019.8729026
14. Gao SE, Lin BS, Wang C-M (2018) "Share Price Trend Prediction Using CRNN with LSTM Structure," 2018 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 2018, pp. 10–13. https://doi.org/10.1109/IS3C.2018.00012
15. Gao T, Chai Y, Liu Y (2017) Applying long short term momory neural networks for predicting stock closing price. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp 575–578. https://doi.org/10.1109/ICSESS.2017.8342981

16. Nivetha RY, Dhaya C (2017) Developing a prediction model for stock analysis. In: Int Conference Technical Advance Computers Commu (ICTACC) 2017:1–3. https://doi.org/10.1109/ICTACC.2017.11
17. Yeze Z, Yiying W (2019) Stock price prediction based on information entropy and artificial neural network. In: 2019 5th International Conference on Information Management (ICIM), pp 248–251. https://doi.org/10.1109/INFOMAN.2019.8714662
18. Hamed IM, Hussein AS, Tolba MF (2011) An intelligent model for stock market prediction. In: The Int Conference Computer Eng Systems 2011:105–110. https://doi.org/10.1109/ICCES.2011.6141021
19. Othan D, Kilimci ZH (2021) "Stock Market Prediction with New Generation Deep Contextualized Word Representations and Deep Learning Models using User Sentiments", in. Int Conference INnovations Intell SysTems App (INISTA) 2021:1–6. https://doi.org/10.1109/INISTA52262.2021.9548419
20. Ghosh P, Neufeld A, Sahoo JK (2022) Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. Financ Res Lett 46:102280. https://doi.org/10.1016/j.frl.2021.102280
21. Kavinnilaa J, Hemalatha E, Jacob MS, Dhanalakshmi R (2021) "Stock Price Prediction Based on LSTM Deep Learning Model," in 2021 International Conference on System, Computation, Automation and Networking (ICSCAN). 1–4. https://doi.org/10.1109/ICSCAN53069.2021.9526491
22. Manh H, Duong B, Manh S, Duong H (2006) The stock market and investment. Fintrop
23. Selvamuthu D, Kumar V, Mishra A (2019) Indian stock market prediction using artificial neural networks on tick data. Financ Innov 5(1):16. https://doi.org/10.1186/s40854-019-0131-7
24. Paul T (2021) "Fintech Empowers Prediction of Stock Market Index Using Artificial Neural Network", in. Int Conference Artificial Intell Computer Sci Technol (ICAICST) 2021:42–46. https://doi.org/10.1109/ICAICST53116.2021.9497817
25. Marchai FL, Martin W, Suhartono Z (2021) "Stock Prices Prediction Using Machine Learning," in 2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE). 79–84. https://doi.org/10.1109/ICITACEE53184.2021.9617222
26. Tuarob S et al (2021) DAViS: a unified solution for data collection, analyzation, and visualization in real-time stock market prediction. Financ Innov 7(1):56. https://doi.org/10.1186/s40854-021-00269-7
27. Singh T, Kalra R, Mishra S, Satakshi, Kumar M (2022) "An efficient real-time stock prediction exploiting incremental learning and deep learning," Evol Syst. https://doi.org/10.1007/s12530-022-09481-x
28. Adlakha N, Ridhima, Katal A (2021) Real time stock market analysis. In: 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), pp 1–5. https://doi.org/10.1109/ICSCAN53069.2021.9526506
29. Vij A, Saxena K, Rana A (2021) Prediction in stock price using of python and machine learning. In: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp 1–4. https://doi.org/10.1109/ICRITO51393.2021.9596513
30. Kan X, Miao M, Cao L, Xu T, Y Li Y, Jiang J (2020) Stock price prediction based on artificial neural network. In: 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pp 182–185. https://doi.org/10.1109/MLBDBI51377.2020.00040
31. Sisodia PS, Gupta A, Y Kumar Y, Ameta GK (2022) Stock market analysis and prediction for nifty50 using lstm deep learning approach. In: 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), vol 2, pp 156–161. https://doi.org/10.1109/ICIPTM54933.2022.9754148
32. Shao G (2022) Stock price prediction based on multifactorial linear models and machine learning approaches. In: 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), pp 319–324. https://doi.org/10.1109/TOCS56154.2022.10016086
33. Pandey M, Litoriya R, Pandey P (2020) Applicability of machine learning methods on mobile app effort estimation: validation and performance evaluation. Int J Softw Eng Knowl Eng 30(1):23–41. https://doi.org/10.1142/S0218194020500023
34. Pandey M, Litoriya R, Pandey P (2019) Novel Approach for Mobile Based App Development Incorporating MAAF. Wirel Pers Commun 107(4):1687–1708. https://doi.org/10.1007/s11277-019-06351-9
35. Litoriya R, Pandey P (2019) "A Predictive Fuzzy Expert System for Crop Disease Diagnostic and Decision Support,". 175–194. https://doi.org/10.4018/978-1-5225-9175-7.ch010
36. Pandey P, Litoriya R (2019) An activity vigilance system for elderly based on fuzzy probability transformations. J Intell Fuzzy Syst 36(3):2481–2494. https://doi.org/10.3233/JIFS-181146

37. Pandey P, Litoriya R (2019) Elderly care through unusual behavior detection: a disaster management approach using iot and intelligence. IBM J Res Dev 64(1):15:1–15. https://doi.org/10.1147/JRD.2019.2947018
38. Bandhu KC, Litoriya R, Bag Z, Barwaniwala AM, Garg M (2022) "Blockchain and smart contract enabled smart and secure electronic voting system," Int J Electron Gov
39. Malviya S, Dave S, Bandhu KCR, Litoriy R (n.d) A Cryptographic Security Mechanism for Dynamic Groups for Public Cloud Environments. J Autom Mob Robot Intell Syst
40. Mathanprasad LM Gunasekara M (2022) Analysing the trend of stock marketand evaluate the performance of market prediction using machine learning approach. In: 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), pp 1–9. https://doi.org/10.1109/ACCAI53970.2022.9752616
41. Hirey M, Unagar J, Prabhu K, Desai R (2022) Analysis of stock price prediction using machine learning algorithms. In: Int Conference Adv Technol (ICONAT), pp 1–4. https://doi.org/10.1109/ICONAT53423.2022.9725888
42. Lakshmanarao A, Babu MR, Gupta C, Lakshmi ASG (2022) Stock price prediction using deep learning and FLASK. In: 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp 1–5. https://doi.org/10.1109/ICSES55317.2022.9914347
43. Wen Y, Lin P, Nie X (2020) Research of Stock Price Prediction Based on PCA-LSTM Model. IOP Conf Ser Mater Sci Eng 790(1):12109. https://doi.org/10.1088/1757-899X/790/1/012109
44. Yick C (2017) Tiingo python. [Online]. Available: https://tiingo-python.readthedocs.io/en/latest/readme.html
45. Chugh A (2023) Deep learning | Introduction to long short term memory. [Online]. Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
46. What is LSTM? Introduction to long short term memory. 2022. [Online]. Available: https://intellipaat.com/blog/what-is-lstm/

Authors and Affiliations

Kailash Chandra Bandhu[1] · Ratnesh Litoriya[1] · Anshita Jain[1] · Anand Vardhan Shukla[1] · Swati Vaidya[1]

✉ Ratnesh Litoriya
litoriya.ratnesh@gmail.com

Kailash Chandra Bandhu
kailashchandra.bandhu@gmail.com

Anshita Jain
anshitajain12345@gmail.com

Anand Vardhan Shukla
theanandam394@gmail.com

Swati Vaidya
swati.dhage28@gmail.com

[1] Department of Computer Science and Engineering, Medi-Caps University, Indore, India