International Journal for Modern Trends in Science and Technology
Volume 11, Issue 10, pages 36-42.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue10.html
DOI: https://doi.org/10.5281/zenodo.17249436

AI for Real-Time Stock Market Analysis

P.Sowjanya[1], D.Swetha Tanusri[2], P.Komala[2], K.Lahari[2]

[1]*Associate Professor, Department of CSE, Vijaya Institute of Technology for Women, Enikepadu, AP, INDIA.*
[2]*Department of CSE, Vijaya Institute of Technology for Women, Enikepadu, AP, INDIA.*

To Cite this Article
P.Sowjanya, D.Swetha Tanusri, P.Komala & K.Lahari (2025). AI for Real-Time Stock Market Analysis. International Journal for Modern Trends in Science and Technology, 11(10), 36-42. https://doi.org/10.5281/zenodo.17249436

Article Info
Received: 05 September 2025; Accepted: 28 September 2025.; Published: 02 October 2025.

KEYWORDS

Stock Market, Machine Learning, Predictions, Cloud Computing.

ABSTRACT

In the world with increasing globalization, where money places a crucial role in determining the expansion and earnings of a company trading places a very crucial role. Multiple companies invest millions and billions of dollars in other countries with an expectation to make profits. In such a risky business Predicting the movement of the market can help companies or individual in making good decisions and can prevent severe loses. In this research paper we will discuss how we can use the computational power of the computer on cloud along with the machine learning algorithms to predict the closing values of the stocks which is a big challenge otherwise. For this purpose, we will use Python as our programming language which supports a lot of ML based Libraries. The models we will be using are SVM (Support Vector Machine), Linear Regression, Random Forest, XG Boost, LSTM for deep learning.

INTRODUCTION

The system of stock trading is fairly simple, someone with excessive cash decides to invest it in the market by buying shares of different companies based on their market performance and market capital with a hope to make good profits in future. This is done by buying the share at lower rate and selling it at a high value. The prediction of 'when' the stock will hit its low and high can make a huge difference in the earnings.

There are already a lot of indicators in the market which somehow helps in market prediction but their accuracy is low, also it requires a lot of knowledge from the user end to analyse these indicators. These indicators also use the past data of the markets like high, lows, volume, P/E ratio etc, to give meaningful insights to the users. In this research paper we will be using some calculators already in the market like gann calculator, tenkan sen for comparing the accuracy of our model. A stock prediction largely depends on its previous data and most importantly the last day data. Using the data extracted from the stock exchange. In order to predict the new price of the day we basically needs 3 parameters i.e. open,high,low to predict the closing price. The closing price will be the value our model will predict based on

its past data and the Final resulbased on its opening price. The models we are using focuses mostly on intra day trading. This project will be using a lot of data which may take days to process and hence it will not be efficient to run it on our regular PC. That's where cloud computing comes into play. The advance high computing GPU's can run our model in minutes and can get us the results as soon as possible. There are two types of Data we can use to train our model

1) Data from stock exchange

2) Tick by Tick data

It should be kept in kind how much of this data has to be used to prevent overfitting or over training of the models. The tick by tick data can be extracted using API's Provided by your stock broker

1.1 Objectives

The primary objectives of the project are as follows:

- To Develop an AI-Based System: Create a robust system that utilizes AI for real-time stock market analysis, enabling users to gain insights into market trends. This involves integrating various AI techniques, such as natural language processing (NLP) for sentiment analysis and machine learning for predictive modeling.
- To Implement Machine Learning Algorithms: Employ various machine learning algorithms, such as regression analysis, decision trees, and neural networks, to perform predictive analysis on stock market data. The project will explore different algorithms to determine which ones yield the most accurate predictions. For example, a comparison between linear regression and more complex models like LSTM (Long Short-Term Memory) networks will be conducted to assess performance.
- To Create a User-Friendly Interface: Design an interface that is easy to use, allowing users to visualize data, access predictions, and understand market trends without requiring extensive technical knowledge. The interface will include features such as drag-and-drop functionality for customizing dashboards and easy navigation through different analysis tools.
- To Ensure Scalability and Flexibility: Build a system that can easily adapt to new data sources and algorithms, ensuring that it remains relevant as market conditions change. This includes the ability to integrate new financial data APIs and machine learning models as they become available.
- To Provide Educational Resources: Alongside the analysis tools, the project will offer educational resources to help users understand stock market concepts, machine learning techniques, and how to interpret the analysis results. This could include tutorials, webinars, and articles on investment strategies.

1.2 Methodology

This basic rule of this project is to use the data from previous day to predict the closing value of the day. Though there are many other factors which effects the price of a stock which are not in our hands like rumours, government policies etc. But working with numbers we should be able to make best model for prediction.

The Training data should be first normalised and data base we will be using is mongo DB for its fast and efficient processing.

The data we have will be in the form of 4 columns O1,H1,L1,C1 i.e. open(for day 1),high(for day

1) ,Low(for day 1),close(for day 1).The last column is

2) our label we need to predict using our models. Therefore, to predict the value for tomorrow we should be having the data till today.

While preparing the data set for training we have to shift the fourth column one block up. So that the data of today has a label of closing value of tomorrow i.e. O1,H1,L1,C2.

We are here using multiple models to predict their accuracy and based on this we can define which to use for the predictions.

2. ARCHITECTURE:

A real-time stock market analysis system powered by Artificial Intelligence (AI) is a sophisticated architecture designed to ingest, process, analyze, and generate insights from a continuous stream of financial data. Here's a breakdown of the key components and their interactions:

2.1. Real-Time Data Ingestion Layer:

• Data Sources: This layer is responsible for collecting data from various sources in real- time. These sources include:

o Stock Exchanges: Direct feeds from exchanges like the National Stock Exchange of India (NSE) provide live

stock prices, order books (bid and ask prices), trading volumes, and trade execution data

o Financial News Providers: Real-time news feeds from sources like Bloomberg, Reuters, and financial news websites provide information on market events.

o Social Media Platforms: Streams of data from platforms like Twitter (X), Reddit, and financial forums can offer insights into market sentiment.

o Financial APIs: APIs from financial data providers (e.g., FactSet, Refinitiv, Alpha Vantage) offer real-time and historical data

o Alternative Data: This can include non-traditional data sources like satellite imagery (e.g., for tracking agricultural output etc)

• Data Ingestion Technologies: Various technologies are used to handle the high velocity and volume of real-time data:

o Message Queues: Systems like Apache Kafka or RabbitMQ act as buffers to handle the incoming data streams

o Stream Processing Platforms: Frameworks like Apache Flink, Apache Spark Streaming, or AWS Kinesis Data Streams enable the continuous processing of data as it arrives.

2.2. Data Preprocessing and Feature Engineering Layer:

• Data Cleaning: Techniques include handling missing data, filtering out irrelevant information, and standardizing data formats.

o Feature Engineering: It involves transforming the raw data into meaningful features that AI models can learn from. Examples include:

o Technical Indicators: Calculated from historical price and volume data, such as Moving Averages (SMA, EMA), Relative Strength Index (RSI), Bollinger Bands, and MACD.

o Sentiment Scores: Derived from news articles, social media posts, and financial reports using Natural Language Processing (NLP) techniques.

o Volatility Measures: Calculated based on historical price fluctuations, such as standard deviation or Average True Range (ATR).

o Fundamental Ratios: Financial ratios derived from company financial statements (e.g., Price-to-Earnings ratio, Debt-to-Equity ratio). While fundamental data isn't strictly real-time, changes and updates are incorporated as they become available.

o Event-Based Features: Indicators based on specific events like earnings announcements, mergers, or regulatory changes.

2.3. AI Model Layer:

• Model Selection: A variety of AI and Machine Learning (ML) algorithms are employed for different analytical tasks:

o Time Series Forecasting Models: For predicting future stock prices, including ARIMA, Prophet, and deep learning models like Recurrent Neural Networks (RNNs), LSTMs, and Transformers.

o Classification Models: For tasks like predicting whether a stock price will go up or down (binary classification) or categorizing price movements into different levels (multi-class classification). Algorithms include Logistic Regression, Support Vector Machines (SVMs), Random Forests, and Gradient Boosting machines (e.g., XGBoost, LightGBM).

o Sentiment Analysis Models: NLP models that analyze text data.These can range from rule-based systems to sophisticated deep learning models like BERT and RoBERTa.

o Anomaly Detection Models: To identify unusual trading patterns or potential market manipulation. Techniques include Isolation Forest, One-Class SVM, and autoencoders.

o Clustering Algorithms: To group stocks with similar performance or characteristics, which can be useful for identifying trends or sector-based movements (e.g., K-Means, DBSCAN).

• Model Training and Evaluation: Models are trained on historical data and continuously evaluated on new real-time data to assess their performance.

• Real-Time Inference: Once trained and validated, the AI models are deployed to make predictions and generate insights on the incoming real-time data. This requires efficient deployment frameworks and infrastructure.

2.4. Insight Generation and Visualization Layer:

• Insight Aggregation: The outputs from various AI models are aggregated and synthesized to generate actionable insights. For example, a "buy" signal might be generated based on a combination of positive sentiment, a bullish technical indicator, and a positive forecast from a time series model.

• Visualization Tools: Real-time dashboards and interactive visualizations are crucial for presenting the analysis and insights to users (traders, investors, analysts). These tools can display:

o Live stock prices and key indicators.

o Sentiment trends for specific stocks or the market.

o Predicted price movements and probabilities.

o Identified trading opportunities or risks.

o Portfolio performance and risk metrics.

• Alerting Systems: Real-time alerts can be triggered based on specific conditions identified by the AI models, such as significant price changes, changes in sentiment, or the detection of anomalies.

2.5. Infrastructure Layer:

• Cloud Computing: Cloud platforms like AWS, Azure, or Google Cloud are often used to provide the scalability, reliability, and processing power required for real-time stock market analysis.

• High-Performance Computing: Processing large volumes of real-time data and running complex AI models demands significant computational resources, often utilizing GPUs or specialized hardware accelerators.

• Low-Latency Network: Ensuring low latency in data ingestion and processing is critical for real-time analysis, especially for high- frequency trading applications.

3. Existing Methods:

- Lack of Real-Time Processing: Traditional stock analysis is based on historical data without incorporating real-time trends.
- Limited Prediction Accuracy: Many conventional models use simple linear regression or moving averages, which fail to capture complex market fluctuations.
- No Sentiment Analysis Integration: Investor sentiment plays a crucial role in stock price movement, but traditional methods do not analyze public opinion from news and social media.
- Manual Data Interpretation: Analysts must manually interpret charts and financial reports, making the process slow and inefficient.

4. PROPOSED SYSTEMS:

- Real-Time Data Analysis: The system fetches live stock market data from Yahoo Finance, Alpha Vantage, or Quandl APIs to provide real-time insights.
- Multi-Model Prediction: Uses ARIMA, LSTM, and Linear Regression models to improve stock price prediction accuracy.
- Sentiment Analysis for Market Mood: Analyzes news articles and social media posts (Twitter, Reddit, etc.) to determine public sentiment toward a stock.
- Interactive Visualization Dashboard: Displays historical price trends, real-time market data, and sentiment scores in a user-friendly web interface.
- Automated Buy/Sell
- Recommendations: Generates trading signals (Buy/Hold/Sell) based on AI model predictions and sentiment scores.
- User-Friendly and Accessible: Even non-technical users can access easy-to-understand visualizations and recommendations.

5. MODELS USED:

1) SVM (Support Vector Machine):

Support Vector Machine(SVM) is a supervised machine learning algorithm which is widely used for both classification and regression problems. However, it is mostly used in classification problems. But in our case, we will be using it for regression purpose because we need a numerical value as our output rather than a category.

It uses a plane on a 2D surface along with a plane to divide the plane in two parts where each class resides on the either sides.

But in our case, we have multiple values of x (Open, high, low) and a single value for y (close value). So, we can use either Gradient Descent or Ordinary least Squares which supports multiple inputs and provides a single output.

Fig 2. Linear Regression

3)Random Forests:

Another supervised machine learning model is Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by using multiple decision trees at the training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression in our case) of the individual trees. It is an extension of

decision tree and keeps track of overfitting of training data.

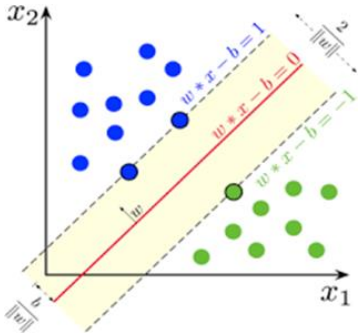

Fig 1. Support Vector Machine

2)Linear Regression:

Basically, the most widely used and simple regression model is Linear Regression. The representation is a linear equation that use or combines a particular set of input values i.e. x the solution will be the predicted output for that set of input values i.e. (y).

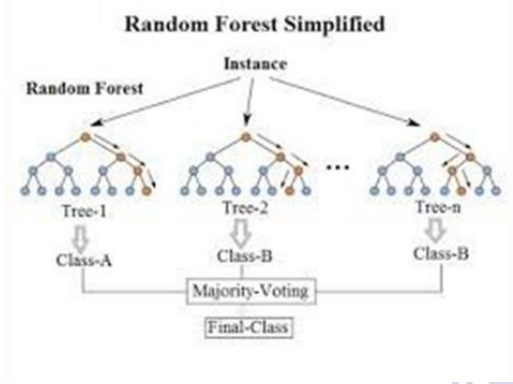

Fig: Random forest

4)XG Boost :

Both the input values (x) and the output value(y) are numeric (i.e. the closing value of the stock).

The linear equation in this model assigns one scaling factor to each value of x, called a coefficient which is represented by the capital Greek letter Beta (B). One additional coefficient called intercept or the bias coefficient is also added which gives the additional degree of freedom i.e. Moving up and down on a 2D plane.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1*x

XG Boost is an implementation of gradient boosted decision trees designed for speed and performance.

It has a perfect combination and balance of software and hardware optimization techniques to provide superior/best results using fewer computing resources in the shortest amount of time.

The two reasons to use XG Boost are also the two goals of the project:

1. Execution Speed. 2. Model Performance. 1. XG Boost Execution Speed Generally, XG Boost is fast. Really fast when compared to other implementations of gradient boosting.

Szilard Pafka performed some objective benchmarks comparing the performance of XG Boost to other implementations of gradient boosting and bagged decision trees.

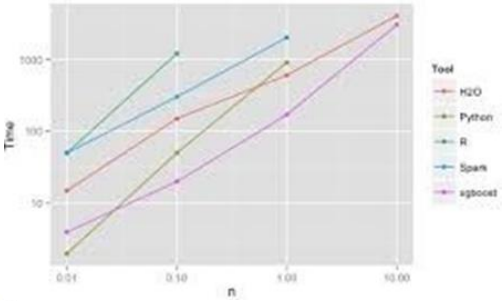

Fig 4. Benchmark Performance of XG Boost

His results showed that XG Boost was almost always faster than the other type of implementation models from R, Spark, Python etc. From this experiment we can conclude that this should be the best choice for machine learning model for our project.

2. XG Boost Model Performance

XG Boost dominates structured or tabular datasets on classification and regression predictive modelling problems.

It has been proved that this model is far better than its other counter parts when it comes to speed and accuracy. When the founder of this model won a ML Competition.

4)LSTM deep learning neural network: Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Till now we have used only regression models which are fast but compromises on the accuracy, the deep learning models provides high accuracy but also takes a lot of time and computational resources. The deep learning algorithms uses multiple players with multi neuron architecture which is what makes it highly computational and time consuming.

The deep learning also has many types of networks and LSTM is one of the feedback neural networks where results backtracks to previous layers unlike feed forward networks. It's hard to implement and use, as compared to single node regressions.

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for remembering the values over a particular time interval and the three gates regulate the flow of information into and out of the cell.

This networks are well-suited for classifying, processing and making predictions based on time series data just like we have in our project. LSTMs is a extended version of a traditional Recurrent Neural Network developed to deal with the exploding and vanishing gradient problems.

This model is specifically used when we want our model to deal with long-term dependencies like in our case the closing value of our stock depends on its all previous values. LSTM is smart model which Nirbhay Narkhede, IJECS Volume 08 Issue 09 September, 2019 Page No.24847-24850 Page 24850 determine how long to hold onto old information, when to remember and forget, and how to make connections between old memory with the new input.

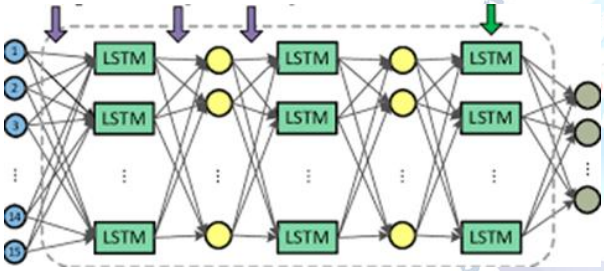

Fig5 LSTM Network

The best way to implement LSTM in our project is to use TensorFlow by google, this library does most of the work for us but still designing this network specific to our project is a big and timeconsuming task.

6. UML Diagrams:

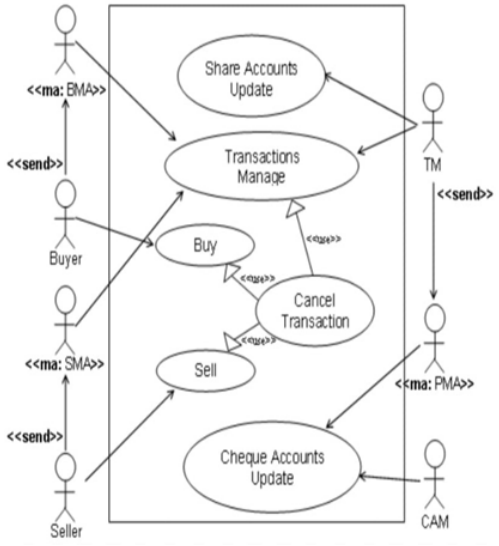

Fig6:Use case diagrams

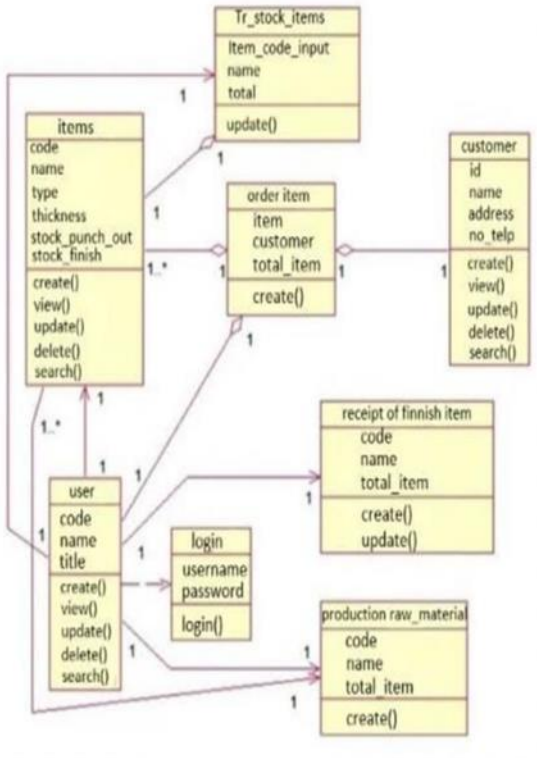

Fig7:Class Diagram

7. RESULTS:

Open	High	Low	Close
2076.9	2077.95	2049	2051.800049
2053.8	2060.2	2035.55	2045.849976
2049	2059	2032.6	2052.199951
2051	2060	2040.75	2056.949951
2055	2055	2025.1	2041.150024
2046.3	2048	1994.15	2000.400024
2001.9	2021.35	1989.55	2011.849976
2019.85	2037.4	2018.95	2029.599976
2021.3	2022.95	1988	1992.199951
1994.9	2009.85	1980.95	1989.199951
1994.85	1998	1956.5	1961.349976
1973.9	1999.85	1910.2	1970.25
1967.25	1972.2	1915	1925.699951
1925.7	1962	1915	1952.400024
1962.95	1978	1955.3	1968.199951
1969.4	1986	1955.15	(predicted value)

Fig 6. Result Output

The final result will be in a text file with the company name along with whether to buy or sell the stock for the given day.

8.TECHNOLOGIES AND FUTURE ENHANCEMENTS:

- Deep Reinforcement Learning for Trading Strategies :

Implement AI based trading bots using reinforcement learning to execute automated trades.

- Advanced Sentiment Analysis with GPT Models:

Use ChatGPT, BERT, or LlaMA to analyze financial news and reports with greater accuracy.

- Multi-Asset Analysis:

Extend the system to cryptocurrency, commodities, and forex markets for diversified trading.

- Blockchain Integration for SecureTransactions :

Use blockchain to record stock trades and prevent fraud in financial transactions.

- Explainable AI (XAI):

Implement explainable AI techniques to show users why a model predicts a certain trend.

- Integration with Algorithmic Trading Platforms :

Connect with platforms like Interactive Brokers or Alpaca API for automatic order execution based on AI predictions.

9.CONCLUSION:

Real-time stock market analysis is a powerful tool for investors, traders, and financial institutions. By leveraging machine learning, natural language processing, and data analytics, we can analyze large amounts of market data, identify patterns, and make informed investment decisions.

The expected scrapped data size should be more than 300GB including historical and tick by tick data. The type of models we are using are expected to take a lot of time to process and predict the new value and hence a powerful GPU on cloud has to be used in order to boost our processing time. Using the predicted value, it will be easy to calculate whether to buy or sell the stock for the given day.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Analytic Vidhya" [Online] Available: https://www.analyticsvidhya.com

[2] Machine Learning Mastery" [Online] Available : https://machinelearningmastery.com/ linear – regression - for – machine - learning/

[3] Medium" [Online] Available: https://medium.com/simple-ai/linearregression-intro-to-machine-learning-66e320dbdaf06

[4] Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601

[5] Long-Short-Term- Memory (LSTM) https://en.m.wikipedia.org/wiki/Long_shortterm_memory

[6] Graves, Alex; Fernández, Santiago; Gomez, Faustino (2006). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks"

[7] Semantic Scholar https://www.semanticscholar.org/paper/Cooccurrence-Feature-Learning-for-Skeletonbased-ZhuLan/4eb753322f13443d0e463e6c7123088734b3583a/figure/0