

Nifty 50 Stock Price Predictor Using Machine Learning Algorithms

Barathvikraman S K¹, Dhivyes Balaji S², Ruban R³, Vikash T⁴

^{1,2,3,4}UG Student

^{1,2,3,4}PG & Research Department of Computer Science,
Sri Ramakrishna College of Arts & Science, Coimbatore, Tamil Nadu, India

Abstract

Stock market price prediction is a critical financial tool that aids investors and analysts in making informed decisions. With advancements in machine learning technologies, the accuracy of these predictions based on historical data has significantly improved. This project aims to predict stock prices for companies listed under the Nifty 50 index using historical data and a Linear Regression model. By employing Python's finance library for data retrieval and Streamlet for web deployment, the project offers an interactive platform for users to predict and visualize stock price trends. The application is designed to serve as an accessible tool for retail investors, enabling them to forecast short-term price movements. Ultimately, this project illustrates how machine learning can be effectively applied in financial forecasting while ensuring user-friendliness and accessibility.

Keywords: Stock market, ML Model, Linear regression

I. INTRODUCTION

Stock prices are inherently volatile and subject to a multitude of influencing factors such as market sentiment, economic indicators, and global events. The demand for accurate stock price prediction has led to the evolution of sophisticated financial models and prediction systems. Historically, these systems were based on statistical models or expert opinions; however, the emergence of machine learning (ML) has paved the way for more precise and automated forecasting. This project focuses on predicting the next day's stock price for Nifty 50 companies. By utilizing historical stock data and a Linear Regression model, we aim to develop a system capable of identifying trends and making price predictions. The integration of machine learning with a user-friendly interface, created with Streamlit, allows users to interact with the model without requiring technical expertise. With this system, retail investors can make better-informed decisions grounded in data-driven predictions. The Nifty 50 serves as a benchmark stock index, comprising 50 of the largest and most liquid stocks in India. These companies are leaders in their sectors, and their performance significantly impacts the Indian economy. Accurate predictions of their stock prices can yield valuable insights for investors and analysts alike. The main objectives of this paper include:

- **Data Collection:** Retrieve and pre-process historical stock data for all Nifty 50 companies from trusted sources, such as Yahoo Finance via the yfinance library.
- **Model Development:** Implement a Linear Regression model to predict future stock prices based on the historical data.
- **Application Development:** Create an interactive web platform using Streamlit, allowing users to select a stock and view its predicted price.
- **Visualization:** Provide visual tools such as graphs and charts to facilitate user understanding of prediction results and historical trends.
- **Evaluation:** Assess the accuracy of the predictive model and refine it to enhance performance.

- **Accessibility:** Ensure that the application remains user-friendly for non-technical users while maintaining transparency in the prediction process.

II. EXISTING SYSTEM

Existing systems for stock price prediction often rely on complex, proprietary algorithms that are not easily accessible to retail investors. Large financial institutions utilize advanced tools like Bloomberg Terminals or Reuters Eikon to analyze market trends and forecast prices. These platforms, however, are typically expensive and challenging for average investors to use. Additionally, many prediction tools currently available depend on statistical methods and fail to leverage the adaptability of machine learning. Such models often struggle to account for market volatility or incorporate new information in real-time. Moreover, free online stock price prediction tools frequently provide basic functionalities and lack the depth required for comprehensive financial analysis. This project aims to address these shortcomings by delivering a transparent, free, and user-friendly prediction tool accessible to a wider audience, particularly retail investors.

III. PROPOSED SYSTEM

The proposed system is a web application leveraging machine learning to predict stock prices for Nifty 50 companies. Key features include:

- **Data Retrieval:** Automated fetching of historical stock data using the yfinance library, capturing essential information such as open, close, high, low prices, and trading volumes.
- **Model Training:** The retrieved data will train a Linear Regression model to forecast the closing price for the next trading day.
- **Prediction:** Users can select a stock from a dropdown menu, prompting the application to predict its price based on historical data.
- **Visualization:** The system will present historical performance data alongside predicted future prices.
- **User Interface:** Streamlit will facilitate a clean, interactive interface that caters to users with varying levels of technical expertise.
- **Future Enhancements:** Potential future extensions could incorporate additional machine learning models (e.g., LSTM, ARIMA) for improved accuracy and the integration of real-time data for intraday price predictions.

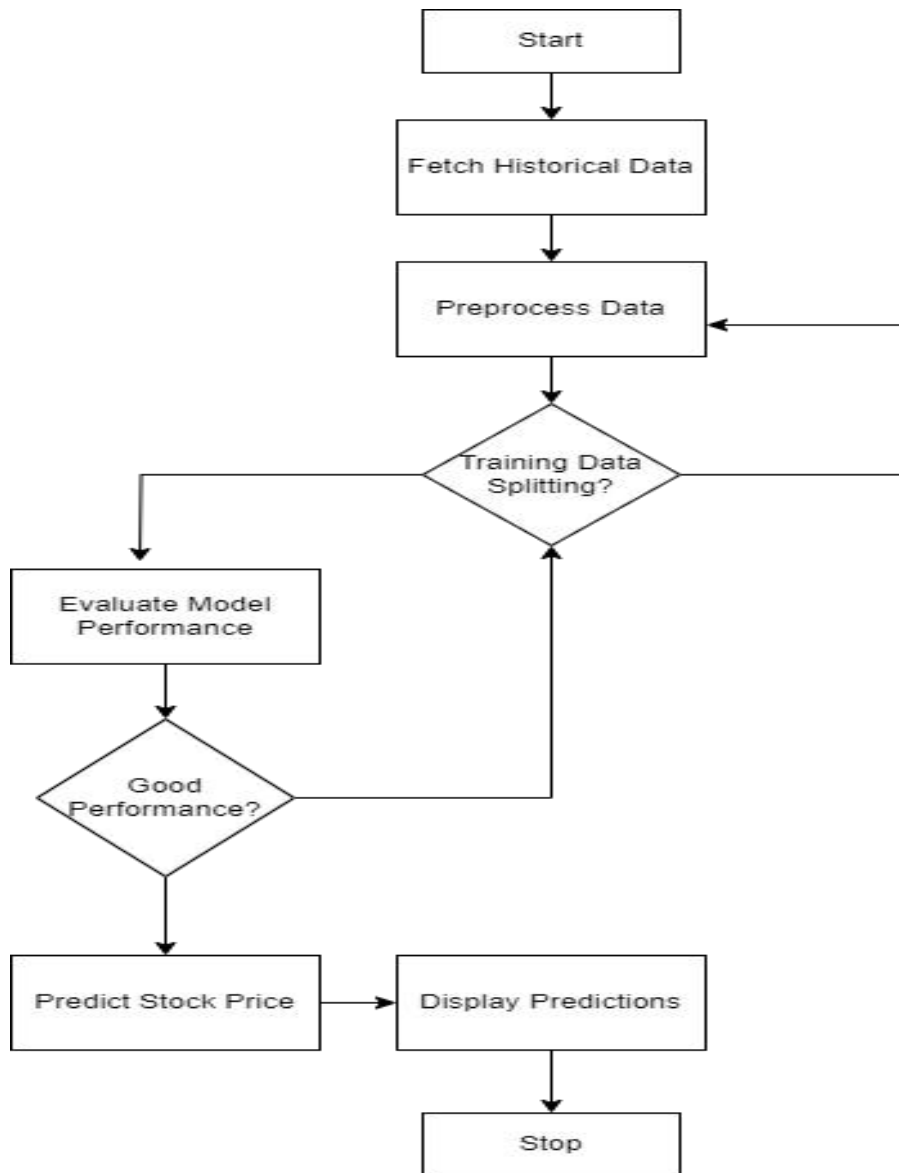


Figure-1 System Architecture

1. Data Collection

- **Source:** Historical stock data for Nifty 50 companies is downloaded using the yfinance library. This data encompasses daily prices (open, high, low, close) and trading volumes.
- **Time Range:** Data is collected over a significant period (e.g., the last 5 years) to equip the model with sufficient patterns and trends.

2. Data Preprocessing

- **Handling Missing Data:** Missing values in the dataset are filled using interpolation techniques or eliminated if deemed irrelevant.
- **Feature Engineering:** New features, such as "day of the year," are added to assist the model in recognizing temporal patterns.
- **Normalization:** Stock prices are scaled to ensure the model's performance is not biased by the price magnitudes.

3. Model Selection:

- **Linear Regression:** This model is selected for its simplicity and effectiveness in predicting continuous variables like stock prices. It assumes a linear relationship between stock price and day of the year.
- **Training:** The data is split into training and testing sets (typically an 80/20 split) for model fitting and evaluation.

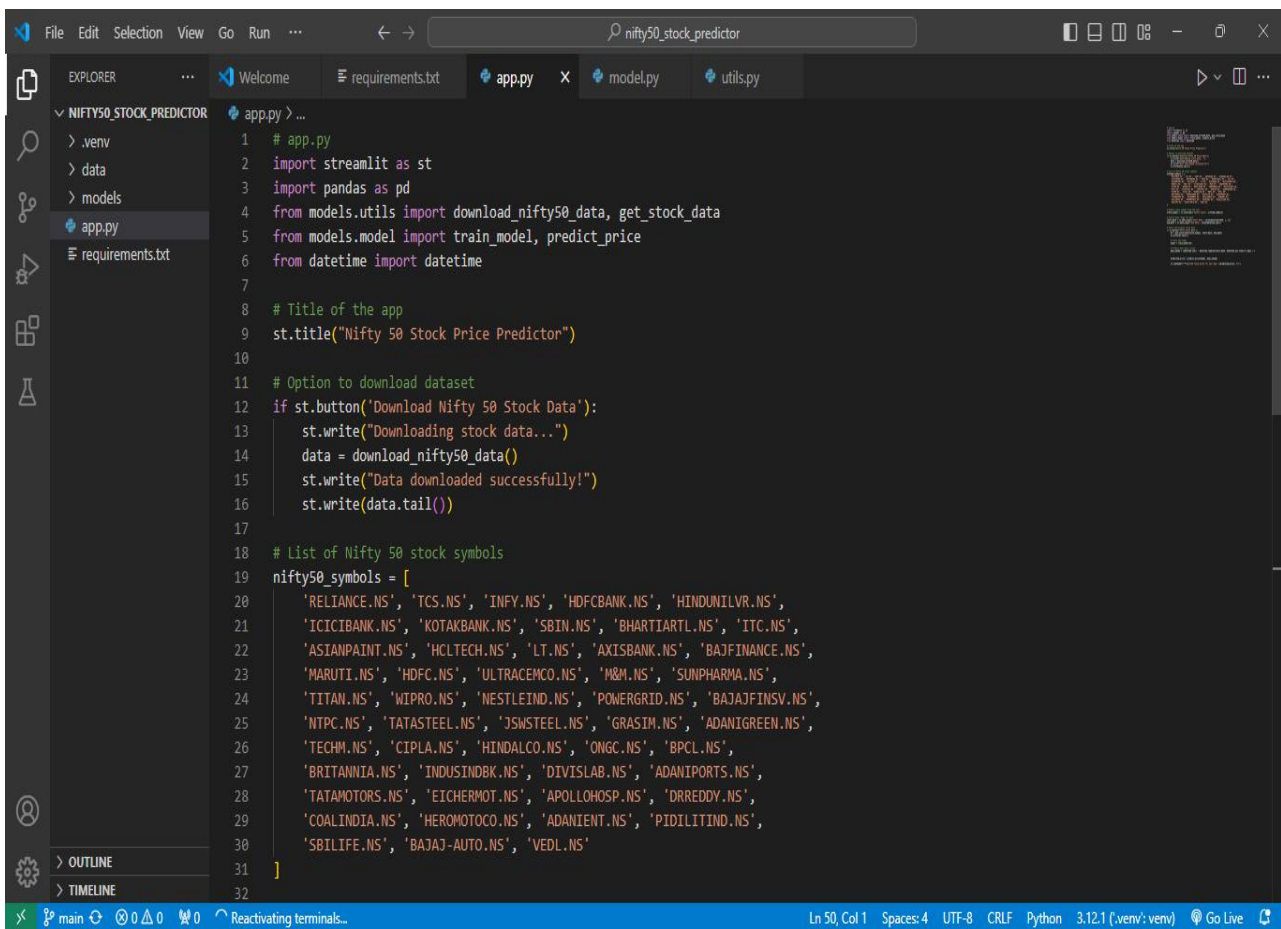
4. Model Evaluation:

- **Metrics:** The model's performance is assessed using metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared values.
- **Cross-Validation:** Techniques such as k-fold cross-validation are implemented to prevent overfitting and ensure good generalization on unseen data.

5. Deployment:

- **Streamlit Application:** The trained model is integrated into a web application using Streamlit, allowing user interaction with stock selections, predictions, and historical analyses.
- **Visualization:** Libraries such as Matplotlib or Plotly are utilized to display stock price trends and prediction results

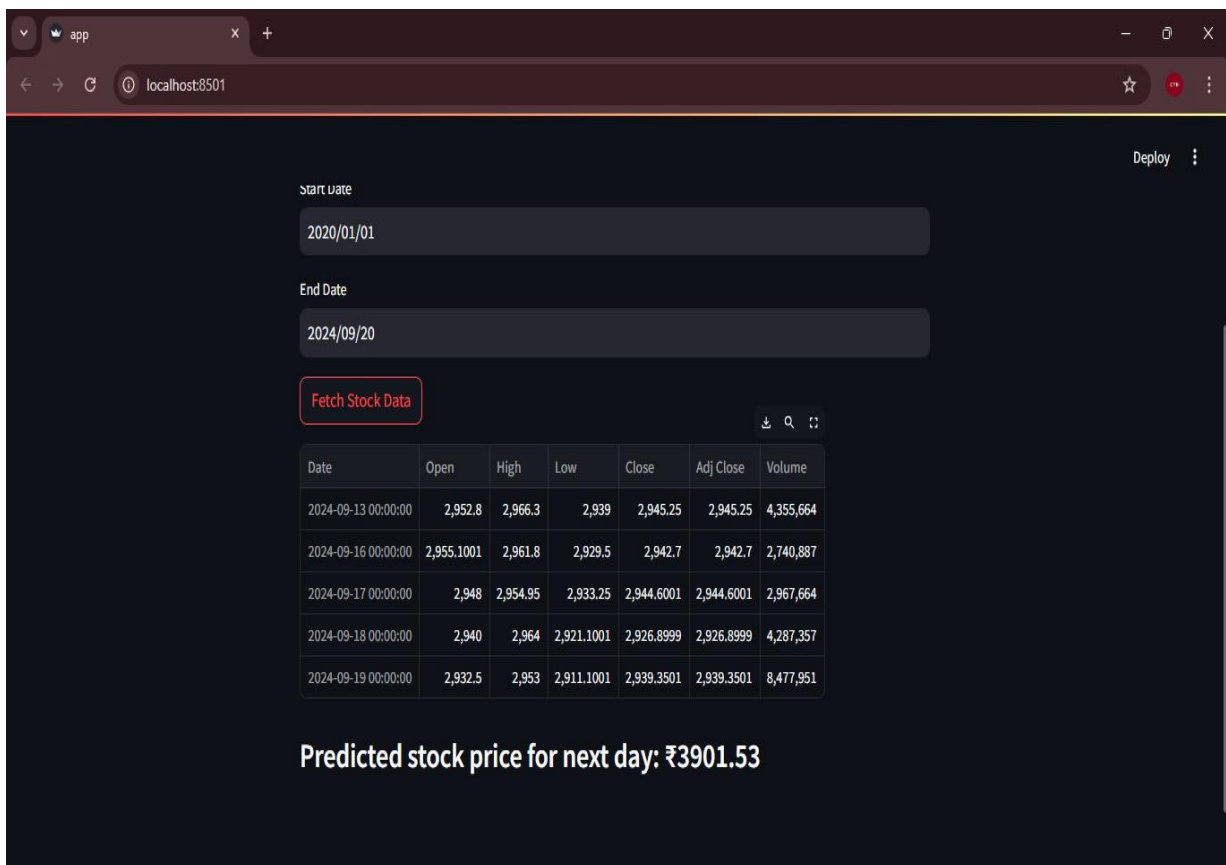
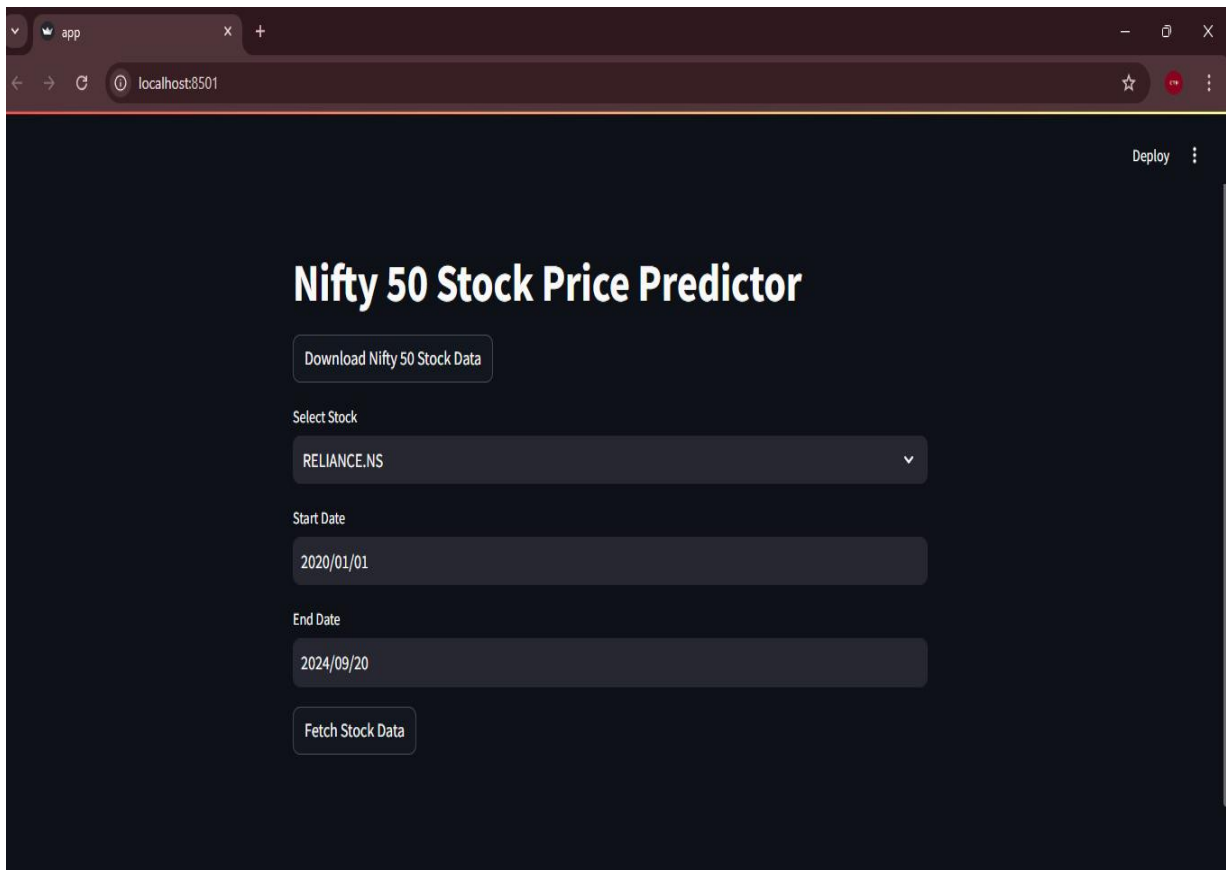
IV. RESULTS



```

File Edit Selection View Go Run ... nifty50_stock_predictor
EXPLORER Welcome requirements.txt app.py model.py utils.py
NIFTY50_STOCK_PREDICTOR app.py > ...
  > .venv
  > data
  > models
  app.py
  requirements.txt

1 # app.py
2 import streamlit as st
3 import pandas as pd
4 from models.utils import download_nifty50_data, get_stock_data
5 from models.model import train_model, predict_price
6 from datetime import datetime
7
8 # Title of the app
9 st.title("Nifty 50 Stock Price Predictor")
10
11 # Option to download dataset
12 if st.button("Download Nifty 50 Stock Data"):
13     st.write("Downloading stock data...")
14     data = download_nifty50_data()
15     st.write("Data downloaded successfully!")
16     st.write(data.tail())
17
18 # List of Nifty 50 stock symbols
19 nifty50_symbols = [
20     'RELIANCE.NS', 'TCS.NS', 'INFY.NS', 'HDFCBANK.NS', 'HINDUNILVR.NS',
21     'ICICIBANK.NS', 'KOTAKBANK.NS', 'SBIN.NS', 'BHARTIARTL.NS', 'ITC.NS',
22     'ASIANPAINT.NS', 'HCLTECH.NS', 'LT.NS', 'AXISBANK.NS', 'BAJFINANCE.NS',
23     'MARUTI.NS', 'HDFC.NS', 'ULTRACEMCO.NS', 'M&M.NS', 'SUNPHARMA.NS',
24     'TITAN.NS', 'WIPRO.NS', 'NESTLEIND.NS', 'POWERGRID.NS', 'BAJAJFINSV.NS',
25     'NTPC.NS', 'TATASTEEL.NS', 'JSWSTEEL.NS', 'GRASIM.NS', 'ADANIGREEN.NS',
26     'TECHM.NS', 'CIPLA.NS', 'HINDALCO.NS', 'ONGC.NS', 'BPCL.NS',
27     'BRITANNIA.NS', 'INDUSINBK.NS', 'DIVISLAB.NS', 'ADANIports.NS',
28     'TATAMOTORS.NS', 'EICHERMOT.NS', 'APOLLOHOSP.NS', 'DRREDDY.NS',
29     'COALINDIA.NS', 'HEROMOTOCO.NS', 'ADANIENR.NS', 'PIDILITIND.NS',
30     'SBILIFE.NS', 'BAJAJ-AUTO.NS', 'VEDL.NS'
31 ]
32
Ln 50, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.1 (.venv: venv) Go Live
  
```



V. CONCLUSION

The stock price predictor system exemplifies the effective utilization of machine learning in financial forecasting. By predicting stock prices for Nifty 50 companies, the project serves as a valuable resource for investors and analysts. The integration of the Linear Regression model with a user-friendly interface ensures accessibility for a broad audience. Although the current model employs simple linear regression, future explorations could involve experimenting with advanced models like Long Short-Term Memory (LSTM) networks to capture non-linear trends and enhance accuracy. Additionally, incorporating real-time data could position the system for intraday trading predictions. This project lays a strong foundation for further research into machine learning applications in finance and opens up pathways for future enhancements.

REFERENCES

- [1] M. Nabipour, P. Nayyeri, H. Jabani, S. S. and A. Mosavi, Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis, in *IEEE Access*, vol. 8, pp. 150199-150212, 2020, doi: 10.1109/AC-CESS.2020.3015966.
- [2] Y. Lin, S. Liu, H. Yang and H. Wu, Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme, in *IEEE Access*, vol. 9, pp. 101433-101446, 2021, doi: 10.1109/ACCESS.2021.3096825.
- [3] S. Saha, J. Gao and R. Gerlach, Stock Ranking Prediction Using List- Wise Approach and Node Embedding Technique, in *IEEE Access*, vol. 9, pp. 88981-88996, 2021, doi: 10.1109/ACCESS.2021.3090834.
- [4] C. Zhang, N. N. A. Sjarif and R. B. Ibrahim, Decision Fusion for Stock Market Prediction: A Systematic Review, in *IEEE Access*, vol. 10, pp. 81364-81379, 2022, doi: 10.1109/ACCESS.2022.3195942.
- [5] D. Napoleon, and M. Praneesh, "Detection of Brain Tumor using Kernel Induced Possiblistic C-Means Clustering," *International Journal of Computer Organization Trends and Technology (IJCOT)*, vol. 3, no. 5, pp. 40-42, 2013.
- [6] Anita Yadava, C K Jhaa, Aditi Sharan Optimizing LSTM for time series production in indian stock market in *Procedia Computer Science* 167 (2020) 20912100
- [7] Kyung Keun Yun, Sang Won Yoon, Daehan Won Prediction of Stock price direction using a hybrid GA-XG Boost algorithm with a three-stage feature engineering process in *Expert Systems with Applications* 186 (2021)115716
- [8] Sondo Kim, Seungmo Ku, Woojin Chang, and Jae Wook Song Predicting the direction of US stock prices using effective Transfer Entropy and ML techniques in *Digital Object Identifier* 10.1109/AC-CESS.2020.3002174
- [9] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Stock market prediction using machine learning in *Conference Paper*, December 2018 DOI: 10.1109/ICSCCC.2018.8703332
- [10] L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du and H. E. Stanley, Which Artificial Intelligence Algorithm Better Predicts the ChineseStock Market? in *IEEE Access*, vol. 6, pp. 48625-48633, 2018, doi: 10.1109/ACCESS.2018.2859809.
- [11] Jin, Zhigang Yang, Yang Liu, Yuhong. (2020). Stock closing price prediction based on sentiment analysis and LSTM in *Neural Computing and Applications*. 32. 10.1007/s00521-019-04504-2.