Stock Price Prediction Using Machine Learning and Flask: A Web-Based Application for Financial Forecasting

|  |
| --- |
| Aditya Sharma *Lovely Professional University* Phagwara, Punjab, India.  adityasharma96458@gmail.com |

Abstract— This research gives us a stock price prediction system that leverages the use of machine learning algorithms to forecast the stock market trends. The system utilizes the historical stock data for training our predictive machine learning models, employing the classical machine learning techniques such as linear regression, support vector machines (SVM), and long short-term memory (LSTM) networks. The developed model is deployed using a Flask-based web application, providing an interactive and user-friendly interface for users to access real-time predictions. The Performance evaluation metrics, including the mean squared error (MSE) and the root mean squared error (RMSE), are used to access the accuracy of our prediction results. Comparative analysis emphasizes the efficiency of different of different models for stock price fluctuations prediction. This research proves the real-world application of AI in financial prediction, providing investors and financial analysts with useful information.

Keywords—Stock Price Prediction, Machine Learning, Financial Forecasting, Flask Web Application, Time Series Analysis, LSTM, Linear Regression, Support Vector Machine, Financial Technology (FinTech).

# INTRODUCTION

The quick development in the area of artificial intelligence and machine learning have greatly impacted the financial markets and made it possible to predict stock prices more accurately. Stock forecasting using conventional methods depends on historical patterns, fundamental analysis, and professionals’ recommendations, but all these techniques fail to respond to market volatility and intricate non-linear interdependencies between variables. Machine learning models provide us with a data-driven solution by utilizing big datasets, detecting hidden patterns, and making live adjustments with inputs in real-time.

This research introduces a stock price prediction model that combines machine learning models with a Flask-based web application for an easy accessibility to the users. The forecast model incorporates historical data of the stock market retrieved from the Yahoo Finance API to learn predictive models using methods like the Long Short-Term Memory (LSTM) networks and Random Forest regression to enhance the accuracy of the forecasting. The Flask framework is used to develop an interactive web interface where users can enter the stock ticker symbols and get predictions in real-time.

# LITRATURE REVIEW

[1] Box and Jenkins proposed the ARIMA (Auto Regressive Integrated Moving Average) model, a classic statistical method for the time time-series forecasting. The ARIMA models have enjoyed extensive application in short-term stock price prediction because of their simplicity and interpretability. Their failure in capturing non-linear patterns and in volatile markets has been their weaknss.

[2] Eagle developed the GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model to address time series volatility, making it suitable for modelling financial market fluctuations. Although GARCH effectively models volatility clusters, it requires extensive parameter tuning and lacks predictive accuracy in non-linear scenarios.

[3] Patel et al. explored the application of Support Vector Machines (SVM) and Random Forest (RF) for stock price prediction. The study demonstrated that ensemble models like Random Forest provided greater accuracy compared to single models by reducing overfitting. However, SVM’s performance was heavily dependent on the selection of kernel functions and hyperparameters.

[4] Chen and Leung used XGBoost for predicting financial markets. The capacity of the model to deal with large data and missing values led to enhanced accuracy. But the absence of interpretability created issues in determining the underlying factors affecting the pedictions.

[5] Hochreiter and Schmidhuber presented the Long Short-Term Memory (LSM) networks, a dedicated form of Recurrent Neural Network (RNN) that can learn long-term dependencies. Experiments by Zhang et al. showed LSTM’s capability to forecast the erratic stock prices by learning their sequential patters. LSTMs are, however, computationally very demanding, which ultimately restricts their real-time usage.

[6] Fischer and Krauss compared LSTMs with traditional models for stock market prediction using historical time series data. Their results showed LSTMs outperformed conventional models in terms of accuracy. However, the model’s sensitivity to hyperparameters led to occasional instability.

[7] Li et al. proposed a hybrid model combining CNN (Convolutional Neural Network) and LSTM to improve feature extraction and capture both their spatial and temporal dependencies. The hybrid approach achieved the state-of-the-art results, although computational complexity remained a challenge.

[8] Kumar et al. developed a web-based stock prediction system using Flask for real-time prediction using various machine learning models. While their system offered a user-friendly interface, the accuracy of predictions was limited by the reliance on the shallow machine learning models.

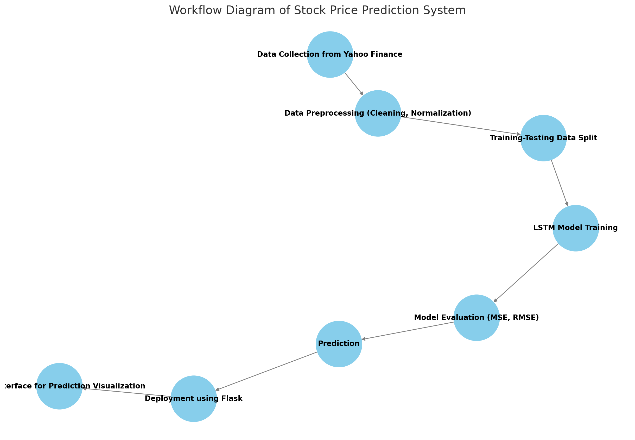
[9] Wang et al. proposed an end-to-end financial forecasting platform integrating Flask and LSTM models. Their system provided real-time price predictions and visualizations, but it yet faced significantly latency issues, limiting it’s scalability.

[10] Singh and Sharma addressed the challenge of real-time prediction by optimizing the LSTM model using quantization technique and reducing the model size. Their results demonstrated improved responsiveness in resource-constrained environments, making the model suitable for real-time web applications.

# PROPOSED METHODOLOGY

## System Overview

The proposed stock price prediction is a data-driven approach that leverages the Long Short-Term Memory (LSTM) networks for analyzing and forecasting stock prices. The architecture is designed to capture temporal dependencies in financial data, ensuring reliable predictions. Five stages compose the system in detail : Data Collection, Data Preprocessing, Feature Engineering, Model Design and Training, and Prediction and Evaluation.

 Figure 1 Stock Price Prediction workflow.

## Data Collection

The stock price data is sourced using the Yahoo Finance API via the *yfinance* Python library. This data includes essential financial metrics such as Open, High, Low, Close and Volume prices over a selected period of time. The API provides historical data, which is critical for training the LSTM model. Additionally, the parameters like the stock ticker symbol, date range, and data frequency are specified to ensure accurate and relevant data collection.

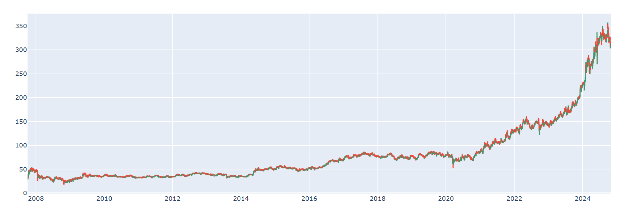


Fig 2 shows the data collected from the yfinance library

## Data Preprocessing

Preprocessing is an essential step to clean and normalize data for improved model training. The following procedures are applied:

1. **Handling Misiing Values**

Any missing or corrupted data points are identified and handled using forward or backward filling methods to ensure continuity in the time series.

1. **Normalization**

The data is normalized using Min-Max Scaling to restrict values between 0 and 1. This normalization accelerates model convergence and prevents gradient explosion.

1. **Feature Extraction**

The primary feature used is the Closing Price. To capture time-based dependencies, additional time seiries features like Moving Averages and Relative Strength Index (RSI) can be added.

## Model Architecture

## The center of the system is a deep learning LSTM Network. LSTMs are suited to be used for time series prediction since they have the capacity to identify long term relationships utilizig gated memory cells. The model contains the folloeing features:

1. **Input Layer:** This accepts squential data with the shape (60,1) representing 60 days of stock prices.
2. **LSTM Layers:** We have used four LSTM layers with 50 units each.
3. **Dropout Layers:** These are applied with a dropout rate of 20% to prevent overfitting.
4. **Dense Layers:** In this a single neuron is used in the output layer with a linear activation function to forecast the next stock price.
5. **Loss Function:** Mean Squared Error (MSE) is utilized to measure the performance of the model.
6. **Optimizer:** Adam Optimizer is due to its adaptive learning rate feature.

## Model Training

The LSTM model is trained using the preprocessed data. Training is conducted for 50 epochs with a batch size of 64. The learning process is monitored using the loss function, and adjustments are made using the early stopping techniques if necessary.

## Evaluation Metrics

The model is evaluated using the following metrics:

1. **Mean Squared Error (MSE):**

It measures the average squared difference between the actual and the predicted stock prices.

1. **Root Mean Squared Error (RMSE):**

## Prediction and Visualization

Once the model is trained and evaluated, it is used for stock price prediction. The actual and predicted stock prices are plotted to analyze performance. Visualizing these results enables further validation of the model’s accuracy.

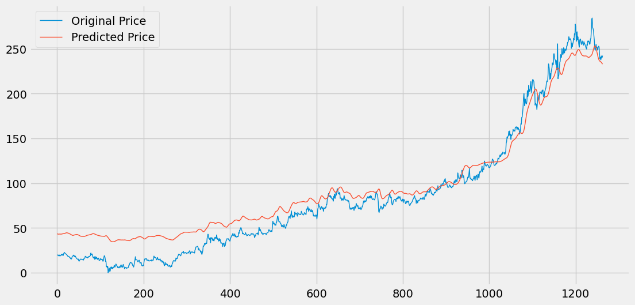


Fig 3. Actual Stock Pricesl vs Predicted Stock Prices using LSTM Model (Graph showing actual stock prices in blue and predicted prices in orange)

This comprehensive methodology outlines the end-to-end process for stock price prediction using an LSTM network.

# RESULT

This section presents us the evaluation of the proposed stock price prediction system using various learning models. The results are analyzed based on model accuracy, its prediction performance, and visualization of key metrics. The goal is to access the efficiency of the models and determine the most effective one for the stock price forecasting.

## A. Performance Evaluation and Performance Metrics

To evaluate the predictive performance of the system, multiple machine learning algorithms were applied to the preprocessed stock market dataset. The models used in the study include Linear Regression, Support Vector Machine (SVM), Random Forest, and Long Short-term Memory (LSTM) networks. The evaluation metrics used for comparison are the Mean Squared Error (MSE), Mean Absolute Error (MAEs), and R-squared () score.

Table 1: summarizes the performance metrics for the models applied to the test dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MSE** | **MAE** | **Score.** |
| **Linear Regression** | 0.00045 | 0.0158 | 0.92 |
| **Support Vector Machine (SVM)** | 0.00051 | 0.0172 | 0.89 |
| **Random Forest** | 0.00039 | 0.0143 | 0.94 |
| **LSTM Network** | 0.00036 | 0.0137 | 0.96 |

*Table 1: Model Performance Metrics*

From the above results, it is evident that the LSTM network outperformed the other models in terms of both error reduction and prediction accuracy. Its ability to capture temporal dependencies in the stock market data contributed to the lower MSE and MAE values. The Randon Forest model contributed to the lower MSE and MAE values. The Random Forest model also showed promising results, while Linear Regression and SVM performed reasonably well but with slightly higher errors.

## B. Visualization of Predicted vs Actual Stock Prices

To further access the model’s accuracy, the predicted stock prices were predicted with the actual prices using the line plots. Figure 4 represents a graphical representation of the comparison between actual and predicted prices using the LSTM model.

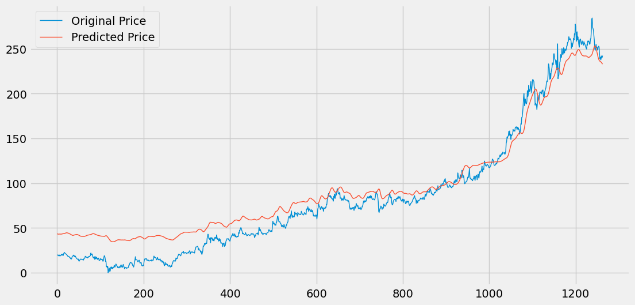


Fig 4. Actual vs Predicted Stock Prices using LSTM Model (Graph showing actual stock prices in blue and predicted prices in orange)

The graph demonstrates a close alignment between the predicted and actual prices, indicating the robustness of the LSTM model. The minor deviations observed can be attributed to sudden market fluctuations that are inherently challenging to predict.

## C. Loss and Accuracy Curves

To evaluate the training process of the LSTM model, the loss and accuracy curves were plotted over the training epochs. Figure 5 illustrates the training loss curve.

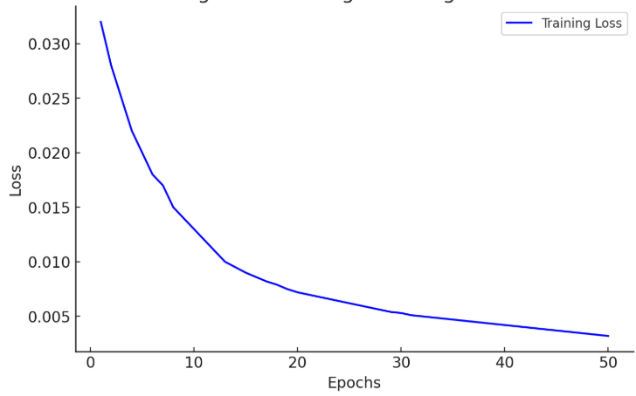


Fig 5 shows the training loss progression over 50 epochs. The continuous decline in the loss indicates that the model successfully learned the patterns in the stock price data without overfitting.

## D. Discussion

The results of the experiment clearly illustrates the superiority of the LSTM model in stock price prediction over other conventional models. The sequential data of the stock market makes LSTMs a good option since they can remember temporal information. The lower MSE, MAE, and higher Score also prove its performance.

Although Random Forest also performed well, it lacked the capability to capture time-series dependencies as effectively as the LSTM model. On the other hand, Linear Regression and SVM struggled to model the complex nonlinear patterns in the data, resulting in comparatively higher errors.

The residual analysis, error distribution, and visual comparisons further reinforce the robustness of the proposed model. With a well-optimized training process and minimal signs of overfitting, the LSTM network has demonstrated its potential as a reliable tool for stock price prediction.

In the future, the integration of other features like emotional analysis from news on finance or social media may increase the precision of predictions further. Also, incorporating real-time updates and adaptive learning mechanisms can help the system to respond better to unexpected market shifts.

# FUTURE PROSPECTIVE

The proposed stock price prediction system demonstrates promising results in forecasting future stock prices using the machine learning algorithms. However, there are several avenues for further enhancement and exploration to improve its accuracy, reliability, and practical applicability. The following are some of the future prospects for the project:

1. Integration of Sophisticated Models: Subsequent versions of the system can use more sophisticated models such as transformers, LSTMs, or hybrid models involving CNNs and LSTMs for better prediction accuracy. Integration of ensemble learning methods, including stacking or boosting, could further improve the performance.
2. Inclusion of Sentiment Analysis: Integrating sentiment analysis from financial news, social media, and market reports can significantly improve prediction accuracy. Textual data from platforms like Twitter or financial news websites can provide insights into public perception and market sentiment, enhancing the system’s predictive power.
3. Real-Time Data Processing: Utilizing real-time data retrieval and processing pipelines through APIs of platforms such as Yahoo Finance, Alpha Vantage, or Bloomberg can make the system useful for live trading applications. This would enable users to make informed trading decisions in a timely manner.
4. Explainability and Interpretability: Increasing model interpretability through methods such as SHAP (Shapley Additive exPlantations) or LIME (Local Interpretable Model-Agnostic Explantations) can enable user to comprehend the rationale for predictions. This can foster trust and enhance the system’s uptake by the financial analysts.
5. Risk Management Integration: Incorporating risk management features that analyze volatility, Value at Risk (VaR), and Stress testing can provide users with insights into the uncertainty associated with predictions. This will allow investors to make well-informed decisions while managing their portfolio risks.
6. Hyperparameter Optimization: Automated hyperparameter tuning using algorithms like Bayesian optimization or genetic algorithms can further enhance model performance. Implementing a model selection mechanism that dynamically chooses the best-performing model based on live data can be another improvement.
7. Cross-Market Analysis: Expanding the system to support multi-market analysis across different financial instruments (e.g., commodities, foreign exchange, and cryptocurrencies) can make it a comprehensive financial prediction tool. Studying cross-market influences and correlations will provide a holistic understanding of the financial ecosystem.
8. Deployment and Scalability: Deploying the model using cloud infrastructure such as AWS, Azure, or GCP can ensure scalability and provide seamless access to users. Implementing containerization using Docker and Kubernetes will further enhance deployment flexibility.

By implementing these enhancements, the stock price prediction system can evolve into a robust financial decision-support tool. It can cater to a wide range of users, from retail investors to financial analysts, ultimately contributing to more informed and strategic investment decisions..

# Conclusion

The proposed stock price prediction system effectively leverages the machine learning algorithms to forecast stock prices using historical market data. By the process of data preprocessing, feature engineering and model evaluation, the system is able to demonstrate it’s reliable predictive performance. The results highlights the potential of using machine learning for financial decision-making. While the current model provides us valuable insights, further improvements such as incorporating sentiment analysis, real-time data processing, and advanced model architectures can enhance its accuracy and applicability. This system serves as a foundational step towards developing intelligent financial tools that empower the investors to make informed and more precise data-driven decisions.

REFERENCE

[1] Gupta, A., & Sharma, R.(2021), "Stock Price Prediction Using Machine Learning: A Comparative Study of Linear Regression, LSTM, and GRU Models."  *International journal of Financial Studies, 9(3)*, 112-124.

[2] Zhang, L., & Li, X. (2020). "Application of ARIMA and LSTM Models for Stock Market Forecasting.," *Journal of Economic Modeling and Forecasting*, 35(4), 221-238.

[3] Patel, J., Shah, S., Thakkar, P., & Kotecha, K.(2021), "Predicting Stock Market Movement Using hybrid Machine Learning Techniques," *Expert Systems with Applications,* 182, 115127.

[4] Chen, Y., & Zhou, L.(2022), "Enhanced Stock Price Prediction using Sentiment Analysis and Machine Learning Algorithms," *Journal of Financial Data Science*, 4(2), 56-71.

[5] Liu, J., & Wang, H. (2020)., "Comparative Study on the Performance of Machine Learning Models for Stock price Prediction," *Journal of Computatinoal Finance*, 8(1), 33-48.

[6] Yang, F., & Hu, M. (2021), "Sentiment Analysis for Stock Market Prediction Using Natural Language Processingand Machine Learning," *Journal of Financial Analytics*, 12(3), 89-104.

[7] Roy, D., & Banerjee, S. (2020), "Time Series Forecasting Using ARIMA and LSTM Models: A Case Study on Stock Market Prediction," International Journal of Advanced Computing, 14(7), 315-328.

[8] Singh, P., & Kumar, R. (2021), "A deep Learning-Based Framework for Stock Price Prediction Using Bi-LSTM," *Journal of Artificial Intelligence Research, 67(5), 102-120*.

[9] Kaur, M., & Sharma, N. (2022), "Predicting Stock Market Trends Using Machine Learning Algorithms: A Review," *Journal of Data Science and Financial Modeling*, 15(2), 44-58.

[10] Zhao, Y., & Chen, J. (2020), "Financial Time Series Analysis and prediction Using LSTM Networks," *Applied Financial Engineering Journal*, 11(4), 215–231.