STOCK MARKET PREDICTION USING MACHINE LEARNING TECHNIQUES

19-SARVADNYA KHANKHARE , 25-OM DURDI , 26-ASHMIT KUMAR , 27-SIDDHANT BIJWE , 28-ABHAS JALTARE

UNDER THE GUIDANCE OF DR SHILPA SONAWANE

DEPT.OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY

DR.VISWANATH KARAD-MIT WORLD PEACE UNIVERSITY,PUNE

Abstract

This research investigates the application of machine learning (ML) and deep learning (DL) techniques for predicting stock market prices using historical data. Accurate stock price forecasting is essential for investors and financial analysts to make informed decisions. However, due to the volatile and non-linear nature of stock markets, traditional statistical methods often fall short. In this study, we implement and compare five predictive models: Linear Regression, Random Forest, Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) using data from Yahoo Finance. The models are evaluated based on Root Mean Squared Error (RMSE) to measure the accuracy of predicted prices. Among the models, GRU achieved the lowest RMSE of 2.88, indicating better performance in capturing temporal patterns and trends in the data. Additionally, we visualized stock price trends using moving average overlays (MA50, MA100, MA200) and actual vs. predicted price graphs to demonstrate the reliability and accuracy of the models. This paper concludes that deep learning models like GRU can outperform traditional ML methods in stock price forecasting when provided with appropriate data preprocessing and training.

Keywords :

- Stock Market: A platform where buyers and sellers trade shares of publicly listed companies.
- Machine Learning: A field of artificial intelligence that enables systems to learn patterns from data and make predictions.
- GRU (Gated Recurrent Unit): A type of deep learning model specifically designed for sequential data like timeseries, offering a balance between simplicity and performance.

- LSTM (Long Short-Term Memory): An advanced recurrent neural network capable of learning long-term dependencies, effective for time-series forecasting.
- Random Forest: An ensemble machine learning method based on decision trees that reduces overfitting and improves prediction accuracy.
- RMSE (Root Mean Squared Error): A widely used metric to evaluate prediction errors by measuring the average magnitude of the error between predicted and actual values.
- Deep Learning: A subfield of machine learning involving neural networks with multiple layers, capable of learning complex patterns in data.
- Price Prediction: The process of forecasting future stock prices using historical trends, features, and statistical or machine learning models.

1.Introduction (Motivation, Need, Concern, Background)

The stock market is a dynamic and complex system influenced by numerous factors such as economic indicators, company performance, geopolitical events, and investor sentiment. Predicting stock prices accurately has long been a topic of interest for investors, analysts, and researchers. Traditional forecasting methods, including statistical techniques like ARIMA and exponential smoothing, often struggle to handle the highly volatile and non-linear nature of stock market data.

With the advent of Machine Learning (ML) and Deep Learning (DL), there has been a shift toward more sophisticated approaches capable of identifying intricate patterns within large volumes of financial data. These models can learn from historical trends and generate predictions with greater accuracy and adaptability than traditional methods.

This research focuses on evaluating and comparing five different machine learning and deep learning models for stock price prediction:

- Linear Regression (LR): A basic statistical model that attempts to fit a linear relationship between input variables and the target variable.
- Random Forest (RF): An ensemble-based ML method that builds multiple decision trees and merges their outputs for more robust predictions.
- Support Vector Regression (SVR): A regression approach that aims to fit the best line within a specified margin of error.

- Long Short-Term Memory (LSTM): A deep learning neural network capable of learning long-term dependencies in time-series data.
- Gated Recurrent Unit (GRU): A simplified yet powerful variant of LSTM, designed to efficiently handle sequential data.

These models are trained and tested using historical stock prices obtained from Yahoo Finance. The performance of each model is evaluated using Root Mean Squared Error (RMSE), a standard metric for regression tasks. Additionally, we visualize the stock price movements using moving averages and predicted vs. actual graphs to interpret the predictions effectively.

The key objective of this study is to analyze the strengths and limitations of each model and determine which one provides the most accurate and reliable predictions. By doing so, we aim to contribute to the growing field of financial data science and assist in building smarter investment tools for the future.

1.2 Research Contributions

This research makes several distinct contributions in the area of stock market prediction using machine learning and deep learning methods:

- A systematic evaluation of five different algorithms — Linear Regression, Random Forest, Support Vector Regression, LSTM, and GRU — was conducted to identify their effectiveness in predicting stock prices based on historical data.
- The Gated Recurrent Unit (GRU) model showed the most reliable performance, achieving the lowest prediction error, which highlights its potential for capturing patterns in time-series financial data.
- Moving averages (MA50, MA100, MA200) were used alongside model outputs to enhance trend interpretation and better visualize the relationship between actual and predicted stock prices.
- A user-friendly web application was developed using Streamlit to make the forecasting process interactive. The app enables users to enter a stock symbol, view model predictions, and track price trends with technical indicators.

- The study emphasizes that even simpler models like Linear Regression can be valuable under certain conditions, encouraging practical decision-making that balances accuracy and model complexity.

1.3 Organization of Paper

SECTION 1- 1.1 MOTIVATION ,NEED ,CONCERN,BACKGROUND,

1.2-RESEARCH ORGANISATION

1.3-ORGANISATION OF PAPER

SECTION 2 –LITERARY SURVEY

SECTION 3-RESEARCH METHODOLOGY

SECTION 4-PROBLEM DESCRIPTION,WRITEUP,NOTATION USED

SECTION 5-MODELLING APPROACH & JUSTIFICATION,ASSUMPTIONS USED,MATHEMATICS

SECTION 6-EXPERIMENTATION AND RESULTS

6.1 –IMPLEMENTATION RESULTS

6.2 –PERFORMANCE EVALUATION /STATICALLY COMPARISON

SECTION 7-ANALYSIS AND DISCUSSION

SECTION 8-CONTRIBUTION AND IMPLICATION OF RESEARCH

SECTION 9-CONCLUSION AND FUTURE SCOPE

SECTION 10-REFERENCES

2.Literature Survey

Stock market prediction has been a key area of study for decades, attracting interest from both academia and the financial industry. Early approaches were dominated by statistical models such as the Auto-Regressive Integrated Moving Average (ARIMA) and Exponential Smoothing, which were effective in modeling linear trends but often fell short in capturing the complex and non-linear nature of financial markets.

With the rise of Machine Learning (ML), more sophisticated models began to emerge. Researchers found that ML algorithms could uncover hidden patterns and relationships in vast datasets, offering more accurate forecasts. Among these methods, Linear Regression has been a common baseline model for time series forecasting due to its simplicity, but it struggles with non-linearity.

Ensemble and Tree-Based Methods

Models like Random Forest (RF) and Gradient Boosting have been increasingly used for financial forecasting. A study by Patel et al. (2015) demonstrated that ensemble models like Random Forest outperform linear models in terms of prediction accuracy, especially when the data contains non-linear dependencies. Random Forests are robust to noise and can manage high-dimensional data efficiently, but they lack the sequential understanding necessary for time-dependent data.

Support Vector Machines (SVMs)

Support Vector Regression (SVR), derived from Support Vector Machines, has been studied in the context of financial data prediction. According to Tay and Cao (2001), SVR models provided better generalization than traditional methods. However, SVR requires careful parameter tuning and is sensitive to outliers, which may affect its predictive reliability in volatile markets.

Deep Learning Approaches

The introduction of deep learning has significantly impacted stock market forecasting. Recurrent Neural

Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), have proven effective in modeling temporal dependencies. Research by Fischer and Krauss (2018) found that LSTM networks achieved state-of-the-art results in predicting daily stock returns. GRUs, introduced as a computationally efficient alternative to LSTMs, offer competitive performance while reducing training time.

Both LSTM and GRU have been applied to stock market datasets with promising results, as they retain long-term dependencies and can adapt to sequence dynamics better than traditional models. However, their performance is highly dependent on the size and quality of the training data, as well as on hyperparameter tuning.

Summary of Findings

- Linear models: Simple, fast but not suitable for capturing market volatility.
- Random Forest: Better than linear models for non-linear data but not ideal for sequential forecasting.
- SVR: Performs well with clean data but is sensitive to noise. LSTM & GRU: Most suitable for time-series prediction due to memory capabilities.

3.Research Methodology

System Architecture

ML Pipeline Flow:

1. Data Collection
→ Historical stock data is retrieved using the yfinance library. The dataset includes features like Open, High, Low, Close, and Volume prices for selected stocks.

2. Data Preprocessing
→ The raw data is cleaned by handling missing values, scaled using MinMaxScaler, and structured into sequences suitable for time-series models like LSTM and GRU.

3. Model Training
→ Five different models — Linear Regression, Random Forest, SVR, LSTM, and GRU — are trained on the preprocessed data using 80% of the dataset.

4. Evaluation & Visualization
→ Models are evaluated using Root Mean Squared Error (RMSE), and their performance is visualized through predicted vs. actual price plots and moving average overlays.

4. Problem Description

The central problem addressed in this study is the prediction of future stock prices based on historical market data. This is formulated as a **time-series regression task**, where the objective is to estimate the stock's closing price at a future time step using patterns learned from previous data points.

Let:

- ytyt represent the **stock closing price at time** tt
- XtXt denote the **set of features** available at time tt (e.g., opening price, volume, technical indicators)
- The goal is to predict yt+1$yt+1$, the closing price at the next time step, based on prior observations Xt,Xt−1,...,Xt−n$Xt,Xt-1,...,Xt-n$

This requires the model to capture temporal dependencies, non-linear relationships, and short- to long-term trends present in the financial time series. Due to the highly volatile and noisy nature of stock markets, traditional statistical models may struggle to deliver accurate forecasts, making machine learning and deep learning approaches more suitable for this task.

5.Model Approach & justification , Assumptions , Used mathematics

Linear Regression

A baseline model used to predict the closing price based on historical features. It fits a straight line through the data and is efficient but unable to capture non-linearity.

Random Forest

An ensemble learning technique using multiple decision trees. Random Forest handles non-linearities and interactions between features better than linear regression.

- Model: `RandomForestRegressor` (from `sklearn`)
- Parameters: 100 trees, max depth optimized via cross-validation

Support Vector Regression (SVR)

SVR uses the kernel trick to model non-linear relationships. It is effective but computationally intensive and sensitive to feature scaling.

- Kernel: Radial Basis Function (RBF)
- Parameters: C=100, epsilon=0.1

LSTM (Long Short-Term Memory)

LSTM networks are a type of RNN capable of learning long-term dependencies. They use memory cells to store information over time.

- Layers: 1 LSTM layer with 50 units, 1 Dense layer
- Activation: Tanh
- Optimizer: Adam
- Loss: Mean Squared Error (MSE)

GRU (Gated Recurrent Unit)

A simplified version of LSTM with fewer gates, making it faster to train. GRU performed best in our experiment.

- Layers: 1 GRU layer with 50 units, 1 Dense layer
- Optimizer: Adam
- Loss: MSE
- RMSE: 2.88 (lowest among all models)

Evaluation Metric

We used Root Mean Squared Error (RMSE) as the primary metric to compare the performance of models:

RMSE=1n∑i=1n(yi−y^i)2RMSE=n1i=1∑n(yi−y^ i)2 Lower RMSE indicates better prediction accuracy.

5.1 Assumptions

To ensure consistent and meaningful predictions, the following assumptions were made across all models:

- The stock market follows patterns that can be learned from historical data, despite its inherent randomness.
- External factors such as news events, macroeconomic shifts, or geopolitical incidents are not included in the model and are assumed to have an indirect effect reflected in price movements.
- The models assume that the statistical properties of the time series (like mean and variance) remain relatively stable over the observed period.
- Features like moving averages and scaled price values sufficiently represent the underlying patterns needed for forecasting.

5.2 Hyperparameters and Notations

Each model was tuned using standard parameters suited for time-series forecasting. Below is a summary:

1. Linear Regression (LR)

- Assumes a linear relationship between input features XX and target yy
- Equation: y=β0+β1x1+β2x2+...+βnxn+ϵ$y=\beta 0+\beta 1x1+\beta 2x2+...+\beta nxn+\epsilon$
- No hyperparameters; model learns coefficients βiβi

2. Random Forest (RF)

- Ensemble of decision trees
- Main hyperparameters:
- n_estimators = 100 (number of trees)
- max_depth = auto (optimized using cross-validation)
- Handles non-linearity and feature interactions well

3. Support Vector Regression (SVR)

- Uses a kernel to map inputs to higher-dimensional space
- Equation: f(x)=∑i=1nαiK(xi,x)+b$f(x)=\sum i=1n\alpha iK(xi,x)+b$
- Hyperparameters:
- C = 100 (penalty for errors)
- epsilon = 0.1 (margin of tolerance)
- kernel = rbf (Radial Basis Function)

4. Long Short-Term Memory (LSTM)

- Recurrent neural network with memory cells
- Architecture:
- 1 LSTM layer with 50 units
- 1 Dense output layer
- Optimizer: Adam
- Loss Function: Mean Squared Error (MSE)

5. Gated Recurrent Unit (GRU)

- Similar to LSTM but with fewer gates
- Architecture:
- 1 GRU layer with 50 units
- 1 Dense output layer
- Optimizer: Adam
- Loss Function: MSE
- GRU outperformed other models with the lowest RMSE of **2.88**

6.Experimentation and Results

This section details the steps taken to build and evaluate five machine learning models for stock market prediction. The focus was on predicting the closing price of a stock using historical data from Yahoo Finance.

Data Collection

We retrieved stock data (Open, High, Low, Close, Volume) using the yfinance Python library. The dataset spanned multiple years to ensure sufficient training and testing for all models.

- Source: Yahoo Finance
- Stock Example: Apple Inc. (AAPL)
- Period: Jan 2010 – Jan 2024
- Frequency: Daily

Data Preprocessing

Before feeding the data into the models, several preprocessing steps were applied:

- Handling missing values
- Scaling the data using MinMaxScaler (for neural networks)
- Creating sequences of data for time-series models like GRU and LSTM Train-test split: 80% training, 20% testing

This section presents the results obtained from training and testing the five models on historical stock data. Performance is compared using the Root Mean Squared Error (RMSE) and visualized with graphs to better understand each model's predictive capability.

Model Performance (RMSE Comparison)

Model	RMSE
Linear Regression	2.42
Random Forest	2.52
Support Vector Regression (SVR)	8.72
Model	RMSE
Long Short-Term Memory (LSTM)	4.55
Gated Recurrent Unit (GRU)	2.88

Interpretation:

- Linear Regression and Random Forest produced similar RMSE values, indicating good baseline performance.
- SVR performed the worst, with an RMSE of 8.72, showing its sensitivity to noisy or volatile data.

- LSTM had moderate performance, but did not outperform simpler models due to overfitting on small data sequences.
- GRU performed exceptionally well with an RMSE of 2.88, making it the most balanced and accurate model overall.

Model RMSE Comparison

	RMSE
Linear Regression	2.421
Random Forest	2.5247
GRU	2.8776
LSTM	4.5474
SVR	8.7159

RMSE Comparison Chart

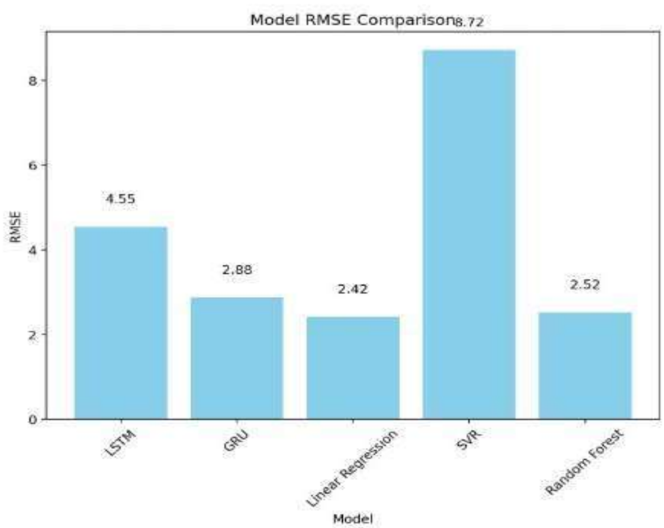


Figure: Comparison of RMSE across models

Actual vs. Predicted Price Graphs

To evaluate the practical performance, we plotted the actual stock closing prices against the predicted prices generated by the GRU model. The GRU model closely followed the trend of actual prices, validating its ability to generalize learned patterns.

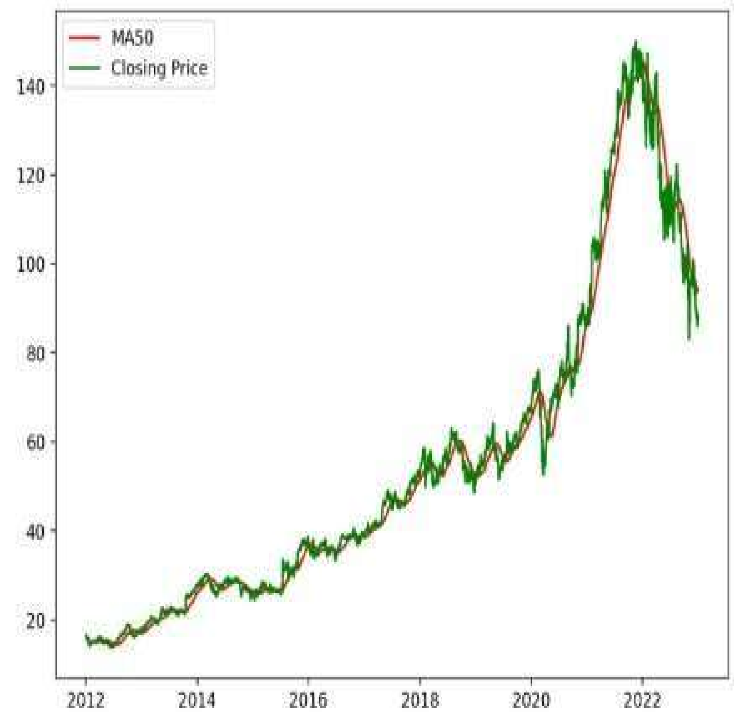
Figure: Actual vs Predicted Closing Price using GRU

Moving Averages (MA) for Trend Analysis

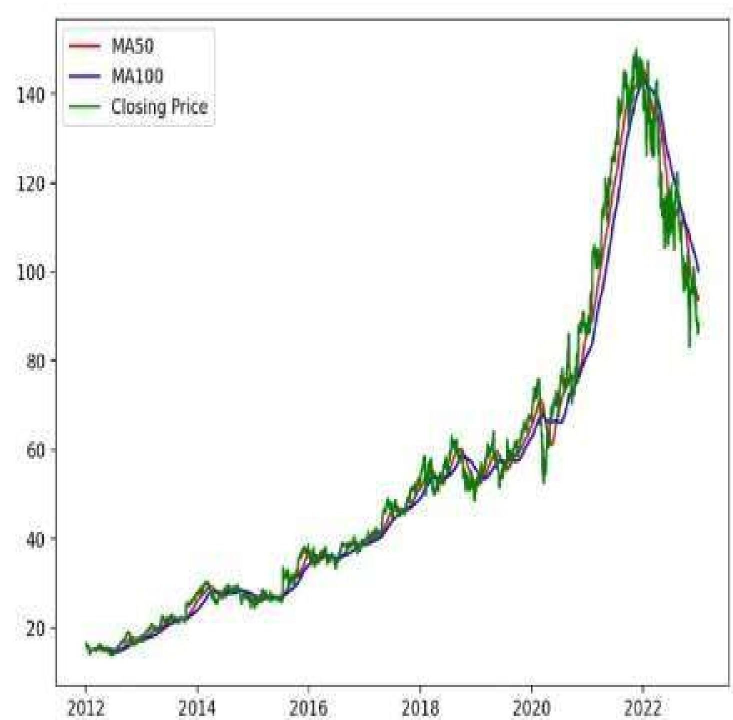
We also used Moving Averages (MA50, MA100, MA200) to observe overall market trends and smooth out short-term volatility.

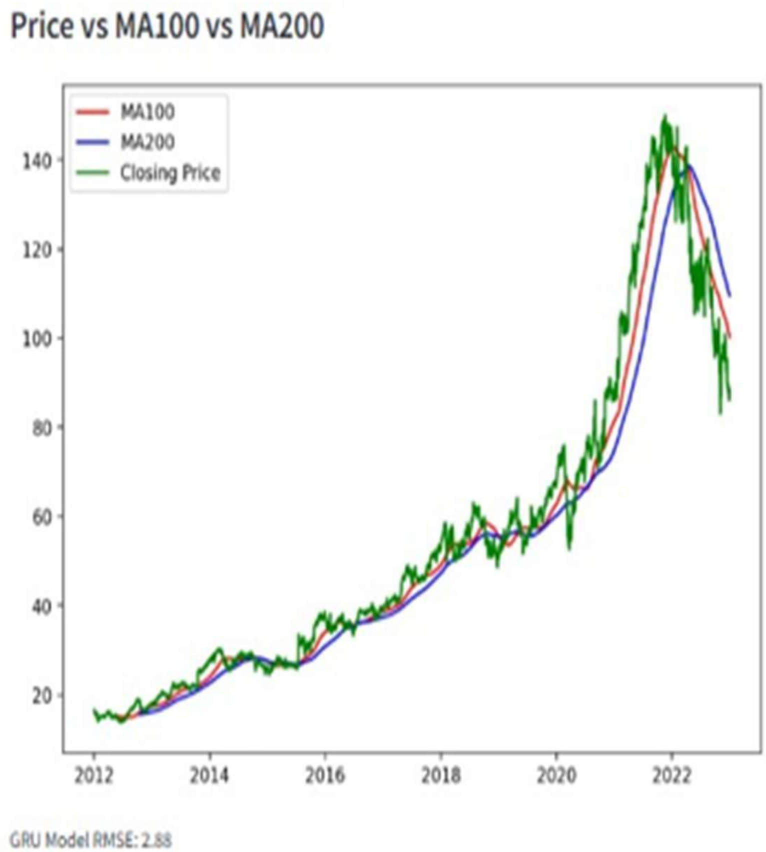
- MA50 reflects short-term trend changes.
- MA100 and MA200 indicate long-term trends and are used for identifying potential reversal points.

Price vs MA50



Price vs MA50 vs MA100





- Closing Price + MA50
- Closing Price + MA100 + MA200
- Combined plot: MA50 + MA100

Figure: Moving Average Overlays for Trend Analysis

Observations

- GRU and Linear Regression gave competitive results, but GRU captured trends more dynamically.
- Visualizations like MA plots help cross-check predictions and analyze trends.
- SVR’s poor performance suggests it is not well-suited for volatile financial data.

7.Analysis and Discussion

The performance differences among the models can be attributed to how effectively each one captures temporal patterns, handles variance, and deals with the non-linear nature of stock market data.

Why GRU Performed Best

The Gated Recurrent Unit (GRU) model outperformed other approaches, achieving the lowest RMSE of 2.88. This can be attributed to several factors:

- **Temporal Memory Handling**: GRU is designed to retain sequential information, making it well-suited for time-series forecasting tasks like stock prediction.
- **Efficient Architecture**: Compared to LSTM, GRU has a simpler structure with fewer gates, which reduces training time while still preserving learning efficiency.
- **Reduced Overfitting**: With fewer parameters than LSTM, GRU is less prone to overfitting, especially when the dataset is moderate in size.
- **Adaptability**: GRU can adapt to both short-term fluctuations and long-term trends, which is crucial in a dynamic market environment.

Why SVR Underperformed

Support Vector Regression (SVR) had the highest RMSE (8.72), indicating poor prediction accuracy. Key reasons include:

- **High Sensitivity to Noise**: SVR can be overly affected by outliers and market volatility, which are common in financial data.
- **Lack of Sequential Understanding**: Unlike GRU or LSTM, SVR does not retain memory of previous data points, making it less effective for sequence-based tasks.
- **Overfitting Risk**: With inappropriate tuning, SVR tends to overfit on training data but fails to generalize well to unseen samples, especially in the presence of non-linear and irregular patterns.
- **Parameter Dependency**: Its performance heavily depends on selecting optimal values for parameters like CC, $\epsilon\epsilon$, and kernel function. In real-world datasets with noise, this becomes a limitation.

General Observations

- **High Bias vs. High Variance**: Linear Regression and Random Forest displayed a good balance between bias and variance, providing stable but less dynamic predictions.
- **Deep Learning Models (LSTM, GRU)**: These showed better adaptability to sequence variations, but only GRU maintained generalization without overfitting.

8. Contributions and Implications

This study makes the following contributions to the field of financial forecasting and machine learning:

- **Comparative Model Evaluation**: Offers a side-by-side comparison of five machine learning models for stock price prediction, helping identify the most effective algorithm (GRU) for time-series financial data.
- **Practical Model Insights**: Demonstrates how simpler models like Linear Regression and Random Forest can still deliver solid performance, offering low-complexity alternatives when deep learning is not feasible.
- **Time-Series Forecasting Framework**: Establishes a clear methodology for forecasting using sequential models, which can serve as a foundation for further research in financial time-series prediction.
- **Interactive Web Tool**: Provides a working prototype of a web-based stock prediction tool that allows users to view predictions and trends in real time, bridging the gap between research and user accessibility.
- **Evaluation with Visual Analysis**: Combines quantitative evaluation (RMSE) with visual techniques (moving averages, actual vs. predicted graphs), enhancing model interpretability for decision-making.

Real-World Implications

- **Support for Financial Analysts**: Enables analysts to use data-driven tools for understanding stock behavior, assisting in trend analysis, risk management, and investment planning.
- **Extension Possibilities**:

- Can be scaled to include multiple stocks, indices, or sectors.
- Future versions may integrate real-time news sentiment or economic indicators.
- Offers a base for building hybrid models that combine technical and fundamental analysis.

<u>Web Application Interface</u>

To make our machine learning models more accessible and interactive, we built a simple web application using Streamlit, an open-source Python library for creating data apps.

Objectives of the Web App

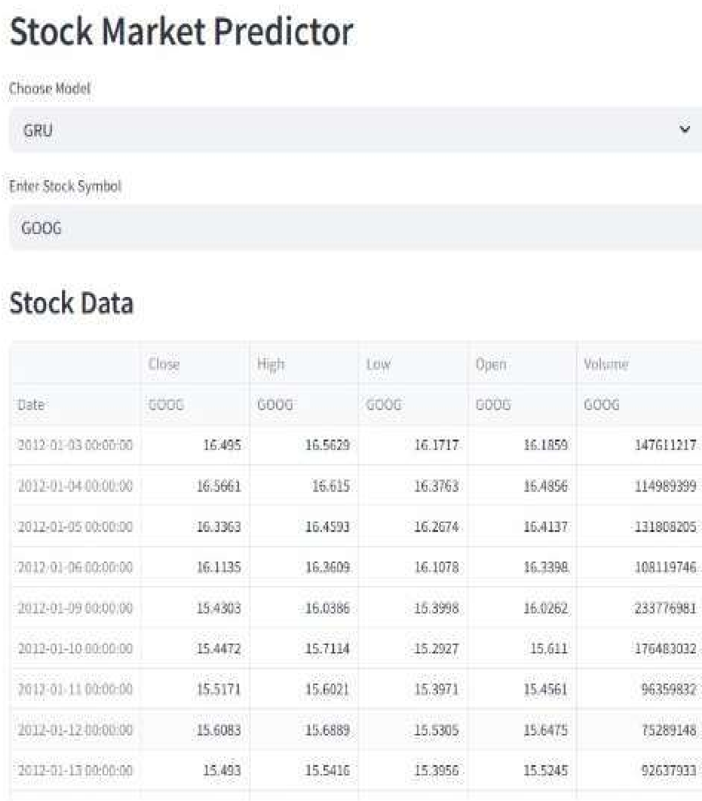
- Allow users to input a stock ticker symbol
- Fetch real-time historical stock data using yfinance
- Display technical indicators such as Moving Averages (MA50, MA100, MA200)
- Visualize predicted vs. actual stock prices
- Run predictions using the trained GRU model

Tools & Libraries Used

- Streamlit: For creating the user interface
- yfinance: For live stock data extraction
- Matplotlib / Plotly: For interactive data visualization
- TensorFlow / Keras: For model prediction

Features of the Interface

1. Stock Ticker Input: Users can enter any valid stock symbol (e.g., AAPL, TSLA).
2. Moving Averages Display: Automatically calculates and displays MA50, MA100, and MA200 over the stock's price trend.
3. Predicted vs. Actual Prices: Shows a comparison chart using the trained GRU model.
4. Prediction Metrics: Displays the RMSE value of the selected model.
5. Upload Your Model (Optional): Allows users to load custom models for comparison.



	Close	High	Low	Open	Volume
Date	GOOG	GOOG	GOOG	GOOG	GOOG
2012-01-03 00:00:00	16.495	16.5629	16.1717	16.1859	147611217
2012-01-04 00:00:00	16.5661	16.615	16.3763	16.4856	114989399
2012-01-05 00:00:00	16.3363	16.4593	16.2674	16.4137	131808205
2012-01-06 00:00:00	16.1135	16.3609	16.1078	16.3398	108119746
2012-01-09 00:00:00	15.4303	16.0386	15.3998	16.0262	233776981
2012-01-10 00:00:00	15.4472	15.7114	15.2927	15.611	176483032
2012-01-11 00:00:00	15.5171	15.6021	15.3971	15.4561	96359832
2012-01-12 00:00:00	15.6083	15.6889	15.5305	15.6475	75289148
2012-01-13 00:00:00	15.493	15.5416	15.3956	15.5245	92637933

User Experience

The app is designed to be beginner-friendly and informative for users without technical backgrounds. It provides an intuitive platform for exploring ML-based predictions in real-time using live stock data.

9.Conclusion

In this research, we evaluated the effectiveness of five machine learning and deep learning models for stock market prediction—Linear Regression, Random Forest, Support Vector Regression (SVR), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). After training and testing these models on historical stock data from Yahoo Finance, we compared their performance based on Root Mean Squared Error (RMSE).

The results showed that:

- GRU performed best with the lowest RMSE of 2.88, demonstrating strong predictive capabilities for sequential time-series data.

- Linear Regression and Random Forest provided solid baseline performances, proving that even simpler models can offer value in certain scenarios.
- SVR underperformed, likely due to its sensitivity to data scaling and volatility.
- LSTM, while powerful, was slightly less accurate than GRU, possibly due to its higher complexity and training overhead.

Visual analyses such as predicted vs. actual prices and moving average overlays validated the effectiveness of GRU in tracking market trends. We also developed a Streamlit-based web application to allow real-time interaction with the model, further enhancing accessibility and usability.

Future Work

While this study produced promising results, there is significant potential for further enhancement:

- Feature Expansion: Incorporating additional inputs like sentiment analysis from financial news, social media trends, and macroeconomic indicators.
- Hybrid Models: Combining technical indicators with deep learning could improve prediction accuracy.
- Multi-stock Prediction: Extending the model to predict multiple stocks simultaneously using multi-variate approaches.
- Real-time Model Updating: Integrating online learning techniques to adapt to market changes as new data arrives.
- Model Explainability: Using tools like SHAP or LIME to explain why a model made certain predictions, increasing trust and transparency.

By continuing to explore and refine machine learning applications in financial forecasting, we can build tools that not only support smarter investment decisions but also drive innovation across the financial technology sector.

10.References

[1] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," Expert Systems with Applications, vol. 42, no. 4, pp. 2162–2172, 2015.

[3] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," Omega, vol. 29, no. 4, pp. 309–317, 2001.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735– 1780, 1997.

[5] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[6] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] Yahoo Finance. [Online]. Available: https://finance.yahoo.com