**Stock Market Prediction using Deep Learning**

**A report on**

**Deep Learning Lab Project**

**[CSE-3281]**

Submitted By

**Tareen Khan - 210962190**

**Shambhavi Sahay - 210962047**

**Simrat Singh - 210962208**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**April 2024**

**Stock Market Prediction using Deep Learning**

Tareen Khan, Shambhavi Sahay, Simrat Singh

Computer science and engineering (artificial intelligence and machine learning)

**Abstract***— This research explores the potential of deep learning to improve stock market prediction accuracy. We specifically compare Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs), implementing validation and regularization techniques to determine the model with the least loss. Traditional forecasting methods struggle to capture the intricate dynamics of the market. By employing these advanced deep learning techniques, this study aims to extract nuanced patterns from historical data for more precise predictions. The findings hold promise for empowering investors and analysts with enhanced insights into market behaviour, highlighting the effectiveness of different neural network architectures in capturing complex temporal relationships in stock market data..*

Keywords***—Deep Learning, Recurrent Neural Networks (RNNs), Gated Recurrent Units, Long Short-Term Memory (LSTM), Stock market prediction, Time Series Forecasting***

1. **Introduction**

The stock market represents a complex and dynamic system, influenced by an array of factors including market sentiment, corporate announcements, and macroeconomic indicators. Traditionally, forecasting stock market movements has been a challenging endeavor for both analysts and investors, owing to the market's volatile nature and the sheer volume of influencing factors.

Historically, various methods have been employed to predict stock market trends. These include fundamental analysis, which focuses on financial statements and economic indicators to evaluate a company’s health, and technical analysis, which relies on statistical trends derived from trading activity such as price movement and volume. More sophisticated techniques like time series analysis using ARIMA models have also been utilized to capture linear relationships in historical data.

However, these traditional methods often fall short in fully capturing the complex, non-linear interdependencies and patterns within the stock market data. This limitation has paved the way for the adoption of deep learning technologies in this field. This research seeks to leverage the capabilities of deep learning to enhance the accuracy of stock market predictions. With the advent of neural network architectures like Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs), it is now possible to model the temporal dynamics of stock prices more effectively. These models can learn from and remember extensive sequences of historical data, making them particularly effective for time series prediction tasks such as stock market forecasting.

1. **Literature review**

**S. Mehtab, J. Sen and S. Dasgupta, “Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models”:**

The paper focuses on various methodologies for stock price prediction, emphasizing the challenge posed by the efficient market hypothesis which suggests that accurate stock price prediction is impossible due to market efficiency. However, the authors discuss various approaches that contradict this hypothesis, demonstrating that with well-designed predictive models, high accuracy in stock price prediction can be achieved. The review touches on the use of traditional time series models like ARIMA and modern deep learning techniques, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, highlighting their effectiveness in capturing complex patterns in highly granular financial data.

The research results reveal the effectiveness of combining CNN and LSTM models to predict stock prices with high accuracy. The models were tested on high-frequency data from the National Stock Exchange of India, recorded every five minutes over two years. The models achieved very low Root Mean Square Error (RMSE) values relative to the mean stock prices, indicating highly accurate predictions. The most accurate model achieved an RMSE to mean ratio of 0.00625, highlighting its precision in forecasting.

In conclusion, the study effectively demonstrates that despite the challenges posed by the efficient market hypothesis, advanced deep learning models like CNNs and LSTMs can indeed predict stock prices with significant accuracy. The paper emphasizes that these models are not only accurate but also efficient in processing large datasets, making them suitable for real-time stock price prediction applications. The authors suggest that these models could be further enhanced by integrating additional data types and refining model architectures to better handle the inherent noise and volatility in financial time series data.

**“Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention based Approach.”** -

**Karan Pardeshi, Sukhpal Singh Gill, Ahmed M. Abdelmoniem :**

The paper begins by acknowledging the complexity of the stock market and the challenges in forecasting its movements due to influences such as geopolitical, social, and economic data. Prior methods such as ARIMA, fundamental and technical analyses have struggled to consistently capture and predict the market dynamics due to their linear and often simplistic assumptions about market behaviour. They cover advancements in machine learning that offer new tools to model these complexities. The use of LSTM (Long Short-Term Memory networks) has been particularly noted for its ability to capture temporal dependencies in time-series data which is crucial for accurate stock price forecasts.

The authors propose a novel hybrid model combining LSTM with a Sequential Self-Attention Mechanism (LSTM-SSAM). This model is designed to enhance the LSTM's ability to remember long-term dependencies by incorporating self-attention mechanisms that weigh the importance of different inputs across the sequence. This theoretically allows the model to focus more on significant data points that could have a greater predictive value on future stock prices.

In conclusion, the experimental results, based on datasets from three stocks (SBIN, HDFCBANK, and BANKBARODA), demonstrated that the proposed LSTM-SSAM model outperforms conventional models like pure LSTM, BiLSTM, CNN, and ARIMA in terms of Root Mean Squared Error (RMSE) and R-square values. This suggests that integrating self-attention with LSTM not only enhances the accuracy of predictions but also effectively captures the complex patterns in stock price movements.

**Touzani, Y, Douzi, K. “An LSTM and GRU based trading strategy adapted to the Moroccan market.”**

This paper details the performance of LSTM and GRU models in predicting stock prices. The authors implemented these models on a selected group of stocks, evaluating their prediction accuracy and efficiency. Key findings include:

* GRU models demonstrated quicker adjustment to changes and provided more accurate predictions compared to LSTM models.
* The application of machine learning methods in stock price prediction significantly enhanced the capability to follow the price trends, even though some stocks showed volatility that was challenging to capture accurately.

The paper states that both LSTM and GRU models are effective in predicting stock prices, with GRU models generally outperforming LSTM in terms of prediction accuracy and response to market volatility. The research confirms the potential of using advanced machine learning techniques for financial forecasting and investment strategy optimization. It also suggests that while these models provide powerful tools for understanding and predicting market dynamics, they must be continually adapted and optimized in response to changing market conditions to maintain their effectiveness.

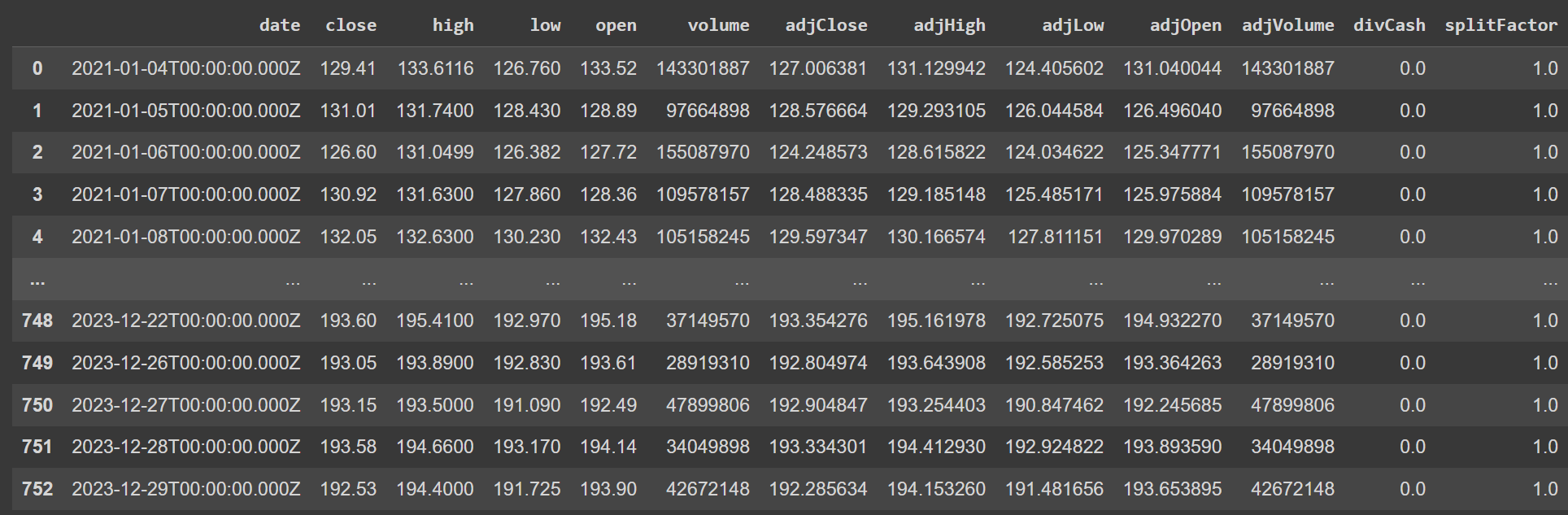
The authors suggest future work could explore integrating additional types of data, such as macroeconomic indicators or news sentiment, to further enhance the models' predictive power and reliability. Additionally, exploring other advanced machine learning techniques could offer new insights and improvements over existing models.

1. **methodology**

**1. Data Acquisition:**

Tiingo is a financial data platform that provides a wide range of tools and information to anybody interested in financial markets, including individual investors, researchers, and developers. One of its key advantages is access to high-quality financial datasets such as stock prices, ETFs, mutual funds, and cryptocurrency data.

We have used Apple’s (AAPL) stock prices data for the time period '01-01-2021' to ’31-12-2023'. which includes date, open, low, high, close, adjusted close, adjusted high, adjusted low, adjusted open prices, trading volume, adjusted volume, dividend cash and split factor columns. We consider the “Close” price as target variable for our predictions for simplicity.



**2. Data Preprocessing:**

1. Normalising the data

The MinMaxScaler from scikit-learn is used to normalize the closing prices to a range between 0 and 1. Normalization is a common preprocessing step for neural networks, helping to speed up training and improve model convergence by ensuring that all input features are on a comparable scale.

1. Creating Sequences and Labels

We then created the input sequences (X) and the corresponding labels (y) that will then be used to train the LSTM. For each iteration, a sequence of length sequence\_length is extracted from the normalized data and added to X. The data point immediately following this sequence is taken as the label and added to y.

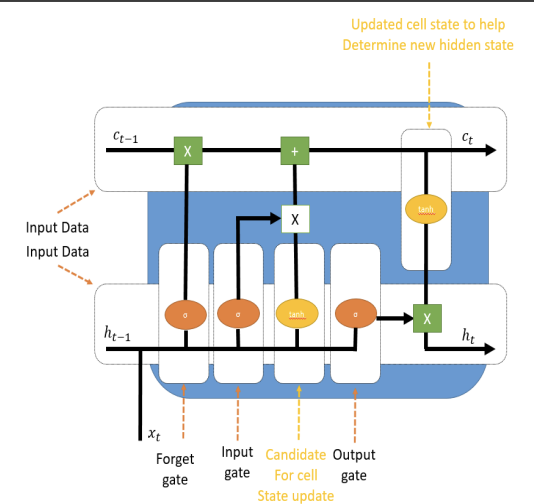
1. Conversion to Numpy Arrays

Finally, the lists of sequences (X) and labels (y) are converted to numpy arrays, which are the expected format for training models in most deep learning frameworks. The function also returns the scaler object, which is necessary for inverse transforming the model's predictions back to their original scale (i.e., denormalizing them) after training.

**3. Model Selection and Architecture:**

**LSTM**:

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that has received a lot of attention in the field of stock market prediction. Unlike standard feedforward neural networks, LSTM networks can detect long-term dependencies and patterns in sequential data, making them ideal for time series analysis, such as stock market data.



*Figure 1: Architecture of Long Short Term Memory.*

**LSTM + CNN**:

Long Short-Term Memory (LSTM) combined with Convolutional Neural Networks (CNN) provides a powerful technique for stock market prediction. CNNs excel at recognising spatial patterns in data, whereas LSTMs capture temporal relationships. By combining these designs, the model can successfully analyse both local and global characteristics in historical market data. This hybrid technique allows for more accurate projections, delivering significant insights to investors and analysts.

**Stacked LSTM**:

Stacking LSTM layers improves stock market predictions by allowing the model to learn complicated patterns from sequential data. This hierarchical technique captures complicated dependencies, allowing for more accurate forecasts of market movements.

**Bidirectional LSTM**:

Bidirectional Long Short-Term Memory (LSTM) networks provide a reliable solution for stock market prediction by processing data sequences in both forward and backward directions. This dual method enables the model to capture both past and future context, resulting in a more complete understanding of temporal trends in market data. Researchers and analysts can use bidirectional LSTM architectures to predict stock market trends with more accuracy and reliability

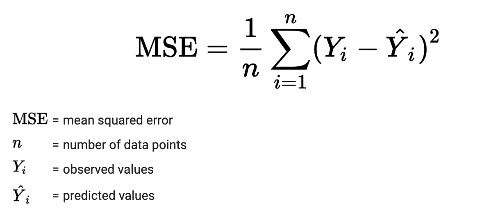
**4. Training the Model:**

We split the data into training, validation, and testing sets. The training set is used to train the model, the validation set is used to monitor performance during training and adjust hyperparameters (learning rate, number of layers, etc.), and the testing set is used for final evaluation after training is complete. A common split is 80% for training, 10% for validation, and 10% for testing.

We trained model on the labelled data. The model learns to identify features in the text that are indicative of hate speech..

We defined our optimizer, loss function and comparison metrics :

Optimizer : Adam

Loss function : M

SE Loss

Comparison metrics : RMSE, MAE and R2

**4. Model Evaluation and Tuning:**

Evaluate the model's performance on the testing set using metrics like RMSE, MAE and R2. These metrics consider how well the model predicts future stock price based on the closing price of the previous 10 days.

RMSE : RMSE is a standard way to measure the error of a model in predicting quantitative data. It represents the square root of the average squared differences between the predicted and actual values. RMSE gives a relatively high weight to large errors due to the squaring of each term, which means that large errors have a disproportionately large effect on RMSE. Thus, RMSE is very useful when large errors are particularly undesirable.

MAE : MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

R2: R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression

Integrating Skorch with scikit-learn's GridSearchCV allows for efficient hyperparameter tuning of LSTM models built with PyTorch, combining the deep learning capabilities of PyTorch with the user-friendly machine learning tools of scikit-learn. Skorch acts as a bridge by wrapping PyTorch models and presenting them as scikit-learn compatible estimators. This enables the use of GridSearchCV for optimizing hyperparameters seamlessly. In this setup, we define the LSTM model in PyTorch, then wrap it using Skorch, and finally, utilize GridSearchCV to search through a predefined grid of hyperparameters, evaluating and selecting the best combination based on specified performance metrics. This approach significantly simplifies the process of model tuning and evaluation, making it accessible without delving deep into the specifics of PyTorch during the optimization phase.

**results and discussion**

Performance Analysis:

Model 1: LSTM with CNN model with combined layers

Model 2: LSTM with CNN and attention mechanism with L2 regularisation

Model 3: Stacked LSTM with 2 layers

The quantitative examination showed that LSTM+CNN performed the best, with the lowest RMSE of 4.618, MAE of 3.785, and greatest R² score of 0.946. The integration of CNN and LSTM fits the data well, and the model performed similarly well on test data, demonstrating high generalisation skills, however the inclusion of an attention mechanism and regularisation to the CNN-LSTM framework prevented the model from capturing complicated data. These findings have major significance for investors and financial analysts, as they provide more accurate forecasts.

**Conclusions**

In In conclusion, this study demonstrates the potential for PyTorch-driven deep learning techniques to increase stock market prediction accuracy. The study uses LSTM, LSTM paired with CNN and stacked LSTM models to extract subtle patterns from historical data, resulting in more accurate market forecasts. These findings provide useful insights for investors and analysts, promising improved decision-making abilities. Continued breakthroughs in deep learning approaches hold the key to further revolutionising stock market forecasting, providing stakeholders with practical insights for successfully navigating financial markets.

**Futurework**

**Using Sentiment Analysis for Better Qualitative Data Integration**: Future study could use sentiment analysis via NLP on numerous web sources to combine qualitative stock market data, hence improving prediction accuracy. Models can acquire a thorough understanding of market dynamics by assessing public perception and sentiment.

**Developing Advisory Triggers for Investment Decisions**: Exploration of algorithmic triggers that advise on appropriate investment timings could combine predictive models with risk assessment techniques to provide actionable investing insights.

**Exploring GNNs and Other Advanced Architectures**: Investigating GNNs and other sophisticated designs could provide new insights by modelling the interconnections of the stock market, potentially outperforming existing models in forecast accuracy.

**References**

*S. Mehtab, J. Sen and S. Dasgupta, "Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1481-1486, doi: 10.1109/ICECA49313.2020.9297652*

*Pardeshi, K., Singh Gill, S., and Abdelmoniem, A. M., “Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention based Approach”, 2023. doi:10.48550/arXiv.2308.04419. 1*

*Touzani, Y., Douzi, K. An LSTM and GRU based trading strategy adapted to the Moroccan market. J Big Data 8, 126 (2021). https://doi.org/10.1186/s40537-021-00512-z*

*Zhuangwei Shi, Yang Hu, Guangliang Mo, Jian Wu “Attention-based CNN-LSTM and XGBoost Hybrid Model for Stock Prediction”*

*Rana, M.; Uddin, M.M.; Hoque, M.M. “Effects of activation functions and optimizers on stock price prediction using LSTM recurrent networks.”*

*Aldhyani, T.H.H.; Alkahtani, H. “A Bidirectional Long Short-Term Memory Model Algorithm for Predicting COVID-19 in Gulf Countries.”*

*Sharma, A.; Bhuriya, D.; Singh, U. “Survey of stock market prediction using machine learning approach.”*