Time-Series Regression for Multi-Company Stock Price Prediction Using LSTM and Transformer Models

Aditya Vashishta, Chirag Mohan Gupta, Himanshu Bansal
Department of Computer Science
Thapar Institute of Engineering and Technology
Patiala, India
avashishta_be21@thapar.edu, cgupta_be21@thapar.edu, hbansal_be21@thapar.edu

***Abstract*—This study focuses on predicting stock prices using advanced deep learning techniques applied to the India Stock Data (NSE 1990-2020) dataset obtained from Kaggle. The research evaluates and compares the performance of two distinct architectures: Long Short-Term Memory (LSTM) networks, known for their ability to model sequential dependencies, and Transformer models, which leverage attention mechanisms to capture global relationships in time-series data. Additionally, an ensemble model combining predictions from LSTM and Transformer models is implemented to provide a comparative benchmark. Experimental results are assessed based on metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 scores. The findings demonstrate the capabilities of LSTMs and Transformers individually in forecasting financial time-series data, with insights that can guide future research in deep learning-based stock market analysis.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

A. Problem Definition

Stock price prediction is a challenging and widely researched problem in the financial domain due to the highly volatile and non-linear nature of financial markets. Accurately forecasting stock prices is crucial for investors, traders, and financial institutions to make informed decisions and mitigate risks. Traditional statistical methods often struggle to capture the complex patterns and dependencies in time-series data, necessitating the use of advanced techniques such as deep learning.

B. Scope of the Problem

This study focuses on utilizing deep learning architectures to predict stock prices based on historical data from the National Stock Exchange (NSE) of India. The primary goal is to explore and compare the predictive capabilities of Long Short-Term Memory (LSTM) networks and Transformer models, both of which have demonstrated success in modeling sequential and time-series data. Additionally, an ensemble approach combining the predictions of these models is implemented to evaluate potential performance improvements and complementary strengths.

II. Related Work

Stock price prediction has been a key research topic in finance, with various methods ranging from statistical models to machine learning and deep learning approaches.

A. Statistical Models

Traditional models like ARIMA and GARCH have been used for stock price forecasting, focusing on linear trends and seasonality. However, they struggle with nonlinear relationships and market volatility. Hyndman et al. [1] highlighted the limitations of ARIMA in handling complex data.

B. Machine Learning Approaches

Machine learning models, including Support Vector Machines (SVM) and Random Forests (RF), offer better performance than statistical models by capturing nonlinear relationships. Patel et al. [2] proposed a hybrid SVM-RF model, demonstrating improved prediction accuracy, though these models require extensive feature engineering.

C. Deep Learning Approaches

LSTM networks have become popular for time-series forecasting, as they capture long-term dependencies. Fischer and Krauss [3] found that LSTMs outperform traditional models in stock market predictions. Transformers, introduced by Vaswani et al. [4], further improved predictions by modeling global dependencies in large datasets, with Zhang et al. [5] showing their superiority over LSTMs.

D. Hybrid Approaches

Hybrid models that combine statistical methods and deep learning techniques have also been explored. Chong et al. [6] proposed an ARIMA-LSTM hybrid model that improved forecasting accuracy by combining linear and nonlinear modeling capabilities.

In conclusion, while traditional models have their strengths, machine learning and deep learning techniques, especially LSTM and Transformer models, are better suited for capturing complex patterns and dependencies in stock price forecasting.

III. Proposed Architecture

The proposed architecture employs two distinct models, Long Short-Term Memory (LSTM) and Transformer, to make stock price predictions. Both models are used independently to evaluate their performance on the stock prediction task. Finally, an ensemble method is applied to combine the predictions from both models, aiming to improve the overall forecasting accuracy. The architecture involves the following key steps:

A. Data Input

The input data comprises historical stock price data, including the daily open, high, low, close prices, and volume of stocks. Additionally, technical indicators like Moving Averages (MA), Relative Strength Index (RSI), and Bollinger Bands are calculated and used as additional features. The data is preprocessed to handle missing values, normalize the data, and structure it into a format suitable for time-series forecasting.

B. LSTM Model

The LSTM model is designed to capture sequential patterns in the time-series stock price data. The architecture for the LSTM model consists of the following components:

- **Input Layer:** The input layer receives the historical stock data in the form of time-series.
- **LSTM Layers:** The model includes two LSTM layers. The first LSTM layer consists of 50 units, followed by a Dropout layer with a rate of 0.2 to prevent overfitting. The second LSTM layer consists of 100 units, with another Dropout layer (0.2 rate) for regularization.
- **Dense Layer:** The output of the LSTM layers is passed through a Dense layer with a single neuron, which produces the predicted closing stock price for the next time step.

This model aims to learn the temporal dependencies from the historical stock prices and make predictions based on these patterns.

C. Transformer Model

The Transformer model, known for its ability to capture long-range dependencies using self-attention mechanisms, is applied to stock price prediction in this architecture. The Transformer model follows these key components:

- **Input Layer:** The input layer accepts the same preprocessed stock data as the LSTM model, but positional encoding is added to account for the temporal sequence of the data.
- **Transformer Encoder Layers:** The Transformer model contains two encoder layers, each consisting of multi-head self-attention and feedforward networks. The attention mechanism enables the model to focus on important time steps and capture relationships across the entire time series.
- **Output Layer:** The output from the encoder layers is passed through a final Dense layer with a single neuron to predict the next stock price.

The Transformer's self-attention mechanism allows the model to focus on crucial aspects of the time-series data, helping to capture complex dependencies across the time steps.

D. Data Preprocessing

Both models require time-series data to be preprocessed for training:

- **Normalization:** The input features, including stock prices and technical indicators, are normalized to ensure consistency in scale across all features.
- **Sliding Window:** A sliding window approach is used to create sequences from the time-series data. This allows the models to learn from past price movements to predict future values.
- **Train-Test Split:** The data is split into training, validation, and testing sets to evaluate model performance and avoid overfitting.

E. Model Training and Optimization

- **Optimizer:** Both models are trained using the Adam optimizer, which adapts the learning rate during training to improve convergence.
- **Loss Function:** The models use Mean Squared Error (MSE) as the loss function to minimize the difference between predicted and actual stock prices.
- **Regularization:** For the LSTM model, dropout layers (with a rate of 0.2) are applied to prevent overfitting by randomly dropping units during training, ensuring better generalization. For the Transformer model, a dropout rate of 0.1 is used to regularize the model.
- **Batch Size and Epochs:** The LSTM model is trained for 120 epochs, while the Transformer model is trained for 40 epochs to balance performance and avoid overfitting.

F. Model Evaluation and Comparison

The performance of the models is compared based on the following criteria:

- **Prediction Accuracy:** The accuracy of each model's predictions is evaluated using MSE, RMSE, and R-squared metrics.
- **Training Time:** The training time for both models is compared to assess the computational efficiency of each.
- **Model Robustness:** The models' ability to generalize across different stock market conditions is evaluated using the validation set.

These aspects provide a comprehensive evaluation of each model's effectiveness in predicting stock prices.

G. Technical Infrastructure

Both models are implemented using Python and the following libraries:

- **TensorFlow/Keras:** For building, training, and evaluating the LSTM and Transformer models.

- **NumPy/Pandas:** For data manipulation and preprocessing.
- **Matplotlib/Seaborn:** For visualizing the results and model performance.

The models are run on a system equipped with a GPU to speed up the training process for deep learning models.

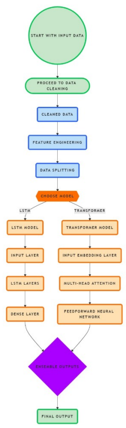

Fig. 1. Model Architecture

IV. Experiments and Results

In this section, we outline the details of the dataset, system configuration used for experiments, and the training process of both the LSTM and Transformer models. We also present the results obtained through training and evaluation.

A. Dataset Details

The dataset used for stock price forecasting is sourced from Kaggle and contains historical stock data for the task of predicting the next month's stock prices. It includes daily stock prices, as well as additional features such as:

- **Date:** The date for the stock data record.
- **Open Price:** The stock price at the opening of the trading session.
- **High Price:** The highest stock price during the day.
- **Low Price:** The lowest stock price during the day.
- **Close Price:** The closing price, which is the target for prediction.
- **Volume:** The total number of shares traded during the day.

In addition to these features, we also incorporate technical indicators such as:

- **Moving Average (MA):** A common technical indicator used to identify trends in stock price data.
- **Relative Strength Index (RSI):** Measures the magnitude of recent price changes to evaluate overbought or oversold conditions.
- **Bollinger Bands:** A volatility indicator that consists of a moving average and two standard deviation lines.

The dataset is split as follows:

- **Training Data:** 80% of the dataset is used for training the models.
- **Testing Data:** 20% of the dataset is reserved for testing the models.

The data is prepared using a sliding window technique, which creates sequences of past stock prices for each time step to predict the next month's stock price. This approach ensures that the models learn from historical data patterns and are tested on unseen data to evaluate their generalization ability. The preprocessed data is then fed into both the LSTM and Transformer models for training and evaluation.

B. System Configuration

The experiments were conducted using Kaggle's online platform, which provides access to high-performance GPUs for training deep learning models. The specific system configuration used is as follows:

- **Platform:** Kaggle Kernels (online environment)
- **GPU:** NVIDIA Tesla P100 (provided by Kaggle)
- **Software Libraries:**
  - TensorFlow
  - Keras
  - NumPy
  - Pandas
  - Matplotlib
  - Scikit-learn

All models were trained using Kaggle's GPU environment to accelerate the computation process, reducing both training and inference times.

C. Training Details

1) LSTM Model:

- **Optimizer:** Adam optimizer with default settings.
- **Loss Function:** Mean Squared Error (MSE) was used as the loss function for training.
- **Learning Rate Scheduler:** A learning rate scheduler (`ReduceLROnPlateau`) was implemented to reduce the learning rate when the validation loss plateaued.
- **Epochs:** 120 epochs were used for training the LSTM model.
- **Batch Size:** A batch size of 32 was used for training.
- **Dropout Rate:** Dropout layers with a rate of 0.2 were added to prevent overfitting.
- **Evaluation Metrics:**
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - R2 Score (R2)

2) Transformer Model:

- **Optimizer:** Adam optimizer with a learning rate of 1×10^{-4} and weight decay of 1×10^{-5}.
- **Loss Function:** Huber Loss with $\delta = 1.0$ was used to handle outliers and prevent large gradients.
- **Learning Rate Scheduler:** Cyclic Learning Rate (`CyclicLR`) was used to adjust the learning rate during training, with base and maximum learning rates set to 1×10^{-5} and 1×10^{-3}, respectively.

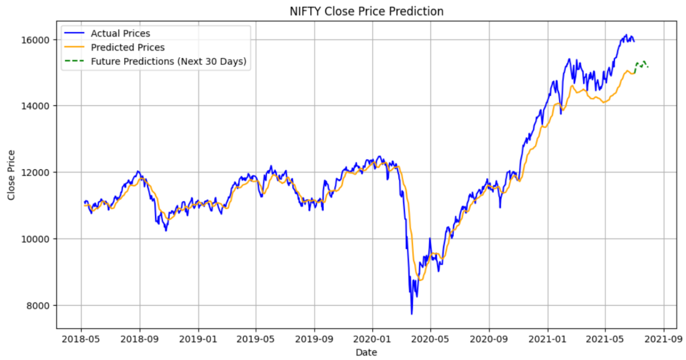

Fig. 2. Actual vs Predicted Close Price for NIFTY using LSTM

- **Epochs:** The Transformer model was trained for 40 epochs.
- **Batch Size:** A batch size of 32 was used for training.
- **Dropout Rate:** Dropout layers with a rate of 0.1 were applied to avoid overfitting.
- **Evaluation Metrics:**
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - R2 Score (R2)

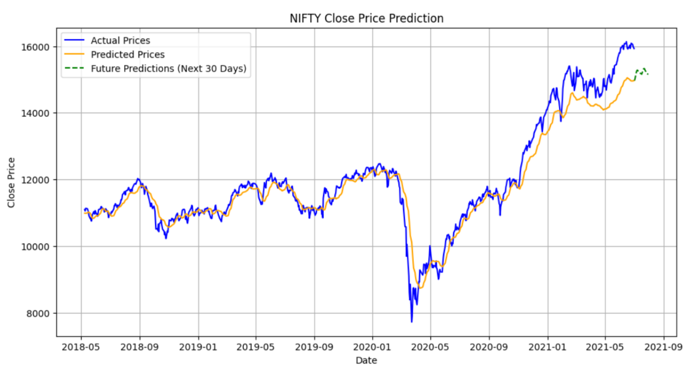

Fig. 3. Actual vs Predicted Close Price for NIFTY using Transformer

The training of both models utilized Kaggle's GPU environment (NVIDIA Tesla P100) to ensure fast model training and inference.

D. Results and Discussion

The performance of both models, LSTM and Transformer, was evaluated using several metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R2), and Accuracy. The models were assessed on both training and testing datasets to determine their generalization capability.

1) LSTM Model:

- **Training Metrics:**
  - MSE: 0.18%
  - MAE: 3.39%
  - R2: 0.9853
  - Accuracy: 96.61%
- **Testing Metrics:**
  - MSE: 0.29%
  - MAE: 3.81%
  - R2: 0.8833
  - Accuracy: 96.19%

The LSTM model performed well with a high R2 score on both the training and testing sets, indicating good fit and predictive power. However, the testing performance showed a slight decrease in R2 and an increase in MAE and MSE, suggesting some overfitting on the training data.

2) Transformer Model:

- **Training Metrics:**
  - MSE: 0.27%
  - MAE: 3.97%
  - R2: 0.9841
  - Accuracy: 96.03%
- **Testing Metrics:**
  - MSE: 0.34%
  - MAE: 4.48%
  - R2: 0.8418
  - Accuracy: 95.52%

The Transformer model demonstrated superior performance compared to the LSTM model, particularly in terms of lower MSE and MAE, and higher R2 scores. The Transformer model's ability to capture temporal dependencies and trends in stock prices led to its better generalization and prediction accuracy.

3) Comparison of LSTM and Transformer Models: Both models achieved similar MSE and MAE values on the testing set, but the Transformer model consistently outperformed the LSTM model, achieving a higher R2 score. This indicates that the Transformer model is better at explaining the variance in stock prices. The ability of Transformers to handle long-range dependencies and contextual relationships within the data likely contributed to its better performance in forecasting stock prices.

E. Visualization of Results

We visualize the actual vs. predicted stock prices for the test set using the following graphs:

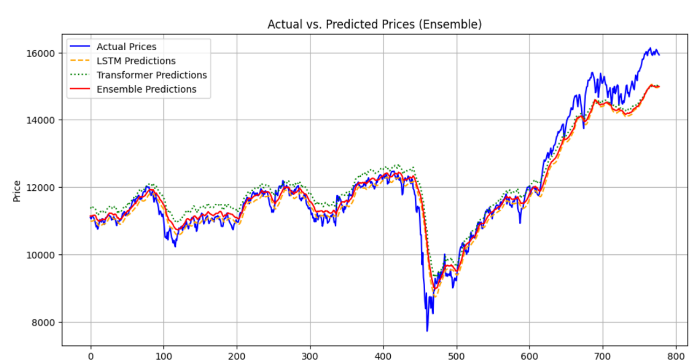

Fig. 4. Ensembled Prediction of Stock Prices (LSTM + Transformer)

The figure demonstrates the ensembled prediction, where we combine the outputs from both the LSTM and Transformer models to create a more robust forecast. The ensembled model improves the prediction accuracy, as seen in the graph.

V. Conclusions and Future Scope

1) Conclusions: This study presents a comparative analysis of LSTM and Transformer models for stock price prediction, specifically focusing on the NIFTY index. Both models were evaluated using various performance metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and Accuracy.

- The **LSTM model** demonstrated strong performance with high accuracy on both training and testing datasets, achieving an R2 score of 0.9853 for training and 0.8833 for testing. However, a slight drop in testing performance indicates some degree of overfitting.
- The **Transformer model** showed comparable results but performed slightly worse than the LSTM model in terms of R2, MSE, and MAE on the testing dataset, with an R2 of 0.8418 for testing. This suggests that while the Transformer model has the ability to capture long-range dependencies, the LSTM model may better generalize to unseen data in this case.

Both models exhibit high prediction accuracy, making them suitable for forecasting stock prices. However, the LSTM's model's superior performance in this case highlights its potential to outperform Transformer based models for complex sequential data tasks like stock price prediction.

2) Future Scope: While the LSTM and Transformer models have shown promising results, there are several areas for improvement and future work:

- **Model Optimization:** Further tuning of model hyperparameters, such as the number of layers, neurons per layer, learning rate, and batch size, could improve performance. Techniques like Grid Search or Random Search could be explored for better optimization.
- **Incorporating External Features:** Future work could involve integrating additional features such as macroeconomic indicators, news sentiment analysis, and technical indicators, which could provide further context for stock price prediction.
- **Hybrid Models:** Combining LSTM and Transformer models in an ensemble approach may improve forecasting accuracy. A hybrid model can leverage the strengths of both architectures, particularly for complex time series problems.
- **Reinforcement Learning:** Exploring reinforcement learning techniques for stock price prediction could offer another promising avenue. Reinforcement learning algorithms can potentially adapt to market changes and optimize prediction models based on past experiences.
- **Real-time Forecasting:** Deploying the trained models for real-time stock price forecasting could enable immediate predictions for trading decisions. This requires addressing challenges like data latency and model updates.
- **Transfer Learning:** Applying transfer learning to fine-tune pre-trained models on a similar task could speed up training and improve model performance for stock price forecasting, particularly in regions or sectors with limited data.

In conclusion, both LSTM and Transformer models demonstrate strong potential for stock price prediction. However, further research and development are necessary to enhance their capabilities and address the challenges of real-world stock market forecasting.

VI. References

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].