Machine Learning Algorithm comparison between Transformer and LSTM for stock market prediction

JACOPO DALLAFIOR STEFANO TONINI
jacopoda @kth.se | stonini @kth.se

October 24, 2024

Abstract

The prediction in the stock market is daunting because financial markets are of a volatile and nonlinear nature. The study compares the performance of the Long Short-Term Memory network with the Transformer based models in stock price forecasting. Historical data for Apple (AAPL) and Microsoft (MSFT) were collected, then further preprocessing was made to ensure quality for any missing data. Both models were implemented with great care in tuning hyperparameters and tested on MAPE and RMSE. We further hypothesize that the performance of the LSTM outperforms the Transformer model since this model is much more effective in dealing with long-term dependencies. The result of this will be used to identify which of these two models best suits the prediction of stock prices, which might be helpful to investors, and at the same time plays an important role in the development of the state-of-the-art in financial machine learning.

Contents

List of Acronyms and Abbreviations

CNN Convolutional Neural Network
LSTM Long Short-Term Memory
MAPE Mean Absolute Percentage Error
NLP natural language processing
RMSE Root Mean Square Error
RNN Recurrent Neural Network
ARIMA Autoregressive integrated moving average
GARCH Generalized AutoRegressive Conditional Heteroskedasticity
GPU Graphics processing unit
TPU Tensor Processing Unit
AAPL Apple
MSFT Microsoft
ML Machine Learning

1 Aims, Objectives, Goals, Research questions, hypotheses

Aims

The aim of this project is to enhance the accuracy of stock market forecasts by utilizing machine learning models. By improving the precision of these predictions, the project seeks to support financial analysts and investors in making more informed decisions, ultimately reducing market volatility risks and optimizing investment strategies. Furthermore, the project aims to apply theoretical knowledge of machine learning to practical implementations within the financial sector, demonstrating how these advanced models can be effectively used to forecast stock prices and improve financial decision-making processes.

Objectives

The objective of this project is to compare the performance of Long Short-Term Memory (LSTM) networks and Transformer-based models in predicting stock prices. The process begins with the collection and preprocessing of historical stock market data from reliable financial sources, ensuring that the data is clean and ready for analysis. Both models will be implemented with carefully tuned hyperparameters to optimize their performance. Once trained, the models will be evaluated using key metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) to assess their predictive accuracy. Ultimately, the aim is to determine which of the two models provides more accurate forecasts, contributing valuable insights into the application of machine learning techniques for stock market prediction.

Goals

The goal of this project is to identify the most effective algorithm between LSTM networks and Transformer-based models for stock price forecasting in the financial sector. By conducting a thorough comparison of their performance, the project aims to confirm or disprove the hypothesis that LSTM networks outperform Transformer-based models in predicting stock prices. The research findings may also offer practical insights that can potentially lead to profitable returns, demonstrating how advanced machine learning models can be leveraged to refine financial decision-making processes and reduce the risks associated with market volatility.

Research Questions

The primary research question of this project is to determine which algorithm performs better in predicting stock prices: LSTM networks or Transformer-based models. Additionally, the project aims to investigate whether LSTM networks provide more accurate stock price predictions compared to Transformer models when evaluated using metrics such as MAPE and RMSE. Another important aspect of the research is to explore how different hyperparameter settings influence the performance of both LSTM and Transformer models in stock price prediction tasks.

Hypotheses

- **Primary Hypothesis:** LSTM networks will predict stock prices more accurately than Transformer-based models, resulting in lower MAPE and RMSE values.

- **Null Hypothesis:** There is no significant difference in the predictive accuracy of LSTM networks and Transformer-based models in forecasting stock prices.

2 Background and rationale

Stock price prediction is a critical challenge within various fields, including economics, business, and computational science, due to the complexity and volatility of financial markets. The ability to accurately predict stock prices offers substantial benefits for investors, decision-makers, and financial analysts. A well-performing prediction model can help minimize risks and enhance investment strategies, making it an area of interest for both academia and industry. Machine learning has become an essential tool in financial forecasting, particularly for stock market prediction, due to its ability to model complex and non-linear patterns in data. Traditionally, LSTM networks have been the go-to models for time series forecasting because of their capacity to capture temporal dependencies in sequential data[1]. However, the advent of the Transformer architecture, renowned for its success in natural language processing by effectively modeling long-range dependencies without the limitations of sequential processing, has sparked interest in its application to time series prediction. Despite this interest, there is a noticeable gap in the literature regarding direct comparisons between Transformer and LSTM models specifically for stock market prediction. This research wants to fill the gap, offering a comparative analysis that can determine which architecture provides superior predictive capabilities in this context. By doing so, we aim to contribute to the field by guiding future research directions and enhancing practical forecasting methods in finance, potentially leading to more informed investment decisions and economic strategies.

3 Theory/literature

3.1 Machine Learning in Stock Market Prediction

Machine learning has gained significant traction in financial markets due to its capability to predict stock prices more accurately than traditional methods. Stock market prediction is inherently challenging due to its non-linear, noisy, and volatile nature. Machine learning models, with their ability to learn complex patterns, have become appropriate tools to tackle these challenges [1].

3.1.1 Challenges in Stock Market Prediction

Stock market data is highly unpredictable due to external factors like economic events, political changes, and investor behavior. Traditional models struggle to capture these non-linear patterns, making machine learning models an increasingly suitable choice.

3.2 Time Series Prediction Methods

Time series forecasting involves predicting future values based on previously observed data points. It has long been applied to stock market prediction to gain insights into future trends. Traditional models like ARIMA and GARCH have been widely used but exhibit limitations in capturing non-linear and complex patterns. With the advent of deep learning, models like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have demonstrated superior performance by overcoming these shortcomings[2].

3.2.1 Traditional Time Series Models

ARIMA and GARCH have been standard methods for forecasting stock market prices, but they have limitations in non-linear scenarios and do not fully capture complex dependencies in the data [3].

3.2.2 Deep Learning Models for Time Series Forecasting

With the introduction of deep learning methods such as RNNs, CNNs, and LSTMs, the ability to capture non-linear dependencies in time series data improved significantly. These models have shown superior performance in stock price prediction tasks[2].

3.3 Long Short-Term Memory Networks

LSTM networks were developed to address the vanishing gradient problem in RNNs, making them capable of retaining long-term dependencies in sequential data. The architecture of LSTMs includes memory cells and gates (input, forget, and output gates), which regulate the flow of information, making them suitable for time series forecasting [4].

3.3.1 LSTM Architecture

The LSTM architecture leverages memory cells and gates to control information flow, which helps maintain relevant information over long sequences of data. This makes it suitable for predicting stock prices that require long-term dependency modeling [5].

3.3.2 Applications in Financial Markets

LSTMs have been successfully applied in various financial market predictions. For example, [**?**] showed that LSTMs outperformed traditional models in predicting stock prices due to their ability to remember dependencies over long sequences [5].

3.4 Transformer Model

Transformers, originally developed for natural language processing (NLP), have been recently adapted for time series forecasting. The key innovation in Transformers is the self-attention mechanism, which, along with the encoder-decoder structure, allows them to efficiently capture long-term dependencies without suffering from the vanishing gradient problem. [6]

3.4.1 Transformer Architecture

The Transformer architecture uses the self-attention mechanism to process information in parallel, allowing for faster computation and the ability to capture long-term dependencies in the data [7].

3.5 Key Differences Between LSTM and Transformer

3.5.1 Sequential Processing vs. Parallel Processing

LSTM

- **Sequential Nature:** LSTMs process data step by step (one time step at a time), which introduces a bottleneck when working with long sequences. Each time step depends on the previous one, and this sequential nature can slow down training, especially when dealing with long sequences.
- **Vanishing Gradients:** Although LSTMs are designed to address the vanishing gradient problem better than standard RNNs, they can still struggle to retain information over very long sequences.
- **Training Time:** Since LSTMs process sequences sequentially, they cannot fully utilize the parallel processing capabilities of modern hardware (like GPUs), which leads to slower training compared to models that can handle data in parallel.

TRANSFORMERS

- **Self-Attention Mechanism:** Transformers use a self-attention mechanism that allows them to attend to all time steps in the sequence simultaneously, rather than one at a time. This parallel processing capability enables faster training, especially for long sequences.

- **Scaling to Long Sequences:** Transformers are designed to efficiently handle much longer sequences, as they can capture dependencies across the entire sequence at once. This makes them particularly well-suited for tasks involving long-range dependencies, such as language modeling or time series predictions like stock market forecasting.

- **Hardware Utilization:** Transformers can take full advantage of GPUs and TPUs for parallel processing, leading to faster training times even on large datasets.

LSTMs process data sequentially, which can limit their performance on very long sequences, while Transformers handle long sequences in parallel, providing faster training times [8].

3.6 Challenges and Limitations in Using LSTM and Transformer for Stock Market Prediction

3.6.1 LSTM Limitations

Although LSTMs are effective, they can suffer from longer training times and difficulty in capturing very long dependencies, which is a challenge in highly volatile market scenarios.

3.6.2 Transformer Limitations

Transformers are relatively new in time series forecasting and can sometimes be computationally intensive for very large datasets, despite their parallelism advantage.

4 Research Methodology

The work will make use of the empirical research methodology based on experimental and quantitative analysis, taking into consideration performance comparison between the LSTM and Transformer-based models for stock market prediction. In this regard, we are going to implement both models in equal conditions and compare them using the RMSE and MAPE to decide which one provides more accurate forecasts.

This methodology initiates itself with an extensive literature review to understand state-of-the-art methods and trace possible gaps in existing studies. This could draw insight from studies such as Nabipour et al.Nabipour et al. [9] and Fischer & Krauss [10]. Furthermore, we proceed with historical stock market data collection through the yfinance library [11], which would be adequate for the analysis. The missing value handling, outlier removal, feature normalization, and then splitting into training, validation, and testing sets are all done for proper cleaning and preprocessing of the data.

In the model development stage, we plan to design and implement an LSTM network as well as a Transformer-based model. The LSTMs will be tailored for timeseries forecasting tasks through the careful setting of hyperparameters: number of layers, units in each layer, dropout rates among others.

The Transformer model is adapted for time series data. At the same time, the optimum number of attention heads, layers, and embedding dimensions are optimized following recent improvements from Lim & Zohren [12] and Wu et al. [13].

Training and validation consists of the process of training both models on the training dataset for pattern recognition and making correct predictions. We further validate our models using the validation set in order to avoid overfitting, applying early stopping and scheduling the learning rate, among other techniques. We will tune hyperparameters based on the outcome of the validation to improve the generalization of models.

Both the models, once trained, will be evaluated on the test dataset by calculating RMSE and MAPE for checking predictive accuracy. We will conduct a comparison analysis whereby we identify which is the superior model and further analyze reasons for the difference between performances. We will be analyzing how well the fitted models work on real-world stock prediction.

Full documentation will be performed for the process: methodologies, experiments, results, and visualization graphs and charts. The reason for comprehensive documentation is to clearly understand our approach and findings.

5 Participants, Procedures, Data collection and analysis

Since this is a computational research project focused on machine learning models and data derived from the stock market, it does not involve any human participants. These procedures will start with the collection of data, where historical stock price data for the selected stocks-AAPL and MSFT-will be collected over a significant period to ensure that it captures all kinds of market conditions. The data will be pulled using the yfinance library [11], which gives access to market data from Yahoo Finance. Data preprocessing will involve handling missing values by either forward-fill or interpolation and the removal of outliers using statistical methods. The dataset will be divided into training (70%), validation (15%), and testing (15%) sets in order for unbiased evaluation to take place.

Model implementation in this work will be based on the Python programming language, supported by libraries for handling data like Pandas and NumPy, among others. Others include TensorFlow and Keras for the implementation of neural networks, and Scikit-learn is helpful for additional utilities in machine learning. The LSTM network will be built with configuration according to prior research on hyperparameter setting about the number of layers and units.

The Transformer will be implemented with adaptation to time-series data and some important hyperparameter tuning-like attention heads and embedding dimension adaptation, based on recent works that provide guidelines in that direction.

Both the models will be trained on the train dataset during training and validation, while performance will be observed using the validation set to avoid overfitting. Techniques for improving model performance may include early stopping and learning rate scheduling.

Then, we will test the models by running them on the testing dataset and computing RMSE and MAPE to quantify the prediction errors. Statistical tests may be used to determine whether such differences in performance between the two models are significant.

Data analysis will include understand model behavior, visualizing predictions versus actual stock prices to qualitatively assess performance, and investigating the impact of hyperparameter settings on model accuracy. Visualization tools like Matplotlib and Seaborn will be used to create graphs and charts that illustrate the findings.

6 Expected outcomes

We would expect the network using LSTM to perform better than the Transformer-based model because it could capture the long-term dependency and handle parallel computation more effectively. The quantitative metrics, such as RMSE and MAPE, which show predictive accuracy in both models, are expected to give a complete basis for making a comparison between these models. With this project, we hope to develop a more thorough understanding of how each model learns from time-series data by identifying strengths and weaknesses in each approach. For instance, the capability of LSTM in handling sequential data can be contrasted with the Transformer using self-attention mechanisms that may capture temporal patterns. The objective, from a practical perspective, is to give actionable insights to investors and financial analysts about the most effective machine learning techniques for the task of stock prediction. We may be working on a predictive tool that can be used in making informed investment decisions that may enhance investment strategy and manage the risks pertaining to the volatility of the market. The contribution to research is that the project adds to the body of knowledge by offering a comparative analysis between LSTM and Transformer models in the context of financial time-series forecasting. This may be helpful for further research into machine learning applications in finance, especially regarding understanding the performances of different architectures when carried out on stock market prediction. We will also check the deployability of the model in a real-world environment with respect to computation and scalability concerns from a practical applicability viewpoint. It is rather an essential sanity check to realize the viability of deploying such models into real live trading applications or into financial analytics platforms.

7 Milestones/schedule, budget

October 1, 2024 Literature Review and State-of-the-Art Research, Enviromental set-up

October 6, 2024 Deepened and developed our project by understanding the possible outcomes and having a better understanding of its positioning in relation to sustainability and the adoption of proper ethics.

October 31, 2024 Starting the validation of the completed ML models.

November 15, 2024 Collection of some results to start defining the answer our research question.

December 6, 2024 Defining the conlusion of the final report

January, 2025 Project Refinement and Finalization

8 Risks

- **Data Quality**: Stock market data may contain missing values or inaccuracies, which could affect model performance even after preprocessing.
- **Overfitting**: The complexity of LSTM and Transformer models increases the risk of overfitting, particularly with limited data, reducing their ability to generalize to new data.
- **Computational Constraints**: Transformer models require significant computational resources, leading to longer training times and higher energy consumption, which could impact project timelines and resource management.
- **Model Interpretability**: These models, especially Transformers, are complex and may be difficult to interpret, posing challenges for explaining predictions in financial decision-making.
- **Market Volatility**: External factors such as economic events or crises can affect the accuracy of stock predictions, regardless of model performance.
- **Ethical Concerns**: The use of these models could unintentionally support companies with poor sustainability practices, requiring careful consideration of their broader impact on ethical investing.
- **Sustainability Risk**: The energy footprint of training and using machine learning models, especially Transformers, is significant. Balancing this footprint with the ethical responsibility of promoting sustainable practices poses a challenge in minimizing environmental impact while maximizing model performance.

9 Outline

References

[1] Z. K. Lawal, H. Yassin, and R. Y. Zakari, "Stock market prediction using supervised machine learning techniques: An overview," in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. Gold Coast, Australia: IEEE, 2020. doi: 10.1109/CSDE50874.2020.9411609 pp. 1–6.

[2] A. Elliot and C. H. Hsu, "Time series prediction: Predicting stock price," arXiv preprint arXiv:1710.05751, 2017, under advisement of Dr. Sang Kim, for his class CS542. Additional author unnamed. [Online]. Available: https://doi.org/10.48550/arXiv.1710.05751

[3] M. Kulkarni, A. Jadha, and D. Dhingra, "Time series data analysis for stock market prediction," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020, p. 5. [Online]. Available: https://ssrn.com/abstract=3569737

[4] B. Gülmez, "Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm," *Expert Systems with Applications*, vol. 227, p. 120346, Oct. 2023. doi: 10.1016/j.eswa.2023.120346. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423001834

[5] Z.-Y. Peng and P.-C. Guo, "A data organization method for lstm and transformer when predicting chinese banking stock prices," *Discrete Dynamics in Nature and Society*, vol. 2022, no. 1, Jan. 2022. doi: 10.1155/2022/7119678. [Online]. Available: https://doi.org/10.1155/2022/7119678

[6] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep transformer model," *Expert Systems with Applications*, vol. 208, p. 118128, Dec. 2022. doi: 10.1016/j.eswa.2022.118128. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422001234

[7] T. Muhammad, A. B. Aftab, M. Ibrahim, M. M. Ahsan, M. M. Muhu, S. I. Khan, and M. S. Alam, "Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market," *Asia-Pacific Financial Markets*, 2023. doi: 10.1142/S146902682350013X. [Online]. Available: https://doi.org/10.1142/S146902682350013X

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1706.03762

[9] M. Nabipour, P. Nayyeri, H. Jabani, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150 199–150 212, 2020. doi: 10.1109/ACCESS.2020.3015966

[10] 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. doi: 10.1016/j.ejor.2017.11.054

[11] R. Aroussi, "yfinance: Yahoo! finance market data downloader," https://pypi.org/project/yfinance/, 2019, accessed: 2023-10-01.

[12] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, p. 20200209, 2021. doi: 10.1098/rsta.2020.0209

[13] H. Wu, T. Xiao, K. Ding, and J. Cao, "Adversarial sparse transformer for time series forecasting," in *Advances in Neural Information Processing Systems*, 2020, pp. 17 105–17 115.

A Data Collection and Preprocessing Code

The following Python code was used to collect and preprocess the stock data for Apple Inc. (AAPL):

```
import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler

# Define the ticker symbol
ticker = 'AAPL'   # Apple Inc.
# ticker = 'MSFT'   # Microsoft

# Get the data
data = yf.download(ticker, start='2008-01-01', end='2024-08-01')

# Save to CSV
data.to_csv('stock_data.csv')

# Load data
data = pd.read_csv('stock_data.csv', index_col='Date', parse_dates=True)
data.fillna(method='ffill', inplace=True)

# Visualize the Dataset
data
```

Listing 1: Python code for data collection and preprocessing

Note: The code above performs the following steps:

- Imports necessary libraries: ‘yfinance‘ for data retrieval, ‘pandas‘ and ‘numpy‘ for data manipulation, and ‘MinMaxScaler‘ from ‘sklearn‘ for future scaling.
- Defines the ticker symbol for Apple Inc. (AAPL). The code for Microsoft (MSFT) is commented out but can be used by uncommenting and commenting the AAPL line.
- Downloads historical stock data from January 1, 2008, to August 1, 2024.
- Saves the downloaded data to a CSV file named ‘stock_data.csv‘.
- Loads the data from the CSV file, setting the ’Date’ column as the index and parsing dates.
- Handles missing values by forward-filling using ‘fillna‘.
- Displays the dataset.

A.1 Sample Output of the Dataset

Below is a sample of the dataset obtained:

Date	**Open**	**High**	**Low**	**Close**	**Adj Close**	**Volume**
2008-01-02	7.116786	7.152143	6.876786	6.958571	5.876341	1,079,178,800
2008-01-03	6.978929	7.049643	6.881786	6.961786	5.879056	842,066,400
2008-01-04	6.837500	6.892857	6.388929	6.430357	5.430276	1,455,832,000
2008-01-07	6.473214	6.557143	6.079643	6.344286	5.357590	2,072,193,200
2008-01-08	6.433571	6.516429	6.100000	6.116071	5.164871	1,523,816,000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2024-07-25	218.929993	220.850006	214.619995	217.490005	217.238556	51,391,200
2024-07-26	218.699997	219.490005	216.009995	217.960007	217.708008	41,601,300
2024-07-29	216.960007	219.300003	215.750000	218.240005	217.987686	36,311,800
2024-07-30	219.190002	220.330002	216.119995	218.800003	218.547043	41,643,800
2024-07-31	221.440002	223.820007	220.630005	222.080002	221.823242	50,036,300

Table 1: Sample Output of the Dataset