# An Overview and Comparative Study of LSTM and Transformer Models on Financial Data

**Mustafa Gültekin[1], Ragıp S. Yılmaz[2]**

[1,2] Department of Industrial and Information Engineering, University of Pavia, Pavia, Italy

E-mail: mustafa.gueltekin01@universitadipavia.it

## Abstract

This paper presents a comparative overview of two machine learning models, Long Short-Term Memory (LSTM) networks and transformer-based architectures in the context of time series stock price prediction. With the financial market's increasing complexity and the expanding volume of financial data, accurate and efficient forecasting models have become more important. This research focuses on giving an overview about these models and evaluating them using data from the most valuable four stocks in the S&P 500 index, obtained from Yahoo Finance. The paper aims to analyze and compare the effectiveness of LSTM and transformer models in predicting next-day stock prices, considering their mechanisms and capabilities while giving a detailed overview from other papers. LSTM networks have been a staple in time series analysis, particularly valued for their ability to capture long-term dependencies and handle the non-linear nature of financial data. Transformers, initially designed for natural language processing, have gained attention for their parallel processing capability and the self-attention mechanism, enabling them to capture complex patterns across extensive sequences effectively. Through experimental setups and evaluations, the models' performance is assessed using standard metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). Our findings indicate that LSTMs are better suited for short-sequence predictions and smaller datasets, where they deliver strong performance. Conversely, Transformers are more adept at managing longer sequences and larger datasets, showing marked improvements in forecasting with increased data. This highlights Transformers' potential for effective stock price forecasting in extensive financial datasets.. The study contributes to the growing literature on the application of advanced machine learning techniques in financial markets. It provides insights into the strengths and limitations of LSTM and transformer models, guiding future research directions, including model interpretability, efficiency, and the integration of diverse data sources for enhanced financial forecasting.

Keywords: LSTM, Transformer Models, Time Series Analysis, Stock Price Prediction, Financial Forecasting, S&P 500.

## 1. Introduction

The stock market, described as "complex, fickle, and dynamic" (Kelany, Aly, & Ismail, 2020), is a foundational element of the global financial landscape, marked by its intricate, dynamic, and unpredictable behaviour. The challenge and allure of accurately predicting stock prices lie in the potential for high returns with the risk of significant losses. The field of time series stock price prediction, which explores the application of machine learning techniques, has evolved notably over the years. Traditionally, stock market forecasting involved statistical methods, but with the rise of deep learning, the focus has shifted toward models that can handle the nonlinear and non-stationary nature of financial time series data. Among these, Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, stand out for their ability to "learn long-term dependencies in time-series data, a crucial feature for financial market analysis" (Kelany, Aly, & Ismail, 2020). This evolution signifies a shift towards more sophisticated approaches and machine learning techniques that offer promising results in forecasting (Verma & Mohapatra, 2020).

Thus, the landscape of financial time series prediction is continually refined by advancements in machine learning, providing deeper insights and enhanced predictive accuracy in the complex domain of stock market analysis.

LSTM models "not only solve the long-term dependence problem of time series prediction effectively but also improve the interpretability of the time series prediction methods based on the neural network" (Zhang et al., 2019), marking a significant advancement in the field by addressing the vanishing gradient problem common in traditional RNNs. This capability allows them to retain information over extended time intervals, which is particularly advantageous in financial contexts where past events can have long-lasting impacts. Their effectiveness in financial time series prediction has been well-documented, with research indicating improved accuracy and reliability in stock market forecasting. However, the landscape of machine learning in time series analysis has witnessed a paradigm shift with the introduction of transformer-based architectures, which have shown "superiority in dealing with such problems, especially long sequence time series input (LSTI) and long sequence time series forecasting (LSTF) problems" (Shen & Wang, 2021). Originally conceived for natural language processing tasks, the transformer model, introduced by the paper Attention Is All You Need, represents a departure from recurrent models. It "dispenses with recurrence and convolutions entirely" (Vaswani et al., 2017), employing a self-attention mechanism that enables the model to weigh the importance of different segments of input data differentially. This approach allows for parallel processing of data and more efficient handling of long-range dependencies, overcoming some limitations inherent in LSTM models.

The transformative impact of the transformer architecture has been profound across various domains, with applications in language understanding, translation, and image recognition, demonstrating its versatility and robustness. Malibari, Katib, and Mehmood (2021) note that while "Transformers are relatively new and while have been applied for NLP and computer vision, they have not been explored much with time-series data." Given its success in these fields, there is growing interest in exploring the applicability of transformer models to financial time series data. Aligning with this research direction, Li, Lv, Liu, and Zhang (2022) propose "a transformer-based attention network framework that uses historical text and stock prices to capture the temporal dependence of financial data," aiming to provide a comparative analysis of LSTM and transformer-based models specifically tailored for stock price prediction.

Motivated by the idea of exploring transformers for financial data, this paper gives an overview about existing research and then focuses on the most valuable four stocks in the S&P 500 index, utilizing data from Yahoo Finance to evaluate the next day closing price predictions by these models. This selection not only provides a rich dataset but also ensures the relevance and applicability of our findings to significant segments of the financial market. By employing data pre-processing techniques, including handling null data and normalization, our study aims to present a comprehensive and robust analysis of these models in a financial context. Through this comparative study, the goal is to provide insights into the strengths and limitations of LSTM and transformer models in stock price prediction, offering valuable guidance for practitioners and researchers in the field of financial data analysis.

## 2. LSTM Models Overview

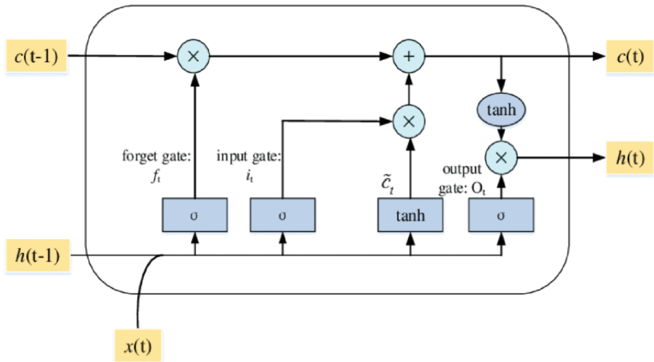



*Figure a The internal structure of a Long-Short Term Memory (LSTM) [9].*

**Advantages of LSTM Models**

-Capability to Capture Long-Term Dependencies: One of the primary advantages of LSTM models is their ability to capture long-term dependencies in time-series data, crucial for financial markets where past events significantly impact future trends. This capability is highlighted by the effectiveness of LSTM networks in renewing interest for time series forecasting systems, as they can capture nonlinear statistical properties, considerably improving forecasting accuracy (Choi & Lee, 2018).

-Robustness to the Vanishing Gradient Problem: LSTMs effectively address the vanishing gradient problem, common in deep neural networks, where gradients diminish exponentially during backpropagation, hindering the learning process. This robustness allows for the efficient training of deep LSTM networks, overcoming obstacles that typically affect RNNs (Hua et al., 2018).

-Flexibility in Feature Learning: LSTMs automatically learn and select relevant features from data, reducing the need for manual feature engineering. This is particularly advantageous in financial time series analysis, where relevant features can be complex and non-intuitive (Yan & Ouyang, 2018).

-Adaptability to Different Time Series Patterns: LSTM models are adaptable to various patterns in financial time series, such as seasonal trends, cyclical fluctuations, and sudden market shifts. This flexibility makes them suitable for diverse market conditions and stock behaviors, demonstrating their effectiveness across different conditions (Yan & Ouyang, 2018).
-Effective in Handling Noisy Financial Data: Financial markets are often characterized by noise and non-stationary signals. LSTM's architecture enables it to filter out such noise and focus on the underlying patterns, making it particularly effective for noisy financial time series (Bao et al., 2017).

**Limitations of LSTM Models**

Complexity and Computational Requirements: LSTM models are known for their intricate architecture, which includes multiple gates (input, output, forget) and parameters to manage the flow and retention of information. This complexity results in significant computational demands, often necessitating extensive training times and substantial computational resources, which can be challenging for real-time financial applications.
Difficulty in Hyperparameter Tuning: The performance optimization of LSTM models heavily relies on the fine-tuning of hyperparameters such as learning rate, number of hidden layers, and batch size. Identifying the optimal configuration requires extensive experimentation and validation, making it a time-consuming and intricate process that can hinder the model's deployment efficiency.
Overfitting in High-Dimensional Data: Given their capacity to learn complex and deep representations, LSTM models are prone to overfitting, especially when dealing with high-dimensional financial datasets. This susceptibility necessitates careful regularization and model validation strategies to ensure that the models generalize well to unseen data.
Limitations in Capturing Sudden Market Changes: While LSTMs excel at learning from historical data, their predictive performance can be compromised by sudden, unpredictable market changes or "black swan" events, which are not well-represented in the training data. This limitation highlights the need for models that can rapidly adapt to new information and market conditions.
Inadequate Interpretability: The "black box" nature of LSTM models poses challenges for interpretability, making it difficult to understand and explain the decision-making process behind their predictions. This aspect is particularly critical in financial contexts where stakeholders require transparency and clarity in predictive modeling.

In conclusion, LSTM models stand out in the arena of financial time series analysis, offering a robust framework capable of capturing long-term dependencies and complex patterns within data. Their structural advantages allow them to address challenges such as the vanishing gradient problem, and their inherent flexibility in feature learning and adaptability to diverse time series patterns make them powerful tools for analyzing the multifaceted nature of financial markets. However, these models are not without limitations. The complexity and computational intensity of LSTMs, coupled with the challenges in hyperparameter tuning and susceptibility to overfitting, present significant hurdles. Moreover, their struggle to quickly adapt to abrupt market changes and the lack of model interpretability can impede their practical application in environments where transparency and speed are crucial. Therefore, while LSTMs represent a significant advancement in predictive modeling, ongoing refinement and innovation are required to fully harness their potential while mitigating their drawbacks in the context of financial forecasting.

## 3. Transformers Overview

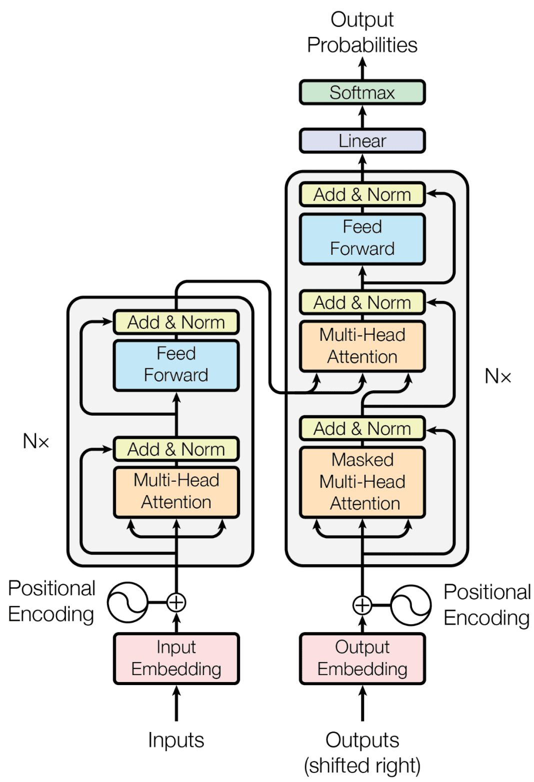



*Figure b The Transformer – model architecture [6].*

The introduction of transformer models marked a significant milestone in the field of deep learning. Developed by Vaswani et al. (2017), transformers were originally designed for tasks in natural language processing (NLP), particularly for challenges such as machine translation and text summarization. As Zhang (2023) articulates, the transformer model utilizes a "self-attention mechanism" and has become "the predominant research direction" due to its dynamic focus capabilities, highlighting its transformative impact across various deep learning tasks.

Origins and Development: The transformer model, marked by its departure from conventional RNN-based architectures like LSTMs, was initially tailored for NLP tasks and has since revolutionized the field. Its impact extends to a variety of applications, indicating its robustness and adaptability beyond its original design (Zhang, 2023).

Core Mechanism - Self-Attention: The central innovation of the transformer model is the self-attention mechanism. This mechanism's ability to process entire sequences of data in parallel and to capture long-range dependencies more effectively than RNNs is a key innovation. It has enabled models like Google Transformer to establish state-of-the-art results in a wide range of NLP tasks (Ham et al., 2021).

**Advancements and Applications**

Broadening the Scope Beyond NLP: While transformers were initially developed for NLP, their application has broadened significantly. Tay et al. (2020) discuss a large selection of models that build upon the original transformer architecture, many of which make improvements around computational and memory efficiency. This expansion has allowed transformers to be applied effectively across a range of domains, including language, vision, and reinforcement learning.

Improvements in Efficiency and Scalability: The transformer architecture, with its ability to process data in parallel and its scalability, has addressed some of the efficiency issues present in RNNs. Peng et al. (2023) have introduced a model that combines the efficient parallelizable training of transformers with the efficient inference of RNNs. This approach is poised to create more efficient models that reconcile trade-offs between computational efficiency and model performance in sequence processing tasks.

**Theoretical Bases**

Multi-Head Attention Mechanism: The multi-head attention mechanism is a key component that allows transformers to simultaneously focus on different parts of the input sequence. This capability is essential for capturing the intricate details and dependencies within the data. As investigated by Mahdavi, Liao, and Thrampoulidis (2023), the multi-head attention mechanism's memorization capacity demonstrates its proficiency in handling diverse sequences and its potential to capture complex dependencies, which is particularly beneficial for tasks requiring a nuanced understanding of context.

Positional Encoding: To maintain sequence order, transformers employ positional encodings. Unlike RNNs, which process data sequentially, transformers use these encodings to inject information about the position of tokens within the sequence. Ruoss et al. (2023) contribute to this aspect by introducing a new encoding scheme that enables transformers to better generalize across varying sequence lengths. This innovation significantly enhances the model's ability to process and learn from longer inputs, a critical advancement for tasks with extended data sequences.

These theoretical bases can offer substantial benefits for financial time series data analysis. The multi-head attention mechanism, with its proven capability to memorize and process various sequence patterns, can discern complex dependencies and contextual relationships inherent in financial markets, where variables are interlinked in intricate ways (Mahdavi, Liao, & Thrampoulidis, 2023). This is crucial for understanding the multifaceted influences on asset prices. Furthermore, the enhanced positional encoding methods allow transformers to handle sequences of varying lengths, which is particularly useful for financial datasets that span different time frames and scales, improving the model's generalization and predictive performance across diverse market conditions (Ruoss et al., 2023). These features could enable more accurate forecasting, risk assessment, and the identification of subtle market trends that traditional models might overlook.

**Transforming Financial Time Series Analysis**

Application in Time Series Data: The adaptation of transformer models for time series data analysis, including financial markets, represents a significant advancement. Their ability to handle long sequences and capture complex dependencies makes them well-suited for analyzing financial time series data, which often contains intricate patterns and relationships. Xu et al. (2023) describe a model that leverages a multiplex attention mechanism and a linear transformer structure to enhance prediction accuracy and model inference speed, a testament to the suitability of transformers for financial applications "A Financial Time-Series Prediction Model Based on Multiplex Attention and Linear Transformer Structure".

Challenges and Potential in Finance: While the potential of transformers in financial time series analysis is vast, challenges such as model interpretability and complexity persist. Emami et al. (2023) tackled these issues with the Modality-aware Transformer, which exploits both categorical text and numerical time series data for forecasting. This approach enhances cross-modal understanding and addresses the challenges inherent in financial datasets, including the need for large amounts of training data "Modality-aware Transformer for Time series Forecasting".

These advancements reflect the transformative potential of applying transformer models to financial time series analysis. By harnessing the power of these models, significant improvements in the accuracy and efficiency of predictive models in finance can be achieved, promising to elevate the field to new heights of analytical capability.

## 4. Transformers vs. LSTM on Time Series Data

To compare the performance of transformer models and LSTM in the context of time series data, especially in financial markets, various factors are considered:

Ability to Handle Sequential Data: LSTM models, with their inherent design, excel at processing sequential data and effectively capturing long-term dependencies within sequences. However, transformers, through innovations such as the Informer model, have been adapted to efficiently address the challenges posed by long sequence time-series forecasting. This model, developed by Zhou et al. (2020), significantly reduces time complexity and memory usage, enhancing the transformer's ability to manage sequential data in domains like finance, thereby offering a competitive alternative to LSTMs "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting".

Learning Long-Term Dependencies: Both LSTM and transformer models are equipped to capture long-term dependencies, a critical aspect of financial time series analysis. However, the methods they employ differ significantly. Murray et al. (2022) provide a comprehensive comparative analysis showing that transformers, despite their differences from LSTMs, consistently yield lower prediction errors across various time series datasets. This underscores the transformers' robust capability in capturing long-term dependencies more effectively than LSTMs, particularly in financial applications "A Comparative Analysis of State-of-the-Art Time Series Forecasting Algorithms".

It should be noted that, Murray et al. (2022), who noted that while transformer models, along with CNN-LSTM, CV-LSTM, and V-LSTM, frequently rank among the top performers across a variety of datasets, it is the GRU and V-LSTM networks that are particularly well-suited to financial closing datasets. Such findings underscore the nuanced nature of model effectiveness across different data types and the importance of model selection in financial analytics.

**Comparative Studies in Time Series Analysis**

In the exploration of time series analysis within financial markets, the comparative efficacies of LSTM and transformer models are scrutinized, uncovering insights that illuminate their respective strengths and challenges. A study conducted by Bilokon and Qiu (2023) meticulously evaluates these models against high-frequency financial datasets. This investigation reveals a nuanced landscape where, despite the prowess of transformers in certain predictive tasks, LSTMs demonstrate superior robustness and reliability in capturing price movements and differences. This suggests an intricate alignment of LSTM architectures with the specific demands of financial time series prediction.Moreover, the intersection of financial market analysis and noise handling unveils the strategic advantage of LSTM models, as illustrated by Cao, Li, and Li (2019). Through the innovative integration of empirical mode decomposition techniques, their research advances a hybrid forecasting approach that significantly enhances noise filtration. This approach thereby augments the predictive precision of LSTM models within the inherently volatile domain of financial data

These scholarly contributions not only enrich the understanding of the dynamic capabilities of LSTM and transformer models in financial time series analysis but also spotlight the critical importance of tailored model selection and preprocessing techniques in navigating the complex terrain of financial forecasting.

**Efficiency and Scalability**

The comparative efficiency and scalability of LSTM and transformer models in processing time series data have been a subject of extensive research. The computational efficiency of transformers, particularly in handling large datasets, has been well-documented. For instance, in the study by Trinh et al. (2022), transformers are shown to be capable of reducing the traditionally long training times associated with LSTM models. This efficiency is achieved through innovations such as combining transformers with convolutional layers, which enhances their ability to learn temporal dependencies quickly and effectively, an essential feature for financial time series forecasting. Moreover, the scalability of transformer models in managing large volumes of data makes them particularly suitable for financial applications. The Informer model, as discussed by Zhou et al. (2020), is a case in point. It introduces a ProbSparse self-attention mechanism that not only improves time complexity but also reduces memory usage. Such enhancements allow it to process extremely long input sequences more efficiently than conventional models, thus addressing a critical need in the analysis of expansive financial datasets. These insights from recent studies underscore the progressive edge that transformer models hold over LSTMs in terms of efficiency and scalability, which is pivotal for the complex and demanding nature of financial time series.

In summary, comparing LSTM and transformer models for predicting financial trends, we see a complex picture. LSTMs are really good at understanding data that comes in a series and can remember things for a long time, which is great for financial information. But, new kinds of transformer models, like the Informer, have been improved to handle big data sets better and faster, challenging LSTMs in this area. Studies show transformers might be better at remembering long-term information than LSTMs and can work with huge amounts of data more efficiently. Yet, LSTMs have their own strengths, especially in certain financial predictions, showing they can be more reliable under specific conditions. The choice between using LSTMs or transformers depends on the job's details, proving that both have their special uses in financial forecasting. This comparison highlights the need to pick the right model carefully, considering the unique demands of analyzing financial markets.

## 5. Experiments and Results

This section of the paper focuses on the results of the experiments and the findings.

### 5.1 Data Preparation

The effectiveness of machine learning models in predicting stock prices hinges on the quality and preparation of the input data. This study focuses on the most valuable four stocks in the S&P 500 index (Figure 1), a choice motivated by their market significance and the rich dataset they provide. The data, sourced from Yahoo Finance, comprises daily stock prices, including opening, closing, high, low, and volume traded. The period under study spans several years, from 1 to 5, depending on the experiment, to capture a wide array of market behaviors.

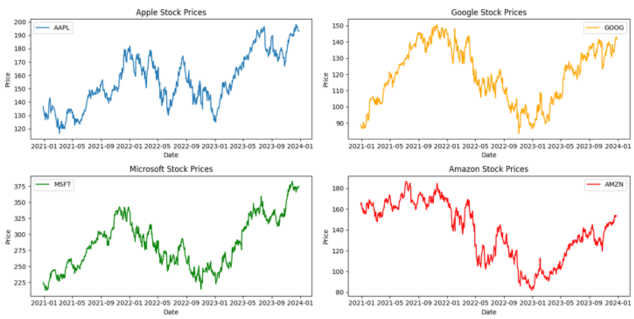



*Figure 1 Data visualization for each company.*

**Preprocess Steps**

Handling Null Data: Handling null data is a critical aspect of data preparation, especially in financial datasets where missing values can lead to inaccurate analyses and predictions. Although the current dataset shows no immediate signs of null values, typically, strategies to handle such cases would involve imputation techniques or the exclusion of incomplete records, depending on the nature and the amount of the missing data. The dataset also appears to be homogeneously typed with float64 for continuous numeric variables, which is appropriate for the quantitative nature of stock market data (Figure 2). Ensuring that all data types are consistent with the expected formats is a necessary step to avoid type-related errors during the computation.

The absence of null values in this dataset is advantageous as it implies that there may be no need for complex imputation strategies, which can introduce bias or inaccuracies, and that the dataset is relatively clean and complete. Such thorough initial data checks underscore the robustness of the data collection process and set a solid foundation for the subsequent scaling and machine learning processes.

Normalization: In the project, the MinMaxScaler from the sklearn.preprocessing module is utilized to normalize the stock market data, ensuring that each feature, such as stock prices or trading volumes, is scaled to a uniform range, typically between 0 and 1. This normalization step is important because it addresses the issue of varying scales across different features, potentially speeding up the convergence of some machine learning algorithms and improving the overall performance of the models by treating all features equally. The normalization process transforms each feature to fall within the specified range while preserving the relationships in the original data. For example, if the original stock prices vary significantly across different tech companies, normalization ensures that these values are adjusted to a comparable scale, thereby facilitating more effective learning by machine learning algorithms. This

```
DatetimeIndex: 755 entries, 2020-12-28 to 2023-12-27
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   AAPL_Open       755 non-null    float64
 1   AAPL_High       755 non-null    float64
 2   AAPL_Low        755 non-null    float64
 3   AAPL_Close      755 non-null    float64
 4   AAPL_Adj Close  755 non-null    float64
 5   AAPL_Volume     755 non-null    int64
 6   GOOG_Open       755 non-null    float64
 7   GOOG_High       755 non-null    float64
 8   GOOG_Low        755 non-null    float64
 9   GOOG_Close      755 non-null    float64
 10  GOOG_Adj Close  755 non-null    float64
 11  GOOG_Volume     755 non-null    int64
 12  MSFT_Open       755 non-null    float64
 13  MSFT_High       755 non-null    float64
 14  MSFT_Low        755 non-null    float64
 15  MSFT_Close      755 non-null    float64
 16  MSFT_Adj Close  755 non-null    float64
 17  MSFT_Volume     755 non-null    int64
 18  AMZN_Open       755 non-null    float64
 19  AMZN_High       755 non-null    float64
 20  AMZN_Low        755 non-null    float64
 21  AMZN_Close      755 non-null    float64
 22  AMZN_Adj Close  755 non-null    float64
 23  AMZN_Volume     755 non-null    int64
dtypes: float64(20), int64(4)
```

*Figure 2 Data Type and Null Analysis*

preprocessing step is particularly vital in financial datasets, where the magnitude of numbers can vary widely, making it a critical task before the data is divided into training and testing sets for further model development and evaluation.

**Sampling Period**

Incorporating techniques from a range of academic papers, our data reconfiguration utilizes a sampling period strategy for providing inputs to our forecasting models. This methodology entails the collection of data points within specific temporal segments be it 7 days, 14 days, or other regular intervals, as illustrated in Figure 3. We compile these segments into sequences that serve as discrete input samples, each sequence aiming to predict the subsequent day's closing stock price. The application of this sampling period framework is critical; it establishes a time-bound structure that captures the sequential progression of the data, equipping our models to discern and learn from the temporal patterns essential for accurate time series predictions.

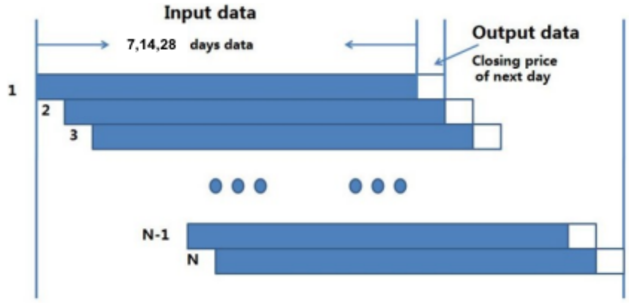



*Figure 1 Diagram of building up the sequence dataset. (Adapted from Ghadimpour et. al., 2022)*

**Correlation Matrix**

To understand the dynamics between various stock market indicators, a comprehensive correlation analysis is done. The intent was to unravel the degree of interconnectivity between diverse metrics within individual stocks. Through meticulous examination, we unveiled a substantial degree of correlation amongst different metrics corresponding to the same stock, which affirmed the potential of these metrics to serve as reliable predictors in our model development. Notably, the trading volume exhibited a comparatively lower correlation with stock prices, albeit still contributing valuable insights that warranted its inclusion in our model. While the correlations varied across different stocks, the consistent internal correlation within the metrics of each stock aligned with our preliminary expectations, strengthening our decision to integrate these findings into the development of our predictive models (Figure 4 - Figures and Tables Section).

### 5.2 Model Comparison

In the subsequent phase of the research, the study shifted its focus towards the comparative analysis of model architectures, specifically the Long Short-Term Memory (LSTM) and Transformer models. The objective was to fine-tune these models to their optimal performance capabilities before initiating a comparison. This was achieved through hyperparameter tuning, where the LSTM's architecture was scrutinized by varying the number of layers, the number of neurons per layer, and the extent of regularization employed. Concurrently, the Transformer model underwent adjustments in the number of attention heads and the dimensionality of each head.

Once the ideal configurations for both models were discovered and the model selection was concluded, the investigation proceeded to test the influence of data size on model efficacy. The first variable was the sequential input size, which entailed increasing the number of observed days, thereby providing the models with a broader temporal context for prediction. The second variable examined was the total volume of data available for training, hypothesizing its potential impact on model accuracy.

Throughout these experimental procedures, the performance of the models was evaluated using four distinct metrics. This rigorous approach was designed to ensure a comprehensive understanding of how data size and model intricacies interact, ultimately contributing to the reliability and validity of the comparative results obtained.

**Evaluation Metrics**

In the evaluation of LSTM and transformer models for financial time series forecasting, we deploy several key metrics to ensure a thorough and nuanced comparison of their predictive performances. Each metric offers unique insights into the models' abilities to capture and predict complex financial dynamics. For a comprehensive evaluation and based on the insights from Chicco, Warrens, and Jurman (2021), who argue for the use of traditional metrics like MSE, MAE, and MAPE, same metrics are chosen for the analysis. This approach not only offers a more detailed view of analyses' effectiveness, including financial time series forecasting but also enhances understanding of a model's predictive accuracy and its capability to capture underlying patterns in financial data.

-Mean Squared Error (MSE): MSE quantifies the average squared difference between the estimated values and the actual value, offering a clear measure of model accuracy. It is especially useful in emphasizing larger errors due to its squaring function. This metric is foundational for assessing the variance in predictions, highlighting models' precision in financial forecasting contexts.

-Mean Absolute Error (MAE): MAE measures the average magnitude of errors in predictions, without considering their direction. This metric provides an intuitive understanding of prediction accuracy, making it a straightforward tool for evaluating model performance in financial time series analysis.
-Mean Absolute Percentage Error (MAPE): MAPE evaluates prediction errors as a percentage, enabling comparisons across different datasets or models. This metric is particularly beneficial in financial time series forecasting, as it allows for the assessment of errors relative to the true values, offering a normalized measure of model effectiveness.
-MAPE-unscaled: Mean Absolute Percentage Error on unscaled dataset.

By incorporating these metrics into our evaluation tables, we aim to present a detailed and nuanced analysis of LSTM and transformer models' performance in financial time series forecasting, ensuring the selection of the most effective model for accurate and reliable financial predictions.

### 5.2.1 LSTM Experiments

To achieve the optimal LSTM model configuration, a systematic approach to hyperparameter tuning was undertaken. The experiment explored a range of architectures, varying the number of hidden layers from one to three, to discern the model's depth impact on performance. Neuron counts in each layer were also adjusted, with configurations set at 16, 32, and 64 neurons, probing the capacity of the network to capture complex patterns in the data. Regularization was introduced via dropout to mitigate the risk of overfitting, ensuring that the model generalizes well to unseen data.
Subsequent experiments were structured to evaluate the effect of sequential data size. The models were fed input sequences that spanned 7, 14, and 28 days, examining the influence of short-term versus long-term temporal dependencies on the forecasting accuracy. Additionally, the total amount of training data was varied across three tiers—1, 3, and 5 years—to assess how the volume of historical data feeds into the model's predictive strength.

**Hyperparameter Tuning**
The hyperparameter tuning phase for LSTM models, (Figure 5 - Figures and Tables Section), was a systematic exploration to identify the most effective network architecture. During this phase, different configurations of hidden layers and neurons were meticulously tested. The column headings in the figure denote the structure of each model variant, where 'L' indicates the number of hidden layers and 'N' signifies the number of neurons. For instance, L3-N16 refers to a model with 3 hidden layers and 16 neurons in each layer.

The analysis revealed a clear outcome: the model denoted as L1-N64, which signifies a single hidden layer with 64 neurons, emerged as the best-performing model (Figure 6). As indicated in figure 5, this configuration achieved the lowest Mean Squared Error (MSE) of 0.0017, which is a strong indicator of its predictive accuracy. Additionally, it recorded the lowest Mean Absolute Error (MAE) of 0.032 and the Mean Absolute Percentage Error (MAPE) of 9.16%, alongside the unscaled MAPE of 1.95%. These metrics collectively suggest that a simpler architecture with one hidden layer of 64 neurons outperformed more complex models, underscoring the effectiveness of a single-layer network with a higher neuron count for the task at hand.

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_7 (LSTM)               (None, 64)                20736

 dense_5 (Dense)             (None, 4)                 260

=================================================================
Total params: 20996 (82.02 KB)
Trainable params: 20996 (82.02 KB)
Non-trainable params: 0 (0.00 Byte)
```
*Figure 6 Summary of best model.*

Continuing the analysis, possible overfitting problem is also checked (Figure 7 - Figures and Tables Section). Beyond the initial sharp decrease, both the training and validation losses show a smooth and stable descent before reaching a plateau. This indicates a well-fitted model: the losses are reducing at a similar rate without large divergences. Importantly, the validation loss does not show an increase, which would have been a clear indicator of overfitting. The leveling off of both losses suggests that the model has reached a point of diminishing returns with respect to learning from the training data. This can be a signal that the model has achieved an optimal balance between bias and variance, learning the underlying patterns in the data without being swayed by noise or outliers. Moreover, the close proximity of the training and validation loss values towards the end of the training process indicates that the model should generalize well to new, unseen data, which is the ultimate goal of a predictive model. This is supported by the model's consistent performance across both training and validation sets.

In conclusion, the graph provides an evidence that the model is performing effectively without overfitting. This kind of loss analysis is important in machine learning to ensure that models not only perform well on their training data but also maintain that performance in practical, real-world applications.

**Sequence Length**
In the continued evaluation of the LSTM model's performance, the effect of sequence length on the model's

predictive accuracy is also explored (Figure 8 - Figures and Tables Section). The table presents a detailed comparison of the LSTM model's performance with different sequence lengths of observed days. The model, L1-N64, appears to excel when utilizing a short sequence length of 7 days. This optimal performance is characterized by the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE), coupled with the most favorable Mean Absolute Percentage Error (MAPE) and unscaled MAPE within this configuration. The columns within Figure 8 represent the varying sequence lengths 7, 14, and 28 days and their corresponding impact on the model's error metrics. Notably, the 7-day sequence length yields the best results, suggesting that for the task of predicting stock prices, a one-week window of recent data is more predictive of future prices than longer historical sequences. This phenomenon can be attributed to the volatile nature of the stock market, where recent trends are often a stronger indication of immediate future movements than distant past data.

It is also critical to acknowledge the significance of the right error metric in financial contexts. While a lower MSE is important, the reduced MAPE for the 7-day sequence is particularly relevant in financial applications where the percentage error directly correlates with the forecasting precision's economic impact. Therefore, the LSTM model's configuration and its associated error metrics should be carefully aligned with the specific requirements and goals of the financial task at hand.

**Amount of Data**

In the final segment of the analysis focusing on the LSTM model, Figure 9 (in Figures and Tables Section) presents the impact of the amount of training data on the model's forecasting ability. The best performing LSTM model, L1-N64, was subjected to datasets covering different time spans to determine the optimal data quantity for training.

Contrary to the results observed with varying sequence lengths, the LSTM model displayed an enhanced performance with an increase in the total amount of training data. Specifically, the model delivered its most accurate predictions when trained with five years of data, outperforming the one and three-year data models across key performance metrics.

The columns in Figure 9 represent the total amount of training data in terms of the number of years—1 year (22-23), 3 years (20-23), and 5 years (18-23). This longer historical view provided by the five-year dataset appears to be instrumental for the LSTM model, allowing it to recognize long-term trends and repetitive patterns that shorter timeframes may not reveal. Notably, the 5-year dataset resulted in the lowest Mean Squared Error (MSE) of 0.0007 and a Mean Absolute Percentage Error (MAPE) of 5.71%, underscoring the value of extensive historical data in bolstering the model's predictive accuracy.

These findings suggest that, for the stocks in question and the specific LSTM architecture employed, an expansive dataset enables a more nuanced understanding of stock price movements, which is critical for accurate forecasting in financial markets.

### 5.2.2 Transformer Experiments

Transitioning from LSTM models to their more contemporary counterparts, the capabilities of Transformer models in time series forecasting is explored. The Transformer architecture, renowned for its success in natural language processing, was adapted for the financial domain through a process of hyperparameter tuning. The model's intricacies, including head size and the number of attention heads, were fine-tuned along with the application of dropout for regularization.

We investigated the influence of sequential data size on the Transformer's performance, experimenting with sequences ranging from a week to a full year's worth of daily data. Additionally, the amount of training data was varied, testing the model on 1, 3, and 5-year datasets. These experiments aimed to identify the Transformer model's ideal configuration and data scope for accurately predicting stock prices.

AS it is done with the LSTM, these steps were essential to tailor the Transformer to the nuances of financial time series data, seeking to determine the optimal combination of hyperparameters, sequence length, and data volume for robust stock market forecasting.

**Hyperparameter Tuning**

To find an optimal Transformer model, the experimenters employed hyperparameter tuning to navigate the architecture's complexities. Figure 10 (in Figures and Tables Section) exhibits the results of this tuning, focusing on the model's head size and the number of attention heads, which are pivotal elements of the Transformer's structure.

The tuning process entailed experimenting with various head sizes and numbers of attention heads to pinpoint the most effective combination for the task at hand. The findings indicated that a Transformer with a head size of 64 and three attention heads, annotated as HS64-NH3 in the figure, yielded the best results. This model configuration outshone others with the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE), as well as a good Mean Absolute Percentage Error (MAPE) and unscaled MAPE, suggesting a high level of accuracy and reliability.

The columns in Figure 10 correspond to different configurations of head sizes and numbers of heads, providing a clear comparison of their respective performances. Notably, the HS64-NH3 model achieved superior results without the need for dropout regularization, which is commonly used to prevent overfitting. This suggests that for the financial time series forecasting task, a Transformer model with a moderately large head size and more attention heads

effectively captures the complexities of the data without overfitting, even in the absence of regularization techniques.
Continuing the analysis, possible overfitting problem is also checked (Figure 11 – Figures and Tables Section). The figure provides a visualization of the training and validation loss for the Transformer model, offering insight into the model's learning progression over epochs and its susceptibility to overfitting. The graph depicts a sharp decline in loss for both training and validation in the initial epochs, indicating that the model is learning effectively from the data. As training progresses, both losses exhibit a plateau, with the validation loss closely tracking the training loss. This convergence of training and validation loss is a positive indication that the model is not overfitting. Overfitting would typically be characterized by a continuous decrease in training loss alongside a stagnation or increase in validation loss, which is not observed in this case. The consistent decline and stabilization of the loss values suggest that the Transformer model, with its chosen hyperparameters, is capturing the underlying patterns in the data without memorizing it. This behavior is crucial for the model's ability to generalize to unseen data, which is the ultimate goal of a robust predictive model. The losses reaching a plateau also imply that additional training might yield marginal gains, indicating an optimal stopping point for the training process.

**Sequence Length**
Figure 12 (in Figures and Tables Section) illustrates the Transformer model's performance across various input sequence lengths. In contrast to the LSTM results, the Transformer showcased significant improvements as the sequence length increased. The model marked as HS64-H3, which indicates a head size of 64 and 3 heads, was particularly adept when processing longer sequences of observed days.
The experiments revealed that the Transformer's best performance was achieved with the largest sequence length of 365 days. This is evidenced by the lowest Mean Squared Error (MSE) of 0.002 and Mean Absolute Error (MAE) of 0.039, alongside the most favorable Mean Absolute Percentage Error (MAPE) of 13.24% and an unscaled MAPE of 2.5%, as highlighted in Figure 12. Such an improvement underscores the model's capacity to capture and utilize long-term dependencies and seasonal patterns inherent in stock price movements.
These findings indicate that, for this particular task, the Transformer benefits from an extensive historical context, allowing for a nuanced analysis of trends and potentially leading to more informed and accurate predictions. The lengthened input sequence provides the Transformer model with a comprehensive view, which is critical in understanding and forecasting the intricate behavior of stock prices.

**Amount of Data**
Figure 13 (in Figures and Tables Section) captures the performance of the best transformer model, denoted as NS64-H3, across different total amounts of training data. Consistent with the observations from increasing the sequence length, the model's accuracy improved with a larger dataset. The five-year dataset, in particular, led to the best model performance, as evidenced by the lower Mean Squared Error (MSE) and Mean Absolute Error (MAE).
While the trend suggests that more data correlates with better performance, the improvement rate was less pronounced compared to the effect of increased sequence length. This indicates that while the model benefits from more data, the advantages gained from each additional year of data diminish as the total dataset grows.
Despite the general trend of improvement, an anomaly was noted in the Mean Absolute Percentage Error (MAPE) when the data span was increased from one to three years. This anomaly was less evident when the data was further increased to five years. Understanding the reason behind this irregularity in the MAPE could be crucial. It raises questions about the model's interaction with the underlying market dynamics during the specific periods covered by the three-year dataset. Further analysis could involve scrutinizing the model's predictions in the context of major financial events or changes in market behavior to gain a deeper understanding of the observed anomaly.
In summary, the results from Figure 13 suggest that the transformer model leverages larger datasets to refine its forecasting capabilities, though the relative benefit of additional data appears to decrease as the amount of data grows. The peculiarities observed in the MAPE underscore the need for a more granular analysis to unravel the complexities of the model's learning process in relation to the temporal characteristics of the financial data.

**Conclusion of Experimental Findings**
In the comparative analysis of LSTM and Transformer models, LSTMs have demonstrated superior performance under the same experimental conditions, both in terms of the number of observed days and the total amount of training data. However, Transformers have shown promising potential, particularly when the input data is increased. They exhibit significant improvements in performance with an increase in the number of observed days a trend not as pronounced in LSTMs.
While LSTMs currently lead in handling large datasets, the observed improvements in Transformer models suggest that they may excel with even larger volumes of data. The capacity of Transformers to better leverage extended sequential input highlights their potential advantage in environments with substantial historical data.

Nevertheless, some anomalies have been noted in the performance metrics, which may be attributed to unique market conditions within the specific years of the training data. These anomalies present an opportunity for future research to dissect the influence of market events on the predictive capabilities of these models.
Future model development could benefit from a more expansive dataset and refined hyperparameter tuning. Exploring different approaches to the sliding window effect may also yield improvements, especially for Transformers. Given that Transformers improve significantly with more extended data sequences, they hold promise for becoming the preferred model in scenarios where vast amounts of historical data are available for training.

In conclusion, while LSTMs currently show a stronger performance in forecasting with a substantial amount of training data and across different observed day lengths, the transformative improvements seen in Transformers with increased data input point towards their potential ascendency in handling large-scale financial time series forecasting in the future. Further investigations into their performance, particularly under varying market conditions and with enhanced hyperparameter optimization, will be essential to fully harness their capabilities.

**Bridging Models and Users**

Following the analytical exploration of LSTM and Transformer models, we have initiated a practical application by creating a Streamlit web application, which can be considered a mini product. Streamlit's open-source framework has enabled us to construct a simple yet functional interface for our project, as demonstrated in Figures 14 and 15 in figures and tables section.
This web application presents a straightforward way for users to observe the outcomes of our LSTM and Transformer models in comparison to actual stock prices. It offers basic functionality for users to select a company and visualize the model's predictions against the true stock values. The app's simplicity is key, it requires no prior knowledge of HTML, CSS, or Javascript, making it accessible to a wide audience.
Currently, the application is basic and has limitations. It does not feature live data or continuous model updates, which are critical for real-time financial decision-making. In future work, enhancing the app by integrating these elements could make it a more valuable tool for users interested in daily investment activities.
The creation of this Streamlit app is a small, yet important step towards applying complex machine learning models in a user-friendly manner. As a preliminary product, it opens up possibilities for non-technical users to interact with advanced data predictions and serves as a starting point for more sophisticated developments in the future.

## 6. Future Work

The exploration of LSTM and transformer models in financial time series prediction, while comprehensive, opens several avenues for future research. This section outlines potential directions for advancing this field of study.

**Enhancing Model Interpretability**

-Interpretable AI in Finance: The development of interpretable Spatio-Temporal Attention Long Short Term Memory models (STA-LSTM), as proposed by Ding et al. (2020), showcases the potential for making LSTM structures transparent. By integrating attention mechanisms, this approach not only retains the model's accuracy but also enhances its interpretability. Although originally applied to flood forecasting, the underlying methodology presents a viable pathway for financial time series analysis, suggesting that complex LSTM architectures can be unraveled for clearer insight into their operational logic.
-Incorporating Explainable AI Techniques: The exploration of Explainable AI (XAI) techniques offers another avenue for demystifying the inner workings of machine learning models in finance. The comparative analysis of XAI methods by Y and Challa (2023) emphasizes the utility of SHAP and LIME in elucidating the features and decision paths models rely on. Such techniques promise to peel back the layers of both LSTM and transformer models, offering stakeholders a window into the rationale behind predictions and thus fostering trust in AI-driven financial systems.

The integration of these interpretative frameworks into financial forecasting models signals a transformative shift towards models that do not merely predict with high accuracy but also operate within an arena of transparency and trustworthiness. This evolution is critical for financial decision-making, where the implications of predictions extend beyond the abstract and into the concrete realms of economics and societal impact.

**Advancing Transformer Models**

The advancement of transformer models for time series analysis, especially in financial markets, has prompted a significant shift towards creating models that balance computational efficiency with predictive accuracy. The emergence of models like the Informer, as developed by Zhou et al. (2020), represents a pivotal step in this direction. By implementing a ProbSparse self-attention mechanism, the Informer model markedly reduces the computational footprint traditionally associated with transformers, offering a solution that is both practical and effective for long sequence time-series forecasting. This innovation is particularly relevant for financial applications where the ability to process extensive sequences efficiently can significantly enhance forecasting capabilities.

Further refining the transformer's application to time series data, Li et al. (2019) propose an adaptation that incorporates a convolutional self-attention mechanism. This approach seeks to mitigate the transformer architecture's inherent insensitivity to the local context, a critical aspect for analyzing time series data. By integrating causal convolution into the generation of queries and keys, their model enhances the transformer's ability to discern local patterns and anomalies within time series, making it more adept at forecasting in environments characterized by rapid changes, such as financial markets.

**Integrating Alternative Data Sources**

The exploration into enhancing financial time series prediction models reveals a growing interest in integrating unstructured and semi-structured data sources. This approach aims to leverage the depth of information available in financial news, social media sentiments, and economic indicators to improve forecasting accuracy.

-Exploiting Unstructured Data: Rajakumar et al. (2019) delve into the impact of utilizing unstructured qualitative data, such as news articles and financial information, on stock market predictions. Their study demonstrates that incorporating such data through a multilayer feedforward network significantly boosts the accuracy of stock price predictions. This method highlights the untapped potential of unstructured data in revealing market sentiments and trends, providing a richer context for prediction models beyond traditional numerical data.

-Combining Quantitative and Qualitative Data: In a complementary approach, Sinha et al. (2021) investigate the benefits of melding machine learning algorithms with diverse stock market data, encompassing both numerical and textual formats. Their application of an unsupervised learning algorithm to this amalgamated data set achieves an impressive 88.7% accuracy in predicting stock prices. This method underscores the value of integrating various data sources for a holistic and nuanced analysis of stock market trends, setting a new benchmark for predictive accuracy.

These investigations underscore a pivotal shift toward more comprehensive models that incorporate a wide array of data types, from the structured numerical to the nuanced textual, promising more sophisticated and accurate tools for navigating the complexities of financial markets.

The exploration of LSTM and transformer models in financial time series prediction has provided substantial insights, yet it also opens avenues for future research. The development of interpretable models such as STA-LSTM suggests that enhancing transparency in complex LSTM architectures is both viable and crucial, especially within the financial sector where decision-making hinges on clarity and trust. Explorable AI (XAI) techniques are also set to play a pivotal role in demystifying the decision-making processes of these advanced models. By employing methods like SHAP and LIME, there is potential to elucidate the intricate features and decision pathways, fostering a deeper trust in the predictions made by AI in financial systems.

On another front, the advancement of transformer models tailored for time series data presents opportunities for improving efficiency in real-time financial applications. Models like the Informer reduce computational demands while maintaining accuracy, offering practical solutions for long sequence forecasting. Furthermore, integrating unstructured data sources such as financial news and social media sentiments with quantitative trading data could yield more holistic approaches to market prediction. This combination of data types aims to leverage the strengths of both LSTM and transformer models in processing diverse datasets. As the field stands at the precipice of these advancements, the integration of alternative data sources, the pursuit of model interpretability, and the refinement of transformer models for time-series data are promising directions that could significantly enhance the accuracy and utility of financial time series prediction.

## 7. Conclusion

Transformers have been making waves across various industries, revolutionizing the way we approach time series data analysis. Historically, industries have relied on LSTM models for their predictive capabilities. However, with the advent of transformers, a shift has occurred, demonstrating significant improvements in performance across numerous domains. Motivated by this advancement, we aimed to conduct a comparative analysis using financial data, focusing specifically on the S&P 500's top four technology companies: Apple, Amazon, Google, and Microsoft. Our goal was to leverage data from Yahoo Finance to experiment with these models, evaluating their effectiveness in the context of financial forecasting. Additionally, we aspired to create a mini product that could assist people in making informed investment decisions regarding such companies, further bridging the gap between theoretical models and practical applications.

Our experiments revealed that while LSTMs show remarkable proficiency in processing sequential data and capturing short-to-medium term dependencies, they are hampered by computational intensity and a susceptibility to overfitting, particularly in high-dimensional datasets. On the other hand, transformer models, with their self-attention mechanism, excel in managing longer sequences and larger datasets, demonstrating superior performance in capturing long-range dependencies. This distinction underscores the importance of model selection based on the specific characteristics of the financial time series data at hand.

The practical application of these findings was further explored through the development of a Streamlit web application, offering a user-friendly interface for comparing model predictions against actual stock prices. This endeavor highlights the potential for these advanced machine learning models to be leveraged in real-world financial analysis tools, despite the noted limitations in real-time data processing and model updating.

Looking ahead, the field of financial forecasting stands on the cusp of significant advancements. The observed strengths and limitations of LSTM and transformer models pave the way for future research directions. Enhancing model interpretability, integrating alternative data sources, and refining model architectures to balance computational efficiency with predictive accuracy are critical areas that promise to elevate the utility and applicability of these models in financial time series forecasting. Moreover, the exploration of hybrid models that amalgamate the strengths of LSTMs and transformers could offer groundbreaking improvements in predictive performance.

In conclusion, our study contributes valuable insights into the capabilities and limitations of LSTM and transformer models in financial forecasting. While LSTMs currently demonstrate robust performance across various datasets, the adaptive improvements and scalability of transformer models signal their potential dominance in future financial time series analysis. Continued innovation and research in this domain are essential to harness the full potential of these advanced machine learning models, ensuring they can meet the evolving demands of financial market analysis.

## References


1. Kelany, O., Aly, S., & Ismail, M. (2020). Deep Learning Model for Financial Time Series Prediction. 2020 14th International Conference on Innovations in Information Technology (IIT), 120-125.
2. Ouahilal, M., Mohajir, M. E., Chahhou, M., & El Mohajir, B. E. (2016). Optimizing stock market price prediction using a hybrid approach based on HP filter and support vector regression. 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt).
3. Verma, N., & Mohapatra, B. (2020). Stock Market Predication Using Machine Learning. Mechanical Engineering eJournal.
4. Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R., & Wu, B. (2019). AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction. IOP Conference Series: Materials Science and Engineering.
5. Shen, L., & Wang, Y. (2021). TCCT: Tightly-Coupled Convolutional Transformer on Time Series Forecasting. ArXiv.
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. ArXiv.
7. Malibari, N., Katib, I. A., & Mehmood, R. (2021). Predicting Stock Closing Prices in Emerging Markets with Transformer Neural Networks: The Saudi Stock Exchange Case. International Journal of Advanced Computer Science and Applications.
8. Li, Y., Lv, S., Liu, X., & Zhang, Q. (2022). Incorporating Transformers and Attention Networks for Stock Movement Prediction. Complexity.
9. Xiaofeng Yuan, Lin Li and Yalin Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3168-3176, 2019.
10. Musial, J., Verstraete, M., & Gobron, N. (2011). Technical Note: Comparing the effectiveness of recent algorithms to fill and smooth incomplete and noisy time series. Atmospheric Chemistry and Physics, 11, 7905-7923.
11. Obaid, H., Dheyab, S., & Sabry, S. (2019). The Impact of Data Pre-Processing Techniques and Dimensionality Reduction on the Accuracy of Machine Learning. 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), 279-283.
12. Ambarwari, A., Adrian, Q., & Herdiyeni, Y. (2020). Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification. , 4, 117-122.
13. Tsang, E., Tao, R., Serguieva, A., & Ma, S. (2017). Profiling high-frequency equity price movements in directional changes. Quantitative Finance, 17, 217 - 225.
14. Choi, J., & Lee, B. (2018). Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting. Mathematical Problems in Engineering.
15. Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2018). Deep Learning with Long Short-Term Memory for Time Series Prediction. IEEE Communications Magazine, 57, 114-119.
16. Yan, H., & Ouyang, H. (2018). Financial Time Series Prediction Based on Deep Learning. *Wireless Personal Communications*, 102, 683-700.
17. Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE, 12.
18. Zhang, J. (2023). Research on transformer and attention in applied algorithms. Applied and Computational Engineering.
19. Ham, T., Lee, Y., Seo, S., Kim, S., Choi, H., Jung, S., & Lee, J. (2021). ELSA: Hardware-Software Co-design for Efficient, Lightweight Self-Attention Mechanism in Neural Networks. 2021 ACM/IEEE 48th Annual

International Symposium on Computer Architecture (ISCA), 692-705.
20. Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2020). Efficient Transformers: A Survey. *ACM Computing Surveys*, 55, 1 - 28.
21. Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M., Kranthikiran, G., He, X., Hou, H., Kazienko, P., Kocoń, J., Kong, J., Koptyra, B., Lau, H., Mantri, K., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., & Zhu, R. (2023). RWKV: Reinventing RNNs for the Transformer Era. , 14048-14077.
22. Mahdavi, S., Liao, R., & Thrampoulidis, C. (2023). Memorization Capacity of Multi-Head Attention in Transformers. *ArXiv*, abs/2306.02010.
23. Ruoss, A., Del'etang, G., Genewein, T., Grau-Moya, J., Csordás, R., Bennani, M., Legg, S., & Veness, J. (2023). Randomized Positional Encodings Boost Length Generalization of Transformers. *ArXiv*, abs/2305.16843.
24. Xu, C., Li, J., Feng, B., & Lu, B. (2023). A Financial Time-Series Prediction Model Based on Multiplex Attention and Linear Transformer Structure. *Applied Sciences*.
25. Emami, H., Dang, X., Shah, Y., & Zerfos, P. (2023). Modality-aware Transformer for Time series Forecasting. *ArXiv*, abs/2310.01232.
26. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2020). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. , 11106-11115.
27. Murray, C., Chaurasia, P., Hollywood, L., & Coyle, D. (2022). A Comparative Analysis of State-of-the-Art Time Series Forecasting Algorithms. *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, 89-95.
28. Chicco, D., Warrens, M., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7.
29. Bilokon, P., & Qiu, Y. (2023). Transformers versus LSTMs for electronic trading. *ArXiv*, abs/2309.11400.
30. Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*.
31. Trinh, M., Behery, M., Emara, M., Lakemeyer, G., Storms, S., & Brecher, C. (2022). Dynamics Modeling of Industrial Robots Using Transformer Networks. *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, 164-171.
32. Ding, Y., Zhu, Y., Feng, J., Zhang, P., & Cheng, Z. (2020). Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing*, 403, 348-359.
33. Y, S., & Challa, M. (2023). A Comparative Analysis of Explainable AI Techniques for Enhanced Model Interpretability. *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, 229-234.
34. LI, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y., & Yan, X. (2019). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting.
35. Rajakumar, M., Jegatheesan, R., Chandy, R., & Sampath, T. (2019). Prediction of Stock Prices Using Unstructured and Semi-structured Qualitative Data – A Neural Network Approach. *International Journal of Intelligent Engineering and Systems*.
36. Sinha, A., Mishra, P., Ramish, M., Mahmood, H., & Upadhyay, K. (2021). Employing Unsupervised Learning Algorithm for Stock Market Analysis and Prediction. *2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT)*, 75-79.
37. Ghadimpour, Mohammadreza & Ebrahimi, Seyed. (2022). Forecasting Financial Time Series Using Deep Learning Networks: Evidence from Long-Short Term Memory and Gated Recurrent Unit. Iranian Journal of Finance. 6. 81-94.

## Figures and Tables

| | AAPL_Open | AAPL_High | AAPL_Low | AAPL_Close | AAPL_Adj Close | AAPL_Volume | GOOG_Open | GOOG_High | GOOG_Low | GOOG_Close | GOOG_Adj Close | GOOG_Volume |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AAPL_Open | 1.0 | 0.997532059018671 | 0.997296760058152 | 0.9935585006998039 | 0.9932906896371949 | -0.3904444609203789 | 0.5157263461735914 | 0.5191388509094096 | 0.5106119879706684 | 0.5104423404100501 | 0.5104423404100501 | -0.14412513977639238 |
| AAPL_High | 0.997532059018671 | 1.0 | 0.9969690716222449 | 0.9970774056559856 | 0.9967091991169075 | -0.36967016971262145 | 0.5110390315445094 | 0.5167503564673651 | 0.5069897138820142 | 0.5089641893435178 | 0.5089641893435178 | -0.13765890567074113 |
| AAPL_Low | 0.997296760058152 | 0.9969690716222449 | 1.0 | 0.9972715650263152 | 0.9970107009579469 | -0.4162130457003344 | 0.5208626753049229 | 0.525320979959701 | 0.5184857653229712 | 0.5192453879364266 | 0.5192453879364266 | -0.15795047999297285 |
| AAPL_Close | 0.9935585006998039 | 0.9970774056559856 | 0.9972715650263152 | 1.0 | 0.999622956224953 | -0.39436272215060925 | 0.5147845953487149 | 0.5214500611449436 | 0.513350286594717 | 0.5173861222104186 | 0.5173861222104186 | -0.15058783725559657 |
| AAPL_Adj Close | 0.9932906896371949 | 0.9967091991169075 | 0.9970107009579469 | 0.999622956224953 | 1.0 | -0.4033890023181046 | 0.5006340751880115 | 0.5073117922640371 | 0.499283876303127 | 0.503290140515592 | 0.503290140515592 | -0.15069378205232017 |
| AAPL_Volume | -0.3904444609203789 | -0.36967016971262145 | -0.4162130457003344 | -0.39436272215060925 | -0.4033890023181046 | 1.0 | -0.14032037220694984 | -0.13314179603529794 | -0.15062219956401227 | -0.1425133251031233 | -0.1425133251031233 | 0.4029395102617401 |
| GOOG_Open | 0.5157263461735914 | 0.5110390315445094 | 0.5208626753049229 | 0.5147845953487149 | 0.5006340751880115 | -0.14032037220694984 | 1.0 | 0.9978607843784123 | 0.9978094228150483 | 0.9947683528575467 | 0.9947683528575467 | -0.24202166066846822 |
| GOOG_High | 0.5191388509094096 | 0.5167503564673651 | 0.525320979959701 | 0.5214500611449436 | 0.5073117922640371 | -0.13314179603529794 | 0.9978607843784123 | 1.0 | 0.9976577996037943 | 0.9976921491054099 | 0.9976921491054099 | -0.22982463004931913 |
| GOOG_Low | 0.5106119879706684 | 0.5069897138820142 | 0.5184857653229712 | 0.513350286594717 | 0.499283876303127 | -0.15062219956401227 | 0.9978094228150483 | 0.9976577996037943 | 1.0 | 0.9978834214262379 | 0.9978834214262379 | -0.2658161477191655 |
| GOOG_Close | 0.5104423404100501 | 0.5089641893435178 | 0.5192453879364266 | 0.5173861222104186 | 0.503290140515592 | -0.1425133251031233 | 0.9947683528575467 | 0.9976921491054099 | 0.9978834214262379 | 1.0 | 1.0 | -0.25232988977773013 |
| GOOG_Adj Close | 0.5104423404100501 | 0.5089641893435178 | 0.5192453879364266 | 0.5173861222104186 | 0.503290140515592 | -0.1425133251031233 | 0.9947683528575467 | 0.9976921491054099 | 0.9978834214262379 | 1.0 | 1.0 | -0.25232988977773013 |
| GOOG_Volume | -0.14412513977639238 | -0.13765890567074113 | -0.15795047999297285 | -0.15058783725559657 | -0.15069378205232017 | 0.4029395102617401 | -0.24202166066846822 | -0.22982463004931913 | -0.2658161477191655 | -0.25232988977773013 | -0.25232988977773013 | 1.0 |
| MSFT_Open | 0.8620592991582099 | 0.8570623440065892 | 0.8681518683912054 | 0.8614490844483509 | 0.8579342828053377 | -0.3724640488709758 | 0.7830611486136233 | 0.7840938866174247 | 0.7826838847088199 | 0.78124150096266 | 0.78124150096266 | -0.22431712738055012 |
| MSFT_High | 0.8637568720228501 | 0.8613086264924521 | 0.8709972068991946 | 0.8667052846966173 | 0.8632475170772841 | -0.3639043340353983 | 0.7783613461098555 | 0.7819361540664109 | 0.7791615067451849 | 0.7800442724257717 | 0.7800442724257717 | -0.21400640075387084 |
| MSFT_Low | 0.854375489812966 | 0.8503744046013603 | 0.8635424831233289 | 0.8577640178297299 | 0.8543246701965477 | -0.3864407951230341 | 0.7820336590613803 | 0.7842581209675473 | 0.7846536054474371 | 0.7841303876105348 | 0.7841303876105348 | -0.24112576994118992 |
| MSFT_Close | 0.8543941105636038 | 0.852783106871201 | 0.8644497095478008 | 0.8622274649628393 | 0.85875503112975 | -0.3758712171170887 | 0.7776728222417361 | 0.7822094208351371 | 0.7811760946403791 | 0.7839017515211472 | 0.7839017515211472 | -0.23076983989655003 |
| MSFT_Adj Close | 0.8676911438396618 | 0.8661248845900369 | 0.8775897193016721 | 0.8753794130418301 | 0.8730724071070917 | -0.39246042485877175 | 0.7534746261417549 | 0.7581541701441304 | 0.7569358071390138 | 0.7596980251588508 | 0.7596980251588508 | -0.2293705728389687 |
| MSFT_Volume | -0.016633182929322605 | -0.00505577892518783 | -0.034186105008529914 | -0.022696918245665224 | -0.02379095255637256 | 0.470716878928922 | -0.15065454636699604 | -0.13774717538083972 | -0.16827053671630707 | -0.1553019462453538 | -0.1553019462453538 | 0.6595620063132489 |
| AMZN_Open | -0.04564237901809211 | -0.05453193906391478 | -0.040607733872313145 | -0.049366188415947414 | -0.0712138727681044 | 0.20081203542314238 | 0.6539027271672164 | 0.64563779832182 | 0.6574799623827576 | 0.6497818602086957 | 0.6497818602086957 | -0.14647097203253134 |
| AMZN_High | -0.04887845371242271 | -0.055731908835496155 | -0.04286182725509403 | -0.04967093785380341 | -0.07164159860243037 | 0.2113729210209454 | 0.6536756688620902 | 0.6473055528225821 | 0.6582516450249799 | 0.652202023591217 | 0.652202023591217 | -0.14058386845005202 |
| AMZN_Low | -0.04725315537928132 | -0.05532474450222455 | -0.03982227382138392 | -0.047795758403424704 | -0.06946156074311932 | 0.1865052890827277 | 0.6521110756161633 | 0.6448586137485414 | 0.6581641360163205 | 0.6511871999804288 | 0.6511871999804288 | -0.1604559577530273 |
| AMZN_Close | -0.050021667752168784 | -0.0561539431868838 | -0.04185213343382593 | -0.04695139201462706 | -0.06875668424156922 | 0.19957286644384709 | 0.6519487530701384 | 0.6464109662701145 | 0.6587205816833871 | 0.654288574492692 | 0.654288574492692 | -0.1527871390065579 |
| AMZN_Adj Close | -0.050021667752168784 | -0.0561539431868838 | -0.04185213343382593 | -0.04695139201462706 | -0.06875668424156922 | 0.19957286644384709 | 0.6519487530701384 | 0.6464109662701145 | 0.6587205816833871 | 0.654288574492692 | 0.654288574492692 | -0.1527871390065579 |
| AMZN_Volume | -0.22770492832534098 | -0.21301757055110412 | -0.24262872809637173 | -0.22967493918662457 | -0.23151033051739653 | 0.5041853159628087 | -0.17409244466855317 | -0.1635936597346549 | -0.18546952610544207 | -0.17549797549933227 | -0.17549797549933227 | 0.49059339973349103 |

| | MSFT_Open | MSFT_High | MSFT_Low | MSFT_Close | MSFT_Adj Close | MSFT_Volume | AMZN_Open | AMZN_High | AMZN_Low | AMZN_Close | AMZN_Adj Close | AMZN_Volume |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AAPL_Open | 0.8620592991582099 | 0.8637568720228501 | 0.854375489812966 | 0.8543941105636038 | 0.8676911438396618 | -0.016633182929322605 | -0.04564237901809211 | -0.04887845371242271 | -0.04725315537928132 | -0.050021667752168784 | -0.050021667752168784 | -0.2277049283253409 |
| AAPL_High | 0.8570623440065892 | 0.8613086264924521 | 0.8503744046013603 | 0.852783106871201 | 0.8661248845900369 | -0.00505577892518783 | -0.05453193906391478 | -0.055731908835496155 | -0.05532474450222455 | -0.0561539431868838 | -0.0561539431868838 | -0.2130175705511041 |
| AAPL_Low | 0.8681518683912054 | 0.8709972068991946 | 0.8635424831233289 | 0.8644497095478008 | 0.8775897193016721 | -0.034186105008529914 | -0.040607733872313145 | -0.04286182725509403 | -0.03982227382138392 | -0.04185213343382593 | -0.04185213343382593 | -0.2426287280963717 |
| AAPL_Close | 0.8614490844483509 | 0.8667052846966173 | 0.8577640178297299 | 0.8622274649628393 | 0.8753794130418301 | -0.022696918245665224 | -0.049366188415947414 | -0.04967093785380341 | -0.047795758403424704 | -0.04695139201462706 | -0.04695139201462706 | -0.2296749391866245 |
| AAPL_Adj Close | 0.8579342828053377 | 0.8632475170772841 | 0.8543246701965477 | 0.85875503112975 | 0.8730724071070917 | -0.02379095255637256 | -0.0712138727681044 | -0.07164159860243037 | -0.06946156074311932 | -0.06875668424156922 | -0.06875668424156922 | -0.2315103305173965 |
| AAPL_Volume | -0.3724640488709758 | -0.3639043340353983 | -0.3864407951230341 | -0.3758712171170887 | -0.39246042485877175 | 0.470716878928922 | 0.20081203542314238 | 0.2113729210209454 | 0.1865052890827277 | 0.19957286644384709 | 0.19957286644384709 | 0.5041853159628087 |
| GOOG_Open | 0.7830611486136233 | 0.7783613461098555 | 0.7820336590613803 | 0.7776728222417361 | 0.7534746261417549 | -0.15065454636699604 | 0.6539027271672164 | 0.6536756688620902 | 0.6521110756161633 | 0.6519487530701384 | 0.6519487530701384 | -0.1740924446685531 |
| GOOG_High | 0.7840938866174247 | 0.7819361540664109 | 0.7842581209675473 | 0.7822094208351371 | 0.7581541701441304 | -0.13774717538083972 | 0.64563779832182 | 0.6473055528225821 | 0.6448586137485414 | 0.6464109662701145 | 0.6464109662701145 | -0.1635936597346549 |
| GOOG_Low | 0.7826838847088199 | 0.7791615067451849 | 0.7846536054474371 | 0.7811760946403791 | 0.7569358071390138 | -0.16827053671630707 | 0.6574799623827576 | 0.6582516450249799 | 0.6581641360163205 | 0.6587205816833871 | 0.6587205816833871 | -0.1854695261054420 |
| GOOG_Close | 0.78124150096266 | 0.7800442724257717 | 0.7841303876105348 | 0.7839017515211472 | 0.7596980251588508 | -0.1553019462453538 | 0.6497818602086957 | 0.652202023591217 | 0.6511871999804288 | 0.654288574492692 | 0.654288574492692 | -0.1754979754993322 |
| GOOG_Adj Close | 0.78124150096266 | 0.7800442724257717 | 0.7841303876105348 | 0.7839017515211472 | 0.7596980251588508 | -0.1553019462453538 | 0.6497818602086957 | 0.652202023591217 | 0.6511871999804288 | 0.654288574492692 | 0.654288574492692 | -0.1754979754993322 |
| GOOG_Volume | -0.22431712738055012 | -0.21400640075387084 | -0.24112576994118992 | -0.23076983989655003 | -0.2293705728389687 | 0.6595620063132489 | -0.14647097203253134 | -0.14058386845005202 | -0.1604559577530273 | -0.1527871390065579 | -0.1527871390065579 | 0.49059339973349103 |
| MSFT_Open | 1.0 | 0.9980197975380641 | 0.9978519517170179 | 0.9949701084854337 | 0.9939033433436916 | -0.09309293616088418 | 0.25801866247854593 | 0.25541762672089185 | 0.2605739046230171 | 0.2577368960605742 | 0.2577368960605742 | -0.2498140915481851 |
| MSFT_High | 0.9980197975380641 | 1.0 | 0.9975532113271057 | 0.9977532444557153 | 0.9968932125191582 | -0.07339055940236741 | 0.24796706974855892 | 0.24699084884687234 | 0.2512820354636898 | 0.25012781145811486 | 0.25012781145811486 | -0.2400953484787537 |
| MSFT_Low | 0.9978519517170179 | 0.9975532113271057 | 1.0 | 0.9978678000724593 | 0.9966692092287023 | -0.11858650047764528 | 0.2631847379245511 | 0.2614135824551257 | 0.26807802674996134 | 0.26588539190873123 | 0.26588539190873123 | -0.263628710166348 |
| MSFT_Close | 0.9949701084854337 | 0.9977532444557153 | 0.9978678000724593 | 1.0 | 0.9988678176584002 | -0.09824833536053854 | 0.25374650472979543 | 0.25362130358179075 | 0.25926545319849087 | 0.25964950563769146 | 0.25964950563769146 | -0.2526710234291158 |
| MSFT_Adj Close | 0.9939033433436916 | 0.9968932125191582 | 0.9966692092287023 | 0.9988678176584002 | 1.0 | -0.09742628481322618 | 0.21352131243421546 | 0.21318988489143134 | 0.21920688967442017 | 0.21940541272339895 | 0.21940541272339895 | -0.2570839045665555 |
| MSFT_Volume | -0.09309293616088418 | -0.07339055940236741 | -0.11858650047764528 | -0.09824833536053854 | -0.09742628481322618 | 1.0 | -0.1463018962595187 | -0.13885097782291647 | -0.1617709278077405 | -0.151332416457868 | -0.151332416457868 | 0.4463717660309609 |
| AMZN_Open | 0.25801866247854593 | 0.24796706974855892 | 0.2631847379245511 | 0.25374650472979543 | 0.21352131243421546 | -0.1463018962595187 | 1.0 | 0.9985458829586706 | 0.9984141487286082 | 0.9961603784480321 | 0.9961603784480321 | -0.0854457234151159 |
| AMZN_High | 0.25541762672089185 | 0.24699084884687234 | 0.2614135824551257 | 0.25362130358179075 | 0.21318988489143134 | -0.13885097782291647 | 0.9985458829586706 | 1.0 | 0.9983890215106924 | 0.9983264516888262 | 0.9983264516888262 | -0.0705272842986231 |
| AMZN_Low | 0.2605739046230171 | 0.2512820354636898 | 0.26807802674996134 | 0.25926545319849087 | 0.21920688967442017 | -0.1617709278077405 | 0.9984141487286082 | 0.9983890215106924 | 1.0 | 0.9984509125999036 | 0.9984509125999036 | -0.1027076466975645 |
| AMZN_Close | 0.2577368960605742 | 0.25012781145811486 | 0.26588539190873123 | 0.25964950563769146 | 0.21940541272339895 | -0.151332416457868 | 0.9961603784480321 | 0.9983264516888262 | 0.9984509125999036 | 1.0 | 1.0 | -0.0872373932143251 |
| AMZN_Adj Close | 0.2577368960605742 | 0.25012781145811486 | 0.26588539190873123 | 0.25964950563769146 | 0.21940541272339895 | -0.151332416457868 | 0.9961603784480321 | 0.9983264516888262 | 0.9984509125999036 | 1.0 | 1.0 | -0.0872373932143251 |
| AMZN_Volume | -0.24981409154818518 | -0.2400953484787537 | -0.263628710166348 | -0.25267102342911585 | -0.25708390456655555 | 0.4463717660309609 | -0.08544572341511592 | -0.07052728429862315 | -0.1027076466975645 | -0.08723739321432515 | -0.08723739321432515 | 1.0 |

*Figure 2 Correlation Matrix*

**Tables and Figures Related to LSTM Models**

| | L3-N16 | L3-N32 | L3-N64 | L3-R-3layers | L3-R-2layers | L2-N16 | L2-N32 | L2-N64 | L1-N16 | L1-N32 | L1-N64-(best model) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MSE | 0.0037 | 0.0025 | 0.024 | 0.00388 | 0.0399 | 0.0024 | 0.0021 | 0.0020 | 0.0019 | 0.018 | 0.0017 |
| MAE | 0.048 | 0.040 | 0.0397 | 0.0496 | 0.0509 | 0.042 | 0.037 | 0.036 | 0.034 | 0.033 | 0.032 |
| MAPE | %15.49 | %12.22 | %10.69 | | | %12.48 | %10.94 | %10.04 | %10.47 | %9.28 | %9.16 |
| MAPE-unscaled | %3.00 | %2.48 | %2.42 | | | %2.49 | %2.28 | %2.19 | %2.11 | %2.03 | %1.95 |

*Figure 3 Parameters: Hidden Layers (0,1,2), number of neuron (16,32,64) and regularization effect.*

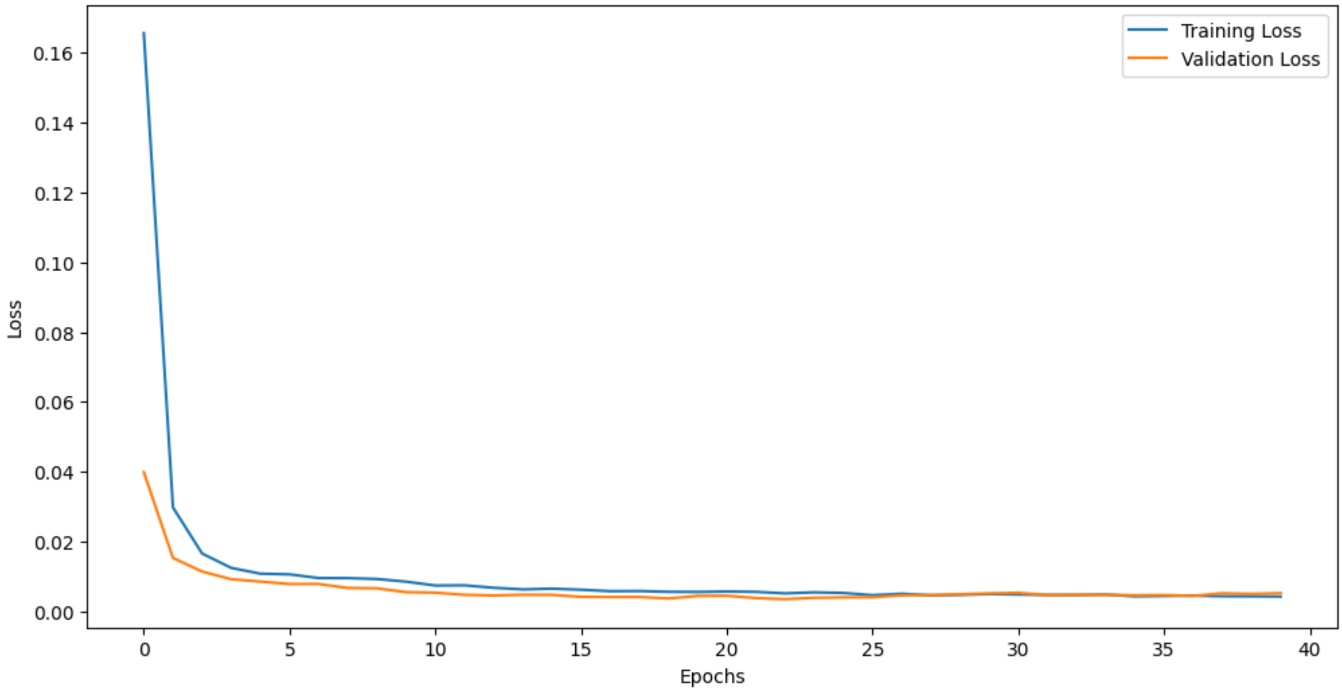



*Figure 7 Training and validation losses over epochs.*

| | 7days | 14 days | 28 days |
|---|---|---|---|
| MSE | 0.0017 | 0.0023 | 0.0020 |
| MAE | 0.032 | 0.038 | 0.035 |
| MAPE | %9.16 | %12.44 | %13.52 |
| MAPE-unscaled | %1.95 | %2.33 | %2.16 |

*Figure 8 The effect of sequence length on the best model L1-N64.*

| | 1 year (22-23) | 3 years (20-23) | 5 years (18-23) |
|---|---|---|---|
| MSE | 0.0029 | 0.0017 | 0.0007 |
| MAE | 0.0423 | 0.032 | 0.0188 |
| MAPE | %10.14 | %9.16 | %5.71 |
| MAPE-unscaled | %2.12 | %1.95 | %2.05 |

*Figure 9 Data period from 1 to 5 years*

**Tables and Figures Related to Transformers**

| | HS32-NH2 | HS32-NH3 | reg 0.2-0.4 | HS64-NH2 | HS64-NH3 | reg 0.2-0.4 | HS128-NH2 | HS128-NH3 | reg 0.2-0.4-0.5 |
|---|---|---|---|---|---|---|---|---|---|
| MSE | 0.014 | 0.015 | 0.023-0.027 | 0.017 | 0.014 | 0.015-0.017 | 0.017 | 0.017 | 0.028-0.018-0.040 |
| MAE | 0.096 | 0.100 | 0.123-0.134 | 0.101 | 0.093 | 0.096-0.109 | 0.102 | 0.103 | 0.124-0.114-0.148 |
| MAPE | %28.67 | %30.42 | | %26.95 | %25.90 | | %29.66 | %28.06 | |
| MAPE-unscaled | %5.90 | %6.23 | | %6.38 | %5.69 | | %6.30 | %6.45 | |

*Figure 10 Hyperparameter Optimization – 3 Years data – Sequence length 7 days.*

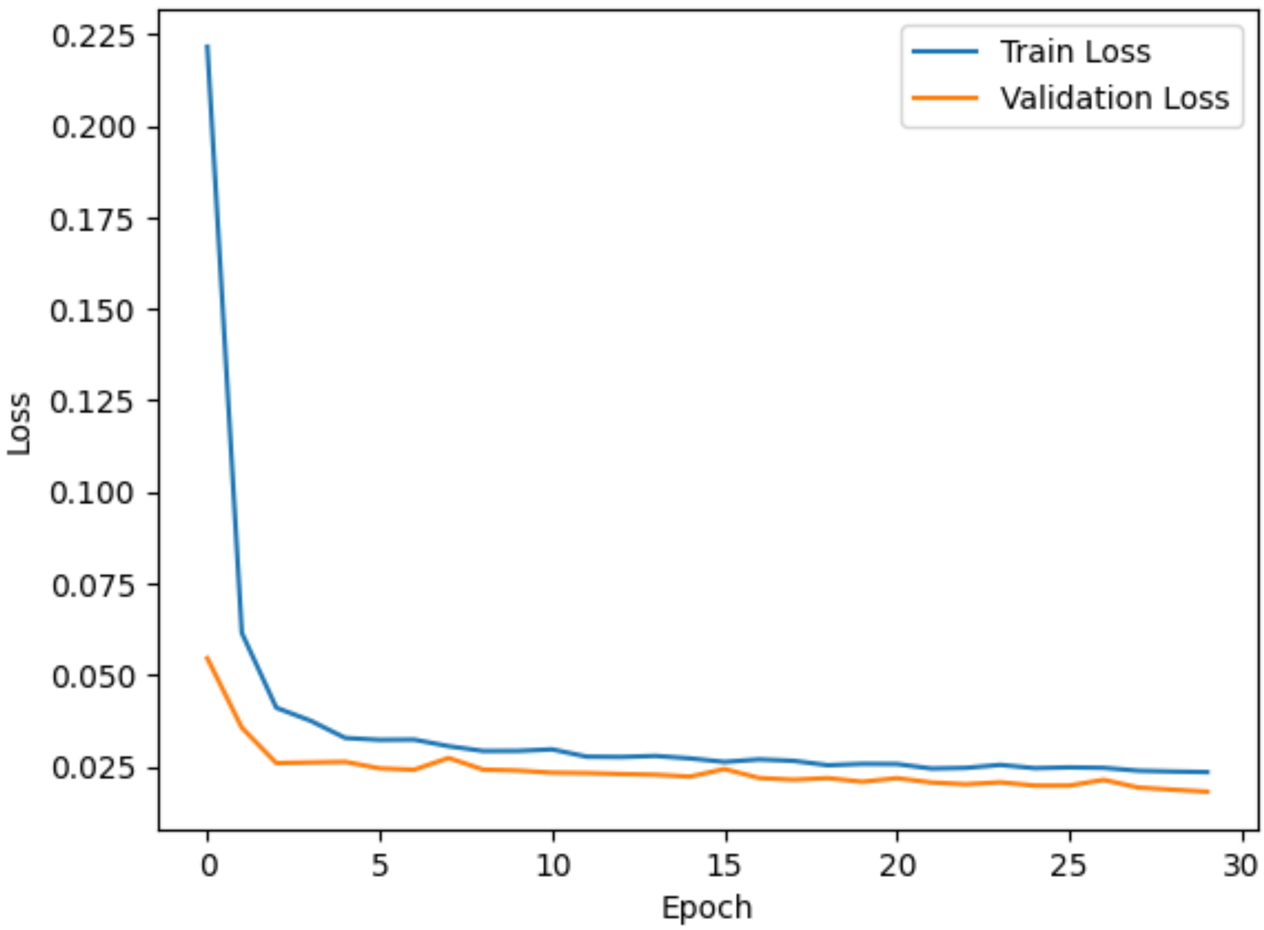



*Figure 11 Training and validation losses over epochs.*

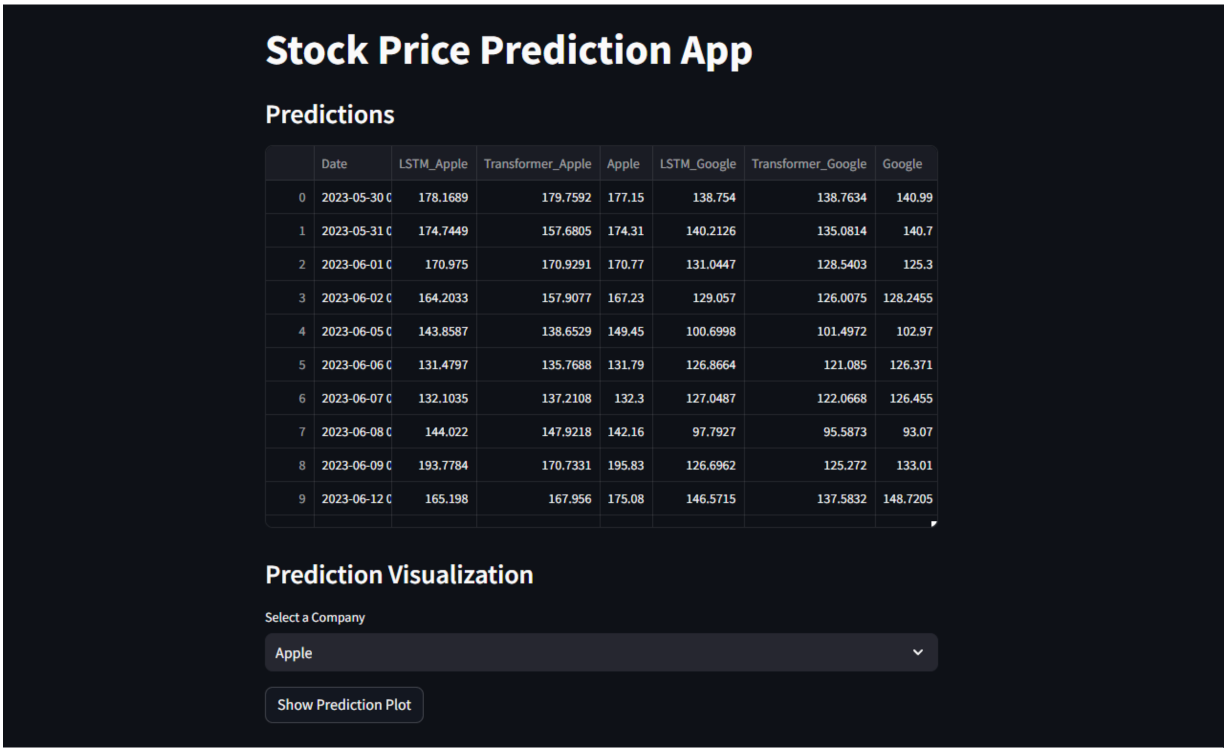
| | 7days | 14 days | 28 days | 112 days | 365 days |
|---|---|---|---|---|---|
| MSE | 0.014 | 0.013 | 0.011 | 0.005 | 0.002 |
| MAE | 0.093 | 0.090 | 0.083 | 0.059 | 0.039 |
| MAPE | %25.90 | %26.14 | %27.71 | %14.97 | %13.24 |
| MAPE-unscaled | %5.69 | %5.58 | %4.99 | %3.61 | %2.5 |

*Figure 12 The effect of sequence length on the best model HS64-NH3.*

| | 1 year (22-23) | 3 years (20-23) | 5 years (18-23) |
|---|---|---|---|
| MSE | 0.008 | 0.005 | 0.003 |
| MAE | 0.077 | 0.059 | 0.042 |
| MAPE | %9.86 | %14.97 | %11.03 |
| MAPE-unscaled | %3.49 | %3.61 | %4.56 |

*Figure 13 Anomalies for MAPE values of the Best Model HS64-NH3 – Sequence Length 112 days.*

**Regarding Streamlit**




| | Date | LSTM_Apple | Transformer_Apple | Apple | LSTM_Google | Transformer_Google | Google |
|---|---|---|---|---|---|---|---|
| 0 | 2023-05-30 0 | 178.1689 | 179.7592 | 177.15 | 138.754 | 138.7634 | 140.99 |
| 1 | 2023-05-31 0 | 174.7449 | 157.6805 | 174.31 | 140.2126 | 135.0814 | 140.7 |
| 2 | 2023-06-01 0 | 170.975 | 170.9291 | 170.77 | 131.0447 | 128.5403 | 125.3 |
| 3 | 2023-06-02 0 | 164.2033 | 157.9077 | 167.23 | 129.057 | 126.0075 | 128.2455 |
| 4 | 2023-06-05 0 | 143.8587 | 138.6529 | 149.45 | 100.6998 | 101.4972 | 102.97 |
| 5 | 2023-06-06 0 | 131.4797 | 135.7688 | 131.79 | 126.8664 | 121.085 | 126.371 |
| 6 | 2023-06-07 0 | 132.1035 | 137.2108 | 132.3 | 127.0487 | 122.0668 | 126.455 |
| 7 | 2023-06-08 0 | 144.022 | 147.9218 | 142.16 | 97.7927 | 95.5873 | 93.07 |
| 8 | 2023-06-09 0 | 193.7784 | 170.7331 | 195.83 | 126.6962 | 125.272 | 133.01 |
| 9 | 2023-06-12 0 | 165.198 | 167.956 | 175.08 | 146.5715 | 137.5832 | 148.7205 |

*Figure 14 Streamlit Stock Price Prediction App*

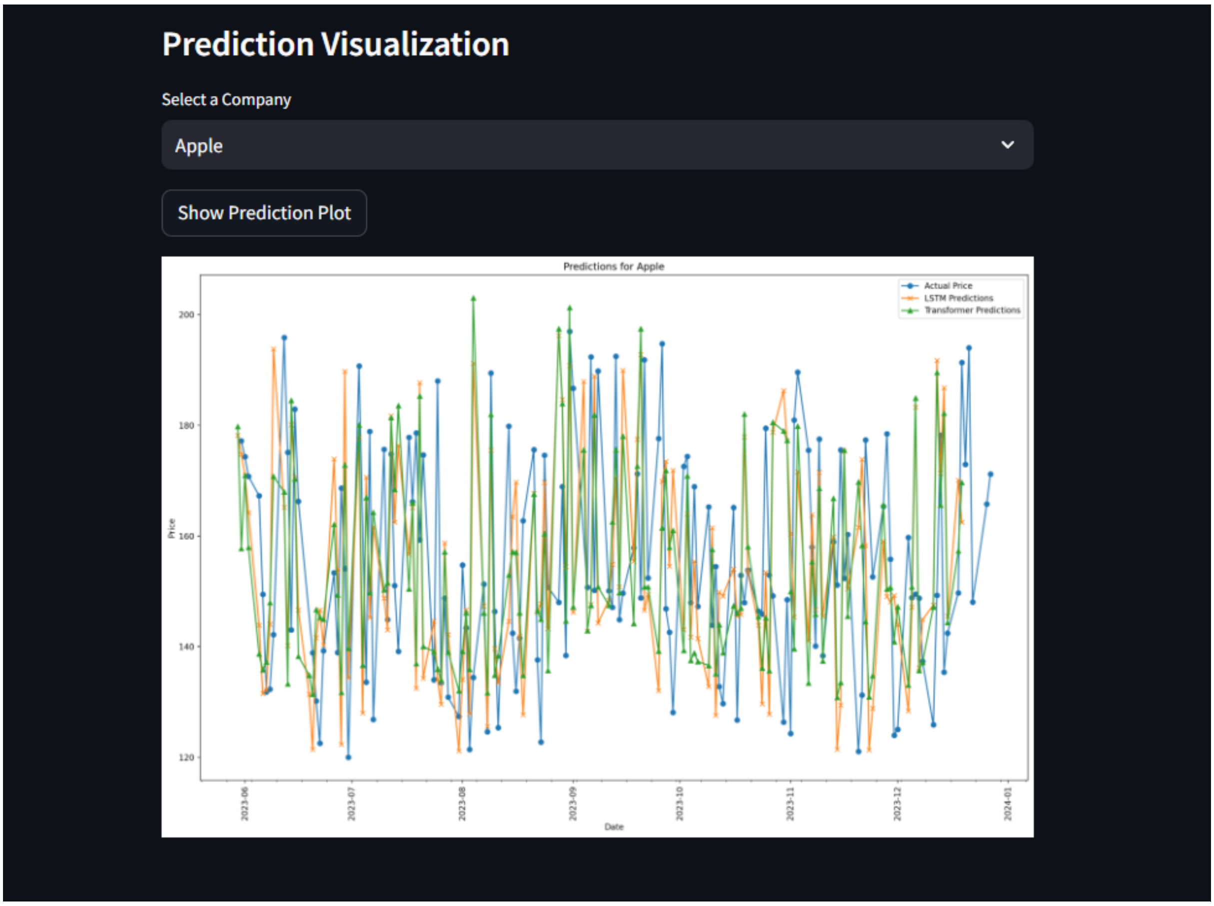


*Figure 15 Streamlit Prediction Visualization*