Deep Learning for Stock Market Predictions

CMPT 420/728 D100

Spring 2025

Amy Cao*
Simon Fraser University
caoamyc@sfu.ca

Johann Tin Tsin Fong*
Simon Fraser University
jtintsin@sfu.ca

Patrick Zhao*
Simon Fraser University
pza28@sfu.ca

* All authors contributed equally to the project

Abstract

This paper explores how different machine learning models— Long Short-Term Memory (LSTM), Temporal Fusion Transformer (TFT), and Temporal Convolutional Network (TCN) perform in predicting stock prices using freely available data. We combined numerical stock data from Alpha Vantage with sentiment analysis from SeekingAlpha, using FinBERT to gauge news sentiment. The dataset includes indicators related to trends, momentum, volatility, and volume, covering the period from 2010 to 2024. We evaluated model performance with Mean Absolute Error (MAE) and R-squared metrics. The TFT model achieved the best results, slightly outperforming LSTM, while TCN lagged behind. We discuss the strengths and weaknesses of each approach and suggest future improvements, including testing with more stocks and developing trading strategies based on the predictions.

1 Introduction

This paper aims to examine how the current state of the art machine learning architectures for time series data fare on predicting stock prices with free data available to the public, and how deep learning techniques can be used to improve the results.

Financial markets are inherently complex, influenced by numerous factors such as:

- **Noise:** Random fluctuations and irrelevant information affecting stock prices.
- **Non-stationarity:** Statistical properties of market data evolving over time.
- **Unpredictable external factors:** News events, economic shifts, and geopolitical tensions impacting market behavior.

To address these challenges, we propose a predictive trading model that integrates numerical stock data with sentiment analysis. Our model consists of two concatenated sub-models, one processing stock price data, and the other analyzing sentiment from financial news.

2 Related Works

The experiment is inspired by a similar experiment by Thomas Fischer and Christopher Krauss[4]. For more information on the model architectures used in this experiment, one can look into the original papers for Long-Term-short-Memory [1], Temporal Fusion Transformer [6], and Temporal Neural Networks [3].

3 Data

The experiment will be using APPL stock data. We collected two types of stock-related data for AAPL constituents using *Alpha Vantage*, an API provider for realtime and historical financial market data [7], and SeekingAlpha [8], another API which provides Market News. Data will be collected from these two API sources and further processed. The resulting data will be split into a 85:15 train:test split.

3.1 Numerical Stock Data

Using the Alpha Vantage API, we pulled the following daily data for each stock: Open, High, Low, Close, and Volume. Data collected covers 2010 January 1st to 2025 January 1st. The data was processed to create more features to better represent patterns in data; we call these features indicators. We used several indicators categorized into four groups: Trend, Momentum, Volatility, and Volume.

3.1.1 Trend

Trend indicators such as 200 day Simple Moving Average (SMA_200), 8 day Exponential Moving Average (EMA_8), Moving Average Convergence Divergence (MACD), and Signal_Line are essential for tracking the overall market direction. The SMA_200 calculates the average closing price over the past 200 days, providing a clear long-term trend highlighting broader market direction. In contrast, the EMA_8 is more responsive to recent price changes, capturing short-term movements. MACD and Signal_Line work together to identify shifts in momentum through crossovers, which often serve as reliable buy or sell signals.

3.1.2 Momentum

Momentum indicators, including Relative Strength Index (RSI) and MACD_Histogram, measure the speed and strength of price movements, helping detect potential trend reversals. The RSI identifies overbought or oversold conditions by comparing recent gains and losses, indicating when a reversal may be imminent. Meanwhile, the MACD_Histogram highlights the strength of a trend by showing the difference between MACD and Signal_Line.

3.1.3 Volatility

Volatility indicators are represented by Bollinger Bands (BB_upper, BB_middle, and BB_lower), which expand and contract based on market volatility. These bands are plotted at standard deviation levels above and below a moving average (BB_middle). When prices touch or breach the upper or lower bands, it may suggest overbought or oversold conditions. Additionally, a sudden widening of the bands typically indicates increased volatility, which could precede significant price movements or breakouts.

3.1.4 Volume

Lastly, Volume serves as a crucial measure of market activity and confirms or contradicts price movements. High volume generally supports strong trends and increases the credibility of breakouts, while low volume suggests weaker or indecisive moves. For instance, if a breakout occurs with high volume, it is more likely to be sustainable, whereas breakouts with low volume are often unreliable. By incorporating these features, our model gains a comprehensive understanding of market behavior, enhancing its ability to make accurate predictions.

3.2 Market News & Sentiment

The intuition behind including article sentiment into training is that stock prices are heavily susceptible to public opinions, which are swayed by article headlines. The portrayal of news and technology in media is one of the few sources of the latest knowledge to non-insiders. Although the articles themselves cannot change the fundamental value of the subject, they can influence how the readers perceive the effects of the subject, which ultimately influences their decision on whether or not to buy or sell a stock.

With the SeekingAlpha API, article data on specified stocks were collected. For each article, we extract the following columns: title, published date, author, summary, and url. This data is passed to a language model, FinBERT [5], to analyze the sentiment regarding a specified stock from each article. The articles are individually assigned a non-discrete sentiment score from 1-100 for the fields "positive", "neutral", and "negative". The values of these three assigned fields shall sum to 100. The dominant sentiment is recorded as well. These 4 fields are used as complementary features to the financial data for training the models. Articles from 2019 December 31st to 2025 January 1st will be used towards Sentimental Data.

The sentiment data is spotted, for there may not be a relevant article to the stock every day. In addition, it needs to have matching number of rows with the numerical stock data in order to be merged and used together as input data. As such, the sentiment data is forward filled to make up for the days where no articles have been pulled from the data source. As the sentiment data begins years after the earliest numerical stock data date, backward-filling shall be employed to extend the number of data rows to match.

4 Methods

The predictive system is built upon three complementary models:

- **LSTM for Financial Time-Series Forecasting:** An LSTM-based model inspired by the work of Fischer and Krauss [4], which aims to predict stock prices based on historical price data. The input is an nxm array, where n represents the number of stocks and m represents associated features.

- **Temporal Fusion Transformer:** The Temporal Fusion Transformer (TFT) architecture is a more modern architecture designed to handle time-series forecasting that have various types of inputs. It combines multiple deep learning ideas such as attention, sequence modeling, etc. into a single architecture, hence the term 'fusion'.

- **Temporal Convolutional Network:** A convolutional network utilizing dilation for temporal data. Our implementation of the Temporal Convolutional Network (TCN) is based on the implementation provided by an empirical study of generic Convolutional and Recurrent Networks for Sequence Modeling by Shaojie Bai, J. Zico Kolter, Vladlen Koltun [3].

For each of these models, they will be trained on the same historical data, and then back-tested on 2024 data. Metrics for the resulting test will be used to compare the performance among the three models. The metrics chosen are Mean Absolute Error (MAE) and R-squared metrics.

5 Model Architecture

The three different architectures implemented as a stock price forecaster are all specialized for temporal data. In this section, we will examine how each architecture meets the use case of retaining short-term patterns in the time-series data to predict subsequent stock prices, briefly summarize the techniques used that are universal to the models, and dive into our specific implementation of each model.

5.1 LSTM

5.1.1 Dropout Layer

Dropout is a regularization technique used to prevent overfitting by randomly deactivating neurons during training. By applying dropout layers after each Bidirectional LSTM layer, the model is encouraged to learn more general patterns rather than memorizing the training data. Tuning the dropout rate helps achieve a balance between learning useful features and avoiding overfitting, enhancing the model's ability to generalize to new data.

5.1.2 Attention Layer

The Attention Layer enhances the model by allowing it to focus more on important parts of the input sequence. It assigns different weights to each time step, effectively prioritizing the most relevant information while minimizing the influence of less important data points. This mechanism improves prediction accuracy and interpretability by highlighting the parts of the sequence that contribute most to the model's decisions.

5.1.3 Bidirectional LSTM

The Bidirectional [1] LSTM processes data from both past and future time steps, giving the model a broader view of trends and patterns. Unlike a standard LSTM, which only processes data sequentially from past to present, a Bidirectional LSTM captures information from both directions. This approach improves pattern recognition, making it especially useful for detecting complex stock price movements.

The combination of Bidirectional LSTM, Dropout, and Attention Layers provides a powerful framework for predicting stock prices. Bidirectional LSTMs capture richer patterns by processing sequences in both directions, while Dropout prevents overfitting by promoting generalization. The Attention Layer further improves accuracy by selectively focusing on relevant information. Together, these techniques enhance the model's ability to generate reliable predictions and trading signals.

5.2 TFT

The Temporal Fusion Transformer (TFT) architecture is a more modern architecture designed to handle time-series forecasting that have various types of inputs. It combines multiple deep learning ideas such as attention, sequence modeling, etc. into a single architecture.

5.2.1 Input Characterization

The TFT architecture categorizes its inputs into three different kinds of features: Static, Known, and Observed.

- Static: Features that don't change over time (ticker, industry, exchange, etc.). In our scenario since we are only predicting the price for one stock this is 1D.
- Known: Features known in advance (Day of the week, month, year, earnings report date, options expiration, etc.).
- Observed: Past inputs that are only known up to the present time-step (open/close price, trading volume, moving averages, etc.).

A separate Variable Selection Network (VSN) is applied for each category. For static variables the VSN output influences the entire pipeline, and for time-varying ones the output produces time-step specific weights. Again, for the purposes of our experiment the static variable essentially has no effect, but if we wanted to predict prices for different stocks, the model would learn different weights for every ticker.

5.2.2 Gated Residual Networks

TFT models allow the model to apply non-linear transformations selectively and use skip connections. This way, the model is able to learn that some layers are unnecessary and reduce the number of operations and used parameters.

5.3 TCN

The TCN model is built with the generic convolution network architecture. To improve performance, attention mechanisms have been included. Batch Normalization and Drop-Out layers were used to prevent overfitting. As neural networks are susceptible to exploding or vanishing gradients, various techniques have been employed to mitigate the possible problem. These include using the activation function LeakyReLU, initializing weights with Kaiming He Initialization [2], and gradient clipping. The model is configured to continue training until it reaches consecutive worsening loss with the technique Early Stopping.

6 Results

The evaluation results for 2024 highlight notable differences in model performance based on MAE and R-squared metrics. Lower MAE values indicate better prediction accuracy, while higher R-squared values (closer to 1) suggest stronger alignment between predictions and actual values. The LSTM and TFT models performed significantly better than the TCN model, with the TFT model slightly outperforming LSTM in both metrics.

Figure 1 presents a comprehensive comparison of actual stock prices against predictions made by the LSTM, TCN, and TFT models over time. Both the LSTM and TFT models closely follow the actual stock prices, while the TCN model exhibits

Table 1: Comparison of Model Performance (2024)

Model	MAE	R-squared
LSTM	4.6359	0.9405
TCN	16.6506	0.4485
TFT	4.4605	0.9499

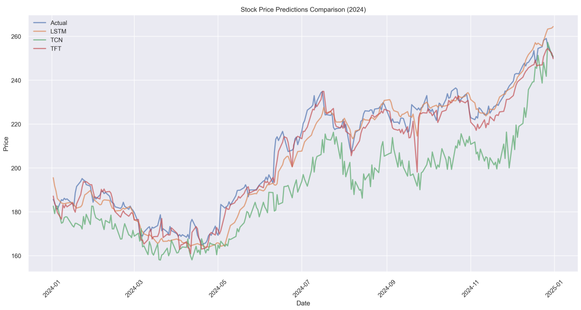

Figure 1: Stock Price Predictions Comparison (2024)

noticeable deviations, particularly during periods of rapid fluctuation.

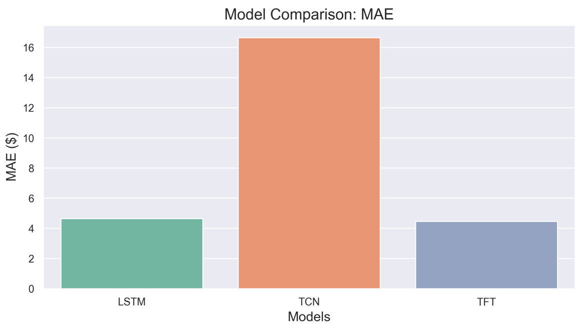

Figure 2: Mean Absolute Error (MAE) Comparison (2024)

Figure 2 displays the Mean Absolute Error for each model, illustrating the clear superiority of the TFT and LSTM models over the TCN model. The TFT model achieves the lowest MAE, reflecting its ability to provide more accurate predictions.

Figure 3 illustrates the R-squared scores for each model. The TFT and LSTM models achieve high R-squared values close to 1.0, indicating strong predictive performance. In contrast, the TCN model exhibits a much lower R-squared score, suggesting poor model fit and limited ability to capture the underlying patterns in the data.

Figure 4 provides scatter plots comparing actual versus predicted values for each model. The

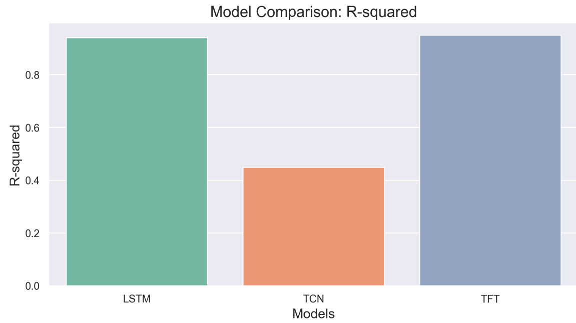

Figure 3: R-squared Comparison of Models (2024)

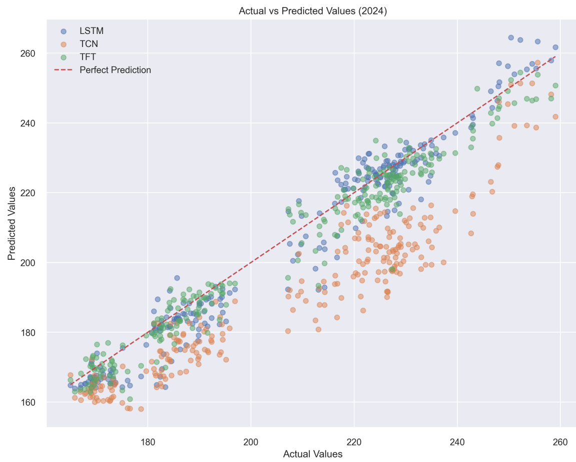

Figure 4: Actual vs Predicted Values (2024)

TFT and LSTM models demonstrate high alignment with the ideal prediction line, whereas the TCN model shows significant scattering, further indicating its poor performance.

7 Conclusion

7.1 Challenges

We had some difficulties determining the objective function of the model (choosing what to max/minimize). Furthermore, each model architecture handled the implementation of the objective in a different way. In collecting data, we were limited by API request rates because we were using the free data sources. Another common obstacle we encountered was compute restrictions; with modest laptops, training the models could take an upward of 30 minutes, which made tuning hyperparameters a very time-consuming task. In the end, we chose to only predict the price for one stock.

7.2 Future Work

In this paper we outlined the performance of models in stock price estimation and indicator generation. While these indicators are useful, applying them in trades is actually a separate problem. Future work could explore applying different trading algorithms to use the predictions we generated and do paper-trading/back-testing. Additionally, we had a 1-dimensional static variable, which was the AAPL stock ticker itself; a direct extension of this work would be to include different markets, industries, and stock tickers for analysis.

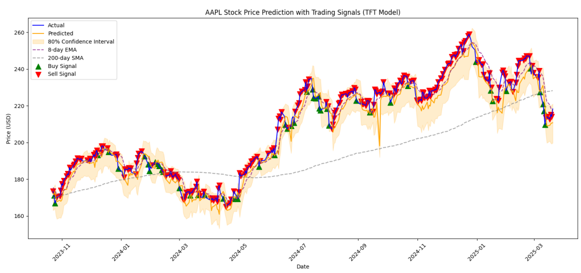

Figure 5: Indicators for TFT model

Finally, the model we used for sentiment analysis, FINBERT, is deprecated by modern GPT models, so another improvement would be replacing our sentiment analysis section.

References

[1] Mike Schuster and Kuldip Paliwal. "Bidirectional recurrent neural networks". In: *Signal Processing, IEEE Transactions on* 45 (Dec. 1997), pp. 2673–2681. DOI: 10.1109/78.650093.

[2] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: 1502.01852 [cs.CV]. URL: https://arxiv.org/abs/1502.01852.

[3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. 2018. arXiv: 1803.01271 [cs.LG]. URL: https://arxiv.org/abs/1803.01271.

[4] Thomas Fischer and Christopher Krauss. "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research* 270.2 (2018), pp. 654–669. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2017.11.054. URL: https://www.sciencedirect.com/science/article/pii/S0377221717310652.

[5] Dogu Araci. "FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models". In: *arXiv preprint* arXiv:1908.10063 (2019). URL: https://arxiv.org/abs/1908.10063.

[6] Bryan Lim et al. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*. 2020. arXiv: 1912.09363 [stat.ML]. URL: https://arxiv.org/abs/1912.09363.

[7] n.a. *Alpha Vantage API Documentation*. URL: https://www.alphavantage.co/documentation/.

[8] n.a. *Stock market analysis tools for investors*. URL: https://seekingalpha.com/.