Stock Movement Modeling with Minimal Transformer

Cole Crescas
Northeastern University
colecrescas@gmail.com

Bronte Sihan Li
Northeastern University
li.siha@northeastern.edu

***Abstract*—In the world of stock market analysis, the application of deep learning methods has surged, incorporating techniques like long short-term memory (LSTM), recurrent and CNN-based networks, and generative adversarial networks (GANs). This project aligns with this trend, focusing on predicting stock price movements during the crucial last ten minutes of trading, utilizing bid and asking price information. Our approach balances model complexity, inference time, and performance, comparing various models including gradient boosting algorithms, LSTMs, and Transformer based methods. Despite observed limitations in these models, our simple Transformer architectures exhibit promising results, showcasing the effectiveness of minimalistic Transformers in capturing temporal dependencies and complex relationships in sequential stock market data with low inference times.**

I. Introduction

In the age of big data, prediction of stock prices with deep learning methods has grown in popularity significantly [2], in particular, techniques including long term short-term memory (LTSM), recurrent and CNN-based deep neural networks, and GANs have been applied to predict both short term and long term trends in the financial markets [12][1]. As stock prediction is a complex system with an incredible amount of intricacies, the space of machine learning approaches continues to evolve, with some studies utilizing feature engineering and leveraging sentiment information and language models [9] while others combine existing methods to develop novel architectures without the need for additional data [3]. In addition, deep reinforcement learning techniques such as deep Q learning have been applied to tackle algorithmic trading problems[10].

Traditional continuous trading systems rely solely on limit order books for matching buys and sells. This can lead to issues with price discovery and volatility, especially during market opens and closes. On the other hand, auction-only systems provide periodic price discovery but lack intra-day liquidity and real-time trade execution. Neither system fully addresses the needs of modern equity trading markets - continuous two-way quotes together with transparent and fair price formation. This leads to the problem of how to best combine elements of the traditional order book and auction matching systems to improve price discovery, volatility control, and liquidity provision in electronic stock trading. Due to the high intensity of closing times of trading days, the challenge lies in developing effective auction matching algorithms and integrating them with the existing order book in a way that improves overall market quality. Accurate predictions can help adjust prices, assess supply and demand dynamics, and identify trading opportunities [5].

For this project, we look at a specific problem posed by this Kaggle competition hosted by Optiver to predict the stock price movements in the last ten minutes of trading given bid and asking price information on a particular day, where the target or ground truth is the 60 second future move in the Weight Average Price (WAP) of the stock, less the 60 second future move of the synthetic index, defined by the following:

- The synthetic index is a custom weighted index of NASDAQ-listed stocks constructed by Optiver for this competition
- The unit of the target is basis points, which is a common unit of measurement in financial markets. A 1 basis point price move is equivalent to a 0.01 percent price move

The target is then defined as follows:

$$Target = (\frac{StockWAP_{t+60}}{StockWAP_t} - \frac{IndexWAP_{t+60}}{IndexWAP_t}) * 10000 \tag{1}$$

In our work, with efficiency and the time sensitivity of stock prediction in mind, we take a minimal approach in our solution design to balance between model complexity, inference time and performance. We demonstrate strong results compared to current public scores for the competition (5.21), with a validation MAE of 5.116.

II. Related Work

Since the introduction of the Transformer architecture [11], there have been numerous efforts in stock movement modeling incorporating it as basic building blocks. [4] proposed a multi-modal framework that leverages financial indicators as well as news to improve prediction accuracy; meanwhile, [13] focuses on finding correlation between stocks and designed a multi-level context approach. Similar to our work, [6] examines performances on stock prediction across various networks suitable for time series and regression, namely, Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) and Temporal Fusion Transformer (TFT) where TFT demonstrated superior results.

In addition, there was a study that claimed linear models often beat Transformers in time series prediction [14] but future work [7] has shown that Transformers have a strong possibility to beat linear models on more complex and larger data sets. However, an interesting finding from this paper was that the "Vanilla Transformer" or simpler model achieved better results then more complex transformers and linear models.

III. Methods

A. Adding Financial Features

The data set from Optiver contains historic data for the daily ten minute closing auction on the NASDAQ stock exchange. The challenge is to predict the future price movements of stocks relative to the price future price movement of a synthetic index composed of NASDAQ-listed stocks. Additional historical data was not used in our case.

Our initial approach was to add common financial features to the existing data set, largely based on the 'bid size', 'ask size', 'imbalance size' and price. In addition to added statistical features, we created windowed features to capture different time ranges.

These features serve multiple purposes, each contributing to a comprehensive understanding of market dynamics. The volume feature, calculated as the sum of bid and ask sizes, is a fundamental metric for assessing market activity and liquidity. The size imbalance feature, representing the ratio of bid size to ask size, offers insights into supply and demand dynamics. The liquidity imbalance feature measures the relative difference between bid and ask sizes, aiding in the identification of liquidity imbalances. The mid price feature, computed as the average of the best bid and ask prices, captures the central tendency of the market.

While this work allowed us to understand the data further but we found that predicting stock price movements using deep learning on the added financial feature data set did not improve performance.

B. Producing Time Series

Next, focusing on the target series from the original data, we designed a "windowed data set" approach to produce a time series for each stock. This method allows us to use overlapping windows as input to the networks, with each window encompassing a sequence of data points serving as sequential features for model learning, while the subsequent data point within the window serves as the target label. By using multiple pivots, we ensure comprehensive coverage by including all possible combinations of stock IDs, date IDs, and time intervals. This organizes stock IDs as columns, while date IDs and time intervals serve as indices, providing a well-organized and structured input for time series analysis and modeling.

C. Deep Learning Approaches

We used both commonly used deep learning based models and classic ML algorithms for this task. The evaluation is done using the Mean Absolute Error (MAE) between the predicted value and the observed target to measure accuracy. Specifically, we achieved good results through LSTMs and Transformer architectures compared to our baseline ML models. First, we get a baseline score using XGB and LightGBM which produced mediocre results even after optimizing for hyper-parameters using a grid search. We briefly explored implementing a basic neural network with two fully connected layers of 12 and 6 nodes respectively but did not achieve improved results.

Stock market data is prone to noise and volatility, especially in the closing minutes. To accommodate this, we used an LSTM structure tailored for regression. The model takes input features with an input size of 200 stocks and processes them through two layers of size 64, one recurrent layer and a dropout probability of 20%. The output from the LSTM layer is then fed into a fully connected layer for the final regression prediction. The model reshapes the input to accommodate sequence processing, and the output is obtained by extracting information from the last time step of the LSTM sequence. The architecture is suitable for capturing temporal dependencies in sequential data and is equipped for regression tasks like predicting stock price movements in closing trading windows.

In addition, we implemented a simple Transformer model designed for sequence-to-sequence tasks and is based on the Transformer architecture. The input features undergo a dimensional reduction using a linear layer (`embedding`). The model consists of two identical Transformer layers (`transformer1` and `transformer2`), each configured with a 8 multi-attention heads and one encoder layer. The first Transformer layer processes the input sequence, and the last output of the sequence is extracted. This output is then passed through a fully connected layer (`fc`) with a dropout rate of 50 percent for additional feature processing. Subsequently, the second Transformer layer processes this refined representation. The final prediction is generated through a linear layer (`decoder`). The dropout layer is incorporated to mitigate over-fitting. The architecture allows the model to capture complex dependencies in sequential data, making it suitable for time varying stock market data.

In addition to implementing architectures that were suited to learning time varying relationships, we experimented with adding Kalman filters to the data. The goal of applying Kalman filters is to recursively estimate the state of a dynamic system by combining noisy measurements with a predicted state model. They aim to provide an optimal estimate that minimizes the mean squared error between the true and estimated states. However, as seen in Table 1, we achieved worse results when applying Kalman filters to the dataset.

1) Why a Transformer?: In the case of stock price movement prediction in high-frequency trading, seconds are vitally important so inference speeds were a consideration in model design. Learning from related work, the use of an Encoder-Decoder transformer architecture is specifically helpful at inference time. This is similar to the "next word prediction machine" that an LLMs are often compared to. A more formal definition of sampling the next token and passing it to the decoder is auto-regressive generation.

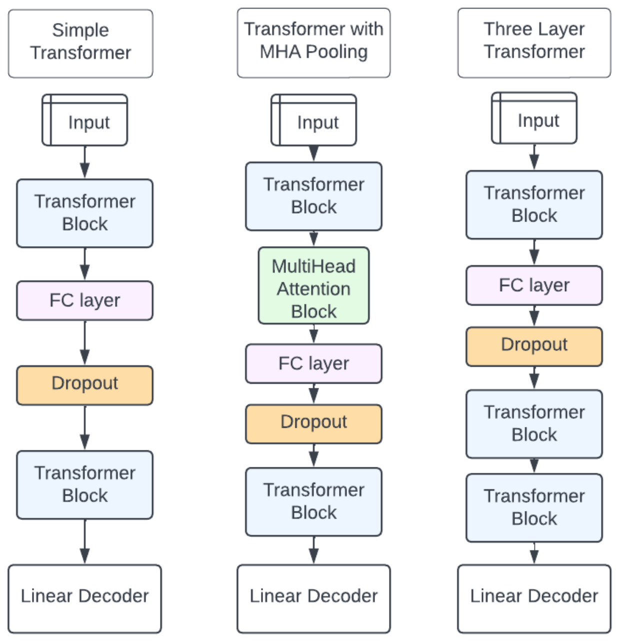

Fig. 1. Diagram of minimal Transformers tested

Secondly, a Transformer proves advantageous for training models on sequences that span numerous time ranges. However, presenting the entire historical context of a time series to the model at once may pose challenges, given the potential limitations imposed by time or memory constraints associated with the attention mechanism. Therefore, a practical approach involves defining a suitable context window and sampling both this window and the subsequent window with a length corresponding to the prediction horizon from the training data when assembling batches for stochastic gradient descent (SGD).

Finally, on our converted time-series data set, the lagging window may result in missing rows. Transformers effectively handle these missing values by incorporating an additional mask to the encoder or decoder, enabling training without the need for in-filling or imputation.

D. Temporal Fusion Transformer

Learning from related work, we proceeded with simpler model architectures to predict stock prices. However, we also wanted to compare against some complex models that are newly released. The Temporal Fusion Transformer (TFT) was released by google [8] in 2023, aimed at learning temporal relationships at different scales. ”TFT uses recurrent layers for local processing and interpretable self-attention layers for long-term dependencies. TFT utilizes specialized components to select relevant features and a series of gating layers to suppress unnecessary components, enabling high performance in a wide range of scenarios [8].”

The TFT model benefits from multiple different components compared to standard LSTMs or Transformers. First, the model has optimized for interpretability by adjusting the methodology of attention heads and expanding on a variable selection block. Instead of having multiple head-specific weights, the weights are shared across all attention heads and then summed vs concatenated. The TFT model can also handle multiple variable types using a separate variable selection block for each type of input: static co-variates, past inputs (time-dependent known and unknown) and known future inputs [8]. These two improvements allow the model to efficiently trace important past time-steps for accurate forecasts, leveraging insights from seasonality and auto-correlation analysis. In addition, the model can detect significant changes in temporal patterns by evaluating the distance between average attention patterns and weights at each point. Below is how the interpretable multi-head attention is formulated:

$$InterpretableMultiHead(Q,K,V) = \frac{1}{h}\sum_{i=1}^{h} head_i W_H$$

$$wherehead_i = Attention(QW_Q^(i), KW_k^(i), VW_v)$$

Secondly, the TFT network substitutes the traditional positional encoding found in Transformers with a Sequence-to-Sequence layer, resembling an LSTM. This choice is particularly suitable for time series data, as it enables the capture of local temporal patterns through recurrent connections. Within the Sequence-to-Sequence block, context vectors play a role in initializing the cell state and hidden state of the initial LSTM unit, along with the static encoder block.

Finally, the model utilizes quantile regression for probabilistic forecasting, offering a range of outcomes instead of a single estimate. This enhances risk assessment and, in turn, can improve the Sharpe ratio, a key metric for evaluating investment performance by accounting for risk. This optimization process compels the model to offer sensible over-estimations for upper quantiles and under-estimations for lower quantiles. The quantiles can be formulated as follows:

$$QL(y,\hat{y},q) = q(y-\hat{y})_+ + (1-q)(\hat{y}-y)_+ \tag{2}$$

IV. Experiments

1) Implementation and Training Details: After our model architectures were finalized, we utilize the Weights and Biases (WandB) library for experiment tracking, logging, and artifact storage. We used a learning rate of 1×10^{-5}, ADAMW as the optimizer with weight decay = 0.05, mean absolute error loss, RELU activation and a warm-up learning rate of 5×10^{-7}. For the loss function and evaluation, we use Mean Absolute Error (MAE), which serves as our key metric for evaluating the accuracy of stock price predictions. We performed hyper-parameter turning on the simple Transformer, results are in Table 2.

Our application of the TFT model utilizes the flexibility to handle multiple targets for prediction. We used our time series data set where each column features a stocks target prices, and

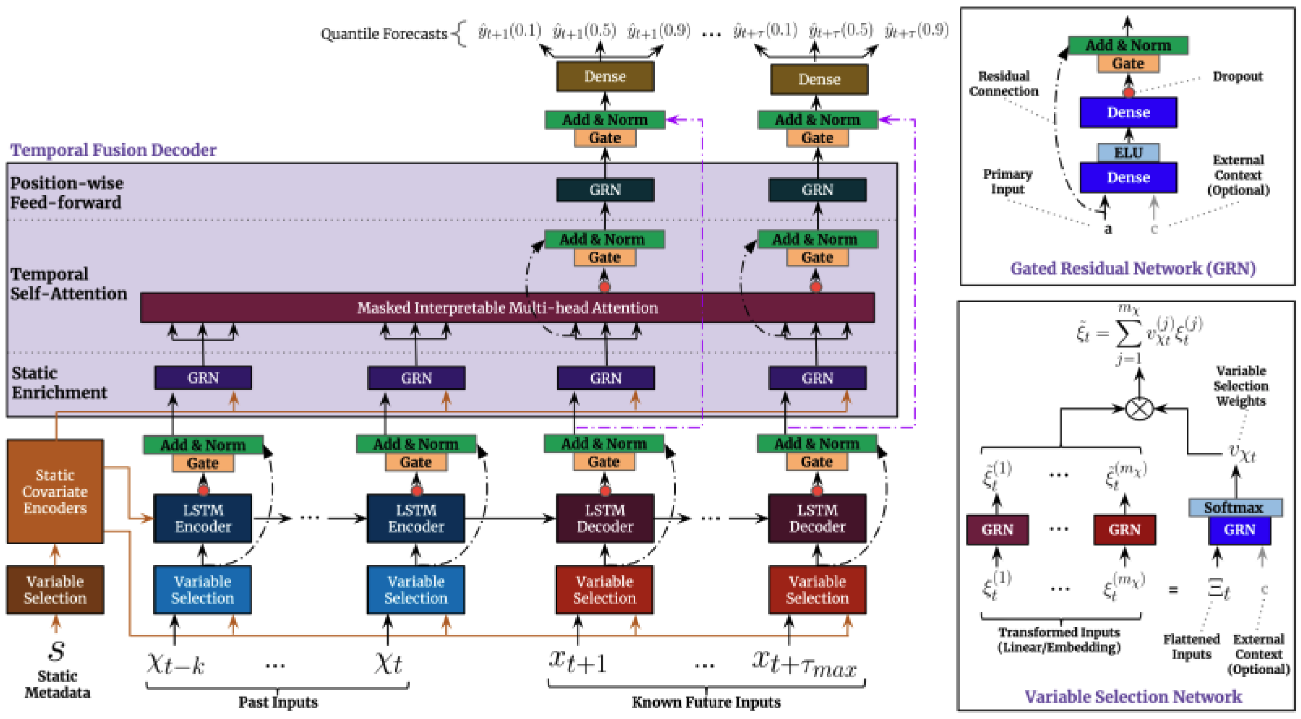

Fig. 2. TFT Architecture [8]

TABLE I
TABLE 1: MODEL COMPARISON

Model	MAE	Link
XGB	6.276	NA
LightGBM	6.202	NA
Neural Net	6.402	NA
LSTM w/Kalman	5.424	WANDB
LSTM	5.519	WANDB
Transformer w/Kalman	5.425	WANDB
Temporal Fusion Transformer	5.361	NA
Transformer	**5.116**	WANDB

TABLE II
TABLE 2: TRANSFORMER PARAMETER TUNING

Iteration	Layers	Pooling?	n_heads	Dropout %	MAE
1	2	No	8	50%	5.415
2	2	Yes	8	50%	5.83
3	3	No	8	50%	5.425
4	2	No	4	50%	5.425
5	2	No	8	25%	**5.192**
6	2	No	8	10%	**5.116**
7	2	No	8	10% and GELU	**5.19**

their lagging window prices. To incorporate, the data into this model, we converted the index into a pseudo time column indicating incremental increases in time done during our data processing steps. After experimenting with using different sets of hyper parameters shown in the table below, we achieved a MAE of 5.36 with the TFT model,

TABLE III
TABLE 2: TRANSFORMER PARAMETER TUNING

Hidden Size	Learning Rate	Epochs	n_heads	Dropout %	MAE
8	0.041	11	2	10%	5.411
64	0.037	12	8	10%	5.361

TABLE IV
MODEL SIZE AND THROUGHPUT COMPARISON

Model	Params	Throughput (samples processed / sec)
LSTM	0.1M	16185
Simple Transformer	6.6M	25417
Transformer with MHA pooling	6.6M	18194
Three layer Transformer	9.8M	22827

V. DISCUSSION AND CONCLUSION

The surge in popularity of deep learning methods for predicting stock prices, utilizing techniques such as LSTM, recurrent and CNN-based networks, and GANs, has set the backdrop for our project [2, 12, 1]. Our focus is on forecasting stock price movements during the last ten minutes of trading, leveraging bid and asking price information. Adopting a minimalist approach, we carefully balanced model complexity, and performance.

In our experimentation, a comparison of models, including XGB and LightGBM highlighted their limitations. However, our proposed LSTM and Transformer architectures yielded promising results, demonstrating the effectiveness of deep learning in capturing temporal dependencies and complex relationships within sequential stock market data.

Our project addresses the challenge of predicting stock price movements during the intense closing minutes of trading. By employing deep learning models, specifically LSTM and Transformer architectures, we achieved competitive performance, surpassing baseline models like XGB and LightGBM. Notably, the Transformer model, especially with a dropout rate of 10%, showcased a validation MAE of 5.116, outperforming other models.

In conclusion, our project demonstrates the potential of simple deep learning in addressing challenges within stock market analysis. The amalgamation of provided financial features with advanced yet simple architectures, such as the Transformer, establishes a robust framework for accurate stock price predictions. Future work may explore optimizing a wide range of hyper-parameters in different model types as well as feature engineering. Additionally, in the realm of high-frequency trading, optimizing inference time for milliseconds improvements could further enhance results.

References

[1] Artha Andriyanto. “Sectoral Stock Prediction Using Convolutional Neural Networks with Candlestick Patterns as input Images”. In: *International Journal of Emerging Trends in Engineering Research* (2020).

[2] Darak Ankita Balaprasad and Sushil Kulkarni. “A Deep Learning System For Short-Term Stock Market Price Trend Prediction”. In: 2022.

[3] Rahul Bhatt et al. “Technique for Forecasting Future Market Movement Using Machine Learning and Deep Learning Algorithms”. In: *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (2023), pp. 471–474.

[4] Divyanshu Daiya and Che Lin. “Stock Movement Prediction and Portfolio Management via Multimodal Learning with Transformer”. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 3305–3309.

[5] Tom Forbes et al. *Optiver - Trading at the Close*. 2023. URL: https://kaggle.com/competitions/optiver-trading-at-the-close.

[6] Xiao-Ping Hu. “Stock Price Prediction Based on Temporal Fusion Transformer”. In: *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)* (2021), pp. 60–66.

[7] Niels Rogge Kashif Rasul Eli Simhayev. *Yes, Transformers are Effective for Time Series Forecasting (+ Autoformer)*. 2023.

[8] Bryan Lima et al. “Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting”. In: *arXiv preprint https://arxiv.org/pdf/1912.09363.pdf* (2023).

[9] Priyank Sonkiya, Vikas Bajpai, and Anukriti Bansal. “Stock price prediction using BERT and GAN”. In: *ArXiv* abs/2107.09055 (2021).

[10] Thibaut Théate and Damien Ernst. “An Application of Deep Reinforcement Learning to Algorithmic Trading”. In: *Expert Syst. Appl.* 173 (2020), p. 114632.

[11] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[12] Feng Wang, Wenxiu Liao, and Shanchao Liang. “Deep Learning Application in Stock Price Prediction”. In: *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)* (2021), pp. 29–32.

[13] Jaemin Yoo et al. “Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021).

[14] Ailing Zeng et al. “Are Transformers Effective for Time Series Forecasting?” In: *International Digital Economy Academy* (2023). {zengailing, leizhang}@idea.edu.cn, {mxchen21, qxu}@cse.cuhk.edu.hk.