Higher Order Transformers: Enhancing Stock Movement Prediction On Multimodal Time-Series Data

Soroush Omranpour
Mila & McGill University
Montreal, Canada
soroush.omranpour@mila.quebec

Guillaume Rabusseau
Mila & University of Montreal
Montreal, Canada
rabussgu@mila.quebec

Reihaneh Rabbany
Mila & McGill University
Montreal, Canada
reihaneh.rabbany@mila.quebec

ABSTRACT

In this paper, we tackle the challenge of predicting stock movements in financial markets by introducing Higher Order Transformers, a novel architecture designed for processing multivariate time-series data. We extend the self-attention mechanism and the transformer architecture to a higher order, effectively capturing complex market dynamics across time and variables. To manage computational complexity, we propose a low-rank approximation of the potentially large attention tensor using tensor decomposition and employ kernel attention, reducing complexity to linear with respect to the data size. Additionally, we present an encoder-decoder model that integrates technical and fundamental analysis, utilizing multimodal signals from historical prices and related tweets. Our experiments on the Stocknet dataset demonstrate the effectiveness of our method, highlighting its potential for enhancing stock movement prediction in financial markets.

KEYWORDS

Stock Movement Prediction, Multivariate Time-Series, Self-Attention, Transformer, Tensor Decomposition, Kernel Attention, Multimodal Signals

1 INTRODUCTION

Predicting stock movements in financial markets is of paramount importance for investors and traders alike, as it enables informed decision-making and enhances profitability. However, this task is inherently challenging due to the stochastic nature of market dynamics, the non-stationarity of stock prices, and the influence of numerous factors beyond historical prices, such as social media sentiment and inter-stock correlations.

Traditional approaches in stock prediction have primarily focused on technical analysis (TA) and fundamental analysis (FA), leveraging historical price data and key financial metrics, respectively [35]. While these methods have provided valuable insights, they often fail to capture the complex interdependencies and the high-dimensional structure of financial data [4]. Recent advancements in machine learning, particularly in natural language processing and graph neural networks, have begun to address these limitations by integrating multimodal data sources, such as news articles and social media sentiment, thereby offering a more nuanced understanding of market dynamics [14, 18].

Despite these advancements, existing models still struggle with the sheer volume and variability of financial data, often resulting in suboptimal predictive performance when dealing with high-dimensional, multivariate time-series data. To address these challenges, we introduce a novel architecture called Higher Order Transformers. This architecture extends the traditional transformer model by incorporating higher-order data structures in the self-attention mechanism, enabling it to capture more complex interrelationships across time and variables.

The contributions of this paper can be summarized as follows:

- We introduce *Higher-Order Transformers*, a novel architecture tailored for processing higher-order data (with more than two dimensions where the last dimension works as the feature dimension).
- We propose a *Multimodal Encoder-Decoder* architecture based on the proposed Higher Order Transformer which integrates both news text and historical price data to provide a holistic understanding of the market dynamics.
- Following a series of experiments and ablation studies, we demonstrate the effectiveness of the proposed method empirically.

2 RELATED WORK

Predicting stock movements has traditionally utilized technical analysis (TA), focusing on historical price data and macroeconomic indicators, with common methods like GARCH and neural networks being prevalent for their ability to identify temporal patterns [4, 34, 35]. However, such methods often fail to account for external factors that can significantly influence market dynamics, thus limiting their overall predictive scope.

Fundamental analysis (FA) attempts to fill this gap by integrating broader market elements such as investor sentiment and news, with the help of advancements in natural language processing (NLP). NLP has enabled more effective sentiment analysis from diverse data sources like news articles and social media, thereby enhancing the FA approaches [1, 10, 14, 16, 32]. Innovative models such as the Hybrid Attention Networks (HAN) and StockNet have emerged, which blend attention mechanisms and variational auto-encoders to analyze text alongside price data, though the assumption of independent stock movements often hampers their effectiveness [15, 32].

Graph Neural Networks (GNNs) have been introduced as a solution to address the interconnected nature of financial markets, by structuring market data into graph formats where nodes represent companies, allowing for enhanced data representation and learning through contextual inter-node relationships [12, 18, 21, 25, 33]. These networks are particularly effective when combined with attention mechanisms and have been expanded upon with financial knowledge graphs to incorporate domain-specific knowledge [17].

Despite progress in graph-based and NLP-enhanced stock prediction methods, fully integrated multimodal approaches that leverage both textual data and inter-stock relationships are still in their infancy. Recent proposals for such multimodal models aim to harness these diverse data sets to improve the predictive accuracy of stock movement models significantly [7, 18–20, 24, 31, 37]. These efforts underscore ongoing challenges and the potential for future advancements in capturing complex market signals and correlations.

3 PROBLEM DEFINITION

In line with previous studies [24, 32], we define the stock movement prediction task as a binary classification problem. For a given stock s, the price movement from day t to $t+1$ is defined as follows:

$$Y_t = \begin{cases} 0 & \text{if } p_t^c < p_{t-1}^c \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where p_t^c represents the adjusted closing price[1] on day t. Here, $Y_t = 0$ indicates that the stock price has decreased, and $Y_t = 1$ indicates that the stock price has increased.

The objective of this task is to predict the price movement Y_{T+1} of a stock s based on its historical price data and related tweets within a time window of T days.

4 PRELIMINARY

Let us define some notations. We use lower-case letters for vectors (*e.g.,* v), upper-case letters for matrices (*e.g.,* M), and calligraphic letters for tensors (*e.g.,*$\mathcal{T}$). We use $\otimes$ to denote the Kronecker product and $\times_i$ for the tensor product along mode i. The notation $[k]$ denotes the set $\{1, \cdots, k\}$ for any integer k.

DEFINITION 1 (TENSOR). *A k-th order tensor* $\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_k}$ *can simply be seen as a multidimensional array.*

DEFINITION 2 (TENSOR MODE AND FIBERS). *The mode-i fibers of* $\mathcal{T}$ *are vectors obtained by fixing all indices except the i-th one:* $\mathcal{T}_{n_1,n_2,\ldots,n_{i-1},:,n_{i+1},\ldots,n_k} \in \mathbb{R}^{N_i}$.

DEFINITION 3 (TENSOR SLICE). *A tensor slice is a two-dimensional section (fragment) of a tensor, obtained by fixing all indexes except for two indices.*

DEFINITION 4 (TENSOR MATRICIZATION). *The i-th mode matricization of a tensor is the matrix having its mode-i fibers as columns and is denoted by* $\mathcal{T}_{(i)}$, *e.g.,* $\mathcal{T}_{(1)} \in \mathbb{R}^{N_1 \times N_2 \cdots N_k}$.

DEFINITION 5 (KRONECKER PRODUCT). *The Kronecker product, denoted by* $\otimes$, *is a binary operation that combines two matrices or tensors to form a larger matrix or tensor. For two matrices* $A \in \mathbb{R}^{m \times n}$ *and* $B \in \mathbb{R}^{p \times q}$, *their Kronecker product* $A \otimes B$ *is defined as:*

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp \times nq}$$

In this paper, we consider multivariate timeseries data in the form of third-order tensors $\mathcal{X} \in \mathbb{R}^{N \times T \times d}$ where N is the number of variables, T is the number of timesteps, and d is the number of hidden/feature dimensions.

5 METHOD

5.1 Tokenization

In this section, we explain the process of tokenizing the input multivariate time-series data. Following prior work [24], we construct a price vector for each stock i at each day t in the form of $x_{i,t} = [p_{i,t}^c, p_{i,t}^h, p_{i,t}^l]$, comprising the stock's adjusted closing price, highest price, and lowest price.

In line with the multivariate time-series forecasting literature [23, 38], we also found it beneficial to add date features, including the day of the month, month of the year, and year. The combination of price and date features forms a six-dimensional vector for each stock on each day.

Inspired by [8], we add stock-specific learnable tokens to the beginning of each time-series and treat them as the common CLS token in transformer encoders. Similar to BERT [9] and ViT [11], we used the hidden state of this special token as the stock representation over the whole time window for the classification task.

5.2 Higher Order Transformer

In this section, we first review the self-attention mechanism in Transformer layers [30]. Then, we extend it to higher orders by tensorizing queries, keys, and values, thereby formulating higher order transformer layers. Given that computing attention on tensors is prohibitively costly, we propose a low-rank approximation using Kronecker decomposition. Additionally, we incorporate the attention kernel trick [6] to significantly reduce the computational complexity.

5.2.1 Standard Transformer Layer

. A Transformer encoder layer comprises two primary components: a self-attention layer $g_{\text{Attn}} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$ and an elementwise feedforward layer $g_{\text{MLP}} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d}$. For a set of n input vectors $X \in \mathbb{R}^{n \times d}$, a Transformer layer computes the following[2]:

$$g_{\text{Attn}}(X)_i = X_i + \sum_{h=1}^{H} \sum_{j=1}^{n} S_{ij}^h X_j W_h^V W_h^O, \tag{2}$$

$$f_{\text{Enc}}(X)_i = g_{\text{Attn}}(X)_i + g_{\text{MLP}}(g_{\text{Attn}}(X))_i, \tag{3}$$

where i denotes the row index of the matrix, H is the number of attention heads, and d_H is the head size. The weight matrices $W_h^V, W_h^K, W_h^Q \in \mathbb{R}^{d \times d_H}$ and $W_h^O \in \mathbb{R}^{d_H \times d}$ are associated with each attention head. The attention coefficients are computed as:

$$S^h = \text{Softmax}\left(\frac{XW_h^Q (XW_h^K)^T}{\sqrt{d}}\right), \tag{4}$$

where the Softmax function is applied row-wise.

5.2.2 Higher Order Transformer Layer

. To generalize the formulation of scaled dot-product attention to higher-order input data $\mathcal{X} \in \mathbb{R}^{N \times T \times d}$, we extend the concept of attention to tensors. We define the transformer encoder layer as a function $f_{\text{Enc}} : \mathbb{R}^{N \times T \times d} \to \mathbb{R}^{N \times T \times d}$, formulated as follows:

[1] https://www.investopedia.com/terms/a/adjusted_closing_price.asp

[2] We omitted normalization after $g_{\text{Attn}}(\cdot)$ and $g_{\text{MLP}}(\cdot)$ for simplicity.

$$g_{\text{Attn}}(\mathcal{X})_{it} = \mathcal{X}_{it} + \sum_{h=1}^{H} \sum_{j,\tau} \mathcal{S}^h_{ijt\tau} W^V_h W^O_h \mathcal{X}_{j\tau}, \tag{5}$$

$$f_{\text{Enc}}(\mathcal{X})_{it} = g_{\text{Attn}}(\mathcal{X})_{it} + g_{\text{MLP}}(g_{\text{Attn}}(\mathcal{X}))_{it}, \tag{6}$$

where $\mathcal{X}_{it} \in \mathbb{R}^d$ denotes the third-mode fiber of $\mathcal{X}$ corresponding to variable i at timestep t. The attention scores are computed as:

$$\mathcal{S}^h_{ijt\tau} = \frac{e^{z^h_{ijt\tau}}}{\sum_{j,\tau} e^{z^h_{ijt\tau}}}, \tag{7}$$

where $\mathcal{S}^h_{ijt\tau}$ denotes the attention score for variable i at time t attending to variable j at time τ on head h and $z^h_{ijt\tau}$ is computed as:

$$z^h_{ijt\tau} = \frac{(W^Q_h \mathcal{X}_{it})^T W^K_h \mathcal{X}_{j\tau}}{\sqrt{d}}. \tag{8}$$

Computing the attention tensor $\mathcal{S} \in \mathbb{R}^{N\times N\times T\times T}$ requires $O(N^2T^2d)$ operations, which can be computationally expensive in most cases.

5.2.3 Low-Rank Approximation With Kronecker Decomposition

. Consider the attention matrix $S \in \mathbb{R}^{NT\times NT}$ as the result of reshaping the tensor $\mathcal{S}$ by flattening the variable and time dimensions. We parameterize the attention matrix S using a rank R Kronecker decomposition with factor matrices $S^{(1)}_i \in \mathbb{R}^{N\times N}$ and $S^{(2)}_i \in \mathbb{R}^{T\times T}$:

$$S \approx \sum_{i=1}^{R} (S^{(1)}_i \otimes S^{(2)}_i), \tag{9}$$

where $S^{(1)}_i$ and $S^{(2)}_i$ represent lower-order attention matrices over the variable and time dimensions, respectively. Having R attention matrices acts similarly to multi-head attention with R heads. In practice, we adopt the multi-head attention mechanism instead of the summation shown in Equation (9), as it uses more parameters and can be more expressive.

THEOREM 5.1. *Given any fourth-order attention tensor $\mathcal{A} \in \mathbb{R}^{N\times N\times T\times T}$, which can be reshaped into a matrix $A \in \mathbb{R}^{NT\times NT}$, there exists a rank R such that matrix A can be expressed as the sum of Kronecker products of matrices $B_i \in \mathbb{R}^{N\times N}$ and $C_i \in \mathbb{R}^{T\times T}$. Formally:*

$$A = \sum_{i=1}^{R} (B_i \otimes C_i), \tag{10}$$

for some $R \leq \min(N^2, T^2)$. As R approaches $\min(N^2, T^2)$, the approximation becomes exact, meaning the Kronecker decomposition is capable of universally representing the original attention tensor $\mathcal{A}$ and matrix A.

PROOF. We begin by reshaping the tensor $\mathcal{A}$ into a matrix $\tilde{A} \in \mathbb{R}^{N^2\times T^2}$. Consider the Singular Value Decomposition (SVD) of $\tilde{A}$:

$$\tilde{A} = U\Sigma V^T, \tag{11}$$

where $U \in \mathbb{R}^{N^2\times N^2}$ and $V \in \mathbb{R}^{T^2\times T^2}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{N^2\times T^2}$ is a diagonal matrix containing the non-negative singular values of $\tilde{A}$. This decomposition allows $\tilde{A}$ to be represented as a sum of outer products of singular vectors scaled by the corresponding singular values, i.e.,

$$\tilde{A} = \sum_{i=1}^{min(N^2,T^2)} \sigma_i u_i v_i^T, \tag{12}$$

where σ_i are the singular values, and $u_i \in \mathbb{R}^{N^2}, v_i \in \mathbb{R}^{T^2}$ are the corresponding left and right singular vectors. For a low-rank approximation, only the first R terms are retained:

$$\tilde{A} \approx \sum_{i=1}^{R} \sigma_i u_i v_i^T. \tag{13}$$

Now, each product $\sigma_i u_i v_i^T$ can be reshaped from vectors u_i and v_i into matrices. Specifically, vector u_i is reshaped into a matrix $\hat{U}_i \in \mathbb{R}^{N\times N}$, and v_i into $\hat{V}_i \in \mathbb{R}^{T\times T}$. The outer product $\sigma_i u_i v_i^T$ corresponds to the Kronecker product of $\hat{U}_i$ and $\hat{V}_i$ when flattened:

$$\sigma_i u_i v_i^T \to \sigma_i(\hat{U}_i \otimes \hat{V}_i). \tag{14}$$

The matrices $B_i = \sqrt{\sigma_i}\hat{U}_i$ and $C_i = \sqrt{\sigma_i}\hat{V}_i$ are then defined, so that $\sigma_i(\hat{U}_i \otimes \hat{V}_i)$ simplifies to $B_i \otimes C_i$. This allows us to express the approximate decomposition of $\tilde{A}$ as:

$$\tilde{A} \approx \sum_{i=1}^{R} B_i \otimes C_i, \tag{15}$$

where $B_i \in \mathbb{R}^{N\times N}$ and $C_i \in \mathbb{R}^{T\times T}$. As R approaches the rank of $\tilde{A}$, this approximation converges to an exact representation. Thus, we verify the theorem's claim that any attention matrix derived from tensor $\mathcal{A}$ can be expressed in terms of Kronecker products, achieving universal representability at full rank. □

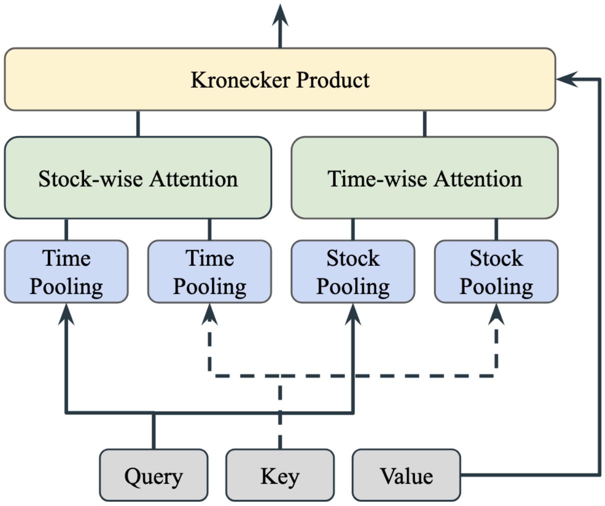

Figure 1: The overview of High Order Attention using Kronecker decomposition.

Now we delve into the computation of the lower-order attention matrices $S^{(1)}$ and $S^{(2)}$. For simplicity, we omit the index i. We project the input tensor $\mathcal{X}$ onto the query, key, and value tensors using mode-3 tensor products: $\mathcal{Q} = \mathcal{X} \times_3 W^Q$, $\mathcal{K} = \mathcal{X} \times_3 W^K$, and

$\mathcal{V} = \mathcal{X} \times_3 W^V$ respectively. Utilizing permutation-invariant pooling functions $f : \mathbb{R}^{N\times T\times d} \to \mathbb{R}^{N\times d}$ and $g : \mathbb{R}^{N\times T\times d} \to \mathbb{R}^{T\times d}$, we define the attention matrices as:

$$S^{(1)} = \text{Softmax}\left(\frac{f(Q)f(\mathcal{K})^T}{\sqrt{d}}\right), \tag{16}$$

$$S^{(2)} = \text{Softmax}\left(\frac{g(Q)g(\mathcal{K})^T}{\sqrt{d}}\right). \tag{17}$$

Common examples of permutation-invariant pooling functions include sum, mean, and product.

The attention matrices are then used to modulate the value tensor $\mathcal{V}$. To maintain computational efficiency, we apply the attention matrices sequentially as follows:

$$g_{\text{Attn}}(Q, \mathcal{K}, \mathcal{V}) = \mathcal{V} \times_1 S^{(1)} \times_2 S^{(2)} \tag{18}$$

Computing and applying these attention matrices incurs a computational cost of $O(d(N^2T + T^2N))$, where the quadratic terms reflect the inherent computational demand of the scaled dot-product attention mechanism, which can become substantial. To mitigate these costs, we employ kernelized linear attention, which achieves linear computational complexity of $O(d^2NT)$.

5.2.4 Linear Attention With Kernel Trick

. Following the work by [6], we approximate the attention matrix $S^{(1)}$ in equation (16) using a suitable kernel function $\phi : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$ as:

$$S^{(1)} \approx Z_1^{-1}\phi(f(Q))\phi(f(\mathcal{K}))^T, \tag{19}$$

where $Z_1 \in \mathbb{R}^{N\times N}$ is a diagonal matrix serving as a normalizing factor. The diagonal elements of Z_1 are defined by:

$$(Z_1)_{ii} = \phi(f(Q))_i \sum_{j\in[N]} \phi(f(\mathcal{K}))_j^T, \tag{20}$$

Substituting into equation (18), we compute:

$$(\mathcal{V} \times_1 S^{(1)})_{:t:} = Z_1^{-1}\phi(g_1(Q))\phi(g_1(\mathcal{K}))^T\mathcal{V}_{:t:}, \tag{21}$$

where $\mathcal{V}_{:t:}$ denotes the tensor slice corresponding to the t-th timestep. The computational sequence initiates by calculating $\phi(f(\mathcal{K}))^T\mathcal{V}_{:t:}$ followed by the subsequent operations, yielding a computational complexity of $O(d^2NT)$. The choice of kernel function ϕ is flexible, and we utilize the same kernel function as in [6], which has been validated both theoretically and empirically. Similarly, the same computational approach is applied to compute $S^{(2)}$ using the same kernel function ϕ.

5.3 Model Architecture

Our model's architecture consists of a multilayer transformer network. Input tensors are transformed through a linear projection layer to align the features with the hidden dimensions required by the model and its attention modules. We adopt pre-normalization techniques, specifically RMSNorm [36], in every layer following the approach suggested by Touvron et al. [28]. Rotary Positional Embedding [27] is applied to the query and key matrices exclusively for computing the temporal attention $S^{(2)}$; stock-wise attention does not involve positional embeddings as the ordering in this dimension is not meaningful.

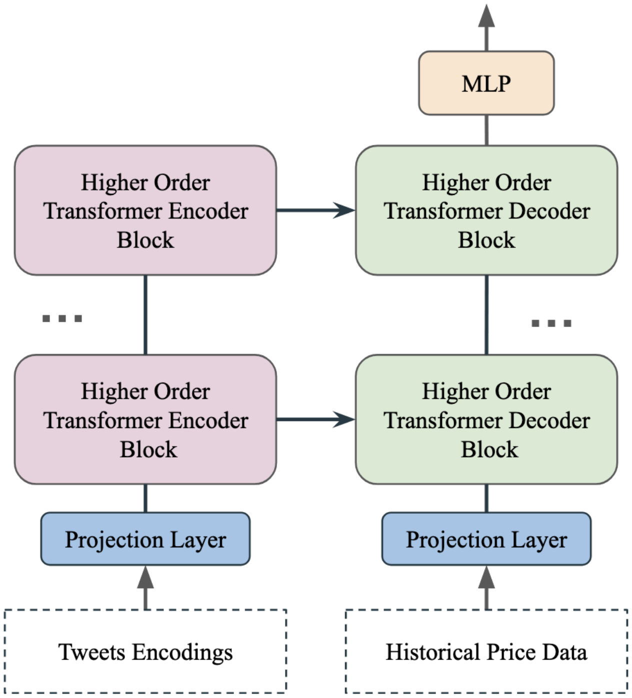
MLP
Higher Order Transformer Encoder Block
Higher Order Transformer Decoder Block
...
Higher Order Transformer Encoder Block
Higher Order Transformer Decoder Block
...
Projection Layer
Projection Layer
Tweets Encodings
Historical Price Data

Figure 2: Multimodal transformer architecture. As depicted in the figure, tweet encodings are fed to the transformer encoder, and the historical price data are given to the transformer decoder.

Inspired by [13, 29], our proposed multimodal model follows an encoder-decoder architecture where the modality of the data differs across encoder and decoder. More specifically, text encodings are processed by the transformer encoder, and price timeseries data by the transformer decoder as presented in figure 2. Cross-attention layers in the network facilitate information blending between these two modalities.

6 EXPERIMENTS

6.1 Dataset

We demonstrate the capability of Higher Order Transformers on stock market movement classification using a multimodal dataset called Stocknet [32] which comprises historical data from 88 stocks extracted from Yahoo Finance[3] and related news crawled from Twitter over two years. We adopt the data processing methodology as outlined in [24, 32], shifting a 5-day lag window along trading days to generate samples. Samples are labeled based on the movement percentage of the closing price, with movements ≥ 0.55% labeled as positive and ≤ 0.5% as negative. Samples lacking either prices or tweets on trading days are discarded to ensure data consistency.

Text encodings are generated using the FinBert model [3] by processing all tweets for each stock on each day through the transformer

[3] https://finance.yahoo.com/

encoder, then averaging over tokens and tweets to produce a vector representation for each stock per day.

The dataset is temporally divided in a 70:10:20 ratio for train, validation, and test splits, spanning from 01/01/2014 to 31/07/2015 for training, 01/08/2015 to 30/09/2015 for validation, and 01/10/2015 to 01/01/2016 for testing.

6.2 Model Configuration

We configured the models using the Adam optimizer with an initial learning rate of 0.0001. Tweet embeddings from FinBert are dimensioned at 768. Training extends for up to 1000 epochs with an early stopping criterion based on validation F1 performance, stopping further training when no improvement is observed. Hyperparameter tuning is conducted through a grid search with the following parameters:

- Hidden dimension: [32, 64, 128]
- Number of attention heads: [1, 4, 8, 16]
- Number of transformer blocks: [2, 4, 6]
- Dropout probability: [0, 0.2, 0.4]

Lastly, we report the performance of the best model based on validation F1 on the test set.

6.3 Metrics

In line with prior research in stock prediction [24, 32], we evaluate classification performance using three metrics: accuracy, F1 score, and Matthew's correlation coefficient (MCC) [4]. We particularly include MCC as it provides a more balanced performance measure that accounts for both positive and negative classes and adjusts for any class imbalance by incorporating True Negatives. This makes it especially valuable in contexts where the data may be skewed. MCC is defined as:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (22)$$

This formula captures the correlation between observed and predicted binary classifications and is regarded as a reliable statistical rate that can deliver a high score only if the predictor successfully predicts the majority of positive and negative instances correctly.

6.4 Baselines

We compare our results with the following baselines:

- **RAND**: A simple predictor to make random guesses about the rise and fall.
- **ARIMA**: AutoRegressive Integrated Moving Average, an advanced technical analysis method using only price signals. [5]
- **HAN**: Hybrid Attention Networks, is a model developed to predict stock trends by analyzing sequences of recent news related to the market. It employs attention mechanisms and self-paced learning to handle the chaotic nature of online content effectively.[15]
- **StockNet**: A Variational Auto-Encoder that jointly exploits text and price signals to predict stock movement. [32]
- **MAN-SF**: Multipronged Attention Network uses tweets, price history, and inter-stock connections to train a GAT network that captures relevant signals across modalities. [24]
- **HATS**: A hierarchical graph attention network for stock movement prediction. [18]
- **Yangjia et al.**: An RNN-based model which uses a co-attention mechanism to capture the sufficient context information between text and prices across every day. [33]
- **ChatGPT (Zero-Shot)**: Using proper prompting on ChatGPT in a zero-shot setting. [31]
- **CapTE**: A Transformer Encoder that extracts the deep semantic features of the social media and then captures the structural relationship of the texts through a capsule network. [20]
- **TEANet**: A Transformer encoder with multiple attention mechanisms that extracts features and fuses text data from social media with stock prices. [37]
- **SLOT**: A self-supervised learning method that embeds stocks and tweets in a shared semantic space to predict stock price movements, overcoming tweet sparsity and noise issues. [26]
- **NL-LSTM**: A novel model that uses neutrosophic logic for more accurate sentiment analysis of social media data, combined with LSTM for forecasting based on sentiment and historical stock data. [2]

Table 1: Classification performance of our method and competitors. The best is in bold, and the second best is underlined. Our model shows the best performance in all evaluation metrics after NL-LSTM.

Method	ACC	F1	MCC
RAND	50.9	50.2	-0.002
ARIMA [2004]	51.4	51.3	-0.021
ChatGPT$_{zs}$ [2023]	52.1	51.7	0.038
HATS [2019]	56.02	56.22	0.117
HAN [2018]	57.6	57.2	0.052
StockNet [2018]	58.2	57.5	0.081
SLOT [2022]	58.7	-	0.2065
MAN-SF [2020a]	60.8	60.5	0.195
Yangjia et al. [2022]	62.6	61.1	0.228
CapTE [2019]	64.22	-	0.348
TEANet [2022]	65.16	-	0.364
NL-LSTM [2024]	**78.5**	-	**0.587**
Ours	72.94	72.23	0.516

[4] Using Torchmetrics library: https://github.com/Lightning-AI/torchmetrics

Table 2: Ablation study on the data modality and attention study shows the effectiveness of using multimodal over unimodal data, and using kernelized over standard attention.

Modality	Attention Type	ACC	F1	MCC
Price	Standard	52.3	51.8	0.049
	Kernelized	53.6	53.3	0.063
Tweet	Standard	69.5	68.0	0.320
	Kernelized	71.0	70.1	0.402
Multimodal	Standard	70.8	70.5	0.391
	Kernelized	**72.9**	**72.2**	**0.516**

6.5 Results

In this section, we analyze the benchmark performance of our model against various baseline models on the StockNet dataset. The results, as summarized in Table 1, demonstrate the superior performance of our model over all of the existing baselines except NL-LSTM, which significantly outperforms the rest of the methods in terms of accuracy, F1, and MCC. NL-LSTM reported the highest accuracy performance on binary stock movement prediction by incorporating fuzzy logic in sentiment analysis [2].

6.6 Ablation Study

We further investigate the impact of different aspects of our model through an ablation study, focusing on the types of attention mechanisms used, the data modalities, and the attention method. The results are provided in Tables 2, and 3 respectively.

The influence of data modality on performance is depicted in Table 2. The multimodal approach that integrates both price data and Twitter news significantly outperforms single-modality approaches, underscoring the benefit of leveraging diverse data sources. Moreover, the text-based models performs better than the timeseries-based models with a significant gap, showing the rich context present in the news data crawled from Twitter for the stock movement prediction task.

Table 2 also explores the effect of using linear versus standard attention mechanisms over different modalities. The results highlight the advantages of linear attention in terms of efficiency and effectiveness, particularly in multimodal settings.

Table 3: Ablation study on the attention dimension shows the effectiveness of attention across both dimensions.

Attention Dimension	ACC	F1	MCC
None	53.5	52.4	0.069
Stock-wise	71.3	69.2	0.495
Time-wise	72.1	71.3	0.507
Both (Higher Order)	**72.9**	**72.2**	**0.516**

Table 3 vividly demonstrates that while applying attention in any single dimension (either stock-wise or time-wise) improves performance metrics compared to using no attention, the most significant enhancements are observed when attention is simultaneously applied across both dimensions.

7 CONCLUSION

In this paper, we presented the Higher Order Transformers, a novel architecture tailored to predict stock movements by processing multimodal stock data. By expanding the self-attention mechanism and transformer architecture to incorporate higher-order interactions, our model adeptly captures the intricate dynamics of financial markets over both stock and time. To address computational constraints, we implemented low-rank approximations through tensor decomposition and integrated kernel attention to achieve linear computational complexity. Extensive testing on the Stocknet dataset demonstrated that our approach significantly surpasses most of the existing models in predicting stock movements. An ablation study further validated the effectiveness of specific architectural components, highlighting their contributory value to the model's performance. Looking ahead, we plan to train our model on other multimodal stock datasets such as ASTOCK [39] and Dhaka Stock Exchange [22], and perform profitability analysis on real-world stock data to further test the practical application and financial viability of our proposed method.

ACKNOWLEDGMENTS

REFERENCES

[1] Jeffrey S Abarbanell and Brian J Bushee. 1997. Fundamental analysis, future earnings, and stock prices. *Journal of accounting research* 35, 1 (1997), 1–24.

[2] Bassant A. Abdelfattah, Saad M. Darwish, and Saleh M. Elkaffas. 2024. Enhancing the Prediction of Stock Market Movement Using Neutrosophic-Logic-Based Sentiment Analysis. *Journal of Theoretical and Applied Electronic Commerce Research* 19, 1 (2024), 116–134. https://doi.org/10.3390/jtaer19010007

[3] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063 [cs.CL]

[4] Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics* 31, 3 (1986), 307–327.

[5] Robert Goodell Brown. 2004. *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation.

[6] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2022. Rethinking Attention with Performers. arXiv:2009.14794 [cs.LG]

[7] Divyanshu Daiya and Che Lin. 2021. 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)*. IEEE, 3305–3309.

[8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision Transformers Need Registers. arXiv:2309.16588 [cs.CV]

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[10] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929 [cs.CV]

[12] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–30.

[13] Ronghang Hu and Amanpreet Singh. 2021. UniT: Multimodal Multitask Learning with a Unified Transformer. arXiv:2102.10772 [cs.CV]

[14] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 261–269.

[15] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2019. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. arXiv:1712.02136 [cs.SI]

[16] Joshi Kalyani, Prof Bharathi, Prof Jyothi, et al. 2016. Stock trend prediction using news sentiment analysis. *arXiv preprint arXiv:1607.01958* (2016).

[17] Natthawut Kertkeidkachorn, Rungsiman Nararatwong, Ziwei Xu, and Ryutaro Ichise. 2023. FinKG: A Core Financial Knowledge Graph for Financial Analysis. In *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*. IEEE, 90–93.

[18] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999* (2019).

[19] Qing Li, Jinghua Tan, Jun Wang, and Hsinchun Chen. 2020. A multimodal event-driven LSTM model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering* 33, 10 (2020), 3323–3337.

[20] Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao, and Liang Yang. 2019. Transformer-Based Capsule Network For Stock Movement Prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen (Eds.). Macao, China, 66–73. https://aclanthology.org/W19-5511

[21] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. 2019. Exploring graph neural networks for stock market predictions with rolling window analysis. *arXiv preprint arXiv:1909.10660* (2019).

[22] Tashreef Muhammad, Anika Bintee Aftab, Muhammad Ibrahim, Md. Mainul Ahsan, Maishameem Meherin Muhu, Shahidul Islam Khan, and Mohammad Shafiul Alam. 2023. Transformer-Based Deep Learning Model for Stock Price Prediction: A Case Study on Bangladesh Stock Market. *International Journal of Computational Intelligence and Applications* 22, 03 (April 2023). https://doi.org/10.1142/s146902682350013x

[23] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. 2024. Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. arXiv:2310.08278 [cs.LG]

[24] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8415–8426.

[25] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2020. Spatiotemporal hypergraph convolution network for stock movement forecasting. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 482–491.

[26] Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. 2022. Accurate Stock Movement Prediction with Self-supervised Learning from Sparse Noisy Tweets. In *2022 IEEE International Conference on Big Data (Big Data)*. 1691–1700. https://doi.org/10.1109/BigData55660.2022.10020720

[27] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864 [cs.CL]

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[29] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal Transformer for Unaligned Multimodal Language Sequences. arXiv:1906.00295 [cs.CL]

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[31] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. 2023. The Wall Street Neophyte: A Zero-Shot Analysis of ChatGPT Over MultiModal Stock Movement Prediction Challenges. arXiv:2304.05351 [cs.CL]

[32] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1970–1979.

[33] Zheng Yangjia, Li Xia, Ma Junteng, and Chen Yuan. 2022. Fundamental Analysis based Neural Network for Stock Movement Prediction. In *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, Maosong Sun, Yang Liu, Wanxiang Che, Yang Feng, Xipeng Qiu, Gaoqi Rao, and Yubo Chen (Eds.). Chinese Information Processing Society of China, Nanchang, China, 973–984. https://aclanthology.org/2022.ccl-1.86

[34] Deniz Can Yıldırım, Ismail Hakkı Toroslu, and Ugo Fiore. 2021. Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators. *Financial Innovation* 7 (2021), 1–36.

[35] Libo Yin and Qingyuan Yang. 2016. Predicting the oil prices: do technical indicators help? *Energy Economics* 56 (2016), 338–350.

[36] Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. arXiv:1910.07467 [cs.LG]

[37] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, and Peide Liu. 2022. Transformer-based attention network for stock movement prediction. *Expert Systems with Applications* 202 (2022), 117239. https://doi.org/10.1016/j.eswa.2022.117239

[38] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. arXiv:2012.07436 [cs.LG]

[39] Jinan Zou, Haiyao Cao, Lingqiao Liu, Yuhao Lin, Ehsan Abbasnejad, and Javen Qinfeng Shi. 2022. Astock: A New Dataset and Automated Stock Trading based on Stock-specific News Analyzing Model. arXiv:2206.06606 [cs.CL]