
FinDe++ : A Novel Decoder-Only Model for Financial Forecasting

Nayasha Gandotra* Gopalakrishnan Thirunellai Venkitachalam[†] Abhi Vakil[‡]

Abstract

Stock price prediction remains challenging due to the inherent volatility of markets driven by unexpected political events, inconsistent news coverage, and limited technical understanding among general investors. Traditional methodologies like moving averages and technical indicators (e.g., RSI, Bollinger Bands) have shown potential but often fall short in accuracy and accessibility for non-specialists. To overcome these limitations, deep learning techniques, particularly Transformer-based architectures, have emerged as promising tools capable of extracting complex, hidden patterns from seemingly unrelated market data.

‘This study explores the efficacy of decoder-only Transformer models—akin to architectures used successfully in generative tasks such as language modeling—for the task of stock price prediction. Specifically, we introduce FinDe++, a novel decoder-only Transformer model for financial prediction. Our model combines mean squared error (MSE) loss with a physics-based loss component inspired by financial market theory, enhancing predictive performance through domain-specific regularization. We also integrate sentiment analysis features obtained by processing textual data using FinBERT, and analyse the efficacy of adding this non-market contextual data on predictive capabilities.

Through comprehensive experiments, we demonstrate that FinDe++ effectively captures underlying market patterns. We analyse sources of error in FinDe++’s predictions, and note scopes of improvement over baselines. This approach presents a robust, accessible solution for financial prediction tasks suitable for both expert practitioners and lay investors.

1 Introduction

Market behavior is influenced by a multitude of unpredictable factors, including political upheavals, macroeconomic shocks, and rapidly evolving news cycles, which collectively introduce significant volatility and undermine the reliability of traditional forecasting approaches. As a result, even seasoned investors face substantial uncertainty when attempting to anticipate future price movements.

While technical indicators such as the Relative Strength Index (RSI) and Bollinger Bands have been widely adopted by institutional investors for decades, their effectiveness is constrained by three key limitations: (1) lagging signals that react tardily to trend reversals, (2) susceptibility to false positives during sustained bullish/bearish regimes, and (3) oversimplified assumptions about linear price relationships. These shortcomings are exacerbated by the increasing complexity of modern markets,

*College of Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. Email: nayeshag@andrew.cmu.edu

[†]College of Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. Email: gopalakt@andrew.cmu.edu

[‡]Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, PA 15213. Email: avakil@andrew.cmu.edu

where traditional chart patterns often fail to account for cross-asset correlations and sentiment-driven microtrends.

These limitations have catalyzed a shift toward data-driven, machine learning-based methodologies, which are better equipped to model the nonlinear and multifaceted dynamics of financial time series. Deep learning models, and in particular Transformer-based architectures, have demonstrated strong potential in extracting complex dependencies and uncovering latent patterns from both structured market data and unstructured textual sources. For example, recent research shows that Transformer models can outperform classical statistical approaches such as ARIMA, especially in volatile and information-rich markets (1). Moreover, integrating sentiment analysis from news and social media with quantitative indicators has been shown to further improve predictive accuracy, offering a more holistic view of market drivers (2)(3).

Existing DL based approaches predominantly utilize encoder-decoder Transformers or recurrent architectures like LSTMs, which struggle with temporal coherence and computational inefficiency in volatile markets (4). Decoder-only architectures address these issues through causal attention masks that prevent look-ahead bias while enabling efficient autoregressive prediction-a capability underexplored in non-pretrained financial models

Hence our major contributions with this project are the following:

- Creating a lightweight decoder-only architecture with custom feature embeddings for time series (stock) data analysis and generation (prediction)
- Improving trend and randomness prediction by using financial research based Physics Loss.
- Providing world context for prediction using sentiment data analysed using FinBERT.

2 Literature Review

In this section, we'd like to highlight previous and central works in the field, while pointing out gaps in literature that we are addressing. We begin by analysing traditional and machine / deep learning based approaches for financial forecasting, paying special attention to LSTM and transformer based models. We then shift focus to talk about Geometric Brownian Motion, which is the financial model that informs our Physics Loss. Finally, we shed light on FinBERT and it's use for sentiment analysis.

2.1 Traditional and RNN/LSTM Based Approaches

In the 1990s, statistical methods such as ARIMA and state space models were the dominant approaches in financial forecasting, reflecting the era's emphasis on linear time series analysis and recursive algorithms for prediction (5). However, their reliance on linear assumptions often limited their effectiveness in capturing the complex, nonlinear behavior of real-world markets (6). Despite this, ARIMA is still widely used as a benchmark for financial forecasting, and hence serves as one of our baselines as well.

In contrast, more recent Deep Learning based methods- including CNNs (7), LSTMs (8), CNN-BiLSTMs (9)- have improved predictive performance by integrating spatial and temporal feature extraction. However, they face the limitation of limited interpretability, computational complexity, and difficulties capturing extremely long-range dependencies inherent in financial time series data.

2.2 Transformer Based Approaches

The application of Transformers to financial forecasting was pioneered by Chen et al. (2022) (10), who first adapted encoder-decoder architectures for volatility prediction using multi-head attention on macroeconomic indicators. Subsequent work predominantly employed encoder-decoder frameworks, such as the Time2Vec-Encoder (11) model which integrates temporal encoding with sectoral correlation features. Parallel advancements introduced encoder-only and decoder-only variants: encoder-only models excel at bidirectional context modeling for regime classification, while decoder-only architectures leverage causal attention to prevent look-ahead bias-a critical advantage for autoregressive price prediction.

Pretrained decoder-only Transformers like Lag-Llama (12) enable efficient zero-shot generalization across market regimes. Unlike encoder-decoder models that require paired input-output sequences,

decoder-only architectures natively support streaming prediction tasks (13). However, these models are computationally heavy and require extensive fine tuning for stock specific tasks.

This presents an opportunity for lightweight decoder-only models which are trained specifically on stock data without look-ahead bias, which we have explored in this project.

2.3 Physics Informed Neural Networks

Physics Informed Neural Networks (PINNs) integrate domain-specific physical laws directly into neural network training through residual loss terms, and have proven valuable in fields such as fluid dynamics (14), solid mechanics (15), and financial modeling (16). For financial modeling, PINNs were renamed to FINNs, which integrate hedging loss and stochastic volatility to yield a lower MAE on common stocks. This encouraged us to consider using physics loss in our implementation as well.

2.4 Sentiment Analysis

Recent works (17),(18) have tried to leverage world information for financial forecasting using FinBERT (19), a BERT (20) variant pretrained on financial corpora. These works note a significant improvement in sudden price prediction from transitional and indicator only features, but still struggle with price variance and overfitting on smaller financial datasets. Despite this, FinBERT remains an integral component in adding world knowledge to the model, which led us to consider using it as well. Details on FinBERT and our use of it are mentioned in section (3.1.1)

3 Methodology

Our methodology is centered on systematically gathering and processing structured and unstructured financial data to enable robust time-series prediction. We begin by collecting historical stock price data using the yfinance API, which includes the open, high, low, close, and volume (OHLCV) values for each trading day. We further enrich this dataset by computing widely used technical indicators such as the Simple Moving Average (SMA), Relative Strength Index (RSI), and the Half-Trend indicator, all of which are well-established in financial analytics.

In addition to these structured numerical indicators, we incorporate unstructured sentiment information using FinBERT, a BERT-based model fine-tuned on financial texts. It assigns discrete sentiment values (+1, 0, -1), helping us quantify market mood and external influences.

To align with sequence modeling requirements, we construct three key input tensors: `memory` (historical feature context), `targets_shifted` (delayed context for decoder autoregression), and `targets_golden` (true values for supervision). These tensors are constructed using sliding windows, and the window size is a tunable hyperparameter.

Together, this enriched, multi-modal input dataset forms the foundation of our modeling pipeline. The processed inputs are then passed into our custom Transformer-based model architecture, described in detail in the following section.

3.1 Dataset

For our primary dataset, we used daily tick data from **yfinance**, which is an open source library allowing online pulling of stock data. For preprocessing, we created 3 major data streams:

These were created by processing windows of the desired size and concatenating then vertically to create the desired shaped vector. The window size is a hyperparameter that can be modified to toggle performance.

In addition to the Open, High, Low, and Close Price (which can be directly pulled from yfinance), we also implemented additional technical indicators, including RSI, HalfTrend, and SMA. The implementational details of these are omitted since they are standard technical indicators in industry.

3.1.1 Sentiment Score with FinBERT

To integrate market sentiment into our predictive framework, we employ FinBERT [19], a specialized pretrained sentiment analysis model developed specifically for financial contexts. FinBERT analyzes

Vector Type	Vector Dimensions	Purpose
targets_golden	$[batch_size, window_size, 1]$	Ground truth Close values for prediction
targets_shifted	$[batch_size, window_size, 1]$	Input context for initializing decoder autoregression
memory	$[batch_size, window_size, d_model, num_feats]$	Encoded historical features providing context for decoding

Table 1: Vector Types, Dimensions, and Their Purposes

textual content from financial reports, news, and social media, assigning sentiment scores of 1 (positive), 0 (neutral), or -1 (negative). Incorporating these sentiment scores provides a valuable contextual dimension, reflecting real-time market conditions and investor sentiment, and enhancing the robustness and accuracy of our predictive model.

Hence, our final list of features can be visualized as follows:

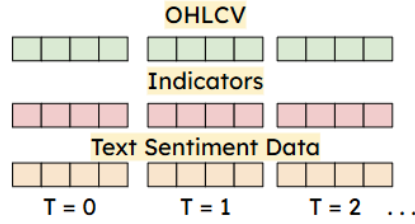


Figure 1: Visualization of features used

This brings us to our inputs, which are time series data of stock features, and our desired output, which is next day closing price.

3.2 Model Description

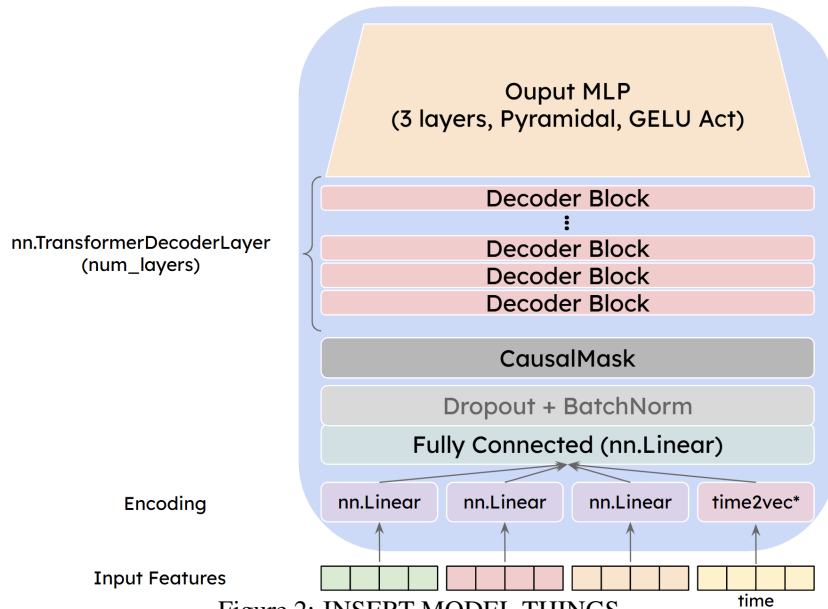


Figure 2: INSERT MODEL THINGS

Our model takes as input a rich sequence of financial features, including technical indicators such as SMA, RSI, and the Half-Trend signal, along with temporal encodings and sentiment scores extracted using FinBERT. These features are precomputed from historical stock data sourced via Yahoo Finance and TradingView.

Initially, we apply feature-specific multilayer perceptron (MLP) encoder layers to transform each indicator into embedding vectors. This step allows us to capture unique data characteristics effectively without unintended interference from other features. Subsequently, we implement Time2Vec embedding [11] to capture complex temporal dynamics inherent in stock market data, enhancing our model’s capability to interpret periodic market fluctuations.

After these feature embeddings, we incorporate absolute positional encoding to clearly represent the sequential order of data points. Transformers, which inherently lack an understanding of sequential positioning, greatly benefit from this explicit encoding, significantly boosting predictive performance by accurately modeling sequential dependencies.

The embeddings obtained from the previous steps are concatenated to form a comprehensive unified vector, which is subsequently projected into a final embedding vector of predefined dimension `d_size`. We further employ batch normalization and dropout layers to mitigate overfitting, enhance model generalization, and stabilize the learning process.

Within the decoder architecture, we utilize a causal attention mask, crucial for autoregressive prediction. This approach prevents the model from accessing future data points during training, preserving the causal relationship essential for realistic financial forecasting. The decoder’s output vectors are finally fed through another multilayer perceptron (MLP) to generate precise predictions of stock prices.

3.2.1 Custom Embeddings

Given the distinct nature and diversity of our input features, it is essential to ensure that each feature’s embedding captures its unique characteristics without inadvertently influencing others. To address this, we implement separate multilayer perceptron (MLP) encoder layers for each individual feature. Each MLP encoder transforms its respective feature into an intermediate embedding vector. These individual embeddings are subsequently concatenated into a single unified vector, which is then projected through a linear transformation into a final embedding vector of dimension `d_size`. This structured approach preserves the unique semantic and numeric properties of each input feature, facilitating effective learning and representation in downstream tasks.

3.2.2 Time2Vec

Time2Vec [11] is a temporal embedding mechanism designed to capture both linear and periodic components of time. Unlike traditional timestamp encodings, Time2Vec is specifically formulated to model patterns with varying frequencies—an essential capability in financial forecasting, where temporal cycles (such as daily volatility, quarterly earnings, or annual trends) significantly impact market behavior.

The Time2Vec embedding comprises two parts: a linear component and a periodic component. For a given scalar time input t , the linear part encodes time using a trainable affine transformation $w_0 t + b_0$, preserving sequential trends. The periodic component is modeled using sinusoidal functions of different frequencies: $\sin(w_1 t + b_1), \sin(w_2 t + b_2), \dots$, where each w_i and b_i are learnable parameters. These periodic functions enable the model to capture recurring patterns in the data, such as seasonality and cyclic behavior.

In our implementation, Time2Vec embeddings are applied to the temporal feature associated with each time step in the financial time series. This enriched representation is then concatenated with the other feature embeddings before being passed into the Transformer decoder. The combination of linear and sinusoidal components allows our model to generalize across both long-term trends and short-term fluctuations in the data.

3.2.3 Positional Encoding

We employ absolute positional encoding to explicitly represent the sequential order of inputs, addressing Transformers’ inherent limitation in recognizing input sequence orientation. This step ensures our model accurately captures both the values and their respective positions in the input data.

3.3 Evaluation Metrics and Loss

Given the desired outcome of predicting Closing price for a stock, we use an evaluation metric of the Mean Absolute Error (MAE) and Mean Squared Error (MSE), which are common error metrics for continuous numerical data. To meet a low error requirement, we develop a hybrid loss function consisting of an MSE component, which reduces the absolute gap between the predictions and targets, and a Physics Loss term, which we will formally define shortly. The behaviour of the model on just MSE, just physics loss, and with the Hybrid loss can be seen in Figure 3.

3.3.1 Physics Loss

Given the limited dataset size—approximately 2500 data points from a decade of daily data—there is a pronounced risk of overfitting. To combat this, we employ a linearization of Geometric Brownian Motion, rooted in financial theory, to provide a domain-specific regularization effect, effectively guiding the model towards realistic market dynamics. Geometric Brownian Motion is a financial model that predicts the next day price of an asset with the following equation:

$$S_t = S_0 \exp \left[\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W_t \right] \quad (1)$$

Where S_t is the next day price, S_0 is the current price, μ and σ are calculated based on historical price, and W_t is a Wiener process variable that is essentially a random variable that mimics the underlying noise/ randomness in the market.

Since this expression is non-linear, and hence computationally expensive to compute, we use a linearized version of this called the Additive Brownian Motion (ABM) equation. This follows the same concept as GBM- that the next day price is defined by some interpolation of the past price, added to a random factor, however it does so in a linear fashion with the following equation:

$$S_t = S_0 + \mu * t + \sigma * W_t \quad (2)$$

3.3.2 Combined loss and evaluation metric

We combine MSE and GBM losses into a unified loss function. This dual-loss strategy allows us to quantitatively and qualitatively evaluate model performance comprehensively, ensuring superior predictive reliability and real-world applicability beyond simple numeric assessments. The final loss can be expressed as follows:

$$\mathcal{L} = w1 \cdot \mathcal{L}_{\text{MSE}} + w2 \cdot \mathcal{L}_{\text{GBM}} \quad (3)$$

Where:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4)$$

$$\mathcal{L}_{\text{GBM}} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \hat{y}_i^{\text{GBM}})^2 \quad (5)$$

Here $\hat{y}_i^{\text{GBM}}$ is the same as S_t in equation 2.

In addition to quantitative loss metrics, we also qualitatively evaluated the forecasts, by looking at general trend following and graph shape fitting. While not exact, this helped us rapidly decide on ablations to run to improve performance.

4 Baseline Selection and Extensions

To effectively evaluate and benchmark the performance of our model, we employ two well-established baselines: (1) a naive baseline and (2) the ARIMA model. The naive baseline simply assumes that today’s stock price is equal to the previous day’s closing price. This approach serves as a fundamental comparison to determine if our sophisticated model adds predictive value beyond trivial predictions. Despite its simplicity, the naive baseline is critical as it represents the minimal effort predictive capability and is frequently used as an initial reference point in financial forecasting tasks.

The second baseline, ARIMA (Auto-Regressive Integrated Moving Average), is an industry-standard statistical forecasting model extensively utilized for financial time series analysis. ARIMA captures temporal dependencies by combining autoregression, differencing (to ensure stationarity), and moving averages. Its mathematical rigor makes it a valuable benchmark as it inherently considers past trends and noises within historical data to make informed predictions. ARIMA has been widely validated in various studies as an effective linear forecasting tool for stock price prediction, making it an essential baseline for our comparison.

Our model significantly differs from these baselines in several ways. Unlike the naive baseline, our decoder-only Transformer model leverages complex deep learning mechanisms to capture intricate market dynamics beyond the simplistic continuity assumption. Compared to ARIMA, our Transformer-based model can inherently handle non-linear and more intricate relationships in financial data through advanced self-attention mechanisms and embedding strategies. Furthermore, our model incorporates additional context such as sentiment analysis and physics-informed loss terms, which allow it to integrate external financial theories and market sentiments effectively. These distinctions enable a robust comparative analysis, highlighting areas where our proposed methodology excels or identifying potential areas for further enhancement.

5 Results

In this section, we present our results across different ablations, and across two different stocks- AAPL and SPX. We evaluate using our metrics- MSE and MAE- the performance of FinDe++ and the different variants against our chosen baselines- ARIMA and next day projection.

5.1 Ablation studies

Parameter	Range Tested	Observed Impact
d_model	100–140	Best results observed in the range 124–132 . Too low reduces expressiveness; too high leads to overfitting.
d_ff	32, 64, 128	64 offers the best trade-off. 32 is insufficient, while 128 adds unnecessary complexity.
d_freq	3–20	All values worked well. Chosen to maintain compatibility with d_model and number of attention heads.
num_layers	2–8	3–5 layers yield optimal performance. Fewer layers underfit; more layers overfit and increase training time.
num_heads	6–24	12–16 heads performed well. Outside this range, error increases significantly, warranting further investigation.
w1 & w2	0.1–25	A ratio between 30–100 (w1:w2) works best. The ideal ratio varies depending on other model settings.

Table 2: Effect of Hyperparameters on FinDe++ Model Performance

Figure 3 shows the results for AAPL, in 3 different modes- figure 3a shows the predictions using just MSE loss, figure 3b shows the predictions with just ABM loss, and figure 3c shows the combined loss predictions. We see that the MSE loss tends to lag behind in terms of following trends, and is unable to cover the sudden spike in price around the 800th data point. On the other hand, the ABM loss based predictions seem to follow the trend a lot better, but have a significant scaling issue. The combined loss gives us the best of both worlds- closer to the raw values, and better at trend following. However, some work needs to be done to improve predictions with this combined loss.

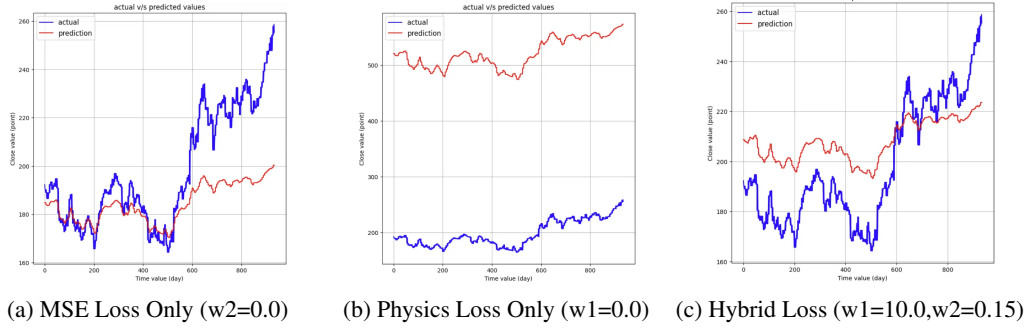


Figure 3: Actual v/s Predicted Close Price for AAPL

We then used SPY data to evaluate the impact of sentiment score on the predictions, since the sentiment data we had was a combination of many stocks, and hence better suited to predict an amalgamation of stocks (like SPY). Qualitatively, between figure 4a and 4b, we see that the results are bad as the sentiment score is not continuous.

Note that the wide, rapid fluctuations in price we observe for SPY are an error in the model that are most likely due to the sensitivity of the model. This is an ongoing challenge (see section: challenges) that we plan on fixing long term.

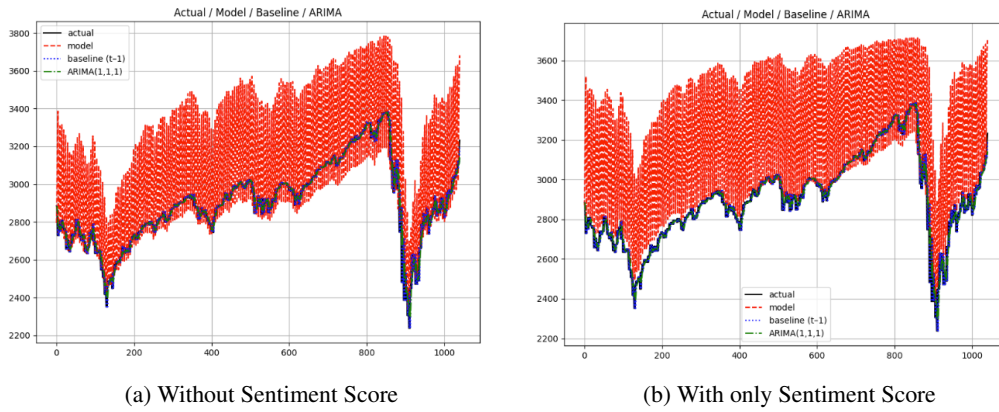


Figure 4: SPY Close Price Predictions v/s Baselines

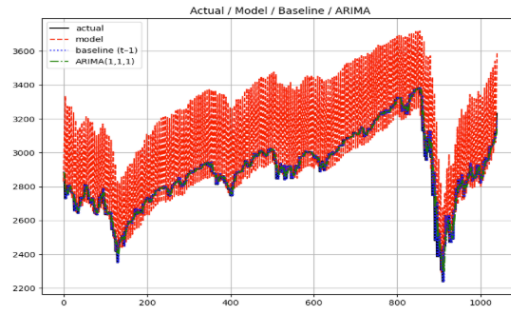


Figure 5: SPY Close Price Predictions v/s Baselines

6 Challenges

Throughout the development and evaluation of FinDe++, we encountered several challenges that have informed our future directions:

Model	Naive	ARIMA	FinDe++	FinDe++ (no sentiment)	Sentiment only
MSE	2449.34	1756.30	71053.65	94875.57	166143.61
MAE	30.05	25.74	217.67	243.07	337.00

Table 3: Results Summary for Ablations Tried

- **Environmental Constraints:** During experimentation, a tornado-induced power outage and disruption of PSC (Pittsburgh Supercomputing Center) services hindered our ability to complete certain ablation studies, particularly those involving variable window sizes. This limited our ability to assess how input window length affects performance.
- **Hyperparameter Sensitivity:** Our model exhibited high sensitivity to hyperparameter choices and feature selection. For example, performance varied significantly between stocks—FinDe++ achieved low variance when predicting Apple stock but showed substantial variance when applied to S&P 500 data. This suggests a need for robust hyperparameter tuning strategies and model generalization techniques.
- **Discrete Sentiment Limitations:** The current sentiment input from FinBERT provides only discrete scores of +1, 0, or -1. While useful, these scores limit gradient flow during backpropagation and do not quantify sentiment intensity. We observed that this hinders the model’s ability to fine-tune its predictions based on nuanced external factors. A more continuous sentiment representation is needed.

7 Future Directions

To enhance the predictive performance and generalizability of our model, several promising directions are under consideration:

- **Reducing Variance:** One key goal is to lower prediction variance through regularization techniques, expanded training data, and refined model architecture. Lowering variance will improve the stability of predictions under volatile market conditions.
- **Embedding Refinement:** We plan to improve the embedding mechanism by exploring richer encodings such as dynamic embeddings that evolve over time, or incorporating attention-based feature weighting to capture interaction effects between features more effectively.
- **Feature Importance Analysis:** Another avenue involves evaluating the contribution of each input feature to model predictions. By leveraging attention weights or SHAP values, we aim to prioritize features that significantly impact prediction quality, thereby enabling more targeted feature engineering.
- **Adaptive Loss Balancing:** Currently, the combination of MSE and GBM loss terms is controlled manually. In future iterations, we aim to enable the model to learn the optimal weighting between these loss terms dynamically during training via meta-learning or reinforcement-based tuning.
- **Handling Temporal Shifts:** We aim to explore techniques for better handling concept drift and structural breaks in financial time series, such as transfer learning across time windows or continual learning frameworks.
- **Evaluation on Broader Benchmarks:** To strengthen external validity, we aim to test our model across different stock sectors, market cap ranges, and temporal spans. This will help establish robustness and uncover model sensitivity across diverse scenarios.
- **Continuous Sentiment Modeling:** To overcome the limitations of FinBERT’s discrete sentiment outputs, we plan to implement or fine-tune a model that provides continuous sentiment scores. This would enhance gradient flow and allow finer sentiment differentiation, improving the model’s interpretability and prediction accuracy.

Collectively, these future directions not only aim to surpass current baseline performances but also advance the broader applicability of decoder-only Transformer models in financial forecasting tasks. Our objective is to make the model not just accurate but interpretable, resilient, and adaptable to the ever-changing dynamics of financial markets.

8 Division of Work

- Nayesha Gandotra implemented the model using Pytorch and made the initial data pipeline for next step prediction.
- Gopalakrishnan Thirunellai Venkitachalam implemented the baselines and performed ablations to improve the model architecture. He also created the custom embeddings for the features.

All the work was performed by Nayesha and Gopal (including the report writing, miscellaneous processes). Abhi tried implementing FinBERT but his implementation could not be used due to errors in data alignment.

9 GitHub Repository

References

- [1] T. Muhammad, A. B. Aftab, M. M. Ahsan, M. M. Muhu, M. Ibrahim, S. I. Khan, and M. S. Alam, "Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market," *International Journal of Computational Intelligence and Applications*, vol. 22, no. 3, p. 2350013, 2023.
- [2] Y. Luan, H. Zhang, C. Zhang, Y. Mu, and W. Wang, "Stock price prediction with sentiment analysis for chinese market," in *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*. Torino, Italia: Association for Computational Linguistics, 2024, pp. 167–177. [Online]. Available: <https://aclanthology.org/2024.finnlp-1.16/>
- [3] S. S. Mehrkian and H. Davari-Ardakani, "An integrated model of sentiment analysis and quantitative index data for predicting stock market trends: A case study of tehran stock exchange," *Expert Syst. Appl.*, vol. 269, no. C, Apr. 2025. [Online]. Available: <https://doi.org/10.1016/j.eswa.2024.126298>
- [4] P. H. Vuong, L. H. Phu, T. H. Van Nguyen, L. N. Duy, P. T. Bao, and T. D. Trinh, "A bibliometric literature review of stock price forecasting: From statistical model to deep learning approach," *Scientific Progress*, vol. 107, no. 1, p. 368504241236557, 2024.
- [5] C. Chatfield, "Forecasting in the 1990s," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 46, no. 4, pp. 461–473, 1997. [Online]. Available: <http://www.jstor.org/stable/2988595>
- [6] A. Sunki, C. SatyaKumar, G. S. Narayana, V. Koppera, and M. Hakeem, "Time series forecasting of stock market using arima, lstm and fb prophet," in *Proceedings of the International Conference on Multidisciplinary Research and Sustainable Development (ICMED 2024), MATEC Web of Conferences*, vol. 392. Hyderabad, India: EDP Sciences, 2024, p. 01163. [Online]. Available: https://www.matec-conferences.org/articles/mateconf/abs/2024/04/mateconf_icmed2024_01163/mateconf_icmed2024_01163.html
- [7] S. Mehtab and J. Sen, "Stock price prediction using convolutional neural networks on a multivariate time series," *arXiv preprint arXiv:2001.09769*, 2020. [Online]. Available: <https://arxiv.org/abs/2001.09769>
- [8] Z. Li, H. Yu, J. Xu, J. Liu, and Y. Mo, "Stock market analysis and prediction using lstm: A case study on technology stocks," *Innovations in Applied Engineering and Technology*, vol. 2, no. 1, pp. 1–6, 2023.
- [9] X. Wang, "Cnn-bilstm-attention algorithm-based stock prices prediction during covid-19," in *Proceedings of the 2024 International Conference on Artificial Intelligence and Communication (ICAIC)*. Atlantis Press, 2024. [Online]. Available: <https://www.atlantis-press.com/proceedings/icaic-24/126003456>

- [10] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, "Transformer-based attention network for stock movement prediction," *Expert Syst. Appl.*, vol. 202, no. C, Sep. 2022. [Online]. Available: <https://doi.org/10.1016/j.eswa.2022.117239>
- [11] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker, "Time2vec: Learning a vector representation of time," 2019. [Online]. Available: <https://arxiv.org/abs/1907.05321>
- [12] K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish, "Lag-llama: Towards foundation models for probabilistic time series forecasting," 2024. [Online]. Available: <https://arxiv.org/abs/2310.08278>
- [13] J. Roberts, "How powerful are decoder-only transformer neural models?" in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jun. 2024, p. 1–8. [Online]. Available: <https://dx.doi.org/10.1109/IJCNN60899.2024.10651286>
- [14] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045782519306814>
- [15] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 379, p. 113741, jun 2021.
- [16] A. M. Aboussalah, X. Li, C. Chi, and R. Patel, "The ai black-scholes: Finance-informed neural network," *arXiv preprint arXiv:2412.12213*, 2024, license: CC BY 4.0. [Online]. Available: <https://arxiv.org/abs/2412.12213>
- [17] T. Jiang and Q. Zeng, "Financial sentiment analysis using finbert with application in predicting stock movement," 2025. [Online]. Available: <https://arxiv.org/abs/2306.02136>
- [18] W. Cheng and S. Chen, "Sentiment analysis of financial texts based on attention mechanism of finbert and bilstm," in *2021 International Conference on Computer Engineering and Application (ICCEA)*, 2021, pp. 73–78.
- [19] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," 2019. [Online]. Available: <https://arxiv.org/abs/1908.10063>
- [20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: <https://arxiv.org/abs/1810.04805>