Auditing Algorithmic Bias in Transformer-Based Trading

Armin Gerami, Ramani Duraiswami
Department of Computer Science, Umiacs
University of Maryland
College Park, MD
[agerami, ramanid]@umd.edu

Abstract

Transformer models have become increasingly popular in financial applications, yet their potential risk making and biases remain under-explored. The purpose of this work is to audit the reliance of the model on volatile data for decision-making, and quantify how the frequency of price movements affects the model's prediction confidence. We employ a transformer model for prediction, and introduce a metric based on Partial Information Decomposition (PID) to measure the influence of each asset on the model's decision making. Our analysis reveals two key observations: first, the model disregards data volatility entirely, and second, it is biased toward data with lower-frequency price movements.

1 Introduction

The recent advancements of advanced artificial intelligence is instigating a paradigm shift in quantitative finance. The auto-regressive Transformer models Vaswani et al. [2017], which have powered recent breakthroughs in natural language processing and computer vision, are now being rapidly adopted for algorithmic stock trading Coelho e Silva et al. [2024]. Architectures such as the "Quantformer" Zhang et al. [2024] leverage sophisticated self-attention mechanisms to capture complex, long-range dependencies within noisy financial time-series, demonstrating empirical superiority over predecessor models like Long Short-Term Memory (LSTM) networks. These models achieve higher predictive accuracy and generate superior cumulative returns, establishing a new state-of-the-art in financial forecasting Guo et al. [2025]. However, their complexity gives rise to a critical challenge: the models operate as opaque "black boxes," where the intricate path from data input to trading decision is largely an enigma, even to their developers Umeaduma [2025]. This lack of transparency poses significant risks for accountability, fairness, and regulatory compliance, creating a need for rigorous auditing methodologies.

This paper trains a Transformer model to predict stock price movements and investigates two core aspects of its decision-making process. First, we audit the trustworthiness of the model's decisions by measuring whether a stock's implied volatility (IV) affects the model's reliance on that stock. Second, we perform an ablation study in which we control the frequency of price movements to simulate various trading speeds for each stock and then measure how this frequency affects the model's reliance on it.

2 Background

This section outlines the analytical frameworks used in our study; Transformer architecture, and PID.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

2.1 Transformer Models

Transformers were designed to overcome the critical limitations of prior recurrent architectures, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Specifically, the sequential nature of RNNs precludes parallelization within training examples and makes it challenging to capture long-range dependencies due to vanishing gradients. The Transformer architecture solves these issues by relying on a parallelizable self-attention mechanism.

The core innovation of the Transformer is the attention mechanism, which allows the model to weigh the importance of all other tokens in an input sequence when computing a representation for a given token. This mechanism operates on a set of queries (Q), keys (K), and values (V), which are linear projections of the input embeddings. To enhance the model's representational power, the Transformer employs Multi-Head Attention, which runs the self-attention mechanism multiple times in parallel with different, learned linear projections for the Q, K, and V vectors. This allows each "head" to focus on different types of relationships within the sequence.

Given a sequence of N tokens, model dimension of C, H heads, and dimension per head of $D = C/H$, the output of each attention head O is derived as

$$\mathbf{O} = \mathbf{AV}, \ \mathbf{A} = \text{Softmax}\left(\mathbf{QK}^T\right), \tag{1}$$

$$\mathbf{o_{ij}} = \frac{\sum_{n=1}^{N} \exp(\mathbf{q}_i.\mathbf{k_n}/\sqrt{D})\mathbf{v_{n,j}}}{\sum_{n=1}^{N} \exp(\mathbf{q}_i.\mathbf{k}_n/\sqrt{D})} = \frac{\sum_{n=1}^{N} f(\mathbf{q}_i.\mathbf{k_n})\mathbf{v_{n,j}}}{\sum_{n=1}^{N} f(\mathbf{q}_i.\mathbf{k}_n)} \tag{2}$$

Then, the output of all the attention heads are concatenated and linearly projected to form the final output of the attention layer.

2.2 Partial Information Decomposition

PID Kolchinsky [2022] is a theoretical framework designed to analyze how a set of input variables provides information about an output variable. Its goal is to move beyond simple mutual information and dissect how the information is structured among the sources. The central idea is to break down the total information into distinct, non-overlapping components, each describing a different mode of interaction:

- **Mutual Information (I)** The mutual information between a source and the target variable.
- **Union Information (U):** The mutual information provided by at least one individual source.
- **Excluded Information (E):** The information in the union of the sources except one particular source.

Given source variables $X_1, X_2, ..., X_k$ and target variable Y, the PID is written as

$$E(X_i \to Y \mid X_1, X_2, ..., X_k) = U(X_1, X_2, ..., X_k;\, Y) - I(X_i;\, Y), \tag{3}$$

$$I(X_i;\, Y) = H(X_i) - H(X_i|Y), \quad H(.) := \text{Shannon Entropy} \tag{4}$$

$$U(X_1, X_2, ..., X_k;\, Y) = \inf_{Q} I(Q;\, Y) \text{ such that } \forall i\, X_i \in Q, \tag{5}$$

where $E(X_i \to Y \mid X_1, X_2, ..., X_k)$ measures "What knowledge does the rest of the group have that X_i is missing".

3 Method

First, we explain how we train our transformer to predict the price movement of a 'target' stock. The model uses the price time-series of the target stock and other relevant 'support' stocks to gain a broader perspective on market movements. Then, we describe how we use PID to measure the influence of each support stock on the transformer's predictions for the target stock.

3.1 Training Transformer

We use historical data from **January 2025** to **June 2025** for training and validation, using the percentage return (pr) of each stock as the input data. The pr is calculated as:

$$pr(t) = \frac{price(t) - price(t-1)}{price(t)}. \tag{6}$$

For simplicity, we only consider pr, although utilizing additional indicators might improve accuracy. We use **NVDA** as the target stock and **AMD**, **MU**, **TSM**, and **INTC** as the support stocks. The pr and IV of these stocks can be found in Appendix A.

Thus, the input to the transformer at each timestep is a five-dimensional vector containing the pr of these five stocks. This vector is then passed to the first attention layer, and we set the timestep as 1 hour. The output of each attention layer is then forwarded to the next, with the final layer producing a 64-dimensional vector. This vector represents a probability distribution over quantized predicted pr values in the range of $[-12.8\%, 12.8\%]$, similar to how an LLM's output is a probability distribution over the possible tokens. This design was chosen over directly predicting a single pr value because a probability distribution is necessary for calculating mutual information during PID. Figure 3 in Appendix A shows the model's prediction result. We should emphasize that focus of this study is on the transformer's decision-making process, and not it's predictive accuracy.

The network has an embedding size of $D = 256$, $H = 8$ attention-heads per attention layer, and four attention layers. Furthermore, we use ROPE Pochet et al. [2023] for positional encoding, and take advantage of add-and-norm residual connection. We set the context length to $N = 64$, meaning the model uses the pr of the target and support stocks from the previous 64 timesteps to predict the target's pr for the next timestep. We use mean Cross-Entropy as our loss function, and apply a dropout rate of 0.1 to prevent overfitting.

3.2 Measuring Influence

Our goal is to: 1. Measure the influence of each support stock on our transformer model's prediction of the target stock's pr. 2. Determine whether the model has less reliance on support stocks with higher IV. 3. Investigate a potential bias in the model towards the trading frequency of the stocks. Let's denote the target and support stocks as $X_1, ..., X_5$, and the model's prediction as Y.

We measure the influence of X_i by calculating the excluded information, $E(X_i \to Y|X)$, using PID. The lower the excluded information, the higher the model's dependence on X_i, and therefore the higher its influence. To derive excluded information, a distribution for the stocks' pr and the model output is needed. We approximate the distribution of the input through quantizing the prs and deriving a histogram over the six month time frame. The probability distribution of the model output is given by the transformer itself, as the final attention layer generates a probability for each value within a predefined set of possible prs.

4 Results

This section presents two key observations from our results: 1. High IV does not discourage the transformer model from relying on a support stock. 2. The model exhibits a bias toward data from stocks with higher trading frequencies. Ideally, our model should have lower reliance on stocks with a higher IV. Therefore, the supports with higher IV should have higher excluded information. However, as we will show in Section 4.1, this is not the case. We then examine how trading frequency affects the model's decision making. Since higher trading frequency can cause greater price variation, we apply a low-pass filter to the price time-series to synthetically smooth these fluctuations, simulating a lower frequency of trades. We find in Section 4.2 that applying this filter to a support stock causes its excluded information to decrease, which indicates an unwanted model bias towards stocks with lower trading frequencies.

4.1 Reliance Independent of IV

In Figure 1 we plot the IV for each of the support stocks, and their excluded information on daily average. Excluded information measures the model's reliance on that stock for the target prediction, and a higher value means lower reliance. We want the model to rely less on stocks with higher IV. However, the results show that the transformer's reliance on a stock is independent of its IV. There are numerous instances that the IV is high while EI is high as well, and the overall correlation between Ei and IV is close to zero. Ideally the EI should be high whenever IV is high to indicate that the model has reduced reliance on a data when the uncertainty is high, and the correlation should be close to 1.

Relying on a support stock with high IV is risky since its price is more likely to experience large, sudden swings Addy et al. [2024]. These movements can influence the model, causing its prediction

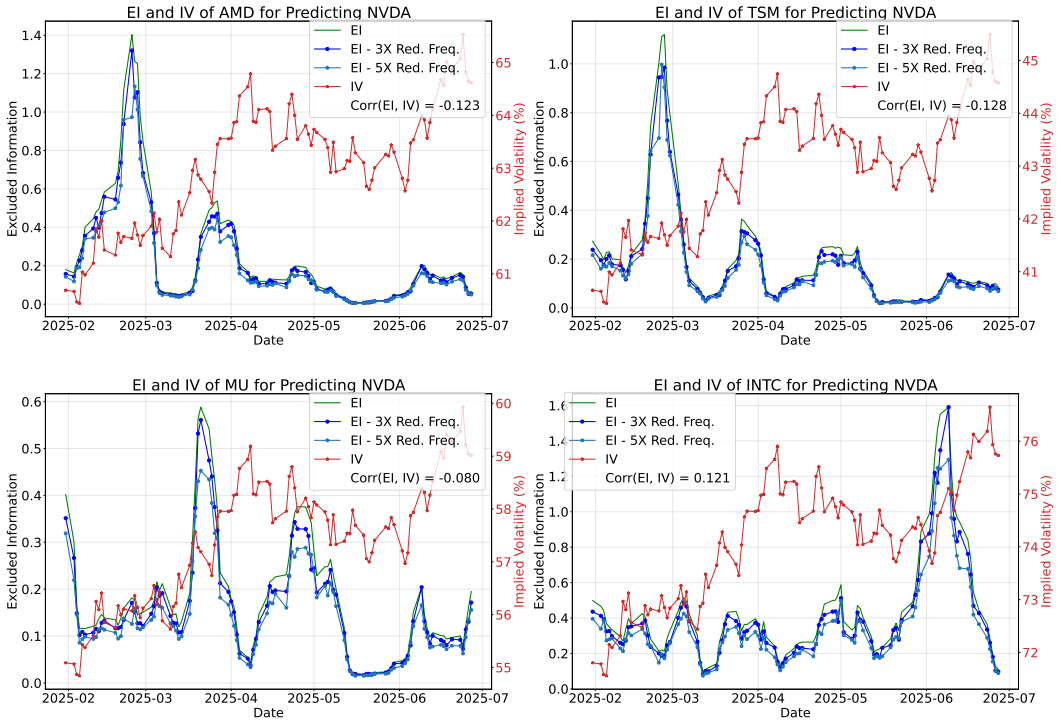

Figure 1: Excluded Information (EI) for each support stock used when predicting the target stock, NVDA. A higher EI value indicates that the model relies less on that stock's data. The right axis displays the 30-day implied volatility (IV) as a proxy for risk. Ideally, high IV should lead to high EI (less reliance on risky data), but the near-zero correlation shown here indicates the model is not discounting this risk. The plot also includes scenarios where support stocks are modified to simulate 3× and 5× reduced trading frequencies for the support stocks, which consistently results in lower EI.

to mirror the support stock's trajectory. If this movement is specific to that stock and not reflective of the broader market, it can lead the model to predict an incorrect pr direction. For example, the model relies most heavily on **AMD** (as shown by its low excluded information) despite it having the highest IV. If company-specific news were to cause a sharp drop in **AMD**'s price, this high reliance would skew the model's prediction for **NVDA** downward, even if the news had no bearing on **NVDA**'s actual value.

4.2 Trade Frequency Bias

To evaluate the effect of trade frequency of the support stocks on our model, we train the model under three sets of condition: 1. Use the same price movement timestep as the target stock (1 hour timesteps), 2. use a $3\times$ reduced frequency (3 hour timesteps), and 3. use $5\times$ reduced frequency (5 hour timesteps). Figure 1 shows the daily average of EI of the support stocks for each of these conditions, as well as the IV. The results show that the $5\times$ reduced frequency consistently achieves lower EI, followed by the $3\times$ reduced frequency.

This indicates that the model prioritizes lower fluctuations, which can be attributed to two phenomena. First, reducing the price movement frequency essentially applies a low-pass filter, which may result in the removal of high-frequency noise. Second, this filtering can also mitigate overfitting. This is because the model avoids learning random patterns as if they were a genuine pattern, which would cause it to perform poorly on new data.

5 Conclusion

In conclusion, this paper takes a step to address the critical "black box" problem of Transformer models in finance, moving towards understanding the logic of these powerful but opaque models. We investigated the model's decision-making by auditing its response to risk (measured via implied

volatility) and analyzing its sensitivity to the frequency of price movements through an ablation study. We found that the model disregards data volatility and is biased toward data with lower-frequency price movements.

References

Wilhelmina Afua Addy, Adeola Olusola Ajayi-Nifise, Binaebi Gloria Bello, Sunday Tubokirifuruar Tula, O Odeyem, and Titilola Falaiye. Algorithmic trading and ai: A review of strategies and market impact. *World Journal of Advanced Engineering Technology and Sciences*, 11(1):258–267, 2024.

Lucas Coelho e Silva, Gustavo de Freitas Fonseca, and Paulo Andre L Castro. Transformers and attention-based networks in quantitative trading: a comprehensive survey. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 822–830, 2024.

Wenhao Guo, Yuda Wang, Zeqiao Huang, Changjiang Zhang, et al. Trading under uncertainty: A distribution-based strategy for futures markets using futurequant transformer. *arXiv preprint arXiv:2505.05595*, 2025.

Artemy Kolchinsky. A novel approach to the partial information decomposition. *Entropy*, 24(3):403, 2022.

Etienne Pochet, Rami Maroun, and Roger Trullo. Roformer for position aware multiple instance learning in whole slide image classification. In *International Workshop on Machine Learning in Medical Imaging*, pages 437–446. Springer, 2023.

CMG Umeaduma. Explainable ai in algorithmic trading: mitigating bias and improving regulatory compliance in finance. *Int J Comput Appl Technol Res*, 14(4):64–79, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhaofeng Zhang, Banghao Chen, Shengxin Zhu, and Nicolas Langrené. Quantformer: from attention to profit with a quantitative transformer trading strategy. *arXiv preprint arXiv:2404.00424*, 2024.

A Supporting Figures

The appendix includes figures to illustrate percentage return (pr) and implied volatility (IV) for the target stock NVDA and each of the support stocks in Figure 2.

Furthermore, Figure 3 shows the trained Transformer model's pr prediction and actual data. The high correlation indicates meaningful model prediction.

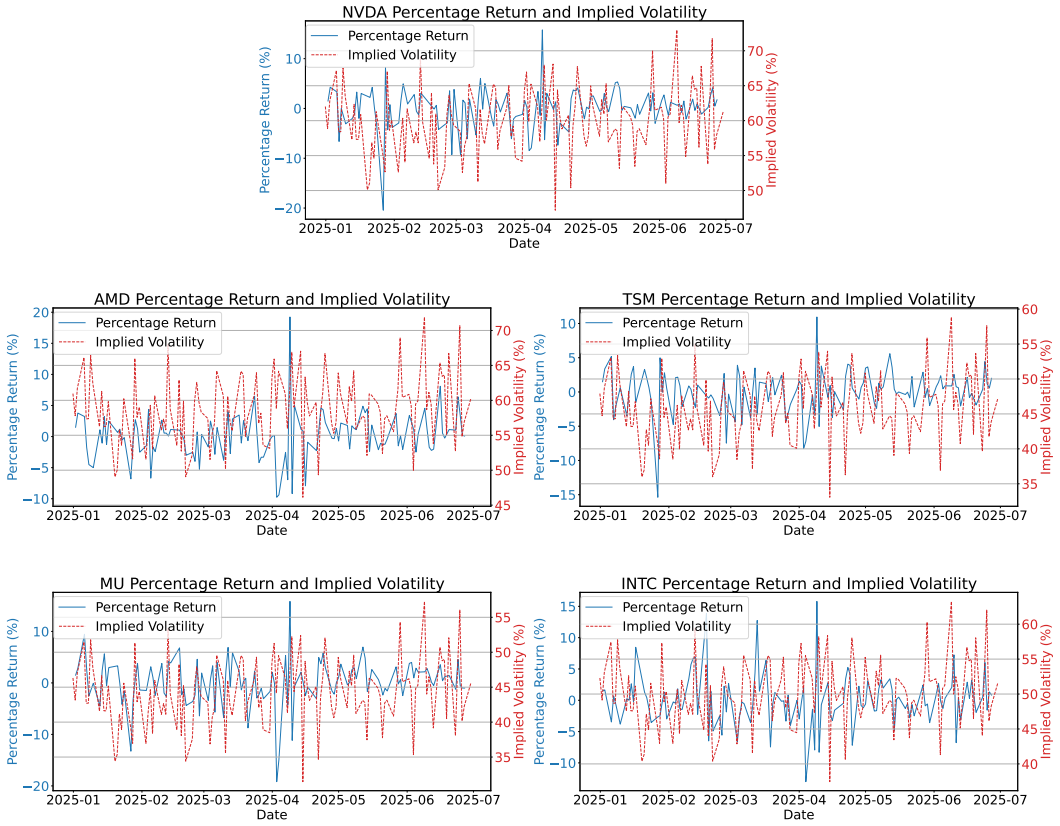

Figure 2: Percentage return and 30-day implied volatility of our target stock (NVDA), and the support stocks (AMD, TSM, MU, INTC).

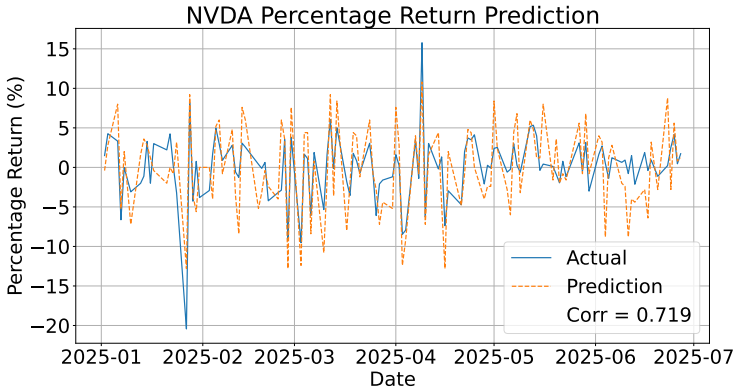

Figure 3: The actual percentage return of the NVDA stock and the trained transformer's prediction.