Daily Market Return Prediction with Transformer *

Yufeng Han† Ryan Huang‡ Guofu Zhou§

This Version: August 2025

Abstract

We apply a Transformer encoder to forecast daily market returns using lagged market returns over horizons of 5, 20, and 60 days. Both the direct model forecasts and post–machine learning forecasts exhibit significant predictive power for next-day returns, while simple averages of past returns do not. Relative to linear predictive regressions, the machine learning forecasts deliver sizable improvements in out-of-sample R-squared. A mean–variance analysis with a risk-aversion coefficient of two shows that the Transformer prediction generates an average return of 30% per annum with a Sharpe ratio of 1.3. The predictability is more pronounced in recessions and periods of elevated investor sentiment. Random Forests and feed-forward Neural Networks also yield economically meaningful, though somewhat weaker, results.

JEL Classification: G12; E44; Q43
Keywords: Transformer

*We thank Chenglong Fu, Guannan Liang, Yingcheng Sun, Qianqian Tong, Chunjiang Zhu for helpful comments. All errors are ours.

†Department of Finance, UNC Charlotte. Email: yhan15@uncc.edu.

‡Department of Computer Science, Department of Economics, UNC Chapel Hill. Email: rhuang1@unc.edu

§Department of Finance, Olin Business School, Washington University. Email: zhou@wustl.edu.

1. Introduction

Artificial Intelligence (AI) has become ubiquitous and is reshaping how researchers and practitioners approach classical topics in finance. Leveraging both traditional machine learning techniques and more advanced deep learning methods, AI can uncover complex, previously undetected patterns in financial data (see Gu, Kelly, and Xiu (2020); Kelly and Xiu (2023)). Among recent advances, models based on the Transformer architecture (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin, 2017) have emerged as particularly powerful and successful. Originally developed for sequence-to-sequence tasks in natural language processing, Transformers rely entirely on attention mechanisms to capture dependencies across input sequences and form the foundation of applications such as ChatGPT.

Given the sequential nature of stock returns, this architecture naturally extends to predicting stock market returns. Canonical return predictors, such as momentum (Carhart, 1997) and mean reversion (De Bondt and Thaler, 1985), are themselves explicit functions of past returns. In effect, future returns appear to "attend" to prior return realizations: when recent past returns are high, momentum suggests high near-term returns; conversely, when long-horizon past returns are high, mean reversion implies that future long-horizon returns may be lower. At short horizons such as the daily level, Lo and MacKinlay (1990) shows that individual stock returns often exhibit negative autocorrelation due to bid–ask bounce. However, these microstructure effects tend to average out in aggregate. At the portfolio or market level, return dynamics are instead shaped by cross-sectional lead–lag relationships, wherein large-cap stocks incorporate market-wide information more rapidly than small-cap stocks. This mechanism can generate positive autocorrelation in portfolio returns, such that high past returns predict high future returns over short horizons.

Motivated by the parallels between return dynamics and the attention mechanism, this paper applies the Transformer architecture to predict daily stock market returns. We focus on the daily market return for two key reasons. First, market return predictability is a

central question in asset pricing, and forecasting at the daily level presents substantially greater challenges than at lower frequencies, such as the monthly horizon. Second, the Transformer is a data-intensive architecture that requires large volumes of training data—far more than the few hundred observations typically available in monthly market return series. Specifically, we evaluate whether the Transformer's predictions based on past daily stock returns outperform a simple average of past daily returns. To provide more context, we also include predictions from other popular ML methods such as Random Forests and traditional feedforward Neural Networks.

We use past returns over different horizons—5, 20, or 60 trading days—as input features for the three machine learning methods. To evaluate predictive power, we begin by regressing the market return at day t+1 on the predictions made at day t for day t+1. We find that the slope coefficients are highly significant in all cases. For instance, the slopes are around 0.7 with t-statistics of 10, 0.6 with t-statistics of 7, and 0.2 with t-statistics of 4 for Transformer, Random Forest, and Neural Network, respectively. In contrast, the slopes are insignificant when the explanatory variables are the average of past daily returns. We then design a marketing timing strategy where we invest in the market if the prediction is positive and invest in the risk-free rate when the prediction is negative. We find improvement in the Shape ratio to more than one when comparing it to simple buy-and-hold, which has a Sharpe ratio of 0.49, or the historical average, which often has a Sharpe ratio of less than one.

These results are encouraging as they suggest that these machine learning methods are effective in processing the information contained in the input features. However, a problem we notice is that the magnitude of the ML prediction does not quite align with the actual returns, which leads to poor out-of-sample R-squares. Prior literature has proposed post-ML calibration to realign the scale (Belloni, Chernozhukov, and Hansen, 2013, 2014; Ribeiro, Singh, and Guestrin, 2016). Following this line of literature, we adopt a post-modeling regression adjustment. Each year, we regress the future observed returns on each ML model's forecasts recursively, using expanding windows, and generate out-of-sample forecasts for the

next year to produce new forecasts that better align with observed returns. The benchmarks here use the same input features, but their unadjusted forecasts come from multiple regression, not ML.

With the post-ML forecast, we confirm that these new forecasts also predict the next-day return positively with high significance. However, they have now achieved a substantial improvement in out-of-sample R-squared compared to the relevant benchmark based on historical daily returns. The post-ML forecasts from Transformer yield out-of-sample R-squares of 0.94%, 1.03%, and 0.97%, respectively, when the input features are the returns from the past 5,20, or 60 days. In contrast, the recalibrated forecasts, generated through multiple regression of historical daily returns, yield an out-of-sample R-squared of close to zero. The out-of-sample R-squares also improve when using predictions from Random Forest and Neural Network, but somewhat weaker. We find that the variance of the forecast errors is relatively small for Transformer when compared to the other two methods. We also conduct a mean-variance analysis using the post-ML forecasts. Here, we can determine the exact long and short positions to invest in the market on any given day. With a risk aversion parameter of two, the Transformer achieves a mean return of approximately 30% per year, accompanied by a Sharpe ratio of around 1.2, which doubles the performance of the benchmarks. Random Forest comes next, followed by a Neural Network.

We find a mixed picture of the performance of the out-of-sample R-squared values of three ML methods by business cycle, sentiment, uncertainty, and VIX. Their predictions tend to perform better during recessions, when uncertainty or VIX is low. Transformer performed better when sentiment is high, but Random Forest and Neural Network perform better when sentiment is low.

We make the following contributions. We are among the first endeavors in the finance literature to apply a Transformer encoder for predicting daily market returns. We find that both the unadjusted predictions and the recalibrated predictions significantly predict future daily returns. In contrast, the simple average of past daily returns in the correspond-

ing window or its recalibrated version fails to do so. Furthermore, the post-ML prediction demonstrates supporting results when we extend the analysis to include mean squared error and mean variance analysis. Random Forest and Neural Network can also improve the performance relative to those benchmarks. We view this as a meaningful endeavor, as the performance of the U. S. stock market index has been stellar and unparalleled by other indexes and most individual stocks in the long term, and return prediction at the daily frequency is largely an open area of research. Our focus is not on whether Transformer outperforms Random Forest, Neural Network, or other ML methods, because one can perhaps use variations of Random Forest or employ deeper neural networks. Overall, our results support the use of ML methods in predicting daily market returns, either because of the nonlinear relations these methods can capture or because the attention mechanism can provide an advantage in this context. A close study is presented by Cong, Tang, Wang, and Zhang (2021), who also utilize the Transformer, but they use monthly data and focus on designing cross-attention among firm-level variables for return prediction at the cross-section.

This paper is organized as follows. Section 2 discusses relevant literature. Section 3 presents methodology, Section 4 discusses our empirical results, and Section 5 concludes. Other supporting results are in the online appendix.

2. Related Literature

This section reviews the literature relevant to why daily market returns may be predictable. A large body of work studies market return predictability using time-series data (Campbell and Shiller, 1988; Campbell and Thompson, 2008a; Welch and Goyal, 2008; Da, Huang, and Yun, 2017). Much of this literature interprets return predictability through the lens of a time-varying discount rate. If aggregate returns are predictable at low frequencies, such as annual or monthly, then some form of predictability must also exist at the daily horizon, since daily returns aggregate into long-horizon returns. The slope from regressing monthly returns

on a predictive variable equals the sum of the slopes from regressing the daily components of the monthly return on the same variable. Savor and Wilson (2013) show that equity risk premia are concentrated on days with important macroeconomic announcements. This evidence suggests that daily returns may be predictable at least on specific days, despite the fact that daily data is noisier.

A second explanation for return predictability is gradual information diffusion, which induces autocorrelation in returns. Lo and MacKinlay (1990) shows that individual stock returns often exhibit negative autocorrelation due to bid–ask bounce. However, these microstructure effects tend to average out in aggregate. At the portfolio or market level, return dynamics are instead shaped by cross-sectional lead–lag relationships, wherein large-cap stocks incorporate market-wide information more rapidly than small-cap stocks. This mechanism can generate positive autocorrelation in portfolio returns, such that high past returns predict high future returns over short horizons. See also Hou (2007). More recently, Baltussen, Van Bekkum, and Da (2019) shows that, as more stocks are indexed, non-fundamental shocks to stocks in the index can cause negative autocorrelation at the index level. Aleti, Bollerslev, and Siggaard (2025) use lagged returns from portfolios sorted by firm characteristics and industry classifications as predictors to predict market returns. Liu and Stentoft (2023) employs lagged market returns along with lagged returns of SP 500 constituents to forecast intraday movements. Dong, Li, Rapach, and Zhou (2022) show that long-short portfolio returns based on firm characteristics can predict future market returns, and the predictive ability appears to stem from asymmetric limits of arbitrage and overpricing dominance.

Behavioral explanations provide another foundation for return predictability. Markets exhibit both underreaction and overreaction, long documented in the cross-section of stock returns. More recently, Da, Hua, Hung, and Peng (2025) found that aggregation attention by both retail investors and institutional investors predicts market returns, but with opposite signs. Daily aggregate retail attention negatively predicts one-week-ahead market returns,

is associated with aggregate retail order imbalance, and tends to flow into equity mutual funds. In contrast, aggregate institutional attention, when observed prior to major news announcements, positively predicts future market-wide returns.

Technical analysis provides a complementary perspective. Brock, Lakonishok, and LeBaron (1992) find that 26 commonly used trading rules generate significant abnormal returns. Lo, Mamaysky, and Wang (2000) develop a kernel regression approach to identify chart patterns and show that certain technical indicators contain incremental predictive power. Sullivan, Timmermann, and White (1999a) evaluate nearly 8,000 trading rules, documenting significant performance even after correcting for data snooping. More recently, Jiang, Kelly, and Xiu (2023) apply deep learning to chart patterns and uncover strong predictability in daily returns.

A growing literature also explores the use of machine learning and deep learning in return prediction. Most studies focus on monthly data and the cross-section of stock returns. Gu et al. (2020) apply classical machine learning and deep learning methods to a broad set of firm characteristics and find strong cross-sectional predictability. Related contributions include Li, Yan, Rossi, and Zheng (2025), Cao, Jiang, Yang, and Zhang (2023), and Li and Tang (2024). Recently, several studies apply sequence models such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and Transformers to asset return prediction. A related paper by Cong et al. (2021) designs cross attention for numerous firm and market-level variables to forecast the cross-section of stock returns. Cong, Tang, Wang, and Zhang (2020) provides a broader overview of deep sequence modeling techniques—including recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and Transformers—and highlights their advantages in capturing temporal dependencies that conventional machine learning methods often overlook. They show that sequence-based models yield superior out-of-sample performance in asset return prediction, underscoring the value of attention mechanisms in financial forecasting. See also Wang (2024).

Despite these advances, the application of sequence models to daily market-level return

prediction remains scarce. The computer science literature contains some efforts that utilize machine learning methods for stock price prediction; however, these studies often lack economic foundations and rarely forecast stock returns. See for example, Yáñez, Kristjanpoller, and Minutolo (2024). Most existing work in the finance literature focuses on long-horizon predictability or the cross-section. In contrast, our approach employs a plain Transformer encoder to forecast daily aggregate returns, relying solely on lagged returns as predictors.

3. Methodology

3.1. Transformer

Introduced by Vaswani et al. (2017), the Transformer relies on a multi-head attention mechanism and has demonstrated superior quality in many applications while being more parallelizable and requiring significantly less training time. Because our goal does not involve translation between languages, the whole encoder-decoder architecture of the transformer is unnecessary. Instead, we will focus on the transformer's encoder component for return prediction.

In the encoder of a plain transformer with a learned matrix, the process of embedding maps a token in an input feature to a dense vector representation. Each token in the input sequence is assigned a position encoding, along with another vector representation. Suppose the length of the input feature is T, and the dimension of the embedding is d; the embedding produces a matrix X of dimension T by d.

By applying linear transformations to X, a head in the transformer attention block will first compute a query matrix (Q) matrix, a key matrix (K), and a value matrix(V) as follows:

$$Q = X\,W_Q, K = X\,W_K, V = X\,W_V \tag{1}$$

where the dimensions of the weight matrix Ws are d by d_k, and the subscriptions relate to

different matrixs. For instance, if the overall dimension of the embedding is 32 and there are four parallel heads, then the length of the embedding on each head is given by the ratio of these two, and that is, d_k is eight. The information in a token is stored in V. Tokens try to find who is more interesting to each other through matrix multiplication of Q and the transpose of K, and once scaled by the dimension of the embedding d_k we have the attention weights, which are the key operation in the multi-head self-attention within each encoder layer. The building block is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V \tag{2}$$

where the function softmax normalizes the scaled attention weights to be probabilities between zero and one. Often, a mask, which takes the form of a lower triangular matrix of ones, is used when calculating attention. The purpose is to allow day t information to pay attention to information from day 1 and day t, but not from future days. This process will build new contextual information for day t using information from days 1, 2, ..., and t. The outputs from multiple heads are concatenated and then added to the original X through a residual connection. After layer normalization, the self-attention output passes through a feed-forward network, whose output is then further processed through a residual connection and layer normalization. Then, the output from iterations of multiple layers can be used for forecasting. The online appendix presents a figure that shows the flow of the Transformer encoder.

The masked attention aims to build a rich and deep representation of the entire input sequence. For instance, on day 3, contextual information will be built using information from days 1, 2, and 3 in the first layer of the transformer encoder. This contextual information and its associated subsequent output will be used to form the contextual information for day 4 at the second layer, and so on. Therefore, day t contextual information utilizes contextual information from previous days in the previous layers, which may help learn

potential temporal stock return patterns to better predict returns at day t+1. When the input feature is 5 days of past return, the contextual information from each of the past five days will be used to make a single numerical prediction for the day 6 return.

3.2. Random Forest

The Random Forest (Breiman, 2001) method is based on regression trees or decision trees. The goal is to identify groups of observations that exhibit similar behavior. To achieve that, random forest partitions the sample by predicting variables sequentially. The value of the outcome variable, in this case, the predicted returns, is calculated as the average within each partition. Suppose in one tree, there are K "leaves" or terminal nodes, and the depth of the tree is L. The predicted return g can be described by the following response function

$$g[z_{i,t}; \theta, K, L] = \sum_{k=1}^{K} \theta_k \mathbf{1}_{z_{i,t} \in C_k(L)} \tag{3}$$

where $C_k(L)$ is one of the K partitions in the training data. The indicator function $\mathbf{1}_{z_{i,t} \in C_k(L)}$ determines if an observation $z_{i,t}$ falls into partition $C_k(L)$. The constant for partition k, θ_k, is the sample average returns in the partition.

To operate the above procedure, one first draws a bootstrapped sample from the data in each iteration. Then, one grows a tree by recursively repeating the following steps until a minimum node size is reached: randomly select a group of forecasting variables, pick the best splitting variables for each node, and split the node into two daughter nodes. As shown in Hastie and Friedman (2009), the criterion for selecting the best variables and splitting involves the minimization of values from impurity functions, which are weighted averages of the sum of mean-squared errors obtained from the daughter nodes. At each step of the tree, the sample is randomly chosen; hence, the name of this method reflects the "random" nature of the process. There are many possible trees; hence there comes the "forest."

Due to its superior performance and improved interpretability, Random Forest is fre-

quently ranked among the top-performing machine learning methods, making it a reliable choice across various applications, such as predictions in tabular data (Grinsztajn, Oyallon, and Varoquaux, 2022; Shwartz-Ziv and Armon, 2022). This robustness is especially valuable in finance when the signal-to-noise ratio is low or when there is overfitting (Donick and Lera, 2021). A recent paper by Shen and Xiu (2025) shows that when signals are weak, random forest outperforms gradient boosting. Other studies show that random forest even outperforms neural network-based methods when the sample is small.

3.3. Neural Network

The neural network, a state-of-the-art machine learning method, addresses complex problems such as computer vision in autonomous driving. Our focus on conventional "feed-forward" networks consists of an input layer, hidden layers, and an output layer. Within the hidden layers, there are nodes and connections among these nodes, which are meant to be analogous to the neurons and synapses found in an animal brain.

In the online appendix, we present a a simple neural network. It has three forecasting variables in the input layer, two hidden layers, and an output layer. The forecasting variables are Transformer before they enter the first hidden layer, which has four nodes. For instance, the first node in the input layer represents a forecasting variable. This scalar X_1 will be Transformer by

$$Z_1 = W_1 \times X_1 + C_1 \tag{4}$$

where W_1 and C_1 are both four-by-one column vectors that transform X_1 into a new input Z_1 of a dimension of four by one before it enters the four nodes in the first hidden layer, respectively. The above equation repeats for other nodes at the input layer, indicating there will be W_2 and C_2 corresponding to the second forecasting variable, and W_3 and C_3 corresponding to the third forecasting variable, etc. The collection of Ws and Cs are the

weighting matrix immediately after the input layer. At each node in the first hidden layer, the input Z_1, as well as all other Zs, is processed via an activation function, which can be the Rectified Linear Unit (ReLU), Sigmoid, or Softplus, etc, and then further processed via the new weighting matrix attached to the first hidden layer, before it enters to the second hidden layer. This process repeats until it hits the output layer. The parameters in the weighting matrix in all layers increase rapidly with the number of layers and nodes used in each layer. They are commonly estimated using backwardation and stochastic gradient descent, in which the learning rate governs the step size.

Our neural network has four hidden layers. The number of neurons in the first hidden layer is optimally chosen to be either 16 or 32. The number of neurons in the second to the fifth hidden layer is fixed at 32, 16, and 8, respectively, following the geometric pyramid rule of Masters (1993). Our activation function is the commonly used ReLU defined as

$$ReLU(Z) = Max(Z, 0) \tag{5}$$

where Max is an operator that takes the larger value of Z and 0. This function encourages the sparsity of active neurons and facilitates faster descent calculation. We choose the adaptive moment estimation (Adam) by Kingma and Ba (2015) for computational efficiency.

3.4. Post-ML Regression

Transformers and other ML methods, such as Neural Networks, use a nonlinear structure to form predictions. Although the nonlinearity may better uncover the complex and predictive relation between past returns and future returns, the ensuing scale of the predictions can be very different from the actual returns, which results in a misalignment, as we show in Section 4. The literature has advocated for additional calibrations, such as reprojection or post-ML regression techniques, to address this issue. These methods generally improve inference and predictive performance as well as interpretability.

For example, Belloni et al. (2013, 2014) recommend applying OLS regression to variables selected by machine learning (e.g., LASSO) to stabilize estimates. Similarly, Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018) propose a "debiased machine learning" framework where an additional calibration step is used to reduce the impact of regularization bias and overfitting in machine-learned nuisance parameters on estimation of the parameter of interest. Post-ML is also used to increase interpretability. For example, Ribeiro et al. (2016) propose LIME (Local Interpretable Model-agnostic Explanations), a novel explanation technique that trains a local, interpretable model (e.g. linear regression) around individual predictions of a black-box model to explain why that specific prediction was made.

Following this line of literature, we adopt a post-modeling regression adjustment. Each year, we regress the future observed returns on each model's forecasts recursively with expanding windows and generate out-of-sample forecasts for the next year to produce calibrated predictions. By regressing returns on these forecasts, we recalibrate them into new forecasts that better align with observed returns, and possibly extract a linear signal from a noisy forecast. This procedure allows the Transformer to flexibly encode latent structure in returns, whereas the second-stage linear projection maps these outputs to the appropriate scale and sign, thus yielding improved out-of-sample R-squares. In essence, we treat the original forecasts from the machine learning methods as predictors to be used in a linear predictive regression. This is also similar to Valaitis and Villa (2024), who use ML to approximate expectations within an economic model, and then use these approximations in a structured way for policy analysis.

4. Empirical Results

4.1. Data and Estimation

Our testing data is the daily market excess return obtained from Professor Fama's website. The sample is from July 1, 1926, to December 29, 2022. We use rolling samples for estimation each time. In all estimations, the input features are either the returns in the past five days, 20 days, or 60 days. The sample used for training and validation is five years, with the last year reserved for validation. For example, the first training sample spans from 1927 (including the second half of 1926) to 1931, with 1931 used for validating the hyperparameters. We then make return predictions for 1932. Note the last block of observations (either 5 days, 20 days or 60 days) in 1931 is used for making the first prediction in 1932 and are not used for validation. We repeat these steps for 1928 to 1932, 1929 to 1933, etc., with their corresponding prediction. This ratio of four between the size of training and validation sample is a reasonable choice and is consistent with (Picard and Berk, 1990).

In the transformer, we set the learning rate to either 1e-4 or 1e-5, the number of embeddings to either 32 or 64, the number of layers to either 2 or 4, and the dropout rates to either 0.1 or 0.2. In the random forest, we set the number of trees to either 250 or 500 and the maximum depth of those trees to either 1, 3, or 5. In the simple Neural Network, we set the number of epochs to 500, and it has five layers with 64 neurons at the first layer. The dropout rate is either 0.1 or 0.2, and the learning rate is either 0.001 or 0.01.

4.2. Unadjusted Forecasts

We begin by evaluating the predictive content of the forecasts generated by the three machine learning methods, employing both predictive regressions and market-timing tests.

4.2.1. Predictive Regressions

Table 1 reports the Fama-MacBeth regression results. We regress the next-day excess returns on the forecasts generated with block inputs of 5-, 20-, and 60-day by the three machine learning methods. For each input block, we use the corresponding average historical returns (5-, 20-, and 60-day averages) as the benchmark. Panel A reports the results for Transformer forecasts. Regardless of input blocks, the second, fifth, and eighth columns show that the forecasts significantly predict future returns, with coefficients around 0.70 and t-statistics larger than 10.00. In sharp contrast, as shown in the first, fourth, and seventh columns, none of the three benchmarks (5-, 20-, and 60-day average returns) is significant, highlighting the usefulness of Transformer in predicting market returns.

However, as pointed out by Mincer and Zarnowitz (1969) and Lewellen (2015), the estimate of the slope coefficient measures the bias of the forecast. If the estimated slope coefficient equals one, the forecast is unbiased since a one-unit increase in the forecast corresponds to a one-unit increase in the realized return on average. In Panel A, the slope coefficients are significantly different from (less than) one, indicating that the forecasts are biased and overstate the time-series variations in the expected return, since a one-unit increase in the forecast is associated with a less-than-one-unit increase in the realized return on average.

We obtain weaker, yet still significant, results when using forecasts from Random Forest (Panel B) and Neural Network (Panel C). For example, the coefficients are around 0.60 with t-statistics around 7.00 for Random Forest, and are less than 0.20 with t-statistics less than 4.00 for the Neural Network. In addition, the adjusted R^2s display the same order, with the Transformer having the largest value and the Neural Network having the smallest R^2.

The third, sixth, and ninth columns of this table present results from multiple regressions that combine simple average past returns and machine learning forecasts. It consistently shows that the former is insignificant, whereas the latter is highly significant. Overall, Table 1 shows that all three machine learning methods can predict future market returns

using only past returns, with Transfomer performing the strongest, followed by Random Forest and Neural Network. However, the forecasts are also biased, with the Transformer having the smallest bias and the Neural Network having the largest bias.

4.2.2. Market Timing

To evaluate the economic significance of these baseline forecasts, we conduct market timing analysis. Specifically, we either invest 100% in the market when the forecasted returns are positive, or invest 100% in the risk-free asset when the forecasted returns are negative.[1]

Table 2 reports the summary statistics of the timing strategies. As benchmarks, we consider similar timing strategies that use past average returns as signals. We also include the performance of the buy-and-hold (BH) strategy for comparison. In our sample, the buy-and-hold strategy yields 8.27% per annum with an annualized standard deviation of 16.8% and an annualized Sharpe ratio of 0.49. It also displays slightly negative skewness and large positive kurtosis.

The timing strategies based on the three forecasts of the Transformer deliver significantly stronger performance. For example, using the forecasts of the 5-day block input, the timing strategy delivers an average return of 14.0% per annum with a standard deviation of 12.3% and a Sharpe ratio of 1.13. In contrast, the corresponding benchmark timing strategy, which uses the average return in the past five days, yields 10.6% per annum with a standard deviation of 11.0% and a Sharpe ratio of 0.96. The difference between the two strategies is highly significant. In the other two cases, the relative performance of the Transformer forecasts is even stronger compared to their respective benchmarks. For example, the forecasts of 60-day block input deliver an average return of 14.0% per annum and a Sharpe ratio of 1.16, whereas the corresponding benchmark that uses the last 60-day average return as timing signals only gains 8.47% per annum on average with a Sharpe ratio of merely 0.77, both much lower than the former. In terms of skewness and kurtosis, as expected, all the timing

[1]We also conduct alternative market timing allowing shorting the market when the forecasted returns are negative. Results, not tabulated, are similar.

strategies have positive skewness and much larger kurtosis.

The last column reports the success rates, which are measured as the percentage of trading days when the timing strategies generate either positive excess returns when the realized excess return is positive or zero excess returns when the realized excess returns are negative. The three timing strategies based on the Transformer forecasts yield success rates around 55%, while their benchmark strategies succeed about 53% during the whole sample period.

Again, similar albeit slightly weaker results are observed for the forecasts from Random Forest and Neural Network. For example, using the forecasts from random forest with 60-day block input, the timing strategy delivers an average return of 12.4% per annum and a Sharpe ratio of 0.95 in contrast to the benchmark strategy with an average return of 9.19% per annum and a Sharpe ratio of 0.75.

In the online appendix, we present the attention matrix obtained from the Transformer for a simpler scenario. In this scenario, we do not roll the samples for brevity and use the past five days of return as input. It allows us to have a peek at the attention matrix. The training and validation sample goes from 1926 to 1974, and the testing sample is from 1975 to 2022. We notice that the average attention matrix across layers is not identical in different heads, albeit the difference seems small. Considering the larger magnitude of the slope coefficient in the first table, we believe that the attention mechanisms and the upper-level feed-forward neural network in the Transformer model both contribute to the success of its predictions.

4.3. Post-ML Forecasts

Results in Table 1 suggest that the unadjusted forecasts are biased. We notice their averages differ considerably from the sample average of the realized returns. As discussed in Subsection 3.4, these machine learning methods often use nonlinear estimation, and the forecast generated can be poorly aligned with the actual returns. Therefore, we adopt a post-modeling regression adjustment, recursively regressing the realized returns on the fore-

casts, and use the new forecasts from the regressions as the post-ML forecasts. We then use the post-ML forecasts to conduct further analysis. To level the playing field, we also apply the same post-modeling regression adjustment to the average returns so that they serve as the new benchmarks.

4.3.1. Predictive Regressions

The test is similar to those in Table 1 but with the explanatory variables now being post-ML forecasts. We present the estimation results in Table 3.

Panel A reports the regression results for Transformer, which are similar to those in Table 1. The benchmarks are still insignificant, while the post-ML forecasts are all highly significant. Furthermore, the coefficients are larger than those in Table 1 (0.80 versus 0.70), suggesting they are less biased, even though the t-stats are smaller, suggesting they are noisier, which also reflects in the slightly smaller adjusted R^2.

We notice substantial improvements for Random Forest and Neural Network. The coefficients are now all above 0.80, whereas they are less than 0.20 for the Neural Network in Table 1. For example, the coefficient for the raw forecast with 60-day block input is only 0.08 in Panel C of Table 1, but is 0.72 for the post-ML forecasts.

4.3.2. Out-of-Sample R^2

A metric popularized by Campbell and Thompson (2008b) to measure out-of-sample statistical performance in return prediction is the out-of-sample R-squared (R^2_{OOS}), defined as

$$R^2_{OOS} = 1 - \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)^2}{\sum_{t=1}^{T}(r_t - \bar{r}_t)^2}, \tag{6}$$

where r_t is realized return, $\bar{r}_t$ is the historical average return, and $\hat{r}_t$ is the post-ML forecast. Since r_t is the daily return, we estimate the average return using all available observations

up to the end of the year before t, and update the estimate every year. Table 4 reports the out-of-sample R-squared for all post-ML forecasts and their respective benchmarks, as well as the differences between the two. We evaluate the significance of R^2_{OOS} using Clark and West's (2007) test and the significance of the difference using Diebold and Mariano's (2002) test as the former can not be applied to test non-nested models.

In Panel A, the three post-ML forecasts from Transformer yield R^2_{OOS} of 0.94%, 1.03%, and 0.97%, respectively, for the three input blocks, all of which are highly significant. In contrast, all benchmark forecasts generate negative but close to zero R^2_{OOS}.[2] As a result, the difference in R^2_{OOS} between post-ML forecasts and benchmark forecasts is all substantial and statistically significant. These results are striking, as it is well-known that market return prediction at a daily frequency is much more challenging than similar prediction at a monthly frequency. This is because daily returns have a much lower signal-to-noise ratio than monthly returns, and at monthly frequency, prior literature has documented numerous variables that help prediction. For example, default variables, term spread, and inflation (Welch and Goyal, 2008). Although not directly comparable, Han, Lu, and Zhou (2025) document an out-of-sample R^2 around 0.60% for the sample period from January 1965 to December 2022 using forecast combination with fourteen economic variables, and their improved methods using trends in the economic variables improve R^2_{OOS} to 0.81% and to maximal 1.52% using Neural Network and trends.

As in previous results, R^2_{OOS} are smaller for Random Forest post-ML forecasts but are still significantly larger than the benchmarks. However, Neural Network forecasts generate the smallest R^2_{OOS} and two out of three are no longer significantly different from the benchmarks, even though they are still statistically significant.

[2]We also compute the R^2_{OOS} for the unadjusted forecasts and find that they are large negative numbers, which is at odds with the strong predictability identified in Table 1 and underscores the importance of alignment of the scale of predictions with actual returns. The past returns yield even larger (in magnitude) negative R^2_{OOS}. Kelly, Malamud, and Zhou (2024) also note the discrepancy between negative out-of-sample R^2 and positive average return and Sharpe ratio when predicting market returns.

4.3.3. Mean Squared Prediction Errors

In this subsection, we compare mean squared prediction errors (MSPE) between the post ML forecasts from each ML method and their respective benchmarks. We first examine the decomposition of mean squared prediction error (MSPE) and then the cumulative difference in the squared errors.

Following Theil, Beerens, Tilanus, and De Leeuw (1966) we can decompose MSPE into two components relevant to forecasting errors,

$$\text{MSPE} = \left(\bar{\hat{e}}\right)^2 + Var(\hat{e}), \tag{7}$$

where $\hat{e}$ signifies the forecast error, $\hat{e}_t = \hat{r}_t - r_t$, $\left(\bar{\hat{e}}\right)^2$ is the squared forecast bias, and $Var(\hat{e})$ is the forecast variance. Of the two components, the forecast variance, $Var(\hat{e})$, is of much larger magnitude than the squared forecast bias, $\left(\bar{\hat{e}}\right)^2$.

Figure 1 plots the error decomposition for the three ML methods. For Transformer, all the benchmarks are clustered in the lower right corner, indicating large variance but small squared bias. We also plot the error decomposition for the historical average, which is also clustered with the benchmarks, consistent with zero out-of-sample R^2s for the benchmarks in Tabel 4. In contrast, all three forecasts have much smaller variances and larger squared bias. Since the variance is several orders of magnitude larger than the squared bias, it explains why the Transformer forecasts have much larger out-of-sample R-squared. Also, consistent with Table 4, forecasts from a 20-day block input have the smallest variance, and thus have the largest R^2_{OOS}, followed by forecasts from a 60-day block and a 5-day block.

For Random Forest, the situation is similar; post-ML forecasts have larger squared errors but smaller error variances than the benchmarks, which explains the large out-of-sample R^2 of the forecasts. Compared to Transformer forecasts, Random Forest forecasts have large error variances and thus smaller R^2_{OOS}s.

For Neural Network, the pattern is different. However, its forecasts have smaller error

variance and large squared errors than the benchmarks, similar to the other two ML methods. They have even smaller squared errors but larger error variances than the forecasts of Transformer and Random Forest, which is consistent with their relatively smaller out-of-sample R^2.

Figure 2 plots and compare the cumulative differences in the squared errors for all the ML forecasts and their corresponding benchmarks, which are defined as follows.

$$\text{Squared error difference} = (r_t - \bar{r}^{HA}_{t|t-1})^2 - (r_t - \hat{r}_{t|t-1})^2, \tag{8}$$

If the second term is larger than the first term, then a negative value for the squared error difference suggests that predictions from post-ML methods are inferior to those from using simple historical averages. We find that overall this difference increases over time and notice large positive values at the end of the sample period for the ML forecasts and close to zero values for their respective benchmarks. This is consistent with the reported out-of-sample R_2 in Table 4. Similarly, Transformer forecasts have the largest ending differences, followed by Random Forest and Neural Network forecasts, which have the smallest ending differences.

Furthermore, for Transformer forecasts, the plots of the cumulative differences are mostly upward sloping, indicating the differences constantly increase. There are more fluctuations for Random Forest and Neural Network.

4.3.4. Mean-Variance Analysis

To further assess the economic value of using the three machine learning methods to predict market excess returns using past returns, we conduct mean-variance analysis using the post-ML forecasts. The weights are given as

$$w_t = \frac{\hat{r}_t}{\gamma \hat{\sigma}^2_t}, \tag{9}$$

where $\hat{r}_t$ is the forecast from one of the ML, $\hat{\sigma}_t^2$ are sample variances estimated using all available observations up to the end of the year before t, and updated every year, and γ is the risk aversion coefficient. We set γ to be five and restrict the weight between -1 and 2.

Table 5 reports the performance of these mean-variance strategies. Again, for each type of forecast with a different input block, we use the forecasts from the corresponding past average returns. Panel A shows that the strategies using the post-ML forecasts from Transformer perform much better than the benchmarks. For example, for the benchmark strategy using forecasts of 60-day past returns, the average return is 5.04% per annum with a standard deviation of 13.3% and a Sharpe ratio of 0.33. In contrast, the forecast from Transformer using 60-day block input delivers an average return of 30.2% per annum with a Sharpe ratio of 1.30. Indeed, the performance for the forecasts from the three input blocks is very similar and slightly increases as the input block size increases. On the contrary, the performance of the benchmarks decreases as the window to calculate the average returns increases.

For Random Forest, the pattern is similar: the post-ML forecasts generate superior performance than the benchmarks. The average returns range from 20.7% to 22.7% per annum, and the Sharpe ratios range from 1.13 to 1.25. For the benchmarks, the average returns are between 5.04% and 7.56% per annum, and the Sharpe ratios are from 0.33 to 0.51.

For Neural Network, even though the post-ML forecasts still significantly outperform the respective benchmarks, their performance declines substantially compared to the other two ML methods. For example, the highest average return, achieved by forecasts from a 5-day block, is 17.6% per annum, and the highest Sharpe ratio achieved by the same forecast is 0.90.

4.4. Impacts of Market and Economic Conditions

In this subsection, we explore the economic channels of return predictability for the post-ML forecasts from the three machine learning methods. Specifically, we investigate whether

or not the predictability of the forecasts depends on the market and economic conditions. In particular, we examine how business cycles, sentiment, economic policy uncertainty, and VIX influence the return predictability.

4.4.1. Business Cycles

To examine whether the predictability of the forecasts changes with the business cycle, we separately estimate the out-of-sample R^2 for the expansion and recession periods, which are defined by NBER business cycle dates. The results are reported in Table 6. For all three ML methods, the out-of-sample R^2s for the post-ML forecasts are generally higher during recession periods than during expansion periods. For example, for the forecasts from a 60-day input block with Transformer, the R^2_{OOS} is 0.76% during the expansion period but rises to 1.51% during the recession period, both highly significant.

4.4.2. Sentiment

A large literature has studied the relation between sentiment and market returns, and researchers often document a negative relation (Baker and Wurgler, 2007; Tetlock, 2007; Baker, Wurgler, and Yuan, 2012; Huang, Jiang, Tu, and Zhou, 2015; Da, Engelberg, and Gao, 2015). Therefore, we explore if sentiment influences the predictive power of the forecasts.

Table 7 reports the out-of-sample R-squared under different sentiment regimes. We divide the whole sample period (the available period is from July 1, 1965, to December 29, 2022) into low and high sentiment periods by the median. The patterns we observe are quite different from those associated with the business cycle. For instance, forecasts from Transformer show that higher R^2_{OOS}s are observed during high-sentiment periods than during low-sentiment periods. They are 1.64%, 1.59%, and 1.20%, respectively, for the three types of input features, when sentiment is high, versus 1.12%, 0.61%, and 1.18% when sentiment is low. However, for Random Forest and Neural Network, the opposite is true: R^2_{OOS}s are substantially higher during the low sentiment periods than the high sentiment periods.

Indeed, R^2_{OOS}s for Neural Network are either insignificant or marginally significant in the high sentiment period, but are all significant in the low sentiment period. This discrepancy may reflect the different predictive performance of the Transformer compared to the other two ML methods.

4.4.3. Econnomic Policy Uncertainty

Researchers have shown that economic policy uncertainty (Baker, Bloom, and Davis, 2016) positively affects market volatility but negatively affects market returns (Pastor and Veronesi, 2012; Brogaard and Detzel, 2015). In this subsection, we test whether EPU affects the predictive power of the forecasts. To this end, we divide the available sample period (January 2, 1985 to December 29, 2022) into two periods with different level of EPU by the median.

Table 8 shows yet a different picture. For all three ML methods and all their forecasts, the out-of-sample R^2s are much higher during low EPU periods than high EPU periods. Indeed, the out-of-sample R^2s are mostly insignificant during high EPU periods but are highly significant during low EPU periods, except for Neural Network, for which R^2_{OOS}s are insignificant. These results suggest that at return prediction at high frequency is more challenging when there are more uncertainties in the economy.

4.4.4. VIX

Similarly, we divide the available sample period (January 2, 1990 to December 29, 2022) of VIX into high VIX and low VIX periods using the median and estimate the out-of-sample R^2 separately for each VIX regime. The results are reported in Table 9. Similar to the results for EPU, the out-of-sample R^2s are much higher in the low VIX regime than in the high VIX regime. Interestingly, the benchmarks now have significantly negative R^2_{OOS} during the available sample period, especially for the one based on 5-day average returns.

5. Conclusion

In this paper, we apply a Transformer encoder to predict future daily market returns using past daily returns. For comparison, we also present results using the popular Random Forest and Neural Network algorithms. We find that the predictions from all three machine learning methods are significant in predicting future daily returns. In contrast, a simple average of past daily returns is unable to predict future daily returns. These results suggest that machine learning is beneficial in predicting daily market returns.

We conduct extensive analysis using the post-ML prediction. We find that the post-ML forecast also predicts the next-day return positively with high significance. These forecasts have a substantial improvement in out-of-sample R-squared compared to the relevant benchmark based on historical daily returns. The out-of-sample R-squares also improve when using predictions from Random Forest and Neural Network, but somewhat weaker. The Transformer also performs best in terms of variance of forecast errors and in a mean-variance analysis. These results suggest that either the attention or the more sophisticated neural network in it is helpful.

Future studies can use more input features, such as technical indicators (Sullivan, Timmermann, and White, 1999b; Bajgrowicz and Scaillet, 2012; Han, Zhou, and Zhu, 2016), or forecast at the stock level with stock-level information and then aggregate to the market (Ferreira and Santa-Clara, 2011), or use other elegant neural networks, such as a convolutional neural network applied to stock price images (Jiang et al., 2023).

References

Aleti, S., Bollerslev, T., Siggaard, M., 2025. Intraday market return predictability culled from the factor zoo. Management Science Forthcoming.

Bajgrowicz, P., Scaillet, O., 2012. Technical trading revisited: False discoveries, persistence tests, and transaction costs. Journal of Financial Economics 106, 473–491.

Baker, M., Wurgler, J., 2007. Investor sentiment in the stock market. Journal of Economic Perspectives 21, 129–151.

Baker, M., Wurgler, J., Yuan, Y., 2012. Global, local, and contagious investor sentiment. Journal of Financial Economics 104, 272–287.

Baker, S. R., Bloom, N., Davis, S. J., 2016. Measuring economic policy uncertainty. Quarterly journal of economics 131, 1593–1636.

Baltussen, G., Van Bekkum, S., Da, Z., 2019. Indexing and stock market serial dependence around the world. Journal of Financial Economics 132, 26–48.

Belloni, A., Chernozhukov, V., Hansen, C., 2013. Inference on treatment effects after selection among high-dimensional controls†. The Review of Economic Studies 81, 608–650.

Belloni, A., Chernozhukov, V., Hansen, C., 2014. High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives 28, 29–50.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Brock, W., Lakonishok, J., LeBaron, B., 1992. Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance 47, 1731–1764.

Brogaard, J., Detzel, A., 2015. The asset-pricing implications of government economic policy uncertainty. Management science 61, 3–18.

Campbell, J. Y., Shiller, R. J., 1988. Stock prices, earnings, and expected dividends. The Journal of Finance 43, 661–676.

Campbell, J. Y., Thompson, S. B., 2008a. Predicting excess stock returns out of sample: Can anything beat the historical average? Review of Financial Studies 21, 1509–1531.

Campbell, J. Y., Thompson, S. B., 2008b. Predicting excess stock returns out of sample: Can anything beat the historical average? Review of Financial Studies 21, 1509–1531.

Cao, S., Jiang, W., Yang, B., Zhang, Alan, L., 2023. How to talk when a machine is listening: Corporate disclosure in the age of AI. Review of Financial Studies 36, 3603–3642.

Carhart, M. M., 1997. On persistence in mutual fund performance. The Journal of Finance 52, 57–82.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J., 2018. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal 21, C1–C68.

Clark, T. E., West, K. D., 2007. Approximately normal tests for equal predictive accuracy in nested models. Journal of Econometrics 138, 291–311, 50th Anniversary Econometric Institute.

Cong, L. W., Tang, K., Wang, J., Zhang, Y., 2020. Deep sequence modeling: Development and applications in asset pricing, working paper, Cornell University.

Cong, W., Tang, K., Wang, J., Zhang, Y., 2021. Alphaportfolio: Direct construction through deep reinforcement learning and interpretable AI, working paper, Cornell University.

Da, Z., Engelberg, J., Gao, P., 2015. The sum of all fears: Investor sentiment and asset prices. Review of Financial Studies 28, 1–32.

Da, Z., Hua, J., Hung, C.-C., Peng, L., 2025. Market returns and a tale of two types of attention. Management Science Forthcoming.

Da, Z., Huang, D., Yun, H., 2017. Industrial electricity usage and stock returns. Journal of Financial and Quantitative Analysis 52, 37–69.

De Bondt, W. F. M., Thaler, R. H., 1985. Does the stock market overreact? The Journal of Finance 40, 793–805.

Diebold, F. X., Mariano, R. S., 2002. Comparing predictive accuracy. Journal of Business & Economic Statistics 20, 134–144.

Dong, X., Li, Y., Rapach, D. E., Zhou, G., 2022. Anomalies and the expected market return. Journal of Finance 77, 639–681.

Donick, D., Lera, S. C., 2021. Uncovering feature interdependencies in high-noise environments with stepwise look-ahead decision forests. Scientific Reports 11, 9238.

Ferreira, M. A., Santa-Clara, P., 2011. Forecasting stock market returns: The sum of the parts is more than the whole. Journal of Financial Economics 100, 505–513.

Grinsztajn, L., Oyallon, E., Varoquaux, G., 2022. Why do tree-based models still outperform deep learning on typical tabular data? 35, 507–520.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learnings. Review of Financial Studies 33, 2223–2273.

Han, Y., Lu, Y. J., Zhou, G., 2025. Macro financial trends and market expected returns.

Han, Y., Zhou, G., Zhu, Y., 2016. A trend factor: Any economic gains from using information over investment horizons? Journal of Financial Economics 122, 352–375.

Hastie, T., R. T., Friedman, J., 2009. The Elements of Statistical Learning. Springer.

Hou, K., 2007. Industry information diffusion and the lead-lag effect in stock returns. Review of Financial Studies 20, 1113–1138.

Huang, D., Jiang, F., Tu, J., Zhou, G., 2015. Investor sentiment aligned: A powerful predictor of stock returns. Review of Financial Studies 28, 791–837.

Jiang, J., Kelly, B., Xiu, D., 2023. (re-)imag(in)ing price trends. Journal of Finance 78, 3193–3249.

Kelly, B., Malamud, S., Zhou, K., 2024. The virtue of complexity in return prediction. The Journal of Finance 79, 459–503.

Kelly, B., Xiu, D., 2023. Financial machine learning. Foundations and Trends® in Finance 13, 205–363.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. 3rd International Conference for Learning Representations .

Lewellen, J., 2015. The cross-section of expected stock returns. Critical Finance Review 4, 1–44.

Li, B., Yan, X., Rossi, A., Zheng, L., 2025. Real-time machine learning in the cross-section of stock returns .

Li, Z., Tang, Y., 2024. Automated volatility forecasting. Management Science forthcoming.

Liu, F., Stentoft, L., 2023. Intraday stock predictability everywhere, available at SSRN: https://ssrn.com/abstract=4496917.

Lo, A. W., MacKinlay, A. C., 1990. An econometric analysis of nonsynchronous trading. Journal of Econometrics 45, 181–211.

Lo, A. W., Mamaysky, H., Wang, J., 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. Journal of Finance 55, 1705–1765.

Masters, T., 1993. Practical neural network recipes in c++. New York: Academic Press .

Mincer, J. A., Zarnowitz, V., 1969. The evaluation of economic forecasts. In: *Economic forecasts and expectations: Analysis of forecasting behavior and performance*, NBER, pp. 3–46.

Newey, W. K., West, K. D., 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation. Econometrica 55, 703–708.

Pastor, L., Veronesi, P., 2012. Uncertainty about government policy and stock prices. The journal of Finance 67, 1219–1264.

Picard, R. R., Berk, K. N., 1990. Data splitting. Am. Statis 44, 140–147.

Ribeiro, M. T., Singh, S., Guestrin, C., 2016. "why should i trust you?" explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Savor, P., Wilson, M., 2013. Asset pricing: A tale of two days. Journal of Financial Economics 113, 171–201.

Shen, Z., Xiu, D., 2025. Can machines learn weak signals? Working paper.

Shwartz-Ziv, R., Armon, A., 2022. Tabular data: Deep learning is not all you need. Information Fusion 81, 84–90.

Sullivan, R., Timmermann, A., White, H., 1999a. Data-snooping, technical trading rule performance, and the bootstrap. Journal of Finance 54, 1647–1691.

Sullivan, R., Timmermann, A., White, H., 1999b. Data-snooping, technical trading rule performance, and the bootstrap. Journal of Finance 54, 1647–1691.

Tetlock, P. C., 2007. Giving content to investor sentiment: The role of media in the stock market. Journal of Finance 62, 1139–1168.

Theil, H., Beerens, G., Tilanus, C., De Leeuw, C. B., 1966. Applied economic forecasting, vol. 4. North-Holland Publishing Company Amsterdam.

Valaitis, V., Villa, A. T., 2024. A machine learning projection method for macro-finance models. Quantitative Economics 15, 145–173.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention Is All You Need, vol. 30.

Wang, Z., 2024. Machine learning for stock return prediction: Transformers or simple neural networks, working paper.

Welch, I., Goyal, A., 2008. A comprehensive look at the empirical performance of equity premium prediction. Review of Financial Studies 21, 1455–1508.

Yáñez, C., Kristjanpoller, W., Minutolo, M. C., 2024. Stock market index prediction using transformer neural network models and frequency decomposition. Neural Computing and Applications 36, 15777–15797.

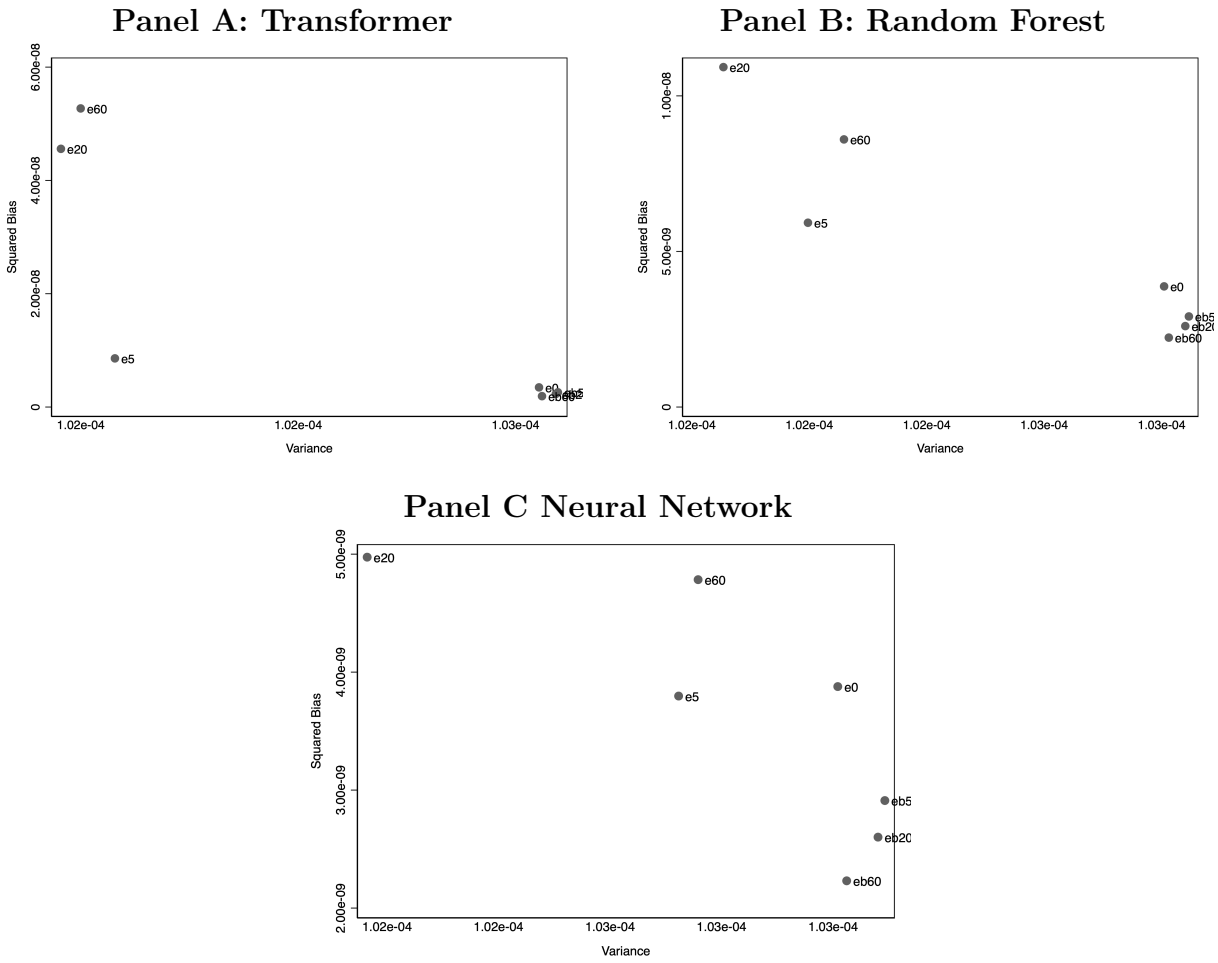
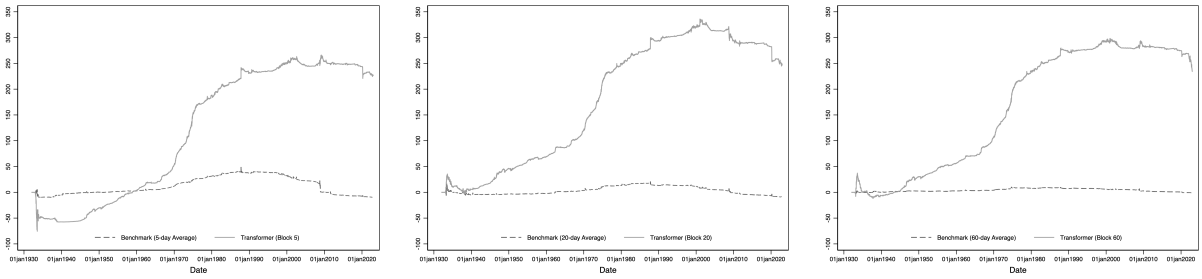


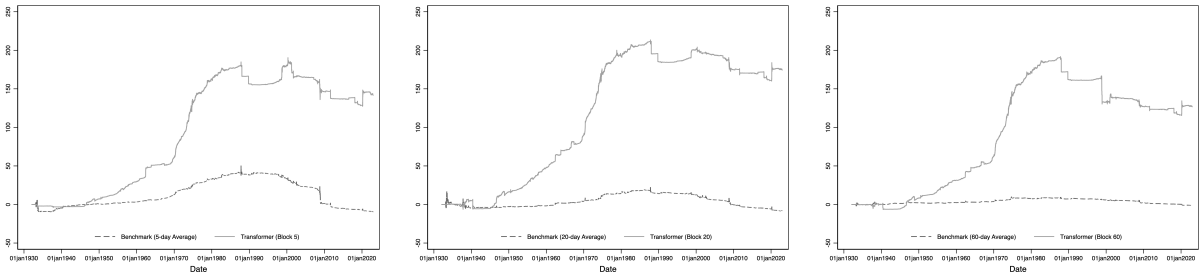
Figure 1: Error Decomposition

Each panel depicts the bias-variance decomposition of the mean squared prediction errors (MSPEs) for Transformer (Panel A), Random Forecast (Panel B), and Neural Nerwork (Panel C). e5, e20, and e60 are for the post-ML forecasts while eb5. eb20, and eb60 are the respective benchmark. The reference is e0, representing the historical average returns suggesting no predictability. The testing period is from January 2, 1932 to December 29, 2022.

Panel A: Transformer



Panel B: Random Forest



Panel C: Neural Network

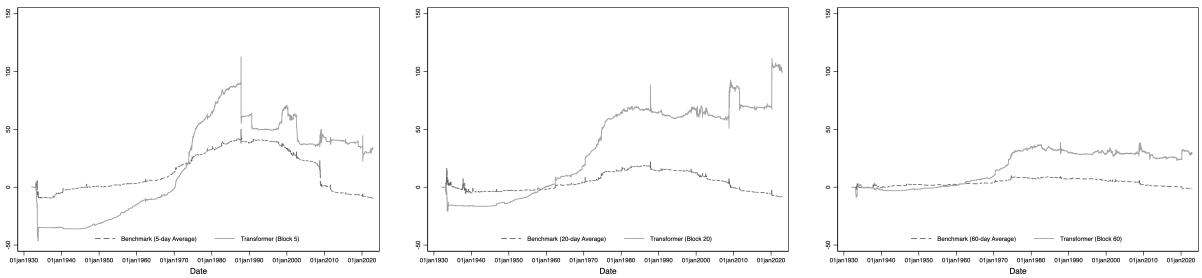


Figure 2: Cumulative Difference in Squared Errors

Each panel depicts the cumulative square prediction error of a competing forecast relative to the historical average benchmark such that

$$\text{squared error difference} = (r_t - \bar{r}^{HA}_{t|t-1})^2 - (r_t - \hat{r}_{t|t-1})^2,$$

where $\bar{r}^{HA}_{t|t-1}$ is the historical mean, r_t is the realized market excess return, and $\hat{r}_{t|t-1}$ is the post-ML forecast based on Transformer (Panel A), Random Forecast (Panel B), and Neural Nerwork (Panel C). The three plots in each panel are for block 5, 20, and 60, respectively. In each plot, the dash line represent the cumulative squared error of the benchmark, i.e., forecasts based on past average daily returns of the same block. The testing period is from January 2, 1932 to December 29, 2022.

Table 1: Regression of Future Returns on Model Forecasted

This table reports the results of regressing observed future returns $(t+1)$ on the forecasted returns (t) directly from Transformer (Panel A), Random Forest (Panel B), and Neural Network (Panel C). fr_5, fr_{20}, and fr_{60} are forecasts from the three ML methods based on blocks of 5-, 20-, and 60-day past returns, and $\bar{r}_5$, $\bar{r}_{20}$, and $\bar{r}_{60}$ are the average returns in the last 5, 20, and 60 days, as the respective benchmarks. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The sample period is from January 02, 1926 to December 29, 2022, with the testing period starting from January 2, 1932.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Panel A: Transformer								
$\widehat{r}_5$	3.25		-0.43						
	(1.17)		(-0.15)						
fr_5		0.70***	0.71***						
		(13.72)	(12.82)						
$\widehat{r}_{20}$				7.12		0.16			
				(1.31)		(0.03)			
fr_{20}					0.71***	0.71***			
					(11.66)	(11.27)			
$\widehat{r}_{60}$							4.47		0.88
							(0.48)		(0.10)
fr_{60}								0.68***	0.69***
								(13.73)	(13.84)
adj. R^2 (%)	0.02	1.10	1.10	0.02	1.03	1.04	-0.00	0.97	1.00

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Panel B: Random Forest								
$\widehat{r}_5$	3.27		1.05						
	(1.18)		(0.37)						
fr_5		0.53***	0.52***						
		(6.49)	(6.32)						
$\widehat{r}_{20}$				7.35		5.78			
				(1.36)		(1.06)			
fr_{20}					0.64***	0.63***			
					(7.60)	(7.57)			
$\widehat{r}_{60}$							4.11		3.34
							(0.44)		(0.36)
fr_{60}								0.54***	0.54***
								(6.40)	(6.40)
adj. R^2 (%)	0.02	0.58	0.58	0.03	0.72	0.74	-0.00	0.54	0.55
	Panel C: Neural Network								
$\widehat{r}_5$	3.27		2.51						
	(1.18)		(0.88)						
fr_5		0.19***	0.19***						
		(2.91)	(2.83)						
$\widehat{r}_{20}$				7.35		6.21			
				(1.36)		(1.15)			
fr_{20}					0.16***	0.17***			
					(3.94)	(3.97)			
$\widehat{r}_{60}$							4.11		4.97
							(0.44)		(0.53)
fr_{60}								0.08***	0.08***
								(3.29)	(3.33)
adj. R^2 (%)	0.02	0.30	0.30	0.03	0.46	0.50	-0.00	0.13	0.14

Table 2: Market Timing Analysis

This table reports the market timing results using the forecasts from the three ML methods: Transformer (Panel A), Random Forest (Panel B), and Neural Network (Panel C). The market timing strategy invests in the market portfolio if the forecasted (excess) returns are positive, and otherwise invests in the risk-free asset. BH: buy-and-hold strtegy. fr_5, fr_{20}, and fr_{60} indicate the timing strategies using forecasts from the three ML methods based on blocks of 5-, 20-, and 60-day past returns, and $\bar{r}_5$, $\bar{r}_{20}$, and $\bar{r}_{60}$ represent the timing strategies using the average returns in the last 5, 20, and 60 days, as the respective benchmarks. *Mean* and *Std Dev* are annualized and in percentage, *Sharpe Ratio* is annualized, *Success Rate* is the percentage of positive returns in the timing strategy. Significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The sample period is from January 02, 1926 to December 29, 2022, with the testing period starting from January 2, 1932.

	Mean	Std Dev	Sharpe Ratio	Skewness	Kurtosis	Success Rate
			Panel A: Transformer			
BH	8.27***	16.8	0.49	-0.15	18.8	.
	(4.78)					
$\bar{r}_5$	10.6***	11.0	0.96	0.87	35.5	53.2
	(9.35)					
fr_5	14.0***	12.3	1.13	0.16	34.5	55.0
	(10.98)					
$\bar{r}_{20}$	8.67***	10.5	0.83	0.02	19.2	53.1
	(8.02)					
fr_{20}	14.8***	12.1	1.22	0.49	35.8	55.0
	(11.84)					
$\bar{r}_{60}$	8.47***	10.9	0.77	0.08	24.2	53.8
	(7.51)					
fr_{60}	14.0***	12.0	1.16	0.51	36.2	55.1
	(11.28)					

	Mean	Std Dev	Sharpe Ratio	Skewness	Kurtosis	Success Rate
			Panel B: Random Forest			
BH	8.39*** (4.88)	16.7	0.50	-0.15	18.9	.
$\bar{r}_5$	10.6*** (9.34)	11.0	0.96	0.85	35.6	53.2
fr_5	12.6*** (9.75)	12.6	1.00	-0.23	19.5	55.3
$\bar{r}_{20}$	8.49*** (7.87)	10.5	0.81	0.01	19.1	53.0
fr_{20}	13.2*** (9.98)	12.8	1.03	0.27	24.6	55.3
$\bar{r}_{60}$	8.19*** (7.24)	11.0	0.75	0.06	24.0	53.7
fr_{60}	12.4*** (9.20)	13.1	0.95	0.23	23.3	55.0
			Panel C: Neural Network			
BH	8.39*** (4.88)	16.7	0.50	-0.15	18.9	.
$\bar{r}_5$	10.6*** (9.34)	11.0	0.96	0.85	35.6	53.2
fr_5	14.2*** (11.24)	12.3	1.16	0.34	33.1	55.3
$\bar{r}_{20}$	8.49*** (7.87)	10.5	0.81	0.01	19.1	53.0
fr_{20}	10.7*** (8.41)	12.3	0.87	0.38	31.9	53.6
$\bar{r}_{60}$	8.19*** (7.24)	11.0	0.75	0.06	24.0	53.7
fr_{60}	7.61*** (6.07)	12.2	0.63	0.02	28.6	52.2

Table 3: Regression of Future Returns on Post-ML Forecasted Returns

This table reports the results of regressing observed future returns $(t+1)$ on the post-ML forecasted returns (t) from Transformer (Panel A), Random Forest (Panel B), and Neural Network (Panel C). To generate post-ML forecasts, at the end of each year we recursively regress the observed returns on the forecasts from the ML, and generate forecasts for the next year. $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ are post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns. For the benchmarks, we similarly regress the observed future returns on the past average returns each year and generate forecasts for the enxt year. $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ are the respective forecasts using the average returns in the last 5, 20, and 60 days. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The sample period is from January 02, 1926 to December 29, 2022, with the testing period starting from January 2, 1932.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Panel A: Transformer								
$\widehat{r}_5$	26.3		-2.76						
	(0.85)		(-0.09)						
$\widehat{fr}_5$		0.80***	0.80***						
		(8.96)	(8.57)						
$\widehat{r}_{20}$				28.3		-28.2			
				(1.00)		(-0.97)			
$\widehat{fr}_{20}$					0.82***	0.84***			
					(12.09)	(11.38)			
$\widehat{r}_{60}$							-0.18		-74.5
							(-0.00)		(-1.55)
$\widehat{fr}_{60}$								0.81***	0.83***
								(13.52)	(13.18)
adj. R^2 (%)	0.01	0.96	0.96	0.01	1.08	1.09	-0.00	1.03	1.05

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	Panel B: Random Forest								
$\widehat{r}_5$	27.0		-14.9						
	(0.89)		(-0.48)						
$\widehat{fr}_5$		0.88***	0.90***						
		(6.32)	(6.19)						
$\widehat{r}_{20}$				29.3		12.7			
				(1.05)		(0.45)			
$\widehat{fr}_{20}$					0.88***	0.87***			
					(7.85)	(7.76)			
$\widehat{r}_{60}$							-2.31		-48.8
							(-0.05)		(-1.02)
$\widehat{fr}_{60}$								0.80***	0.81***
								(6.49)	(6.49)
adj. R^2 (%)	0.01	0.56	0.56	0.01	0.70	0.70	-0.00	0.52	0.53
	Panel C: Neural Network								
$\widehat{r}_5$	27.0		17.7						
	(0.89)		(0.55)						
$\widehat{fr}_5$		0.61*	0.60*						
		(1.80)	(1.74)						
$\widehat{r}_{20}$				29.3		24.1			
				(1.05)		(0.85)			
$\widehat{fr}_{20}$					0.95***	0.94***			
					(3.29)	(3.25)			
$\widehat{r}_{60}$							-2.31		-16.3
							(-0.05)		(-0.33)
$\widehat{fr}_{60}$								0.72***	0.72***
								(2.78)	(2.77)
adj. R^2 (%)	0.01	0.17	0.17	0.01	0.37	0.37	-0.00	0.09	0.09

Table 4: Out-of-Sample R-Square

This table reports the out-of-sample R^2 using the post-ML forecasts, which are computed using the historical average as the reference. At the end of each year, the historical average is computed using all past daily returns and used for the next year. We also report the difference (ΔR^2_{OOS}) in the out-of-sample R^2 between the post-ML forecasts and the corresponding benchmarks for the three ML methods: Transformer (Panel A), Random Forest (Panel B), and Neural Network (Panel C). $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ indicate results for post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns, respectively. We test the statistical significance of R^2_{OOS} using Clark and West (2007) test and of the differences using Diebold and Mariano (2002) test. $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ represent the respective benchmark results. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The sample period is from January 02, 1926 to December 29, 2022, with the testing period starting from January 2, 1932.

	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}
	Panel A: Transformer		**Panel B: Random Forest**		**Panel C: Neural Network**	
$\widehat{r}_5$	-0.04		-0.04		-0.04	
	(1.15)		(1.19)		(1.19)	
$\widehat{fr}_5$	0.94***	0.98***	0.59***	0.63***	0.14**	0.18
	(9.01)	(3.27)	(7.24)	(3.19)	(2.15)	(0.53)
$\widehat{r}_{20}$	-0.04		-0.03		-0.03	
	(1.34)		(1.39)		(1.39)	
$\widehat{fr}_{20}$	1.03***	1.06***	0.72***	0.76***	0.41***	0.44*
	(11.46)	(4.39)	(9.22)	(3.66)	(3.22)	(1.65)
$\widehat{r}_{60}$	-0.01		-0.01		-0.01	
	(0.73)		(0.69)		(0.69)	
$\widehat{fr}_{60}$	0.97***	0.98***	0.53***	0.53***	0.12***	0.13
	(12.66)	(4.89)	(8.38)	(2.95)	(3.04)	(1.16)

Table 5: Mean-Variance Analysis

This table reports performance of the mean-variance (MV) strategies based on the post-ML forecasts for Transformer (Panel A), Random Forecast (Panel B), and Neural Nerwork (Panel C). We compute the weights using the post-ML forecasts as the expected excess returns and variancees estimated recursively each year. The relative risk aversion coefficient is set to 5 and weights are restricted between -100% and 200%. fr_5, fr_{20}, and fr_{60} indicate the MV strategies using forecasts from the three ML methods based on blocks of 5-, 20-, and 60-day past returns, and $\bar{r}_5$, $\bar{r}_{20}$, and $\bar{r}_{60}$ represent the corresponding MV benchmark strategies. *Mean* and *Std Dev* are annualized and in percentage, *Sharpe Ratio* is annualized, Significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The sample period is from January 02, 1926 to December 29, 2022, with the testing period starting from January 2, 1932.

	Mean	Std Dev	Sharpe Ratio	Skewness	Kurtosis
	Panel A: Transformer				
$\widehat{r}_5$	7.56***	16.4	0.50	-17.9	21.9
	(4.87)				
$\widehat{fr}_5$	27.7***	22.2	1.24	-24.0	22.7
	(11.97)				
$\widehat{r}_{20}$	7.56***	15.5	0.44	-14.7	21.9
	(4.24)				
$\widehat{fr}_{20}$	30.2***	22.9	1.29	-24.0	22.7
	(12.47)				
$\widehat{r}_{60}$	5.04***	13.3	0.33	-14.7	21.9
	(3.16)				
$\widehat{fr}_{60}$	30.2***	22.4	1.30	-24.0	22.7
	(12.62)				

	Mean	Std Dev	Sharpe Ratio	Skewness	Kurtosis
	Panel B: Random Forest				
$\widehat{r}_5$	7.56*** (4.93)	16.5	0.51	-17.9	21.9
$\widehat{fr}_5$	22.7*** (11.16)	18.7	1.15	-16.6	21.9
$\widehat{r}_{20}$	7.56*** (4.38)	15.6	0.45	-14.7	21.9
$\widehat{fr}_{20}$	22.7*** (12.12)	19.1	1.25	-16.6	21.9
$\widehat{r}_{60}$	5.04*** (3.25)	13.3	0.33	-14.7	21.9
$\widehat{fr}_{60}$	20.2*** (10.93)	18.6	1.13	-17.9	21.9
	Panel C: Neural Network				
$\widehat{r}_5$	7.56*** (4.93)	16.5	0.51	-17.9	21.9
$\widehat{fr}_5$	17.6*** (8.76)	19.3	0.90	-18.6	21.9
$\widehat{r}_{20}$	7.56*** (4.38)	15.6	0.45	-14.7	21.9
$\widehat{fr}_{20}$	12.6*** (6.56)	20.1	0.68	-17.9	22.7
$\widehat{r}_{60}$	5.04*** (3.25)	13.3	0.33	-14.7	21.9
$\widehat{fr}_{60}$	10.1*** (4.57)	18.7	0.47	-16.6	22.7

Table 6: Out-of-Sample R-Square Under Business Cycle

This table reports the out-of-sample R-Square under different stages of Business Cycle. We divided the available sample period into periods of Expansion and Recession. $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ are post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns, and $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ are the respective forecasts using the average returns in the last 5, 20, and 60 days. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The entire sample period is from July 1, 1926 to December 29, 2022. The testing period starting from January 2, 1932.

	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}
	Panel A: Expansion					
	Panel A1 Transformer		Panel A2 Random Forest		Panel A3 Neural Network	
$\widehat{r}_5$	-0.05		-0.05		-0.05	
	(1.06)		(1.04)		(1.04)	
$\widehat{fr}_5$	0.84***	0.89**	0.59***	0.65***	-0.03	0.02
	(7.35)	(2.56)	(7.52)	(3.22)	(1.14)	(0.05)
$\widehat{r}_{20}$	-0.06		-0.06		-0.06	
	(1.09)		(1.10)		(1.10)	
$\widehat{fr}_{20}$	1.08***	1.14***	0.59***	0.65***	0.05**	0.11
	(11.30)	(4.65)	(8.61)	(2.93)	(1.97)	(0.45)
$\widehat{r}_{60}$	-0.03		-0.03		-0.03	
	(0.10)		(0.09)		(0.09)	
$\widehat{fr}_{60}$	0.76***	0.79***	0.38***	0.41*	0.11***	0.14
	(10.98)	(3.53)	(7.07)	(1.96)	(3.10)	(1.29)
	Panel B: Recession					
	Panel B1 Transformer		Panel B2 Random Forest		Panel B3 Neural Network	
$\widehat{r}_5$	-0.02		-0.01		-0.01	
	(0.53)		(0.61)		(0.61)	
$\widehat{fr}_5$	1.20***	1.22**	0.57***	0.58	0.56**	0.57
	(5.25)	(2.05)	(3.04)	(1.24)	(2.40)	(0.98)
$\widehat{r}_{20}$	0.02		0.03		0.03	
	(0.79)		(0.85)		(0.85)	
$\widehat{fr}_{20}$	0.89***	0.87	1.05***	1.02**	1.30**	1.27*
	(4.24)	(1.50)	(4.52)	(2.20)	(2.55)	(1.76)
$\widehat{r}_{60}$	0.06		0.05		0.05	
	(0.92)		(0.87)		(0.87)	
$\widehat{fr}_{60}$	1.51***	1.46***	0.88***	0.83**	0.14	0.09
	(6.52)	(3.43)	(4.63)	(2.37)	(1.17)	(0.33)

Table 7: Out-of-Sample R-Square Under Sentiment

This table reports the out-of-sample R-Square under different stages of Sentiment. We divided the available sample period into periods of Low Sentiment and High Sentiment using the median of Sentiment. $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ are post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns, and $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ are the respective forecasts using the average returns in the last 5, 20, and 60 days. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The entire sample period is from July 1, 1926 to December 29, 2022. The testing period starts from July 1, 1965 due to availability of Sentiment data.

	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}
	Panel A: Low Sentiment					
	Panel A1 Transformer		Panel A2 Random Forest		Panel A3 Neural Network	
$\widehat{r}_5$	-0.19		-0.19		-0.19	
	(-0.23)		(-0.20)		(-0.20)	
$\widehat{fr}_5$	1.12***	1.31***	0.84***	1.03***	0.58***	0.77*
	(7.68)	(2.86)	(5.10)	(2.58)	(2.98)	(1.67)
$\widehat{r}_{20}$	-0.04		-0.04		-0.04	
	(0.45)		(0.48)		(0.48)	
$\widehat{fr}_{20}$	0.61***	0.66	0.85***	0.90**	0.92**	0.96
	(5.99)	(1.43)	(5.78)	(2.27)	(2.51)	(1.59)
$\widehat{r}_{60}$	-0.04		-0.04		-0.04	
	(-0.07)		(-0.10)		(-0.10)	
$\widehat{fr}_{60}$	1.18***	1.22***	0.90***	0.93**	0.19*	0.23
	(8.86)	(3.62)	(6.09)	(2.55)	(1.90)	(1.01)
	Panel B: High Sentiment					
	Panel B1 Transformer		Panel B2 Random Forest		Panel B3 Neural Network	
$\widehat{r}_5$	0.02		0.01		0.01	
	(1.20)		(1.18)		(1.18)	
$\widehat{fr}_5$	1.64***	1.62***	0.30***	0.29	-0.12	-0.13
	(7.61)	(4.00)	(3.01)	(0.68)	(0.71)	(-0.13)
$\widehat{r}_{20}$	-0.10		-0.10		-0.10	
	(0.23)		(0.25)		(0.25)	
$\widehat{fr}_{20}$	1.59***	1.69***	0.47***	0.57	0.20*	0.30
	(7.15)	(3.89)	(4.35)	(1.59)	(1.73)	(0.67)
$\widehat{r}_{60}$	-0.01		-0.01		-0.01	
	(0.39)		(0.39)		(0.39)	
$\widehat{fr}_{60}$	1.20***	1.22***	0.06***	0.07	0.09*	0.10
	(7.34)	(3.35)	(3.11)	(0.18)	(1.95)	(0.45)

Table 8: Out-of-Sample R-Square Under Economic Policy Uncertainty

This table reports the out-of-sample R-Square under different stages of Economic Policy Uncertainty. We divided the available sample period into periods of Low EPU and High EPU using the median of EPU. $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ are post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns, and $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ are the respective forecasts using the average returns in the last 5, 20, and 60 days. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The entire sample period is from July 1, 1926 to December 29, 2022. The testing period starts from January 2, 1985 due to availability of EPU data.

	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}
	Panel A: Low EPU					
	Panel A1 Transformer		Panel A2 Random Forest		Panel A3 Neural Network	
$\widehat{r}_5$	-0.23		-0.24		-0.24	
	(-0.33)		(-0.36)		(-0.36)	
$\widehat{fr}_5$	0.41***	0.64	0.45***	0.69**	0.01	0.26
	(4.04)	(1.48)	(2.93)	(2.20)	(1.32)	(0.71)
$\widehat{r}_{20}$	-0.21		-0.22		-0.22	
	(-1.02)		(-1.08)		(-1.08)	
$\widehat{fr}_{20}$	0.65***	0.85*	0.17**	0.39*	-0.20	0.02
	(4.12)	(1.93)	(2.23)	(1.73)	(0.90)	(0.06)
$\widehat{r}_{60}$	-0.10		-0.10		-0.10	
	(-1.21)		(-1.28)		(-1.28)	
$\widehat{fr}_{60}$	0.27***	0.37	-0.48	-0.38	0.03	0.14
	(3.59)	(0.95)	(0.41)	(-0.86)	(1.57)	(0.68)
	Panel B: High EPU					
	Panel B1 Transformer		Panel B2 Random Forest		Panel B3 Neural Network	
$\widehat{r}_5$	-0.45		-0.46		-0.46	
	(-1.56)		(-1.57)		(-1.57)	
$\widehat{fr}_5$	-0.01*	0.44	-0.60	-0.14	-0.62	-0.16
	(1.73)	(0.97)	(-0.37)	(-0.32)	(-0.04)	(-0.19)
$\widehat{r}_{20}$	-0.20		-0.20		-0.20	
	(-0.94)		(-0.90)		(-0.90)	
$\widehat{fr}_{20}$	-0.57	-0.36	-0.45	-0.25	0.45	0.65
	(0.54)	(-0.78)	(-0.31)	(-0.64)	(1.53)	(0.99)
$\widehat{r}_{60}$	-0.07		-0.07		-0.07	
	(-0.81)		(-0.80)		(-0.80)	
$\widehat{fr}_{60}$	-0.36	-0.29	-0.48	-0.41	-0.07	0.00
	(1.38)	(-0.91)	(-0.27)	(-1.20)	(0.58)	(0.00)

Table 9: Out-of-Sample R-Square Under VIX

This table reports the out-of-sample R-Square under different stages of VIX. We divided the available sample period into periods of Low VIX and High VIX using the median of VIX. $\widehat{fr}_5$, $\widehat{fr}_{20}$, and $\widehat{fr}_{60}$ are post-ML forecasts based on blocks of 5-, 20-, and 60-day past returns, and $\widehat{r}_5$, $\widehat{r}_{20}$, and $\widehat{r}_{60}$ are the respective forecasts using the average returns in the last 5, 20, and 60 days. Newey and West (1987) robust t-statistics are in parentheses and significance at the 1%, 5%, and 10% levels is given by an ***, and **, and an *, respectively. The entire sample period is from July 1, 1926 to December 29, 2022. The testing period starts from January 2, 1990 due to availability of VIX data.

	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}	R^2_{OOS}	ΔR^2_{OOS}
	Panel A: Low VIX					
	Panel A1 Transformer		Panel A2 Random Forest		Panel A3 Neural Network	
$\widehat{r}_5$	-0.40**		-0.39**		-0.39**	
	(-2.04)		(-1.98)		(-1.98)	
$\widehat{fr}_5$	0.42***	0.82**	0.32***	0.71***	0.11*	0.50***
	(4.28)	(2.30)	(2.76)	(3.61)	(1.93)	(2.71)
$\widehat{r}_{20}$	-0.26**		-0.27**		-0.27**	
	(-2.16)		(-2.17)		(-2.17)	
$\widehat{fr}_{20}$	0.44***	0.71**	0.36***	0.63***	-0.23	0.04
	(4.26)	(2.11)	(2.85)	(3.04)	(0.72)	(0.15)
$\widehat{r}_{60}$	-0.07		-0.07		-0.07	
	(-1.03)		(-1.08)		(-1.08)	
$\widehat{fr}_{60}$	0.35***	0.42	0.29***	0.35**	-0.29	-0.23
	(3.92)	(1.53)	(2.67)	(2.42)	(-0.28)	(-1.34)
	Panel B: High VIX					
	Panel B1 Transformer		Panel B2 Random Forest		Panel B3 Neural Network	
$\widehat{r}_5$	-0.43*		-0.44*		-0.44*	
	(-1.84)		(-1.88)		(-1.88)	
$\widehat{fr}_5$	-0.13*	0.30	-0.21	0.23	-0.32	0.12
	(1.83)	(0.75)	(0.61)	(0.63)	(0.48)	(0.26)
$\widehat{r}_{20}$	-0.19		-0.19		-0.19	
	(-1.09)		(-1.06)		(-1.06)	
$\widehat{fr}_{20}$	-0.57	-0.39	-0.17	0.01	0.40*	0.59
	(0.68)	(-0.92)	(0.07)	(0.05)	(1.70)	(1.06)
$\widehat{r}_{60}$	-0.08		-0.08		-0.08	
	(-1.08)		(-1.12)		(-1.12)	
$\widehat{fr}_{60}$	-0.46	-0.38	-0.42	-0.34	0.04	0.12
	(1.38)	(-1.24)	(0.03)	(-1.12)	(1.02)	(0.61)

Internet Appendix

This online appendix provides additional results. Figure IA1 shows the flow of the Transformer encoder. Figure IA2 shows the flow of a multiple-layer perceptron neural network. Figure IA3 shows the actual attention matrix obtained in a Transformer estimation.

We present the attention matrix obtained from the Transformer for a simpler scenario where we do not roll the samples for brevity. In this scenario, we use the past five days of return as input. The training and validation sample goes from 1926 to 1974, and the testing sample is from 1975 to 2022. Figure IA3 shows the attention weights for all heads in all layers. The training involves four layers, resulting in a four-by-four matrix, where the first four refer to the four layers and the second four refer to the four heads. We will focus on the lower triangular part, as attention is masked here. The left corner is the attention matrix for the first head in the first layer. Each row refers to the attention weights when we use the past one day, two days, up to five days to predict the return on the next day. In the first row, the attention is one because there is only one day to pay attention to. In the second row, the two colors are similar but not identical, with the second square slightly brighter. In the remaining rows, we observe that the attention weights are also similar but again not identical. These patterns appear to hold for the remaining heads and layers. There are variations among heads and layers, but no particular head or layer exhibits particularly large values.

Table IA1 presents the average of the attention weights. For the second row, the average is 0.4946 and 0.5054 for the two squares, respectively. And for the fifth row, the averages are 0.1960, 0.2017, 0.2048, 0.2009, and 0.1976, respectively.

Figure IA1: Encoder

This figure plots the encoder of the transformer. Inputs are embedded into vectors and receive positional encoding before entering the Transformer blocks or layers. Nx represents lawyers suggesting there are similar blocks behind the one shown.

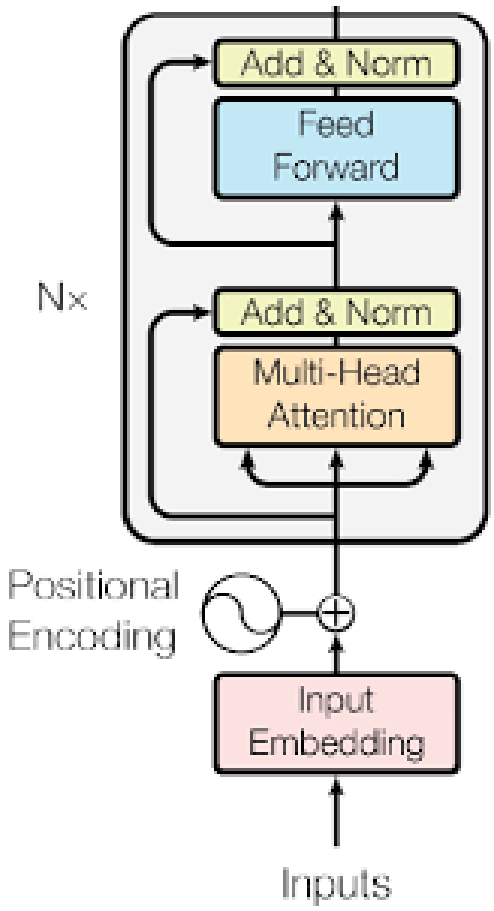


Figure IA2: Neural Network

This figure plot illustrates the nodes and their connections in a "feed-forward" network. The input layer consists of three forecasting features or three nodes. The first hidden layer has four nodes. The second hidden layer has two nodes. The output layer has one node.

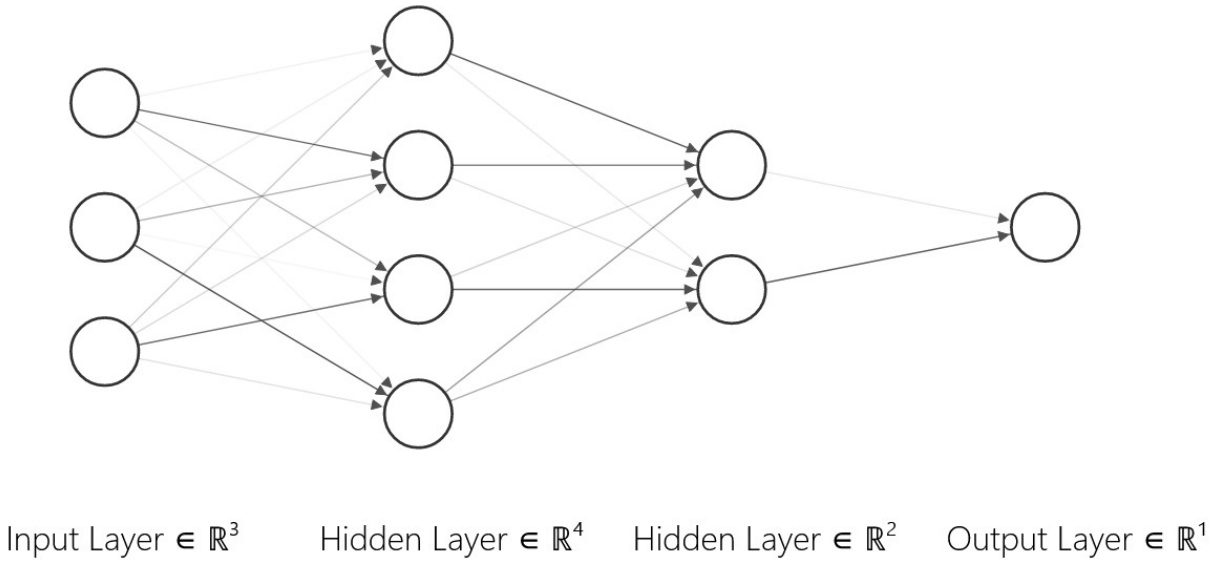


Figure IA3: Attention Weights – Detailed

This plot illustrates the attention weights for each of the four heads in each of the four layers in the Transformer encoder.

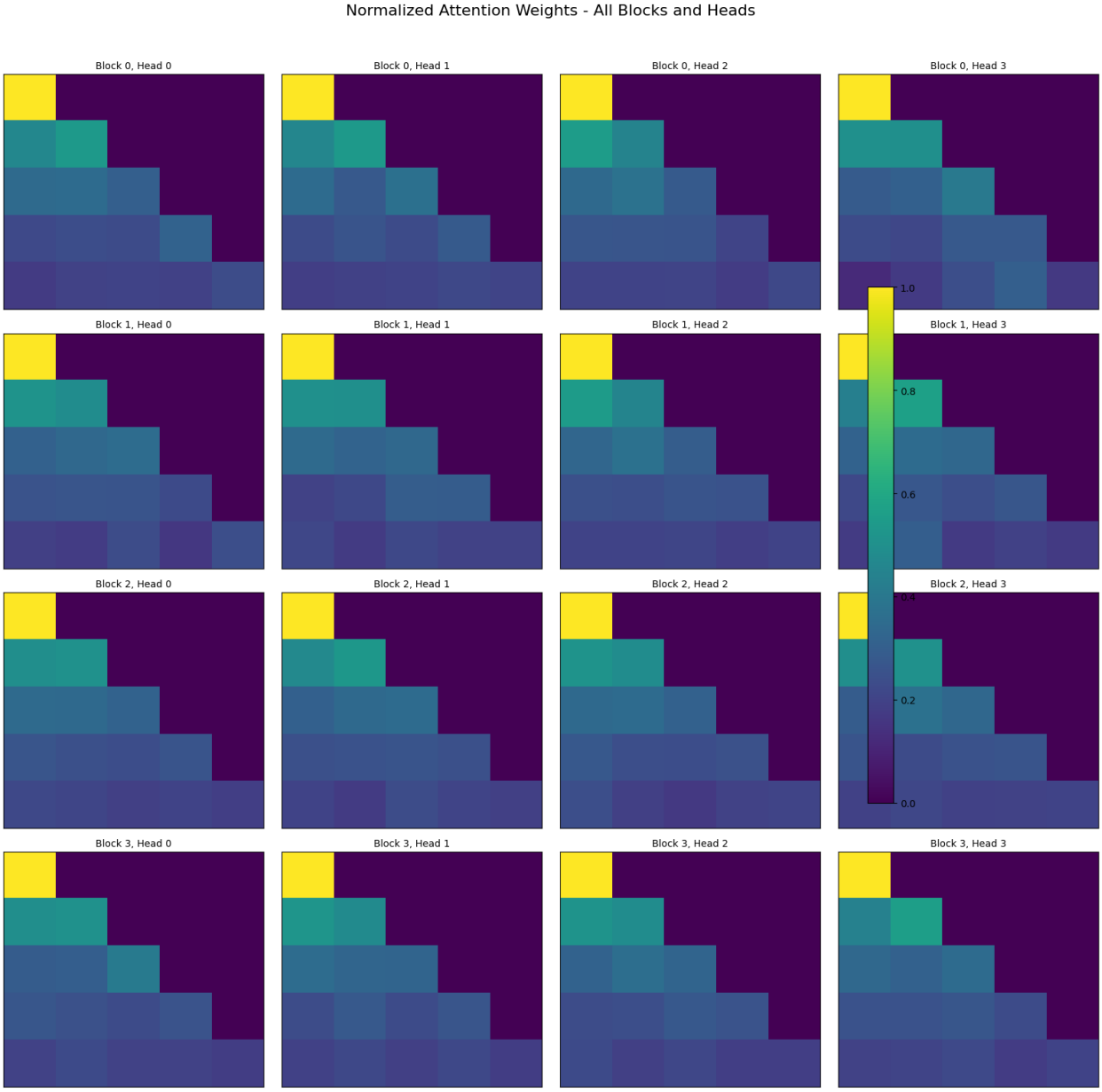


Table IA1: Average Attention Weights

Ret[t-4]	Ret[t-3]	Ret[t-2]	Ret[t-1]	Ret[t]
1.0000				
0.4946	0.5054			
0.3255	0.3376	0.3369		
0.2431	0.2466	0.2520	0.2583	
0.1950	0.2017	0.2048	0.2009	0.1976

The model is trained to use returns from day $t-4$ to day t to predict the return on day $t+1$. The attention mechanism is masked to ensure that each prediction only uses information from the present and prior days. The rows in the table represent different prediction scenarios. For example, the second row shows the average attention weights when using returns from days $t-4$ and $t-3$ to predict the return on day $t-2$; the fourth row corresponds to using returns from days $t-4$ to $t-1$ to predict the return on day t; and the fifth row uses returns from days $t-4$ to t to predict the return on day $t+1$.