Machine Learning Based Long-Short Strategies on Industry Returns Predictions

Firdavs Nasriddinov, Manuel Rodriguez

June 9, 2024

1 Abstract

This paper investigates the effectiveness of machine learning models, particularly Long Short-Term Memory (LSTM) networks, in predicting financial market trends and generating large and significant excess returns. Using a rolling window approach, models were trained on historical data from 1926 to the current year and tested on the subsequent year. Various portfolio strategies, including Max-Min Long-Short, Weighted Long-Short, and Max Long, were evaluated, and models were benchmarked against CAPM, FF3, and FF6. The optimal model, an LSTM trained on value-weighted 49 Industry Portfolio Returns with a 5-day lag and a Max-Min Long-Short strategy, consistently outperformed other configurations, achieving significant annual alphas of 51.57% (1980-1995), 17.43% (1995-2010), and 10.65% (2010-2020), with corresponding Sharpe ratios of 1.49, 0.40, and 0.38. This study highlights the importance of model configuration and diversification in achieving high performance and demonstrates the potential of LSTM models in developing advanced trading strategies that deliver consistent, excess returns across diverse market conditions.

2 Introduction

The hedge fund industry has consistently leveraged a broad array of resources to gain an edge in markets. One of the most recent and transformative applications in this endeavor has been the integration of artificial intelligence and machine learning. The adoption of machine learning techniques has led to a significant evolution in traditional stock arbitrage strategies, which have become less effective. This shift can be attributed to the transition from trading algorithms that relied on human intuition to those that employ highly optimized computational models.

Since the introduction of the Feed Forward Neural Network (FFNN) [1], these networks have been extensively utilized across various domains due to their capability as universal function approximators. FFNNs have demonstrated substantial utility in finance, as evidenced by numerous studies. Long Short Term Memory (LSTM) networks, a specific type of recurrent neural network, have shown superior performance over FFNNs in tasks such as predicting next-day stock movements [6] and conducting financial sentiment analysis for stock market predictions [4]. Additionally, Gradient Boosting Decision Trees (GBDT) [8] have been explored within the same domain, outperforming FFNNs in predicting next-day stock movements [13] and forecasting daily returns of the S&P 500 [10].

The primary objective of this is to develop robust trading strategies based on the next-day returns predictions of machine learning models on value-weighted Fama French Industry Portfolio prior lag returns. We implement and compare three distinct models: Feed Forward Neural Networks (FFNN), Long Short-Term Memory (LSTM) networks, and Gradient Boosting Decision Trees (GBDT). Each model is designed to capture different aspects of market behavior and is evaluated on its ability to generate excess returns, or alpha, beyond traditional benchmarks.

Our strategy involves a comprehensive rolling window approach, where models are trained on historical data from 1926 to the current year and then tested on the subsequent year. This process is repeated annually to assess the models' performance in real-world, dynamic market conditions. We examine various portfolio creation strategies, including Max-Min Long-Short, Weighted Long-Short, and Max Long, to determine the most effective method for leveraging model predictions. Additionally, we benchmark our models against the Capital Asset Pricing Model (CAPM), Fama-French 3-Factor (FF3), and Fama-French 5-Factor + Momentum (FF6) models to evaluate their relative performance.

Through this analysis, we aim to identify machine learning models that consistently deliver superior returns across different market regimes, including bullish and bearish periods. Our findings will contribute to the understanding of how advanced computational techniques can enhance trading strategies and potentially improve financial decision-making.

2.1 Data

Our strategy was based on trading the value-weighted Fama French Industry Portfolios [7]. From the Fama French website [?], The portfolios are created by "...assign[ing] each NYSE, AMEX, and NASDAQ stock to an industry portfolio at the end of June of year t based on its four-digit SIC code at that time. (We use Compustat SIC codes for the fiscal year ending in calendar year t-1. Whenever Compustat SIC codes are not available, we use CRSP SIC codes for June of year t.) We then compute returns from July of t to June of t+1." An example list of the industries is given in Table 1.

Additionally, we use the Fama French Data Library [7] to get the Fama French 3 Factor (FF3) and Fama French 5 Factor + Momentum (FF6) factor returns to compare our models against.

Non-Durable Goods	Durable Goods	Manufacturing	Energy	High Technology
Telecommunications	Retail Shops	Health	Utilities	Other

Table 1: Example of categories: 10-industry portfolio categories.

2.2 Models

2.2.1 *Feed Forward Neural Network (FFNN)*

Introduced in the early 1990s [1], Feed Forward Neural Networks (FFNN) are a foundational architecture in the field of machine learning. These networks consist of an input layer, one or more hidden layers, and an output layer, where the data flows in a single direction from the input to the output. FFNNs are widely used for tasks such as regression and classification due to their simplicity and effectiveness in modeling complex relationships between inputs and outputs. Despite their simplicity, FFNNs serve as a robust baseline model for more complex architectures.

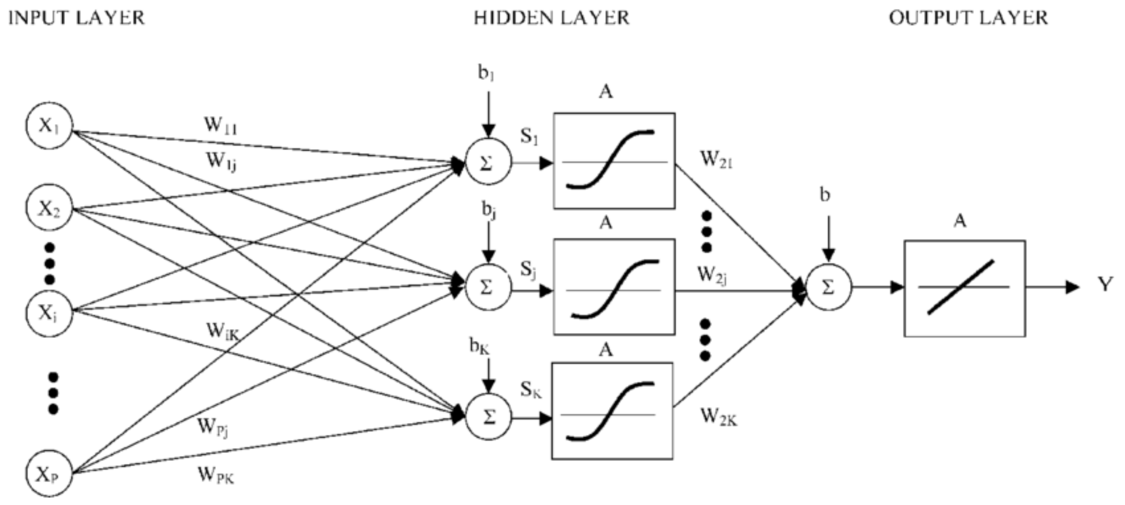


Figure 1: Feed Forward Neural Network (FFNN) simple visualization [12]

FFNN Structure An FFNN consists of the following components:

(1) Input Layer: The input layer receives the raw input features (denoted as $X_1, X_2, \ldots, X_p$). The number of neurons in this layer corresponds to the number of input features.

(2) Hidden Layers: The hidden layers transform the inputs using weighted connections and activation functions. Each neuron in a hidden layer computes a weighted sum of its inputs, adds a bias term, and applies an activation function (e.g., sigmoid, tanh, ReLU).

(3) Output Layer: The output layer produces the final predictions (denoted as Y). The number of neurons in the output layer depends on the nature of the task (e.g., a single neuron for regression, multiple neurons for classification).

FFNN Computation During a single iteration of data passing through a network, at each layer, the FFNN performs the following computations:

(1) Weighted Sum: Each neuron computes a weighted sum of its inputs. For neuron j in layer l, the weighted sum is given by:

$$S_j^{(l)} = \sum_i W_{ij}^{(l-1)} \cdot A_i^{(l-1)} + b_j^{(l)}$$

where $W_{ij}^{(l-1)}$ is the weight connecting neuron i in layer $l-1$ to neuron j in layer l, $A_i^{(l-1)}$ is the activation from the previous layer, and $b_j^{(l)}$ is the bias term.

(2) Activation Function: The weighted sum is passed through an activation function A to introduce non-linearity. Common activation functions include the sigmoid function, tanh, and ReLU. For neuron j in layer l, the activation is:

$$A_j^{(l)} = A(S_j^{(l)})$$

In our training we use the ReLU activation function:

$$A(x) = \max(0, x)$$

(3) Output: In the output layer, the activations are combined to produce the final output, Y.

FFNN Training The FFNN is trained by minimizing a loss function, which measures the discrepancy between the predicted outputs and the actual target values. The loss function is minimized using an optimization algorithm, such as stochastic gradient descent (SGD). In this project, Mean Squared Error (MSE) loss and the Adam optimizer were used. During training, the following steps are performed iteratively:

(1) Forward Propagation: Compute the activations for each layer, from the input layer to the output layer.

(2) Loss Calculation: Calculate the loss based on the difference between the predicted outputs and the actual target values.

(3) Backward Propagation: Compute the gradients of the loss with respect to the weights and biases using backpropagation.

(4) Weight Update: Update the weights and biases using the computed gradients to minimize the loss.

This process continues until the loss converges to a minimum value or a predefined number of iterations is reached.

The simplicity of the FFNN's architecture makes it a powerful baseline model for various tasks, including regression and classification, providing a solid foundation for comparison with more complex models. For our implementation we use the PyTorch `nn.Linear` module.

2.2.2 *Long Short Term Memory (LSTM)*

The LSTM Recurrent Neural Network was introduced in [9] over 20 years ago but since has become a popular model for time series regression due to its ability to take in and learn from sequences in past data. LSTM networks are a type of recurrent neural network (RNN) designed to capture long-term dependencies and mitigate the vanishing gradient problem commonly encountered in traditional RNNs. LSTMs are particularly effective for sequence prediction tasks such as language modeling, time series forecasting, and speech recognition. Here's a detailed explanation of how LSTMs work:

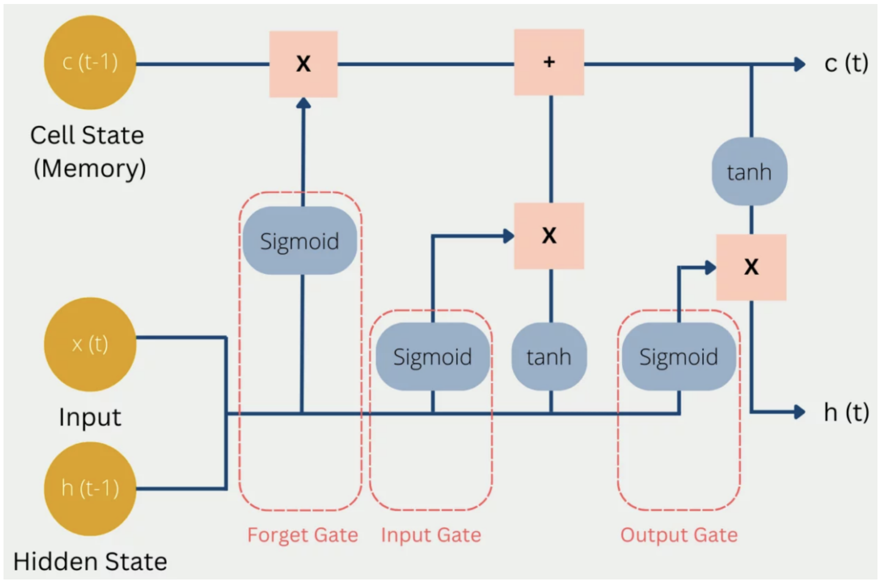


Figure 2: Long Short Term Memory (LSTM) simple visualization of a single unit [2].

LSTM Structure An LSTM unit consists of a cell state, hidden state, and three main gates: the input gate, forget gate, and output gate. These gates control the flow of information, enabling the network to retain or discard information over long sequences.

(1) Cell State (c_t): The cell state is a memory component that carries information across time steps. It is the main pathway through which information flows in the LSTM, with only minor linear interactions. This design helps in retaining long-term dependencies.

(2) Hidden State (h_t): The hidden state is the output of the LSTM unit at each time step. It combines information from the cell state and the current input to produce the output.

(3) Gates:

   (a) Forget Gate (f_t): This gate determines what proportion of the previous cell state should be carried forward. It takes the hidden state from the previous time step (h_{t-1}) and the current input (x_t), and outputs a number between 0 and 1 for each number in the cell state c_{t-1}.

   (b) Input Gate (i_t) and Candidate Cell State ($\tilde{c}_t$): The input gate decides how much of the new information (candidate cell state) should be added to the cell state. The candidate cell state ($\tilde{c}_t$) is created using the current input and the previous hidden state.

   (c) Output Gate (o_t): This gate determines the output of the current cell. It uses the current input and the previous hidden state to compute the output.

LSTM Computation During a single iteration of data passing through a network, the LSTM performs the following computations at each unit:

(1) Forget Gate: Decide what information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

(2) Input Gate and Candidate Cell State: Determine the new information to add to the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

(3) Update Cell State: Update the cell state with the new information.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

(4) Output Gate: Determine the output based on the updated cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

LSTM Training Training an LSTM involves similar steps to training an FFNN, with additional considerations for the sequential nature of the data. The primary goal is to minimize a loss function that measures the discrepancy between the predicted and actual target values. Here are the steps involved in training an LSTM:

(1) Forward Propagation: At each time step, compute the activations and cell states for each LSTM unit, from the input layer through the hidden layers to the output layer. This involves calculating the forget gate, input gate, candidate cell state, cell state, output gate, and hidden state.

(2) Loss Calculation: Calculate the loss based on the difference between the predicted outputs and the actual target values. The Mean Squared Error (MSE) loss is commonly used for regression tasks, while cross-entropy loss is often used for classification tasks.

(3) Backward Propagation Through Time (BPTT): Compute the gradients of the loss with respect to the weights and biases through backpropagation through time. This involves unrolling the LSTM network over the entire sequence and computing gradients at each time step, taking into account the dependencies between time steps.

(4) Weight Update: Update the weights and biases using an optimization algorithm, such as Adam or stochastic gradient descent (SGD). The weights are updated by applying the computed gradients to minimize the loss. This step is performed iteratively for each training batch until the loss converges to a minimum value or a predefined number of iterations is reached.

LSTMs address the vanishing gradient problem by maintaining a more constant error gradient, which allows them to learn long-term dependencies more effectively than traditional RNNs. The gating mechanisms enable the network to decide what information to retain, what to update, and what to output at each time step, making LSTMs powerful for tasks involving sequential data. In our implementation we use the PyTorch `nn.LSTM` module

2.2.3 *Gradient Boosting Decision Tree (GBDT) Regressor*

Introduced in the early 2000s [8], the Gradient Boosting Decision Tree (GBDT) is a powerful ensemble learning technique that combines the predictions of multiple decision trees to produce a more accurate and robust model. The `xgboost` library is widely used for implementing GBDT due to its efficiency and scalability. GBDT is particularly effective for regression and classification tasks.

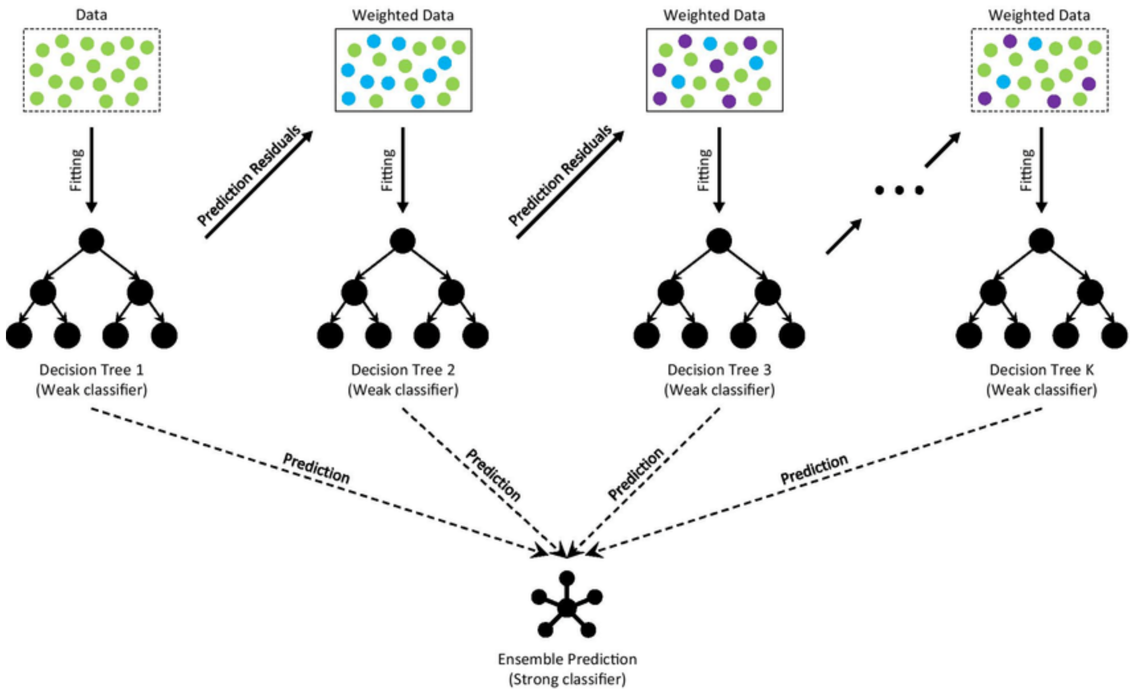


Figure 3: Gradient Boosting Decision Tree (GBDT) visualization [3]

GBDT Structure A GBDT model consists of an ensemble of decision trees, where each tree is trained to correct the errors of its predecessor. The main components of GBDT include:

(1) Base Learners: Each base learner is a decision tree, typically a shallow tree with limited depth to avoid overfitting.

(2) Ensemble Method: The ensemble method involves training multiple decision trees sequentially, where each subsequent tree is trained on the residual errors of the previous trees.

GBDT Computation The GBDT performs the following computations during training and prediction:

(1) Initialization: The model starts with an initial prediction, usually the mean of the target values for regression tasks.

(2) Training Iterations: For each iteration m:

  (a) Residual Calculation: Compute the residual errors between the actual target values and the current predictions.
$$r_i^{(m)} = y_i - \hat{y}_i^{(m-1)}$$
  where y_i is the actual target value and $\hat{y}_i^{(m-1)}$ is the prediction from the previous iteration.

  (b) Fit Base Learner: Train a new decision tree to predict the residuals $r_i^{(m)}$.

  (c) Update Predictions: Update the predictions by adding the weighted predictions of the new tree.
$$\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \nu f_m(x_i)$$
  where ν is the learning rate and $f_m(x_i)$ is the prediction from the m-th tree.

(3) Final Prediction: The final prediction is the sum of the initial prediction and the weighted predictions of all the trees.
$$\hat{y}_i = \hat{y}_0 + \sum_{m=1}^{M} \nu f_m(x_i)$$

GBDT Training The training process for GBDT involves minimizing a loss function through an iterative process. The main steps are:

(1) Initialization: Initialize the model with the mean of the target values.

(2) Iterative Training: For each iteration, compute the residuals, fit a new decision tree to the residuals, and update the model's predictions.

(3) Loss Minimization: The residuals represent the gradients of the loss function with respect to the predictions. By fitting trees to these residuals, GBDT effectively performs gradient descent in the function space.

(4) Hyperparameter Tuning: Optimize hyperparameters such as the number of trees, tree depth, and learning rate to improve the model's performance and prevent overfitting.

Gradient Boosting Decision Trees, implemented using the `xgboost` library, offer a powerful approach to regression tasks by sequentially improving the model through the correction of residual errors. The ensemble nature of GBDT allows it to capture complex patterns and interactions in high dimensional data. This method is powerful because it aims to create a model which reduces the high biases of each weak regressor while maintaining their low variances, thereby creating a reliable and accurate model.

2.2.4 Benchmarks

To evaluate the performance of our machine learning models, we compare them against well-established financial models: the Capital Asset Pricing Model (CAPM), the Fama-French 3-Factor (FF3) model, and the Fama-French 5-Factor + Momentum (FF6) model. These benchmarks provide a solid foundation for understanding the added value of our models in terms of excess returns (alpha) and risk-adjusted performance.

The CAPM is a foundational model that describes the relationship between systematic risk and expected return for assets, particularly stocks. It serves as a baseline for expected returns based on the risk-free rate, the stock's beta, and the expected market return.

The FF3 model expands on CAPM by including two additional factors: the size of companies (small vs. large) and the book-to-market value (value vs. growth). This model helps capture the variations in stock returns that cannot be explained by market risk alone.

The FF6 model further extends the FF3 model by adding three more factors: profitability, investment patterns, and momentum. This comprehensive model provides a more detailed view of the factors influencing stock returns and serves as a rigorous benchmark for evaluating the performance of advanced machine learning models.

Range (Years)	Market Type	Description
1980-1980	↗ Bullish	Market growth in early 1980.
1981-1981	↘ Bearish	High inflation and interest rates.
1982-1986	↗ Bullish	Post-recession recovery and growth.
1987-1987	↘ Bearish	Black Monday market crash.
1988-1989	↗ Bullish	Market rebound after crash.
1990-1990	↘ Bearish	Recession due to Gulf War.
1991-1999	↗ Bullish	Dot-com bubble driven growth.
2000-2002	↘ Bearish	Dot-com bubble burst.
2003-2007	↗ Bullish	Economic growth and low rates.
2008-2008	↘ Bearish	Global financial crisis.
2009-2017	↗ Bullish	Recovery post-2008 crisis.
2018-2018	↘ Bearish	Trade war and global slowdown.
2019-2019	↗ Bullish	Continued economic recovery.
2020-2020	↘ Bearish	COVID-19 pandemic impact.
2021-2021	↗ Bullish	Post-pandemic economic recovery.
2022-2022	↘ Bearish	Inflation and interest rate concerns.
2023-2023	↗ Bullish	Continued economic rebound.

Table 2: Years when the market was either bullish or bearish and a potential reason why for each.

We also distinguish between different market conditions by categorizing years as either bullish (strong market returns) or bearish (weak market returns) in Table 2. This classification allows us to analyze how our models perform under varying economic conditions. For instance, we mark the year 2020 as bearish due to the COVID-19 pandemic's impact on markets, despite the eventual recovery.

By comparing our models' performance against these benchmarks, we aim to determine their ability to generate significant and consistent alpha. We perform a detailed analysis of the models' alphas and betas during bullish and bearish periods to assess their robustness and adaptability to different market environments. This comprehensive evaluation ensures that our findings are grounded in rigorous financial theory and practical market dynamics.

2.3 Motivation

In this study, we aim to explore the potential of machine learning models in improving trading strategies based on the Fama French Industry Portfolios, leveraging their diverse and comprehensive data to develop robust, adaptable, and high-performing investment models. The motivation for this study stems from the desire to enhance trading strategies through the application of advanced machine learning techniques. The Fama French Industry Portfolios provide a robust framework for developing diversified investment strategies, given their comprehensive coverage of various industries and the availability of extensive historical data. These portfolios are not only well-documented but also accessible through exchange-traded funds (ETFs), making them practical for real-world trading applications.

Our decision to utilize the Fama French Industry Portfolios is based on their ability to represent a diverse set of market sectors, which allows for a broad interaction between industry-specific trends and behaviors. This diversity is crucial for creating a well-rounded strategy that can potentially mitigate risks and capitalize on opportunities across different market segments. By leveraging the rich historical data available on Kenneth French's website, we can effectively train and backtest our models, ensuring their reliability and robustness.

For all industry portfolios, the Fama French website has data for equal-weighted and value-weighted. We decided on using value-weighted because value-weighted portfolios give more weight to companies with higher market caps, which more accurately reflects the impact of larger companies on the market and aligns the portfolio more closely with market indices and real-world investment scenarios. This approach tends to reduce overall volatility and offer better risk management as larger companies are generally more stable and less susceptible to sharp declines compared to smaller, more volatile stocks. Moreover, value-weighted portfolios provide a more accurate representation of the economic footprint of industries, as larger companies contribute more significantly to economic activity. Data for equal-weighted portfolio model performance is in appendix.

We chose to focus on three machine learning models—FFNN, LSTM networks, and GBDT regressors—each bringing unique strengths to the table. The FFNN serves as a baseline due to its simplicity and proven effectiveness in financial applications. The LSTM model is selected for its ability to capture temporal dependencies in sequential data, which is particularly relevant for financial time series. The GBDT model is included for its robustness and capability to handle high-dimensional data, often outperforming simpler models in various regression tasks.

Our approach includes a thorough comparison of these models against traditional financial benchmarks like the CAPM, FF3, and FF6 models. We also assess the performance of our models across different market regimes—bullish and bearish—to understand their effectiveness under varying economic conditions. By doing so, we aim to identify machine learning models that not only generate consistent alpha but also adapt well to different market environments, thereby providing a reliable tool for enhancing trading strategies.

3 Related Work

The prediction of next-day stock price movements has been a significant area of research within the field of financial machine learning. One of the pioneering works in this domain is by Fischer and Krauss (2018), who demonstrated the superiority of Long Short-Term Memory (LSTM) networks over traditional feedforward neural networks (FFNN) for predicting next-day stock movements. Their study showed that LSTMs could effectively capture temporal dependencies in financial time series data, leading to improved prediction accuracy compared to FFNNs [6]. Another notable study by Weng, Ahmed, and Megahed (2017) explored the application of Gradient Boosting Decision Trees (GBDT) for next-day stock movement prediction. They found that GBDTs outperformed FFNNs by leveraging the model's ability to handle high-dimensional data and capture complex non-linear relationships between features. This study highlighted the potential of ensemble learning techniques in financial prediction tasks [13]. Furthermore, Eachempati et al. (2019) utilized LSTM networks for financial sentiment analysis, demonstrating how integrating sentiment data from news and social media can enhance the prediction of stock price movements. Their work underscored the importance of incorporating diverse data sources to improve the robustness and accuracy of predictive models in finance [4].

The development of industry trading strategies has also garnered considerable attention in financial research. A seminal work by Fama and French (1997) introduced the Fama French Industry Portfolios, which have become a standard framework for analyzing industry-specific returns. Their methodology involves constructing portfolios based on industry classifications and has been widely adopted in both academic research and practical trading strategies [5]. Recent studies have leveraged machine learning models to enhance industry trading strategies. For instance, Nevasalmi et al. (2020) applied GBDTs to forecast daily returns of the S&P 500 and demonstrated the model's ability to outperform traditional linear models. Their approach involved using a comprehensive set of features, including technical indicators and macroeconomic variables, to capture industry-specific trends and generate more accurate predictions [10]. Additionally, the work by Tsantekidis et al. (2017) on using deep learning models for financial prediction emphasized the effectiveness of LSTM networks in capturing the sequential nature of financial data. They applied their models to various industry sectors and showed that deep learning models could significantly enhance the performance of industry-specific trading strategies compared to conventional approaches [11]. Overall, the integration of advanced machine learning techniques in industry trading strategies has demonstrated substantial potential in generating superior returns. These studies highlight the importance of using diverse and comprehensive datasets, as well as sophisticated model architectures, to capture the intricate dynamics of financial markets and improve the efficacy of trading strategies.

4 Methods

Our methodology involves a systematic rolling window training and testing strategy to rigorously evaluate the performance of our machine learning models. We start by training each model on historical data from 1926 up to, but not including, the current year. The trained model is then tested on data from the current year up to, but not including, the next year. This process is repeated annually, progressively moving the training window forward by one year. This approach mimics real-world conditions where models are continually updated with new data, ensuring their relevance and adaptability over time.

To identify the optimal data and model configurations, we exhaustively test all permutations of the following parameters:

(1) Number of Industries: 5, 10, 12, 17, 30, 48, 49

(2) Model Architecture: FFNN, LSTM, GBDT

(3) Day Lag: 5, 22

(4) Portfolio Strategy Type: Max-Min Long-Short, Weighted Long-Short, Max Long

(5) Industry Data Weighting: Value, Equal

4.1 Data Processing

We preprocess the Fama French Industry Portfolios data to ensure it is suitable for input into our models. Initially, we clean the data by removing any missing values (marked as -99.99 in the dataset) and normalize the valid data points. Each model is provided with the past 5 (week) or 22 (month) days of data to predict the returns of the next day. The dataset is then split into training and testing sets, where the training set spans from 1926 to the year before the testing year. The testing set, comprising the current year, is used solely for evaluating model performance during training epochs without influencing the training process.

4.2 Training

4.2.1 *FFNN*

The FFNN model comprises four linear layers: an input layer, two hidden layers, and an output layer. The input layer has neurons corresponding to the number of industries multiplied by the number of days of lag data. The hidden layers contain 32 and 16 neurons, respectively, while the output layer has neurons corresponding to the number of industries. The model uses Mean Squared Error (MSE) loss, ReLU activation functions, and the Adam optimizer with a learning rate of 0.001. A dropout rate of 0.1 is applied after each linear layer, and the model is trained for 5 epochs.

4.2.2 *LSTM*

The LSTM model consists of a single LSTM layer with 50 hidden units and a subsequent linear output layer. The LSTM layer processes the sequential data, feeding its output through a ReLU activation function before reaching the final linear layer. The model uses MSE loss and the Adam optimizer with a learning rate of 0.001. Training is conducted over 5 epochs.

4.2.3 *GBDT*

The GBDT model employs the `xgboost` library's XGBRegressor, configured for regression tasks using squared error loss. The objective function is set to `reg:squarederror`, reflecting its focus on minimizing the squared error during training.

4.3 Backtesting

Our backtesting evaluates the performance of our models over an extended historical period. We implement a rolling window approach, where models are continuously updated and tested on new data each year, mimicking real-world conditions. Initially, the models are trained on data from 1926 to 1979 and tested on data from 1980. Subsequently, the models are retrained with data extended to include 1980 and tested on data from 1981. This iterative process continues up to 2020, with additional plots and tables provided for data up to 2024 in the appendix. This exclusion accounts for the unique market conditions during and post the COVID-19 pandemic; we will note though that our strategy performs just as well post pandemic.

For each testing year, daily predictions are made using the past n days of data. These predictions inform the creation of portfolios based on one of three strategies:

(1) Max-Min Long-Short: Long the top decile of industries with the highest predicted returns and short the bottom decile with the lowest predicted returns, with weights adjusted so the sum of the weights is zero.

(2) Weighted Long-Short: Long industries predicted to have positive returns and short those predicted to have negative returns, with weights proportional to the predicted returns and adjusted to sum to zero.

(3) Max Long: Long the single industry predicted to have the highest return.

After generating predictions for all testing years, we benchmark our models by calculating alphas, betas, and Sharpe ratios. Linear regression is used to compare model portfolio returns to the FF6 benchmark returns, determining the significance of the alpha and the extent to which our models are explained by the benchmark. To gain deeper insights, we also analyze the performance of our models across different market conditions, categorizing years as bullish or bearish. This allows us to evaluate how our models perform in varying economic environments.

Furthermore, we assess model performance over different time periods: 1980-1995, 1996-2010, and 2011-2020. This segmented analysis helps us understand how the models adapt to changing market dynamics over decades. By comparing alphas, betas, and Sharpe ratios across these periods, we can identify trends and evaluate the consistency of model performance. This comprehensive evaluation ensures a thorough understanding of the robustness and effectiveness of our models across multiple dimensions and market conditions.

5 Results

Our highest alpha producing strategy had configuration that utilized the LSTM model trained on the value-weighted Fama French 49 Industry Portfolio Returns using a 5-day lag, with returns calculated using the Max-Min Long-Short strategy. Let this configuration be labeled as H. To evaluate the performance of our optimal strategy, we compared it to other configurations by varying one parameter at a time and analyzing the results. We provide visualizations of cumulative returns, bar graphs depicting yearly raw returns, and tables summarizing benchmark analysis.

We show our best strategy's returns in blue and the market's returns in black. The green and red regions on the plots represent times of bullish and bearish market movements, respectively. The accompanying bar graphs in Figures 9, 10, 12, and 14 illustrate the raw strategy returns and market returns for each month during the testing period. In the tables, we denote significant alphas with an asterisk (*) next to their corresponding value and compute all values, except for those in Table 9, during the periods from 1980 to 1995, 1995 to 2010, and 2010 to 2020, respectively.

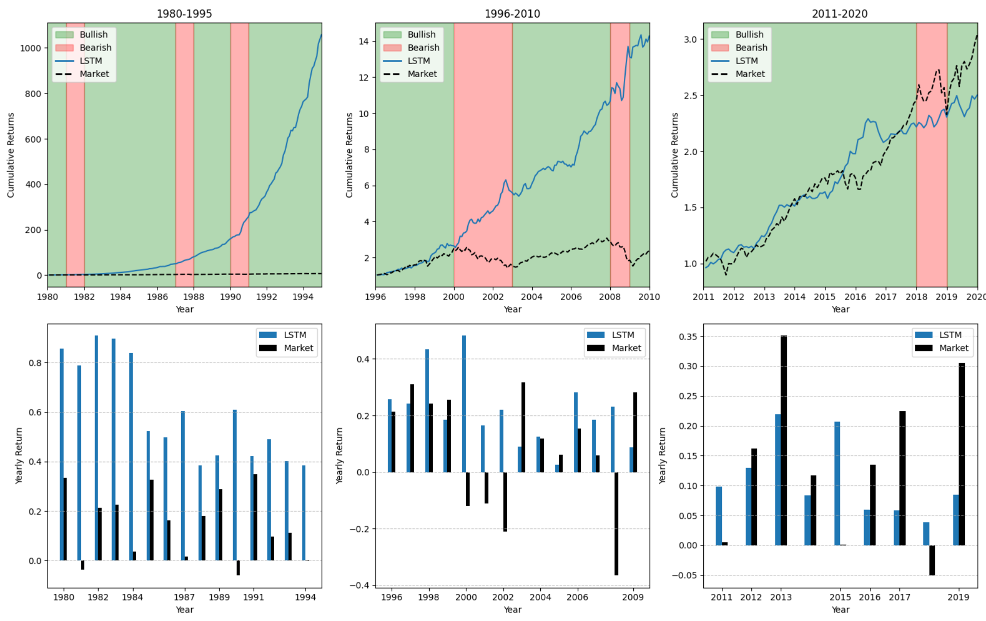


Figure 4: Monthly cumulative returns and annualized raw returns of best strategy in three separate time periods.

	α (Annual)	β	Sharpe Ratio
CAPM	0.10* (28.54)	0	0.67
FF3	0.10* (28.36)	0.01	0.67
FF6	0.09* (27.69)	0.02	0.67

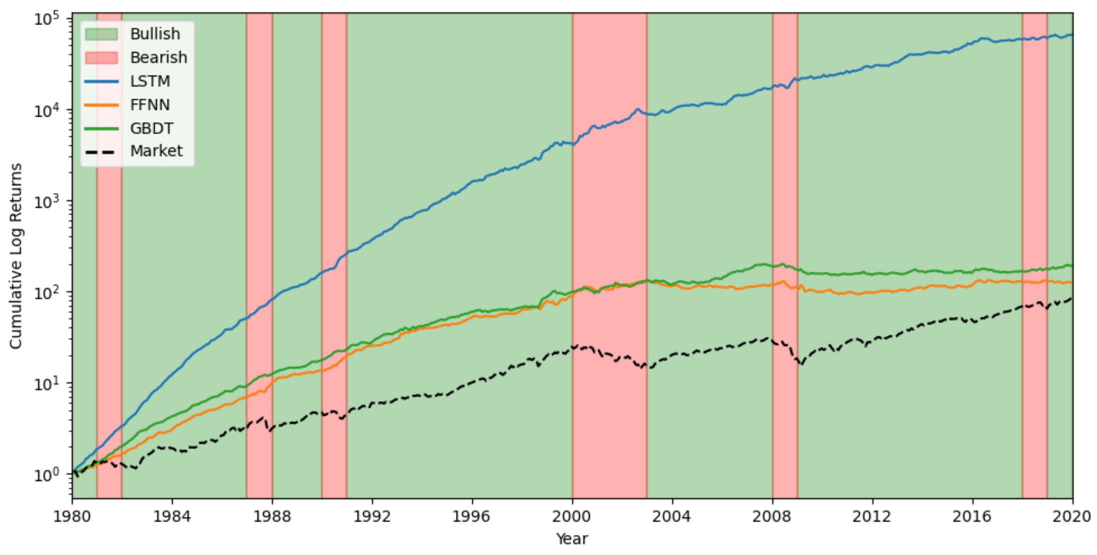
Table 3: Best strategy CAPM, FF3, and FF6 benchmarks. α is daily and in percent. * signifies significant α.

	Period	α (Annual)	β	Sharpe Ratio
Total	1980-1995	0.16* (51.57)	0.01	1.49
	1995-2010	0.06* (17.43)	0.03	0.40
	2010-2020	0.04* (10.65)	0.04	0.38
Bullish	1980-1995	0.15* (50.32)	-0.00	1.05
	1995-2010	0.05* (15.36)	0.03	0.33
	2010-2020	0.04* (11.49)	0.04	0.38
Bearish	1980-1995	0.17* (58.50)	0	0.45
	1995-2010	0.07* (21.03)	0.02	0.26
	2010-2019	0.02 (4.35)	0.05	0.09

Table 4: Best strategy FF6 benchmarks for bullish and bearish markets as well as for different time periods. * signifies significant α.

The strategy performs exceptionally well in the first time period (1980-1995), consistently outperforming the market with raw returns. During this period, it achieves a significant annual alpha of over 50% across all market conditions, with the highest overall Sharpe ratio of 1.49. This indicates a very favorable risk-adjusted return compared to the market. In the second period (1995-2010), the strategy continues to generate significant alpha, though at a reduced level of around 17% annually. Despite this reduction, the Sharpe ratio remains reasonably high at 0.40, indicating a decrease in performance potentially due to more use of similar strategies for trading. In the third period (2010-2020), while the strategy's annual alpha remains significant, it is the lowest among all time periods at approximately 10%, with a Sharpe of ratio of 0.38. This suggests that while the strategy continues to add value, its effectiveness has diminished over time.

Notably, the market beta is consistently near zero across all time periods, benchmarks, and market conditions. This implies that the strategy's returns are largely independent of market movements. Additionally, there is minimal discrepancy in alpha values across the CAPM, FF3, and FF6 benchmarks, suggesting that the factors included in these models do not significantly explain the strategy's alpha.



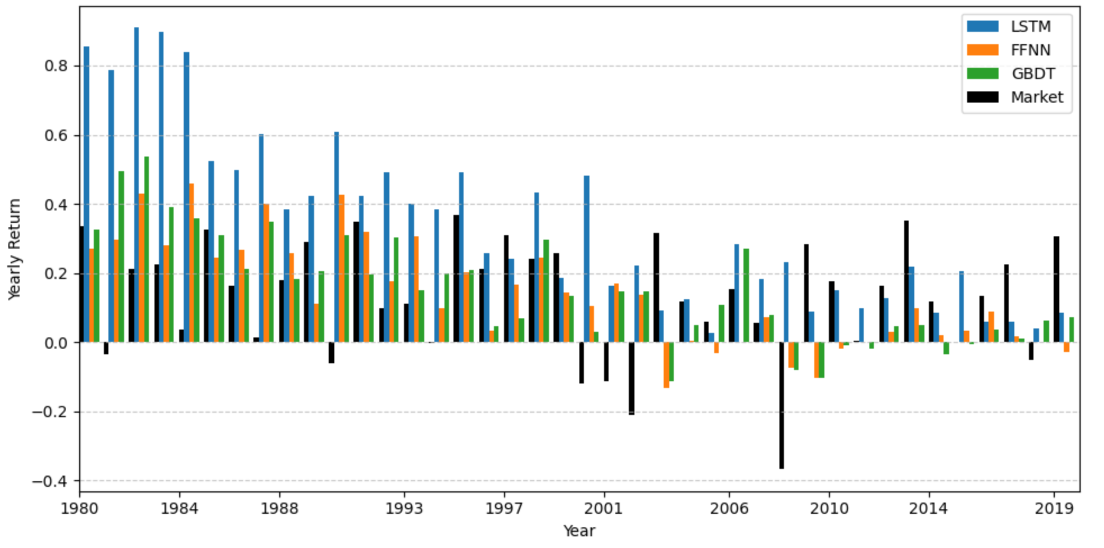


Figure 5: Monthly cumulative returns and raw annual returns for varying model types.

	Period	α (Annual)	β	Sharpe Ratio
LSTM	1980-1995	0.16* (51.57)	0.01	1.49
	1995-2010	0.06* (17.43)	0.03	0.40
	2010-2020	0.04* (10.65)	0.04	0.38
FFNN	1980-1995	0.07* (20.70)	-0.04	0.70
	1995-2010	0.01 (1.60)	0.01	0.05
	2010-2020	0.01* (3.35)	-0.04	0.10
GBDT	1980-1995	0.07* (21.38)	0.05	0.80
	1995-2010	0.02* (4.49)	0.01	0.12
	2010-2020	0.01* (1.66)	0.02	0.09

Table 5: FF6 benchmark for varying model types. * signifies significant α.

The comparison of different model architectures, as depicted in Figure 5, highlights the performance variations between LSTM, FFNN, and GBDT models over several decades. The LSTM model consistently outperforms the other architectures, especially in the earlier period (1980-1995). In the subsequent periods (1995-2010 and 2010-2020), while the LSTM's performance diminishes slightly, it still performs better than the other two model types. This consistent performance across different time periods underscores the robustness and adaptability of the LSTM model in financial market predictions, indicating the LSTM's superior ability to capture complex temporal patterns and generate high risk-adjusted returns.

In comparison, the FFNN and GBDT models exhibit more variable performance. The FFNN model shows a significant annual alpha of 20.70% with a Sharpe ratio of 0.70 in the first period, but its effectiveness decreases considerably in later periods, with only marginally positive alphas and lower Sharpe ratios. Similarly, the GBDT model achieves a 21.38% annual alpha and a Sharpe ratio of 0.80 in the first period, but its performance also wanes over time, with much lower alphas in the subsequent periods. These results suggest that while FFNN and GBDT models can provide some benefits in specific conditions, they lack the consistent performance and robustness demonstrated by the LSTM model.

	Period	α	α (Annual)	β	Sharpe Ratio
49 Industries	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2020	0.04*	10.65	0.04	0.38
48 Industries	1980-1995	0.16*	53.09	0.02	1.37
	1995-2010	0.08*	23.28	0.02	0.48
	2010-2020	0.02*	5.53	0.03	0.20
30 Industries	1980-1995	0.12*	39.05	0.02	1.17
	1995-2010	0.04*	10.07	-0.02	0.21
	2010-2020	0.01*	2.59	-0.02	0.06
17 Industries	1980-1995	0.18*	58.77	-0.03	1.29
	1995-2010	0.08*	21.97	-0.03	0.40
	2010-2020	0.01*	3.68	-0.00	0.09
12 Industries	1980-1995	0.19*	63.97	-0.05	1.43
	1995-2010	0.03*	8.76	-0.01	0.19
	2010-2020	-0.00	-1.03	-0.02	-0.04
10 Industries	1980-1995	0.24*	88.93	-0.06	1.36
	1995-2010	-0.00	-0.39	-0.05	0.01
	2010-2020	0.01	1.63	-0.01	0.04
5 Industries	1980-1995	0.1*	29.82	-0.05	0.75
	1995-2010	0.00	1.3	0.01	0.03
	2010-2020	0.00	1.14	-0.02	0.03

Table 6: FF6 benchmark for varying number of industries in different time periods. * signifies significant α.

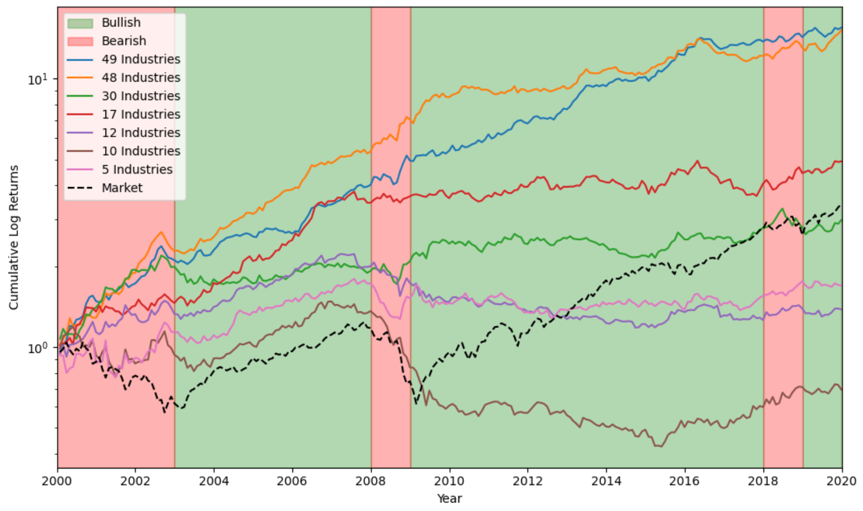
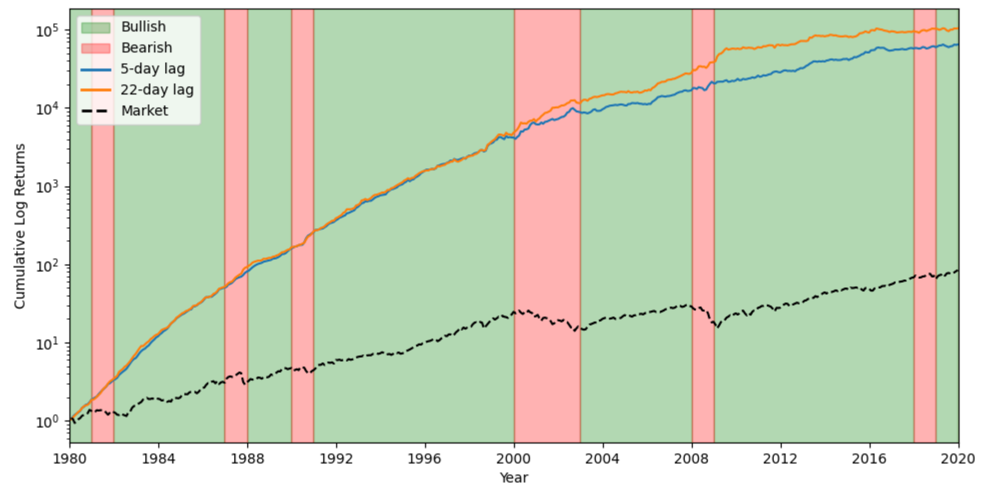


Figure 6: Monthly cumulative returns for varying number of industries.

Across different periods, models trained with a higher number of industries, specifically the 49-industry and 48-industry configurations, demonstrated robust performance with significant alpha and strong Sharpe ratios. This suggests that incorporating a broad range of industries allows the model to capture diverse market dynamics, thereby enhancing its ability to generate returns and manage risk. These configurations consistently yielded high returns, especially in earlier periods (1980-1995), where market inefficiencies could have been more prevalent, allowing well-diversified models to capitalize on a wider array of opportunities.

On the other hand, configurations with fewer industries, such as the 5-industry and 10-industry models, showed more variable performance. These models achieved remarkable results in the first period but struggled to maintain their performance in subsequent periods (1995-2010 and 2010-2020). The decline in performance can be attributed to the reduced diversification, making these models more susceptible to industry-specific risks and less able to exploit cross-industry opportunities. This trend underscores the importance of diversification in constructing resilient trading strategies that can adapt to changing market conditions over time.

Comparing the 48-industry and 49-industry configurations, both demonstrate robust performance with significant alpha and strong Sharpe ratios. However, the 49-industry model is preferred due to its higher alpha in the 2010-2020 period. The key difference between these two configurations is that the 48-industry model has a single category for Computers, while the 49-industry model splits this into Software and Hardware. This additional granularity likely allows the 49-industry model to better capture the distinct dynamics within the technology sector in the 2010s. We see that the 48-industries produced an annual alpha of around 5.5% while the 49-industry produced an alpha of 10.65% in the 2010s with Sharpe ratios of 0.20 and 0.38, respectively. Thus, we can argue that the 49-industry strategy is better to deploy than the 48-industry in current market conditions.



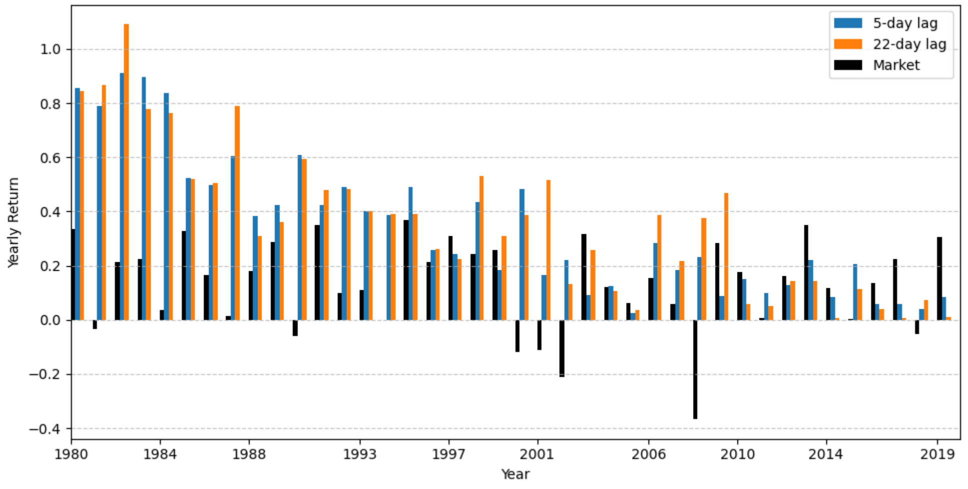


Figure 7: Monthly cumulative returns and raw annual returns for prior week (5) and prior month (22) input data.

	Period	α	α (Annual)	β	Sharpe Ratio
5-day lag	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2020	0.04*	10.65	0.04	0.38
22-day lag	1980-1995	0.16*	51.73	-0.00	1.34
	1995-2010	0.09*	26.99	0.01	0.55
	2010-2020	0.02*	5.66	0.05	0.20

Table 7: FF6 benchmark for varying day-lag input data. * signifies significant α.

The comparison between models using 5-day and 22-day lag input data reveals interesting insights into their performance across different periods. In the first period (1980-1995), both configurations achieve significant alpha, with the 5-day lag model producing an annual alpha of 51.57% and a Sharpe ratio of 1.49, while the 22-day lag model achieves an annual alpha of 51.73% with a Sharpe ratio of 1.34. During the second period (1995-2010), the 5-day lag model shows an annual alpha of 17.43% and a Sharpe ratio of 0.40, whereas the 22-day lag model outperforms with an annual alpha of 26.99% and a Sharpe ratio of 0.55. However, in the third period (2010-2020), the performance of the 5-day lag model remains relatively strong with an annual alpha of 10.65% and a Sharpe ratio of 0.38, while the 22-day lag model's performance drops to an annual alpha of 5.66% and a Sharpe ratio of 0.20.

Given these results, the 5-day lag model is preferable, especially considering its superior performance in the most recent period (2010-2020). This period is crucial as it reflects the model's ability to adapt to more recent market dynamics and conditions. Additionally, the 5-day lag model consistently demonstrates better risk-adjusted returns as indicated by its higher Sharpe ratios across all periods, particularly in the 2010-2020 period. The higher Sharpe ratio indicates that the 5-day lag model provides better returns per unit of risk compared to the 22-day lag model. Therefore, despite the 22-day lag model's slightly better performance in the second period, the 5-day lag model's more consistent and robust performance, especially in recent years, makes it a more reliable choice for developing trading strategies.

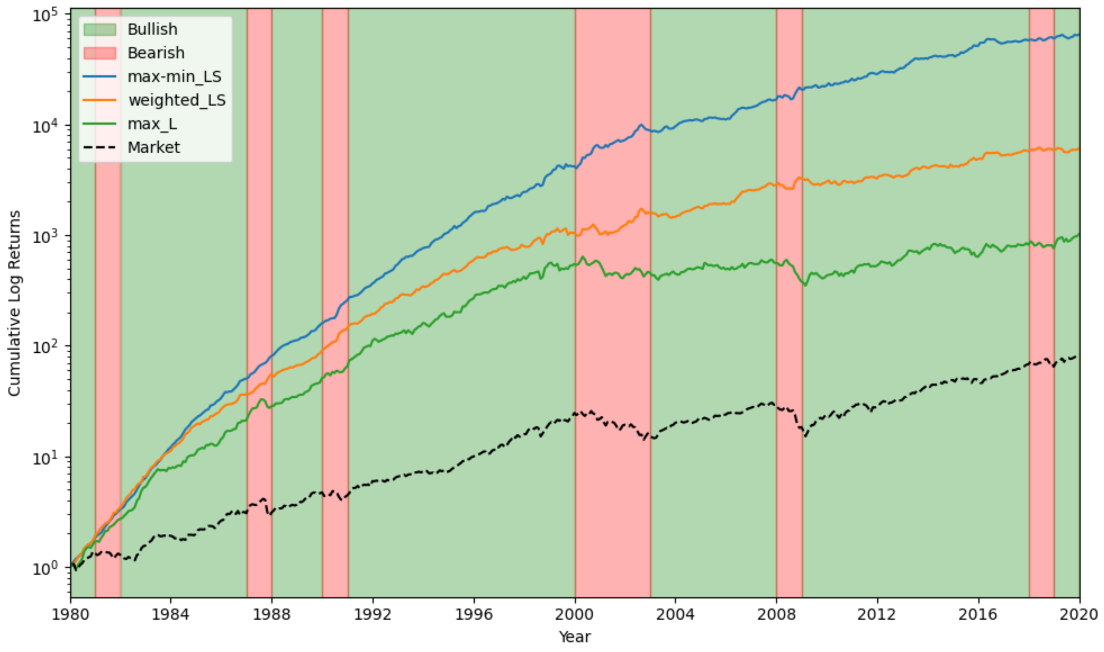


Figure 8: Monthly cumulative returns for varying portfolio trading strategies.

	Period	α	α (Annual)	β	Sharpe Ratio
max-min Long-Short	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2020	0.04*	10.65	0.04	0.38
weighted Long-Short	1980-1995	0.13*	42.38	-0.05	1.00
	1995-2010	0.03*	9.44	0.04	0.20
	2010-2020	0.03*	7.28	0.08	0.25
max Long	1980-1995	0.09*	28.35	0.68	0.57
	1995-2010	-0.01*	-3.86	0.65	0.03
	2010-2020	-0.00	-0.33	0.73	0.17

Table 8: FF6 benchmark for varying portfolio trading strategies. * signifies significant α.

The comparison of portfolio trading strategies indicates that the Max-Min Long-Short (LS) strategy consistently performs the best across all periods. This strategy achieves significant alphas and the highest Sharpe ratios in each period. The primary reason for its superior performance is likely due to its method of trading based on the highest absolute value predicted returns. By going long on the top decile of industries with the highest predicted returns and shorting the bottom decile with the lowest predicted returns, the strategy effectively captures significant market movements and exploits both bullish and bearish opportunities, leading to more robust returns.

In contrast, the Weighted Long-Short (LS) strategy, while still profitable, does not perform as well because it weights positions proportionally to predicted returns, which may dilute the impact of extreme predictions and lead to less optimal allocation. This approach can result in less pronounced returns compared to the Max-Min LS strategy. The Max Long strategy, on the other hand, shows the highest beta and lower Sharpe ratios, indicating higher volatility and risk. This is because it only takes long positions on the single industry with the highest predicted return, without any hedging through short positions. This lack of diversification and risk management results in higher exposure to market fluctuations, leading to less stable and lower risk-adjusted returns.

6 Discussion

The results of our analysis show that machine learning models, particularly the Long Short-Term Memory (LSTM) networks, can effectively capture market trends and generate consistent positive returns across different market regimes.

Our findings indicate that the LSTM model, when trained on value-weighted Fama French 49 Industry Portfolio Returns using a 5-day lag and applied through the Max-Min Long-Short strategy, produced the highest alpha. This model consistently generated significant positive returns across various market conditions, including both bullish and bearish periods. For example, the LSTM model achieved an annual alpha of 51.57% during 1980-1995, 17.43% during 1995-2010, and 10.65% during 2010-2020. The robustness of this model is further highlighted by its low beta values, indicating minimal correlation with market movements and demonstrating its ability to provide excess returns independent of overall market trends.

The data emphasizes the superiority of the LSTM model for financial market predictions due to its ability to adapt and maintain high performance across different market regimes. The consistent alpha generation and favorable Sharpe ratios of the LSTM model highlight its effectiveness in capturing market trends and providing reliable risk-adjusted returns. In contrast, the more variable performance of the FFNN and GBDT models suggests that they may be less suited for long-term, dynamic market conditions, reinforcing the preference for LSTM in developing advanced trading strategies. For instance, the Sharpe ratios for the LSTM model were 1.49, 0.40, and 0.38 across the three periods, respectively.

Moreover, our analysis across different market periods reveals that the model's performance remains strong, though it exhibits a decline in alpha over time. This decline may be attributed to the increasing prevalence of machine learning for market analysis and prediction and the overall AI boom in the late 2010s.

Our study also underscores the importance of model configuration and parameter selection. The number of industries, input day lags, portfolio strategy type, and industry data weighting significantly impact model performance. For instance, models trained on a larger number of industries (e.g., 49 industries) generally outperformed those trained on fewer industries (e.g., 5 industries) due to increased diversification. In fact, from Figure 11 the lower industry models taper off and end up with little to no returns in latter years, likely because of their simplicity being factored in to the market.

Interestingly, there does not seem to be a significant difference between the number of Day Lags as seen in Table 12. We experimented with a 1 Day Lag (not included in results) which performed worse than both current models, but the similarity between the weekly (5 Day) and monthly (22 Day) Lags could mean that the models can learn sufficiently well using only the past week's worth of data.

As seen in Figure 13, our chosen portfolio strategy of Max-Min Long Short was also important to model performance. By choosing to invest only in the top and bottom deciles, we minimize our exposure to uncertainty by choosing to short or long industries that are particularly well or poorly, respectively.

6.1 Future Research and Limitations

6.1.1 *Future Research*

In future research, we aim to implement several enhancements to improve our models' performance and robustness. (1) temporal data weighting. By giving more weight to recent market data in the training process of each model, we hypothesize that the models will have better predictive power, as more recent information is likely to be more relevant for future returns. (2) class imbalance handling. Given the market's tendency to rise over time, there is a bias in returns for each industry to be positive. By balancing the classes in our models, we could potentially improve their prediction abilities. (3) new input features, such as industry-wide earnings data, news sentiment, and technical indicators like moving averages, Relative Strength Index (RSI), and Moving Average Convergence/Divergence (MACD).

6.1.2 *Risks and Limitations*

There are several limitations to our study that must be acknowledged. (1) transaction fees were not accounted for in our backtest, which will lower the alpha for higher industry count portfolios. (2) we trained our models on a yearly basis, but since this is a daily strategy it would be most accurate to train daily to get most accurate alphas. (3) while we used the Fama French Industry Portfolios as a proxy for trading, actual implementation would require finding ETFs to trade, which may not perfectly reflect industry returns and could reduce the models' predictive power. (4) we did not incorporate any regularization checks on our models' outputs. This means that if a model predicts an extraordinarily high return for a single industry, it could lead to disproportionate investments in that industry, posing significant risks. Implementing stop losses or reweighting/resampling model predictions could mitigate this issue. (4) our models were insensitive to intraday data, relying solely on daily data. This exposure to overnight or intraday risk means that our models may not react promptly to significant market announcements requiring immediate buys or sells. Addressing these limitations will be crucial for translating our models' theoretical success into practical, real-world trading strategies.

7 Conclusion

Our research demonstrates that machine learning models, especially Long Short-Term Memory (LSTM) networks, can effectively capture market trends and generate consistent positive returns across different market regimes. The LSTM model, trained on value-weighted Fama French 49 Industry Portfolio Returns using a 5-day lag and applied through the Max-Min Long-Short strategy, produced the highest alpha. Specifically, this model achieved significant annual alphas of 51.57% during the period from 1980 to 1995, 17.43% from 1995 to 2010, and 10.65% from 2010 to 2020. These results highlight the model's ability to generate excess returns independent of overall market movements, as indicated by its consistently low beta values near zero.

The robustness of the LSTM model is further evidenced by its high Sharpe ratios, which measure the risk-adjusted returns. The Sharpe ratios were 1.49 for the period 1980-1995, 0.40 for 1995-2010, and 0.38 for 2010-2020, underscoring the model's effectiveness in providing reliable returns with manageable risk. The comparative analysis showed that the LSTM model outperformed other architectures, such as Feed Forward Neural Networks (FFNN) and Gradient Boosting Decision Trees (GBDT), which exhibited more variable performance and lower Sharpe ratios.

Moreover, our analysis across different market periods revealed that while the LSTM model's alpha declined over time, it remained significant, reflecting the model's adaptability to changing market conditions. The number of industries included in the model also played a crucial role, with the 49-industry configuration consistently outperforming those with fewer industries due to increased diversification. Further, the 22-day lag did for all data before 2010 but after the 5-day lag did better.

Our chosen portfolio strategy, the Max-Min Long-Short, proved to be the most effective, leveraging the highest and lowest predicted returns to maximize gains and minimize risks. This strategy consistently yielded higher alphas and Sharpe ratios compared to the Weighted Long-Short and Max Long strategies.

In terms of future research, we propose several enhancements to improve model performance and robustness, including temporal data weighting, class imbalance handling, and the introduction of new features such as industry-wide earnings data, news sentiment, and technical indicators. These additions may provide a more comprehensive understanding of market dynamics, leading to more accurate predictions. We also acknowledge several limitations in our study, such as the exclusion of transaction fees, the need for more frequent model training, the practical challenges of trading Fama French Industry Portfolios through ETFs, the lack of regularization checks on model outputs, and the insensitivity to intraday data. Addressing these limitations will be crucial for translating our models' theoretical success into practical, real-world trading strategies.

References

[1] G. Bebis and M. Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31, 1994.

[2] Database Camp. Long short-term memory networks (lstm)- simply explained! `https://databasecamp.de/en/ml/lstms`, 2023. Accessed: 2024-06-08.

[3] Haowen Deng, Youyou Zhou, Lin Wang, and Cheng Zhang. Ensemble learning for the early prediction of neonatal jaundice with genetic features. *BMC Medical Informatics and Decision Making*, 21, 12 2021.

[4] Prajwal Eachempati, Praveen Ranjan Srivastava, Ajay Kumar, Kim Hua Tan, and Shivam Gupta. Validating the impact of accounting disclosures on stock market: A deep neural network approach. *Technological Forecasting and Social Change*, 170:120903, 2021.

[5] Eugene F. Fama and Kenneth R. French. Industry costs of equity. *Journal of Financial Economics*, 43(2):153–193, 1997.

[6] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[7] Kenneth R. French. Data library. `https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`, 2024. Accessed: 2024-06-07.

[8] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] Lauri Nevasalmi. Forecasting multinomial stock returns using machine learning methods. *SSRN Electronic Journal*, 2020.

[11] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. pages 7–12, 07 2017.

[12] Aleksandra Vuckovic, Vlada Radivojevic, Andrew Chen, and Dejan Popović. Eeg drowsiness 2002 medengphy, 03 2015.

[13] Bin Weng, Mohamed A. Ahmed, and Fadel M. Megahed. Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, 79:153–163, 2017.

8 Appendix

Redo of all results but with data up to 2024. The final plots and tables are for a comparison of equal-weighted and value-weighted industry portfolios.

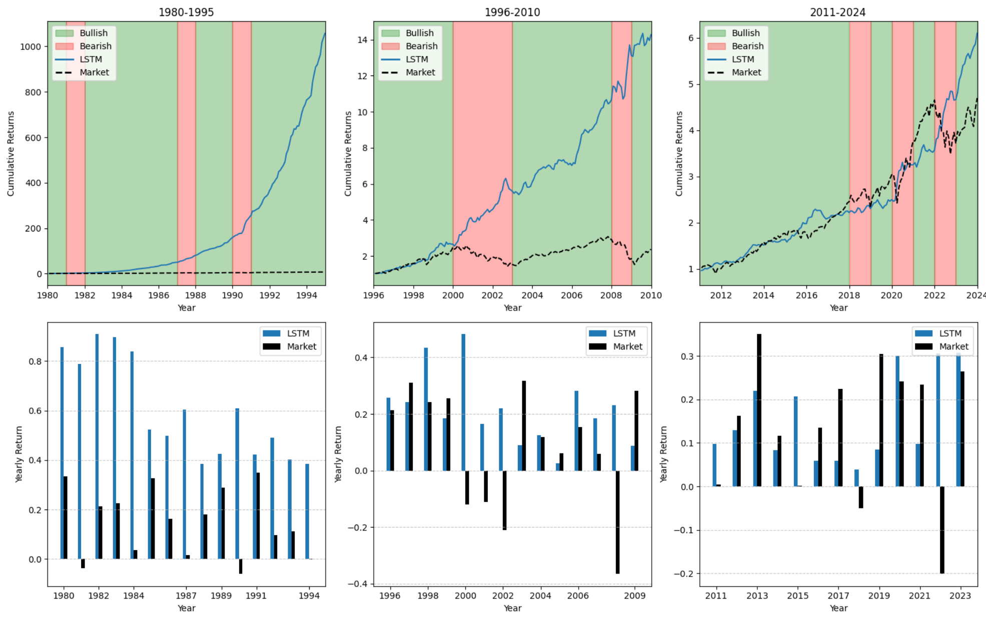
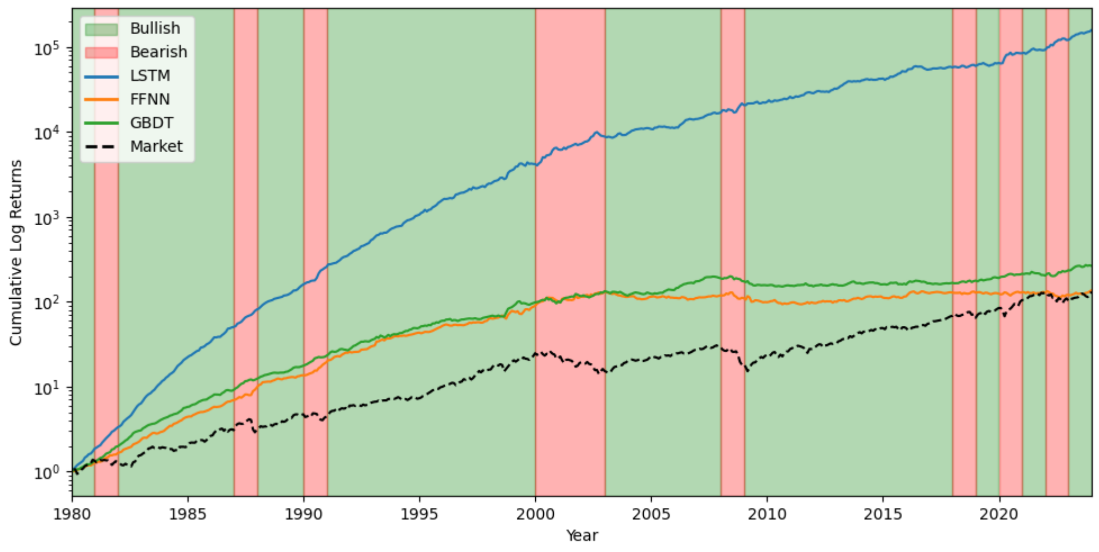


Figure 9: Monthly cumulative returns and annualized raw returns of best strategy in three separate time periods for data up to 2024.

	α (Annual)	β	Sharpe Ratio
CAPM	0.09* (28.18)	0.01	0.65
FF3	0.09* (27.00)	0.01	0.65
FF6	0.09* (27.47)	0.02	0.65

Table 9: Best strategy CAPM, FF3, and FF6 benchmarks. α is daily and in percent. * signifies significant α.

	Period	α (Annual)	β	Sharpe Ratio
Total	1980-1995	0.16* (51.57)	0.01	1.49
	1995-2010	0.06* (17.43)	0.03	0.40
	2010-2024	0.05* (14.69)	0.03	0.41
Bullish	1980-1995	0.15* (50.32)	-0.00	1.05
	1995-2010	0.05* (15.36)	0.03	0.33
	2010-2024	0.04* (12.37)	0.04	0.38
Bearish	1980-1995	0.17* (58.5)	0	0.45
	1995-2010	0.07* (21.03)	0.02	0.26
	2010-2014	0.07* (21.07)	0.03	0.29



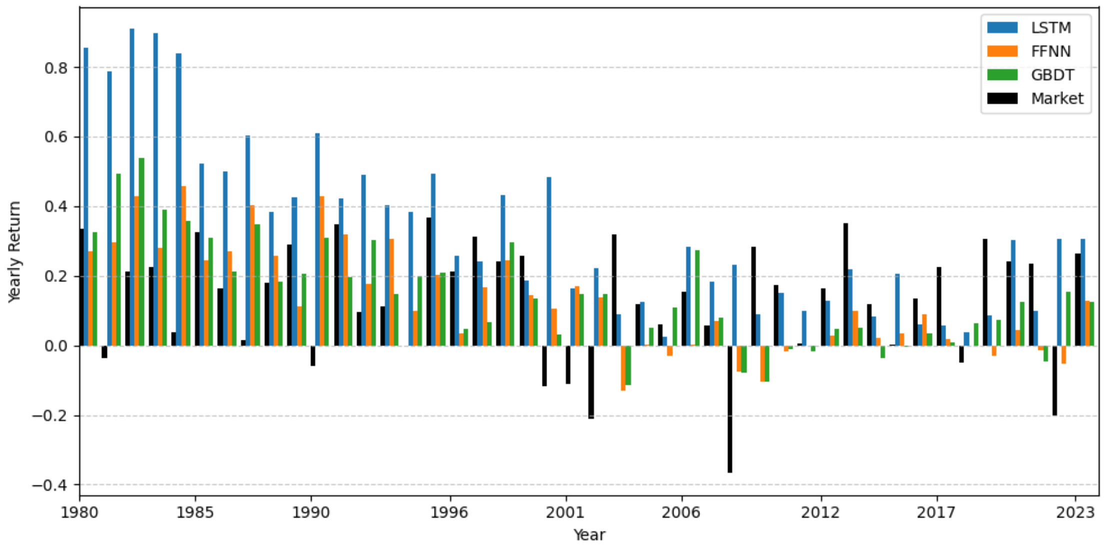


Figure 10: Monthly cumulative returns and raw annual returns for varying model types for data up to 2024.

	Period	α (Annual)	β	Sharpe Ratio
LSTM	1980-1995	0.16* (51.57)	0.01	1.49
	1995-2010	0.06* (17.43)	0.03	0.40
	2010-2024	0.05* (14.69)	0.03	0.41
FFNN	1980-1995	0.07* (20.70)	-0.04	0.70
	1995-2010	0.01 (1.60)	0.01	0.05
	2010-2024	0.01* (3.31)	-0.08	0.08
GBDT	1980-1995	0.07* (21.38)	0.05	0.80
	1995-2010	0.02* (4.49)	0.01	0.12
	2010-2024	0.01* (3.23)	0.03	0.14

Table 10: Model Architecture Comparison

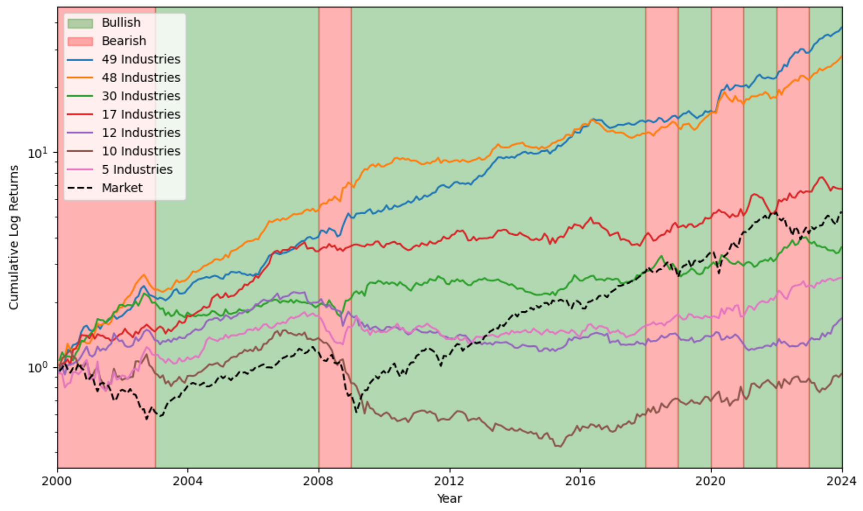


Figure 11: Monthly cumulative returns for varying number of industries for data up to 2024.

	Period	α	α (Annual)	β	Sharpe Ratio
49 Industries	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2024	0.05*	14.69	0.03	0.41
48 Industries	1980-1995	0.16*	53.09	0.02	1.37
	1995-2010	0.08*	23.28	0.02	0.48
	2010-2024	0.03*	8.8	0.01	0.24
30 Industries	1980-1995	0.12*	39.05	0.02	1.17
	1995-2010	0.04*	10.07	-0.02	0.21
	2010-2024	0.01*	3.06	-0.02	0.07
17 Industries	1980-1995	0.18*	58.77	-0.03	1.29
	1995-2010	0.08*	21.97	-0.03	0.40
	2010-2024	0.02*	5.31	-0.02	0.11
12 Industries	1980-1995	0.19*	63.97	-0.05	1.43
	1995-2010	0.03*	8.76	-0.01	0.19
	2010-2024	0.00	0.8	-0.04	0.01
10 Industries	1980-1995	0.24*	88.93	-0.06	1.36
	1995-2010	-0.00	-0.39	-0.05	0.01
	2010-2024	0.02*	4.43	-0.04	0.07
5 Industries	1980-1995	0.1*	29.82	-0.05	0.75
	1995-2010	0.00	1.3	0.01	0.03
	2010-2024	0.02*	4.28	-0.03	0.10

Table 11: FF6 benchmark for varying number of industries in different time periods. * signifies significant α.

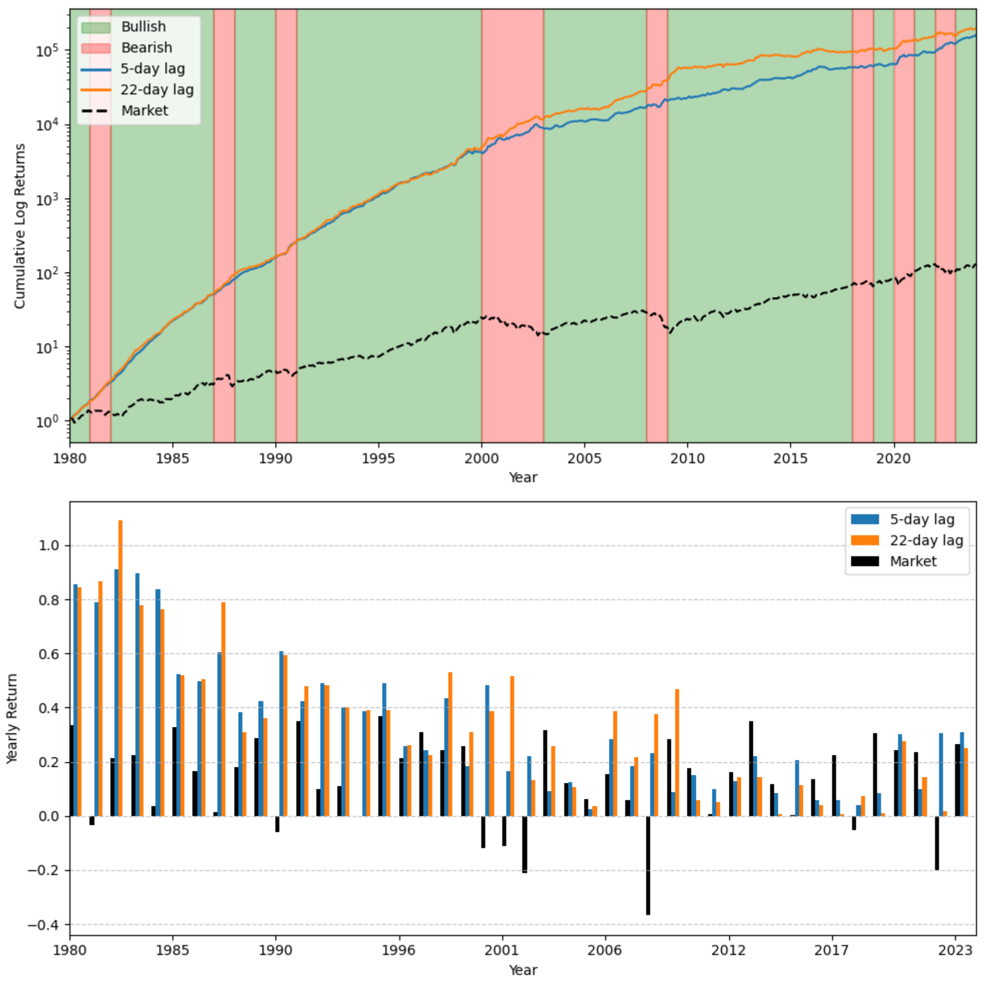


Figure 12: Monthly cumulative returns and raw annual returns for prior week (5) and prior month (22) input data up to 2024.

	Period	α	α (Annual)	β	Sharpe Ratio
5-day lag	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2024	0.05*	14.69	0.03	0.41
22-day lag	1980-1995	0.16*	51.73	-0.00	1.34
	1995-2010	0.09*	26.99	0.01	0.55
	2010-2024	0.03*	8.38	0.06	0.25

Table 12: FF6 benchmark for varying day-lag input data. * signifies significant α.

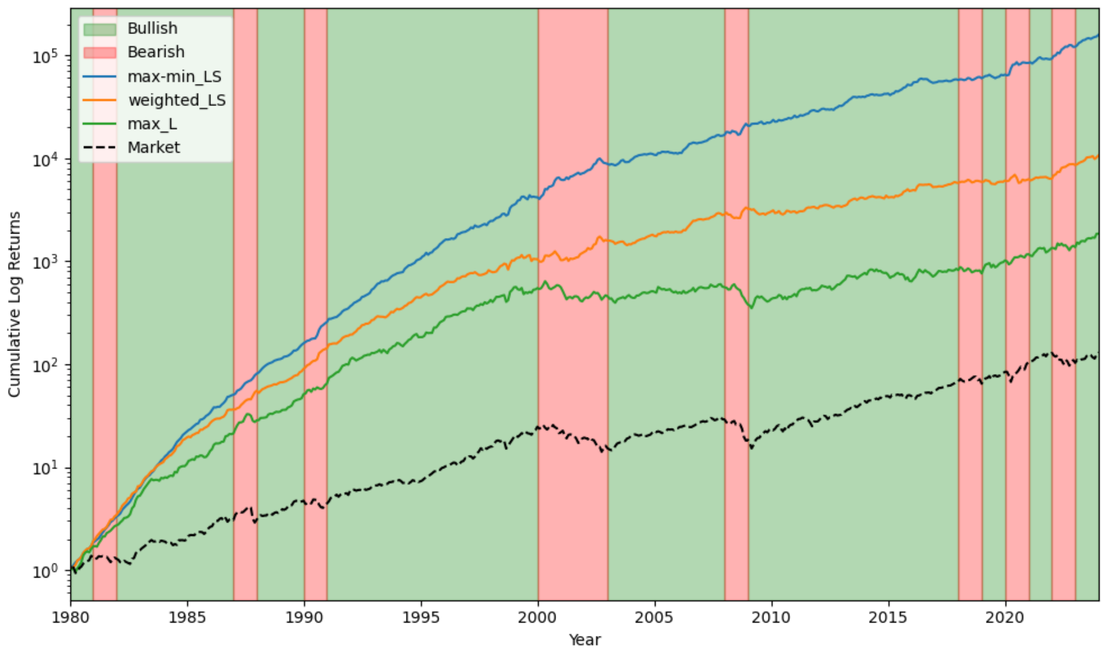


Figure 13: Monthly cumulative returns for varying portfolio trading strategies for data up to 2024.

	Period	α	α (Annual)	β	Sharpe Ratio
max-min Long-Short	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2024	0.05*	14.69	0.03	0.41
weighted Long-Short	1980-1995	0.13*	42.38	-0.05	1.00
	1995-2010	0.03*	9.44	0.04	0.20
	2010-2024	0.03*	9.43	0.07	0.27
max Long	1980-1995	0.09*	28.35	0.68	0.57
	1995-2010	-0.01*	-3.86	0.65	0.03
	2010-2020	0.01	2.11	0.70	0.20

Table 13: FF6 benchmark for varying portfolio trading strategies. * signifies significant α.

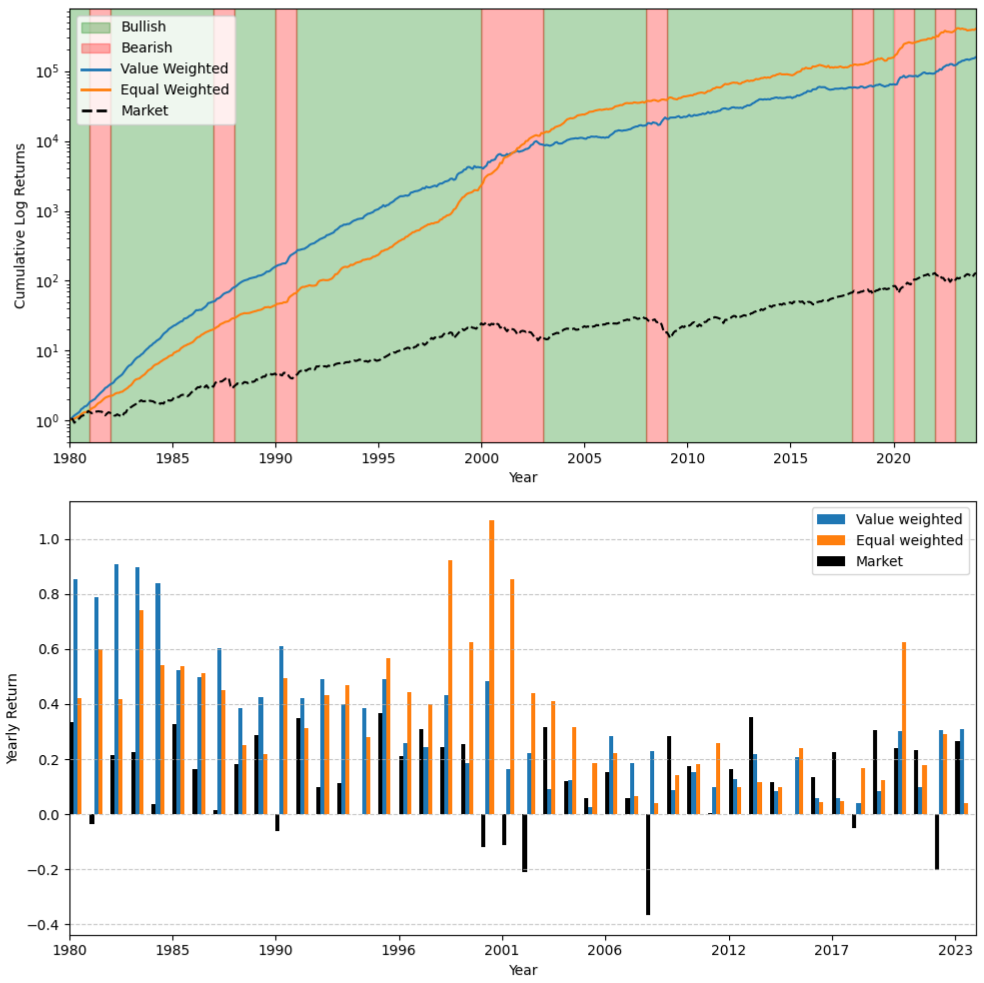


Figure 14: Monthly cumulative and raw annual returns for equal and value-weighted data.

	Period	Alpha	Alpha (Annualized)	Beta	Sharpe Ratio
Value weighted	1980-1995	0.16*	51.57	0.01	1.49
	1995-2010	0.06*	17.43	0.03	0.40
	2010-2024	0.05*	14.69	0.03	0.41
Equal weighted	1980-1995	0.12*	35.46	0.01	1.08
	1995-2010	0.13*	39.2	-0.04	0.79
	2010-2024	0.06*	17.46	-0.01	0.45

Table 14: Industry Data Weighing Comparison

In Figure 14, the equal weighted industries data actually outperforms our model. This is expected and can be explained by the SMB (Small Minus Big) Fama French factor which holds that large cap companies are overvalued compared to their smaller cap counterparts. The value weighted industries will not be able to capture this arbitrage opportunity, but we chose it regardless because in a real world scenario we may not be able to buy all stocks equally and it is risky to buy large quantities of a small cap stock at scale regardless of returns because of volume concerns. Additionally, ETFs are more likely to be found matching the value than equal portfolio which would make it easier and more practical to trade this strategy. Even without ETFs, we would likely have higher transaction fees for the equal weighted portfolios because they would result in making more trades since they are buying more of small cap stocks to match the equity in large cap stocks. For more information about why we chose value against equal weighted Industry data, refer to the bottom of the appendix.

The results indicate that the equal-weighted portfolio strategy outperforms the value-weighted strategy in terms of both alpha and Sharpe ratio, particularly in the periods after 1995. For the period 1995-2010, the equal-weighted strategy achieves an annual alpha of 39.2% and a Sharpe ratio of 0.79, compared to the value-weighted strategy's 17.43% alpha and 0.40 Sharpe ratio. Similarly, in the period 2010-2020, the equal-weighted strategy continues to show superior performance with an annual alpha of 13.87% and a Sharpe ratio of 0.41, while the value-weighted strategy generates an annual alpha of 10.65% and a Sharpe ratio of 0.38. This consistent outperformance by the equal-weighted strategy can be attributed to its higher exposure to smaller-cap stocks, which tend to yield higher returns due to the size premium effect.

Despite the higher returns, we still prefer the value-weighted strategy for practical reasons related to trading. In real-world scenarios, it is more challenging to trade equal-weighted portfolios because it requires buying and maintaining larger positions in smaller-cap stocks, which can be difficult due to volume constraints and increased market impact. Additionally, value-weighted portfolios align more closely with ETFs, making them easier and more cost-effective to trade. Even if not trading on ETFs but purely stocks by industry, frequent rebalancing needed to maintain equal weights in the portfolio also leads to higher transaction fees, which can erode net returns. The value-weighted strategy, though not capturing the full arbitrage opportunity presented by the size factor, offers a more practical and feasible approach for large-scale trading, providing stable and consistent returns with lower operational complexities and costs.