Neural Networks in Asset Management: Identifying Inefficiencies with Systemic Market Factors

A THESIS

Presented to

The Faculty of the Department of Economics and Business

The Colorado College

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Arts

By:

Anderson (AJ) Fabbri

May 2025

Neural Networks in Asset Management: Identifying Inefficiencies with Systemic Market Factors

Anderson Fabbri

May 2025

Economics

Abstract

Pushing the boundaries of the efficient market hypothesis, this thesis explores how well instances of multi-layered sequential long short-term memory (MLS-LSTM) recurrent neural networks augmented with Fama-French factors generate alpha—overperformance compared to a buy-and-hold strategy. Testing 30 assets including large-cap stocks, small-caps, and cryptocurrencies, this study finds that incorporating Fama-French factors boosts alpha generation by an average of 45.26 percentage points with statistical significance. Models demonstrate their highest alpha with cryptocurrencies, consistent with their lower informational efficiency, but statistical significance for the effects of market segmentation varies by analysis method. Robust analytical techniques including ANOVA, multiple linear regression, and Sharpe ratios are used to validate results. Findings affirm that incorporating broad market information can improve machine learning performance, while limitations include sample size, timeframe, and computational constraints. Enriched, well-designed neural networks have a high potential to exploit market inefficiencies, particularly with volatile low-information assets like cryptocurrencies.

KEYWORDS: Long Short-Term Memory Networks, Efficient Market Hypothesis, Alpha Generation, Financial Forecasting, Fama-French Factors, Cryptocurrencies

JEL CODES: C45, G14, G17, G11

ON MY HONOR, I HAVE NEITHER GIVEN NOR RECEIVED

UNAUTHORIZED AID ON THIS THESIS

Anderson James Fabbri

Signature

Table of Contents

1. Introduction

1.1 Motivation

How can financial markets, designed to optimize economies, do their job successfully if not truly optimized? The efficient market hypothesis, developed in the 1960s, states that markets are already efficient and translate new information into price action in real time, so it is impossible to reliably generate returns above a buy-and-hold strategy. With increased computational power and data availability, scholars and investors have begun to challenge this through empirical research on applied machine learning in financial markets. Thus far, however, existing papers on the subject have remained confined to technical analysis, approaching financial markets as a computing problem somehow detached from the economics driving those very markets. This presents the opportunity to explore whether incorporating macroeconomic and broad market indicators, like Fama-French factors, can enhance predictive power. Plus, existing studies have confined their research to limited asset samples, with several published papers studying machine learning on fewer than five assets or indices. There may be structural differences between asset groups which the broader world of machine learning research has not yet considered.

1.2 Hypotheses

Machine learning offers new opportunities to address the limitations of prior models used to inform asset management. This study proposes two hypotheses to test how macroeconomic factors impact the viability of using advanced neural networks for this purpose.

Primary hypothesis: Multi-layered sequential long short-term memory (MLS-LSTM) networks enhanced with Fama-French factor data outperform counterpart models using only technical inputs in identifying inefficiencies and generating alpha across market segments.

This hypothesis tests the premise that incorporating systemic market information, such as macroeconomic and factor-based anomalies, enhances the predictive power of neural network models to generate excess returns over a buy-and-hold strategy. By testing the efficacy of an approach with holistic input features, the study will test the efficient market hypothesis, especially its strong and semi-strong forms. These claim that markets incorporate all publicly available information into price action and, in the case of the strong form hypothesis, that prices fully reflect even insider information. Generating significant alpha in this study would imply that these forms of the hypothesis do not hold. It also addresses gaps in traditional econometric approaches that struggle to account for non-linear time-dependent relationships in large datasets.

Secondary hypothesis: MLS-LSTM networks will perform better in low-information market segments, such as small-cap stocks and cryptocurrencies, compared to high-information segments like large-cap stocks.

This will test if the model generates greater excess returns in market segments with relatively low levels of informational availability and, thus, high information asymmetry. Speculation, limited trading volumes, and lower scrutiny may generate inefficiencies for advanced predictive models to exploit.

From a practical perspective, better predictive models improve investors' ability to identify opportunities for alpha generation. The findings from testing these hypotheses could inform the further development of dynamic data-driven investment strategies that outperform static approaches designed for regression and valuation estimates.

1.3 Experimental Approach

How will this study work? First, models are trained on a daily dataset with price, volume, dividend, and stock split data (if applicable) for every tested asset, accessed from the Yahoo Finance database. To represent different market segments, assets tested include 10 top large-cap stocks from the SPDR S&P 500 ETF (SPY), 10 stocks from the Schwab U.S. Small-Cap ETF (SCHA), and 10 of the top cryptocurrencies by market cap. For each ticker, one model is trained with a technical-only dataset and one with a Fama-French factor-augmented dataset to test the impacts of including broad market information. Each model is tested in a trading simulation and compared with a buy-and-hold strategy over a two-year testing period. Finally, the overall regression and framing of the study is a categorical analysis of model performance.

Unlike many Economics theses, this paper has more layers than just a series of regressions in its analysis section. To be clear, this study examines the MLS-LSTM model architecture by testing various instances of that architecture, one for each input dataset. The machine learning models used are technically regression models, but this study tests their aggregate performance over a trading period that is separate from training data, specifically by analyzing mean squared error, mean absolute error, and alpha returns. A final ordinary least squares regression is conducted using categorical dummy variables to evaluate model performance, along with paired T-tests and ANOVA. Sharpe ratios are calculated on a strategy level, using overall return metrics over the testing period to calculate volatility. This research design, which analyzes neural networks through the lenses of Fama-French inclusion and information segments, is designed to provide a novel, integrated view of how the economic theories of the efficient market hypothesis and asymmetric information intersect with data-driven approaches to investing.

This study expects that MLS-LSTM networks augmented with Fama-French factors will outperform their technical-only counterparts in alpha generation compared to a buy-and-hold strategy across all tested market segments, with particularly strong performance in the low-information segments—small-cap stocks and cryptocurrencies—due to more prevalent inefficiencies.

1.4 Roadmap

This thesis is organized into six parts to explore the following research questions: Does an MLS-LSTM informed by stock price movements and Fama-French factors outperform technical-only versions of those models? And do they capture market inefficiencies differently across low- and high-information market segments?

Part 1 (Introduction) frames the study, its hypotheses, and its importance in the context of economic theory and financial markets. Part 2 (Literature Review) examines the history of stock price prediction as well as existing theories, model types, and prior approaches in the field, identifying the gaps this study will address. Part 3 (Theory & Model) presents the study's theoretical foundation and the MLS-LSTM architecture's framework and mathematics. Part 4 (Methodology) outlines the study's sample selection, data preprocessing, and how the final models will be assessed. Part 5 (Results & Analysis) details the statistical techniques used to analyze and validate model performance by category, including ANOVA, Sharpe ratios, and multiple linear regression. Finally, Part 6 (Conclusion) ends with key insights from the analysis, the practical implications of this study, and ideas for future research directions.

2. Literature Review

2.1 Financial Markets and Prediction Challenges

We hear about financial markets everywhere, from academia to news to short-form social media interviews with strangers on the street. These systems are clearly relevant to many in society, and it seems, on the surface, that the sole reason is because people can use them to grow their wealth. However, financial markets also play a pivotal role in optimizing economies. According to Mishkin (2019), the purpose of financial markets is to maximize economic welfare by shifting money from lender-savers to borrower-spenders who have opportunities to deploy it at times and in ways that increase productivity and output. Countries with markets that succeed in this are generally much wealthier than those with poorly functioning markets (p. 52).

Since markets are engines through which investors can gain or lose money, there is a massive emphasis on earning the highest returns possible. So much so that entities from banks to hedge funds to retail investors incessantly pursue the holy grail of "beating the market"—outperforming a major index like the S&P 500 over a certain time horizon. And investors can theoretically do this if they are able to predict what will happen to asset prices more effectively than other market participants. Early market theorists throughout the 1900s believed this to be impossible.

First, Louis Bachelier (1900) proposed the random walk theory: that stocks follow random, unpredictable patterns, at least in the short to medium term, thus making it impossible to predict exactly what will happen to them. He begins his paper with a problem that persists in its relevance today, over 120 years later. Although later scholars have disputed the random walk theory, in some ways, Bachelier was prophetic. In the following passage (translated from French), he lays the groundwork for the issues studied in this thesis:

> "In addition to somewhat natural causes of variations, artificial causes also intervene: the stock market acts on itself, and a current movement is a function not only of past movements but current price positioning. The determination of these movements depends on an infinite number of factors: it is therefore impossible to hope for mathematical prediction. The calculus of probabilities will never be able to apply to stock market movements, and stock market movements will never be an exact science. But it is possible to mathematically study the static state of the market at a given moment, that is to say, to determine the probability of price variations that the market admits at that moment." (p. 21)

But what if we did have an algorithm that could determine current movements as functions of previous ones, price positions, and more? What if it were possible to study (while not infinite) many hidden factors in conjunction with time series fluctuations? Due to recent advances in machine learning and computing power, Bachelier's impossible is the modern era's possible. Accurately predicting stock price movements could revolutionize the investing landscape.

Building on the random walk theory, Eugene Fama pioneered the efficient market hypothesis in the 1960s. It essentially argues that prices are unpredictable because new information immediately becomes priced in (Bao et al., 2025, p. 1). Stocks reflect the random walk as prices constantly update to reflect new information. If the random walk were predictable, the market would not be efficient. The implication of this is that it is impossible to earn returns in excess of a buy-and-hold strategy by predicting price movements; since the market reflects all available information, the best any individual investor can theoretically do over the long run is to buy and hold. Put another way, since the market "knows" everything there is to know, the best decision is to ride the market. For many, this works just fine, and the common strategy to buy and hold an index-linked exchange traded fund consistently earns higher risk-adjusted returns than an aggregate of actively managed funds (Crane et al., 2018, p. 62). Through the lens of the efficient market hypothesis, major indices outperform active funds because they intrinsically contain more information; again, it's the problem of the market versus the individual.

With the advent of big data and expanded information access through technology, scholars have called into question the empirical effectiveness of the aforementioned theories. Fama himself acknowledged the limitations of his hypothesis in a 1970 paper, writing that although "the model stands up rather well to the data, it is obviously an extreme null hypothesis. And, like any other extreme null hypothesis, we do not expect it to be literally true" (Fama, 1970 p. 388). Due to a recent body of research around the financial market applications of algorithmic developments from the 2010s and especially the 2020s, it seems that many researchers also do not expect the efficient market hypothesis to be a firm rule. Through data-driven methods that use past information to predict future price movements, it is now more possible than ever to hope for a helpful degree of mathematical predictability in financial markets.

The main early data-driven methods for stock price prediction were linear regression and time series decomposition (Bao et al. 2025). These models are designed to run with minimal computational requirements and perform inaccurately due to the nonlinearity of stock prices (p. 2). As computing power grew over time, researchers developed new methods that were more powerful but still rooted in traditional econometrics. Specifically, the autoregressive conditional heteroscedasticity (ARCH) and autoregressive integrated moving average (ARIMA) models took over the forefront of the field. The autoregression component of both models refers to predicting future values based on past ones, integrating lagged terms in an attempt to account for time series fluctuations (Bagrecha et al., 2024, p. 6). ARCH adds a conditional heteroscedasticity method which accounts for the clustering of stock price volatility where variance depends on that of previous periods. Differently, ARIMA adds a moving average mechanism to model the autocorrelation of series residuals in addition to autoregression, and its integration component refers to the use of differencing to eliminate non-stationarity (Bao et al., 2025, p. 2). Both methods are univariate, limiting their generalizability due to their failure to account for exogenous conditions or multiple interrelated variables. While multiple linear regression can mathematically deduce causal relationships between variables, it often fails to account for the impacts of time series. Plus, the time series ARCH and ARIMA models, which build on the concepts of traditional regression, end up functioning as little more than technical analysis due to being univariate.

To address these flaws, Kumar et al. (2014) incorporated concepts from the emerging field of machine learning and attempted to improve the ARIMA model. They experimented with three hybrid models on India's Nifty 50 index, one combining ARIMA with support vector machines, one with artificial neural networks, and a final one with random forests (p. 284). These combined networks all performed slightly better than standalone models (p. 302), revealing the potential benefits of incorporating a time series component in conjunction with machine learning's strengths in classification and pattern recognition. However, even these hybrids depend on careful feature selection and engineering to ensure homoscedasticity, stationarity, and a lack of multicollinearity, leaving room for subjectivity and inconsistent performance across varying market conditions and studies.

2.2 The Rise of Recurrent Neural Network Architectures

Although neural networks performed worse in Kumar et al.'s study than hybrid models, the optimization of recurrent neural networks (RNNs) was a major milestone in the field of data-driven stock forecasting. This study will specifically address a promising variant of the RNN, the long short-term memory network (LSTM). According to Bao et al.'s (2025) review study, the RNN-LSTM architecture has served as the most consistently effective type for predictive time series regression tasks with numeric inputs and outputs. This is due to its flexibility, high generalizability, and low computational load relative to other cutting-edge neural network architectures, which include the convolutional neural network (CNN), transformer, graph neural network (GNN), generative adversarial network (GAN), and large language model (LLM) (pp. 4-12).

RNNs are designed to process sequential data, combining basic neural network architecture with the capacity to retain information across multiple time steps. The methods section of this paper discusses the algorithms behind RNNs and LSTMs in greater detail. In a 2019 study, Berradi et al. used vanilla RNNs to model major market indices like the S&P 500, finding them to be effective for short-term predictions with shorter input and output windows compared to what LSTMs can handle (p. 56).

Like traditional neural networks, vanilla RNNs contain three main components: an input layer, a sequence of hidden layers, and an output layer. Put simply, the hidden layers model relationships between input variables in ways akin to creating abstract interaction terms. They then transmit this information to the output layer, which recompiles it into a regression or prediction for the dependent variable. Through the gradient descent training process, neural networks tune their weights—akin to a matrix of beta coefficients from linear regression—to minimize prediction error. These weights transform the values of variables as they pass through a network's layers. In RNN architecture, each node within the hidden layers is connected not only to the previous layer's output but also to a vector held in memory that stores information from previous time steps—its "hidden state" (Berradi et al., 2019, p. 57). The network processes data step by step, adjusting its existing hidden state with each new vector of input variables to model trends across a sequence. This added capability increases the complexity, accuracy, and computational load for processing sequential data (Bao et al., 2025, p. 4). While not perfect, the RNN's ability to capture some temporal patterns already made it much more effective for stock price forecasting compared to traditional neural networks.

The fatal flaw of vanilla RNNs is the "vanishing gradient" problem, in which the gradient, or rate at which the training process of model weight adjustments minimizes error, declines toward a limit of zero. This causes issues with large datasets and modeling long-term trends (Bao et al., 2025, p.4). To combat this problem and yield better results over longer time horizons, Hochreiter and Schmidhuber developed the LSTM architecture (1997, p. 1743). By incorporating input, forget, and output gates to control vector information memory in multiple cell states, not just the hidden state, LSTMs are better able to combat the vanishing gradient problem and decide which information is pertinent to a network's pattern recognition. Thus, they are more powerful in modeling long-term dependencies within variables (Hochreiter & Schmidhuber, 1997, p. 1735). This study will leverage the RNN-LSTM framework to capture complex hidden patterns within stock prices, benefiting from the algorithm's high level of adaptability to (relatively) efficiently model fluctuations within datasets.

2.3 Other Architectures in Financial Market Prediction

In recent years, researchers have explored several other neural network architectures for financial time series predictions, the most effective among these being convolutional neural networks (CNNs) and Transformer models. Originally designed for image processing and classification, CNNs have shown some promise in stock forecasting (Bao et al., 2025, p. 7). By processing market data through abstract spatial filters, CNNs can capture localized patterns; however, they lack the memory capability of RNNs and therefore struggle with long-term trends that are prominent in asset markets (p. 8).

Transformer architecture, central to the design of large language models (LLMs) like OpenAI's ChatGPT and Anthropic's Claude, uses a mathematical self-attention mechanism to capture dependencies between disparate data points, using abstract vectors in high-dimensional space. By modeling vector relationships within sequential data, they are theoretically effective for capturing long-range dependencies (Bao et al., 2025, p. 8). In practice, due to their high complexity and computational requirements, Transformers have struggled with long sequence data, performing better in short-term prediction tasks with high-frequency entries (p. 10). While transformers have potential for capturing complex relationships, their very high computational requirements and limited generalizability make them less efficient than LSTMs for long sequential tasks.

2.4 Limitations and Research Gaps

Several unresolved challenges continue to limit the full potential of neural networks in financial forecasting. As outlined by Bao et al. (2025), chief among these is a lack of adequate data representation (p. 16). Most studies covering neural networks thus far have relied on one-dimensional time series data or two-dimensional data made up solely of technical information like prices and trading volumes. While using these indicators allows models to capture basic market dynamics, they often fail to account for external events such as supply shocks, political shifts, or macroeconomic policy changes. This can result in high error rates and noisy prediction sequences (Bao et al., 2025, p. 16). To address this, the solution of including more data sources becomes apparent. Assuming adequate hyperparameter optimization, neural networks tend to have lower prediction error with more data, as they have more real sources from which to extract complex relationships. Incorporating additional features should improve a model's robustness and ability to generalize across different market conditions (Bao et al., 2025, p. 16). Zhou et al. tested this with additional data sources including internet search trend data, technical indicators, and news sentiment scores (2020, p. 1). They found that, through employing support vector machines as their prediction algorithm, multiple data sources tended to yield higher stock prediction accuracy than single-source data (p. 14). While this study shows promising results with multiple data sources, it neither uses LSTM architecture nor incorporates financial data (pp. 4-5). To address this gap, this study proposes a time series neural network with stock splits, dividends, interest rate data, and Fama-French factors as additional inputs beyond price and volume history.

Neural networks' lack of interpretability inherent to the black box problem has slowed their widespread adoption in industry. The black box problem refers to the fact that, due to layers of complex vectors, model weights, and matrix multiplication, it is nearly impossible to deduce exactly how or why a neural network predicts one value over another. To address this challenge, several methods have emerged, chief among them being Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). LIME provides local interpretability by explaining individual predictions, estimating the model's behavior at a specific instance, typically with linear regression (Bao et al., 2025, p. 17). On the other hand, SHAP is helpful for global interpretability, assigning importance scores to input features based on their levels of marginal contribution to a model's predictions (p. 17). The algorithm calculates Shapley values with weighted averages of the difference values between model predictions with all features and with individual features removed (Sun et al., 2023, p. 3). However, the Python packages required to implement these techniques are slow and buggy at best.

Another issue is that studies around this topic are inconsistent in the evaluation metrics they choose to use. Bao et al. point out that most studies, in addition to not including financial indicators, neglect financial evaluation metrics like the Sharpe ratio and return metrics (2025, p. 17). The lack of standardized datasets makes it difficult to compare results across studies, since markets are always changing as time passes and researchers often pull data from many different sources. Plus, while some studies test neural networks in regression tasks, predicting price as a continuous variable, others pursue classification tasks, predicting which direction a price will move with probabilistic modeling. Since both are different categories of neural network predictions, they use different evaluation metrics. This study will test a specialized multi-layered sequential LSTM on the continuous stock price variable of multiple tickers, using the aforementioned financial evaluation metrics, plus alpha returns and regression metrics like mean squared error and mean absolute error.

3. Theory & Model

3.1 Model Introduction

This section presents the theoretical and mathematical frameworks that guide the predictive multi-layered sequential long short-term memory network (MLS-LSTM) developed in this thesis. By grounding the study in established financial theories and explaining how the model addresses their limitations, this section will justify the selection of the proposed neural network architecture as well as its input features. Essentially, it will explore the question: Does feeding information about the broader market improve methods that were previously used in technical analysis?

Theoretical frameworks, as explored in the literature review section, provide lenses through which we can interpret complex market dynamics, whether they be why prices move as they do, like in the efficient market hypothesis, or why they may be impossible to predict, as outlined in the random walk theory. In pursuit of alpha, researchers and corporations have poured extensive monetary and human capital into developing models that attempt to offer a structured analysis of the interactions between market forces and asset price movements.

Fama's decades-old efficient market hypothesis (EMH) serves as the foundation for this study. If markets are fully efficient, there should be no way to generate returns in excess of a buy-and-hold strategy; however, if alpha is possible by using a relatively accurate model's predictions to determine entry and exit points, then semi-strong and strong-form EMH should be false. The study will test whether time-series neural network architecture itself enhances accuracy and if incorporating Fama-French factors as inputs yields an even stronger and more generalizable model architecture. More recent empirical research, including by Fama himself (1970), suggests that financial markets contain inefficiencies, especially in low-information states; asset prices are often inefficient due to asymmetric information between market participants. These findings, and the potential to further disprove the strong-form EMH, motivate this paper's hypotheses about broad market factors and information segments.

3.2 Model Framework

Using a Python 3 Jupyter Notebook, Keras with TensorFlow as a backend, and Bayesian optimization with the Keras tuner, this thesis will explain and implement a MLS-LSTM RNN. The next part of this section will break down every component of the model's abbreviated name. For now, it is important to note that this model architecture was chosen after an extensive literature review, through which its benefits to accuracy, computational efficiency, and ability to capture abstract temporal dependencies in continuous data became apparent. Plus, by employing Bayesian optimization with the Keras tuner, the program fine-tunes hyperparameters to appropriately adjust models for each stock dataset tested. Incorporating Fama-French factors allows a given model to capture broad market dynamics, including size, value, and momentum indicators across many stocks at once. Most previous papers about neural networks in stock forecasting have focused on technical analysis, which relies on historical price and volume data; however, it fails to account for macroeconomic and systemic influences. This thesis tests whether a holistic approach, feeding the model indicators connected to the broader market and real economy, improves its predictive power over models that rely solely on technical factors.

Unlike human analysts, who often rely on static or long-term strategies, neural networks can dynamically adapt to patterns in the data they are trained on. By repeatedly updating model weights in gradient descent, LSTMs can identify patterns over time and can learn how X impacts Y under various input sequences. Through their hidden layers, neural networks can detect complex and seemingly abstract interactions among variables that humans will miss, particularly when dealing with the added dimension of time. This capability implies that LSTMs should outperform human-driven strategies or models based on traditional regression like ARIMA, especially in large datasets with fluctuating multivariate inputs.

The modern LSTM framework offers several advantages over prior approaches. First, the univariate time series econometric ARIMA and ARCH models struggle to incorporate external factors or manage non-stationary data without differencing. In contrast, LSTM architecture does not require differencing to handle non-stationary data due to its dynamic feature scaling and memory gate mechanisms. While stock prices do reflect some external variables, the inability of ARIMA and ARCH to break down feature interactions limits these models' generalizability. Next, LSTMs have been shown to outperform basic neural networks, which, despite effectively capturing nonlinear relationships, lack a memory mechanism and are thus unable to model the time dependencies inherent to sequential data.

As explained in the literature review, while some studies have successfully combined traditional econometric techniques like ARIMA with machine learning algorithms, these hybrid models are prone to overfitting to noise in periods of volatility. They also often require extensive and specialized feature engineering, making it difficult and time-consuming to conduct an experiment with a hybrid model architecture over multiple trials and datasets. In contrast, the sequential LSTM architecture inherently handles patterns with time dependencies and benefits from automated feature learning. Plus, since the architecture can handle non-stationary data and thus needs far less feature engineering, it reduces the computational load and researcher input necessary when training and testing the model on various datasets. Plus, the multi-layered nature of this study's LSTM allows the model to capture dependencies over multiple time horizons, and the incorporation of a basic dense neural network layer also allows the model to learn from nonlinear relationships without time dynamics. Widely used in multivariate quantitative research applications, the LSTM is an ideal choice for the complex task of stock price prediction.

3.3 Assumptions and Limitations

This study operates on several key assumptions, notably markets' partial predictability, the explanatory power of Fama-French factors, and sufficient gradient descent. First, while stock price movements exhibit a degree of random behavior, patterns can emerge over time and under specific conditions, such as positioning within the business cycle. Fama-French factors, which bring broad market dynamics into the picture, will allow the model to move past pure technical analysis. The Fama-French five-factor model has shown that risks stemming from inefficiencies, such as those associated with small-cap and value stocks, can persist over time (Fama & French, 2014, p. 31). Finally, we assume that the gradient descent algorithm with the Adam optimizer updates model weights such that loss function errors converge to a sufficient minimum. It is important to note that the algorithm may not always find the global minimum due to the complexity of input features; however, even in cases where it fails to do so, it reaches one that will still yield adequate prediction accuracy for the algorithmic testing purposes of this study.

The proposed model also faces several challenges that may impact its performance and interpretability, including the risk of overfitting, dependence on data availability, and the black box problem. First, note that assets in low-information market segments pose a heightened risk of overfitting due to sparse or noisy data. Here, the LSTM network may learn and fit to noise rather than meaningful causal patterns, reducing its predictive power. The approach in this study mitigates this risk by using techniques such as sliding window training, dropout regularization, and cross-validation, but it remains a potential limitation. All training will be conducted on the first 81% of a price dataset, with the next 9% used for validation to compute gradient descent error metrics, and the final and most recent 10% will be reserved for testing generalizability. This will ensure that the experiment actually tests learned patterns on completely new, unseen data.

Next, the model's performance will rely on the availability and quality of historical data; luckily, this should not pose much of an issue as Python's yfinance package pulls directly from all available Yahoo Finance data, which is extensive and reputable. Finally, while the literature has shown that LSTMs excel in capturing complex nonlinear relationships, their black box nature can make it difficult to interpret how specific inputs influence predictions. This issue arises from the many layers of hidden vectors and matrix math within neural networks and remains relatively uninterpretable for humans. Methods like the permutation importance method, which tests versions of the model with different variables omitted to rank factors' individual contributions to predictive accuracy, can apply here. However, this method can be resource-intensive and time-consuming, and falls outside the scope and focus of this study. This study will also use simple dummy variable regressions on model accuracy metrics to see how incorporating Fama-French factors, and how market segments of different information levels, impact returns.

3.4 The Math

Given the LSTM architecture's centrality to this study, this section will explain all math which the architecture adds beyond traditional recurrent neural networks (RNNs). While RNN mechanisms incorporate a temporal mechanism, processing inputs based on information from the neural network's hidden state at previous timesteps, LSTMs add specialized ways to control which information is important and to minimize the impacts of noisy inputs. This makes them unique in their ability to address the vanishing gradient problem that affects vanilla RNNs, in which the model training process fails to adequately diminish error because of an inability to model long-term relationships.

Before diving further into this, let's define a few terms. First, the cell state is the network's memory component that allows it to remember, forget, and adjust information based on its gate mechanisms. It carries important information about feature relationships forward in time, allowing the model to tune weights depending on which dynamics are most important. Similarly, the hidden state represents the model's short-term memory at a given point in time. While the cell state is the main holding place for long term information, the hidden state periodically updates with the parts of the cell state that are most relevant to near-term predictive dynamics. Finally, activation functions determine how a node processes its input into an output.

By applying nonlinear transformations to weight matrices, LSTM cells allow machine learning models to approximate complex dynamics that may not follow set patterns. They allow the network to combine long and short-term memory to make more accurate predictions on multivariate time-series data, hence the long short-term memory network name. LSTMs use two main activation functions: The sigmoid (σ) function outputs values between 0 and 1, helping the model control the degree to which information passes through a gate and impacts model predictions, while the hyperbolic tangent (tanh) function outputs a range from -1 to 1, allowing the model to introduce a new directional dimension into the cell state (Hochreither & Schmidhuber, 1997, p. 1752)

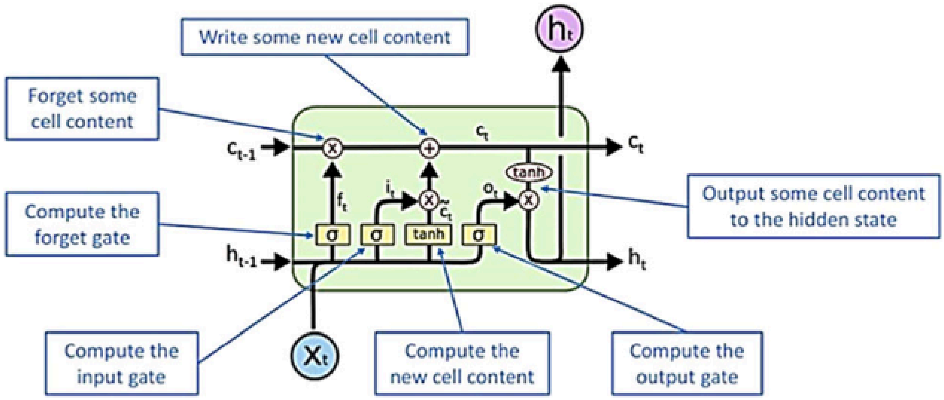


Fig. 1 Hidden layer neuron structure of LSTM deep neural network

As seen in Figure 1 from Yan et al.'s (2021) paper, each LSTM node operates on a series of six functions. Bao et al. (2025) write the functions as follows:

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

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

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

$$g_t = tanh(W_g \bullet [h_{t-1}, x_t] + b_g)$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet g_t$$

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

Where σ represents the sigmoid activation function, *tanh* is the hyperbolic tangent activation function, *W* is a weight matrix, and $[h_{t-1}, x_t]$ is a concatenated matrix of the previous hidden state and the current input data entry. Instead of dot products, LSTM math uses element-wise multiplication.

Next, i_t is equal to the output of a node's input gate, which effectively decides which new information is important enough for the model to retain. Note that this step does not drop the weights assigned to a variable relationship; instead, the matrix computation determines the degree of influence a variable or interaction term has on the node's predictive dynamics. It technically takes the candidate cell state's output and determines the degree to which proposed new information will enter the cell state. The forget gate, f_t, is similar to the input gate in its mechanism for deciding which information to omit from the cell state's vectors. However, unlike the input gate, this function operates element-wise to determine whether to retain or omit values from the prior cell state. Higher values are more likely to stick around in the new cell state. The output gate, o_t, determines the degree to which a given piece of information influences the next hidden state.

To circle back to activation functions; if, for example, the input gate's sigmoid activation function outputs 0.8, then the node will choose to keep 80% of the candidate information and discard the rest. Meanwhile, the hyperbolic tangent function introduces positive and negative information, helping the model understand patterns such as how a certain sequence of input variables impacts the direction of future stock prices.

g_t, the candidate cell state, uses a hypertangent activation function to generate information about the direction of relationships between variables. It essentially acts as a proposal for how to update the cell state. This matrix then joins that of the input gate, the forget gate, and the previous cell state to calculate the current updated cell state, c_t, at the most recent timestep in model optimization. In this step, the input gate effectively acts as a filter to decide which information from g_t is most relevant to updating the model's understanding of patterns in the data. Finally, the new hidden state, h_t, represents the Hadamard (element-wise) product, which is the hypertangent vector of the current cell state times the output gate's values. h_t acts as the output for the current timestep and provides short-term memory for the next timestep.

3.5 The Process

While LSTM nodes provide an answer to time dependency and vanishing gradient issues, more processes contribute to building any given model. From digital applications of linear regression to cutting-edge neural networks and language models, gradient descent is the foundational algorithm for the training process in machine learning. It works by iteratively adjusting model weights (think β coefficients in linear regression) to minimize the error between its predictions and actual values as measured by a loss function; in this study's case, mean squared error. By computing partial derivatives of the loss function with respect to the model's weights, the algorithm defines the direction, or gradient, in multidimensional space where the error decreases the fastest. The algorithm then updates weights according to this direction and the process repeats until it reaches convergence, the point at which model error stops noticeably decreasing.

Like many other studies of neural networks, this one uses the Adam optimizer, a gradient descent algorithm that enhances the vanilla version by using momentum to accelerate gradient updates when appropriate as well as adaptive learning rates to control their size. Adam is able to adjust learning rates for parameters separately from each other based on past gradients, providing benefits to accuracy and stability in the search for optimal model weights (Kingma & Ba, 2017, p. 7). While not a unique focus of this thesis, the optimizer's adaptability makes it well-suited to handle large, multivariate time-series datasets.

This study integrates Fama-French factors into the model as part of its input feature set, giving the neural network more information to draw from at each time step. The LSTM model uses these factors along with technical indicators like stock prices and volumes to create input vectors of the following format:

X_t = [$Open_t$, $Hight_t$, Low_t, $Volume_t$, $Dividends_t$, Stock $Splits_t$, Mkt-RF_t, SMB_t, HML_t, RMW_t, CMA_t, RF_t]

Where the following variables, pulled in real-time from the yfinance API, represent values corresponding to time period t (one trading day in this case):

Open: Opening price of the asset.

High: Highest price of the asset.

Low: Lowest price of the asset.

Volume: Trading volume of the asset.

Dividends: Dividend payouts for the asset, if applicable.

Stock Splits: Stock split factor, if applicable.

Next, the following variables represent Fama-French factors up through the most recent month of data available (October 2024 in this case), each corresponding to time period t (Fama & French, 2024):

Mkt-RF_t: Market risk premium, the excess return of the stock market over the risk-free rate from US treasuries.

SMB_t: Small Minus Big, which captures the return performance differential between small-cap and large-cap stocks.

HML_t: High Minus Low, the differential of returns between value stocks with high book-to-market ratios and growth stocks with low book-to-market ratios.

RMW_t: Robus Minus Weak, the return differential between firms with robust operating profitability versus those with weak or unreliable profitability.

CMA_t: Conservative Minus Aggressive, the return differential between conservative firms with low asset growth and aggressive firms with high asset growth.

RF_t: Risk-free rate of return on one-month U.S. Treasury bills, providing a baseline metric.

The target variable, $Close_t$, reflects the closing price of the asset at the end of period t. This paper's model predicts future closing prices based on input vectors over a defined user-adjustable time horizon.

By incorporating Fama-French factors in addition to the aforementioned technical indicators, the input vector provides a comprehensive representation of market conditions, both as they pertain to the individual asset being examined and the broader stock market. By adding inputs grounded in the performance of real companies and macroeconomic metrics, the LSTM network is able to implement a holistic approach to capturing market inefficiencies and seeking alpha in defiance of the efficient market hypothesis. This approach bridges technical analysis and economic theory to capture patterns stemming from both micro-level asset performance and macro-level market trends.

3.6 Market Segmentation by Information

This study will test time series neural network application through the lens of information states, which refers to the prevalence of quality data that investors can use to make decisions. Following the logic of the EMH, stocks with high levels of information should be more difficult for a machine to identify inefficiencies in, as more data can be incorporated into the price in real time. This study will test that idea; if it turns out to be false, the dynamic could imply that the logic underpinning the EMH's semi-strong and strong forms is untrue in reality. Assets will fall into broad market segments based on their information states.

First, high-information assets have robust, widely available data surrounding them and high market activity. These include large-cap stocks, which a range of institutional and individual investors track and trade, as well as those with high trading volumes, indicating events or trends that reflect high levels of focus from the market. In this segment, prices should reflect all publicly available information and hold less opportunity for alpha, according to EMH, due to high scrutiny and competition among market participants. On the other hand, low-information assets have relatively sparse data, which could point to lower market efficiency. Examples include small-cap stocks, with lower investor participation, and emerging asset categories like cryptocurrencies, which lack the direct links to reality that stock prices do.

A major source of inefficiency in these segments is the problem of asymmetric information, where some market participants have better access to knowledge about an asset than others, leading to multiple investor groups pushing the market price in different directions based on their levels of knowledge. Thus, the opportunity for inefficiency arbitrage arises among those who can exploit the impacts of noise and limited competition in a market. In high-information segments, high competition and trading volume among informed participants should reduce pricing inefficiencies and more closely align prices with market equilibrium values. On the other hand, fewer participants and lower data availability should make it more likely for mispricings to endure in low-information segments. Cryptocurrencies will be an interesting market segment to examine under this framework as, due to their relative lack of basis in real economics or financials, most market participants have equally low levels of information. While cryptocurrency markets may be inefficient, they could also have fewer opportunities for arbitrage arising from asymmetric information.

LSTMs hold potential advantages for both high- and low-information market segments, and this study's experiment will quantify the differences and sizes of those advantages. In high-information markets, the abundance of data will complement the strengths of neural network architecture in general, as more training data tends to yield greater predictive accuracy. For the time-series component specifically, a longer horizon of past data will allow the model to more accurately capture both short-term and long-term sequential dependencies. While useful, Fama-French factors may have a relatively smaller impact on assets falling into this category since the broad market dynamics that they aim to capture could be closely reflected in prices due to greater scrutiny and trading volumes. In low-information markets, the LSTM architecture could perform exceptionally well by detecting subtle patterns and variable relationships that human analysts using simpler models would miss. These segments often contain assets with noisy price data, and the LSTM's ability to filter out irrelevant information and the impacts of multicollinearity through its memory gate mechanisms and hidden layers makes it very well-suited for this segment. Plus, including Fama-French factors as inputs should allow the model to account for systemic inefficiencies that are more likely to impact low-information assets.

4. Methodology

4.1 Methodology Introduction

This section serves as a blueprint for the implementation of the models in this study, outlining the sources and decisions behind data collection, timeframes, and analytical techniques. This study's methodology rigorously tests the research question about the predictive performance of multi-layered sequential long short-term memory network (MLS-LSTM) models across different informational environments. The hypotheses, outlined in the Introduction section, concern predictive capability in low-information versus high-information markets and whether including Fama-French factors improves generalizability and performance. They stem from the academic debate about market efficiency and the emerging potential for machine learning to uncover arbitrage opportunities in asset pricing. Using the following framework, this thesis will contribute insights to the intersection of financial markets, game theory, and machine learning, while addressing the challenges and limitations inherent to predictive modeling.

4.2 Sample Selection

This study uses stratified sampling of assets to represent both high-information and low-information market segments. All asset data are pulled in real-time from Yahoo Finance using Python's yfinance library, using the maximum possible timeframe to give each training run of the neural network its highest possibility of successfully learning patterns in the input data. Then, all models are tested on an equal timeframe of the most recent two years of data for equivalent evaluation. The first category is high-information stocks, which include 10 of the top holdings from the SPDR S&P 500 ETF (SPY), which tracks the S&P 500 index. These are highly scrutinized and exhibit high trading volumes, pointing to greater market efficiency. There were errors encountered in retrieving data from Berkshire Hathaway Class B (BRK.B), one of the top 10 holdings, so it was replaced with number 11, JPMorgan Chase (JPM). The tickers tested in this category are AAPL, NVDA, MSFT, AMZN, META, GOOGL, TSLA, GOOG, AVGO, and JPM. Next, for low-information stocks, the study takes the top 10 holdings from the Schwab U.S. Small-Cap ETF (SCHA), due to its reputability, performance, and effective reflection of the small-cap market segment according to Evens at Morningstar (Evens, 2024). These small-cap stocks tend to have less market coverage and lower trading volumes than their large-cap counterparts, implying higher levels of inefficiency. The tickers are AFRM, SF, BWXT, DUOL, EVR, AIT, IOT, EXP, ALSN, and PRI. The third category, tested as a subset of the low-information segment, consists of the top 10 cryptocurrencies by market cap excluding stablecoins. These assets are rife with speculation, inefficiency, and limited regulation due to their still emerging nature as a category and their lack of ties to the conventional economy. The cryptocurrency tickers tested are BTC-USD, ETH-USD, SOL-USD, XRP-USD, DOGE-USD, ADA-USD, TRX-USD, SHIB-USB, and AVAX-USD.

Table 1
Full company and cryptocurrency token names.

Small-Cap Stock Tickers	Small-Cap Stock Names	Large-Cap Stock Tickers	Large-Cap Stock Names	Cryptocurrency Tickers	Cryptocurrency Names
AFRM	Affirm Holdings	AAPL	Apple	BTC-USD	Bitcoin
SF	Stifel Financial	NVDA	NVIDIA	ETH-USD	Ethereum
BWXT	BWX Technologies	MSFT	Microsoft	SOL-USD	Solana
DUOL	Duolingo	AMZN	Amazon	BNB-USD	Binance Coin
EVR	Evercore	META	Meta Platforms	XRP-USD	XRP (Ripple)
AIT	Applied Industrial Technologies	GOOGL	Alphabet (Class A)	DOGE-USD	Dogecoin
IOT	Samsara	TSLA	Tesla	ADA-USD	Cardano
EXP	Eagle Materials	GOOG	Alphabet (Class C)	TRX-USD	TRON
ALSN	Allison Transmission	AVGO	Broadcom	SHIB-USD	Shiba Inu
PRI	Primerica	JPM	JPMorgan Chase	AVAX-USD	Avalanche

4.3 Data Sources and Biases

As mentioned in the prior “Sample Selection” subsection, this study pulls from the comprehensive Yahoo Finance database, which one can access using the yfinance library in Python. Refer to the subsection of the Theory/Model section titled “The Process” for a detailed description of the input vectors used. The maximum possible historical timeframe was chosen for each asset, allowing the LSTM network to learn from an extensive range of market cycles and volatility environments. Generally, more training data, and thus more deep learning, allows neural networks to better generalize out of sample.

For testing the architecture with Fama-French factors, a Python script automatically merges the data by date with a set of daily Fama-French Five Factor inputs from January 1, 1980, through the end of November 2024. This enriches a given dataset with information about broad market anomalies, including the performance of companies that fall into various categories captured in the factors. During training, the LSTM is able to learn which, and to what degree, factor values influence stock prices. In order to avoid making up data, the final Fama-French factors copy over cells up to a given testing date; for example, if one were training and testing a model on stock data through December 5, 2024, the script would copy over the Fama-French values from November 30, 2024. This is not perfectly accurate. However, since the data copied are beyond the training and validation samples, it does not negatively impact the performance of model learning, and it has a negligible impact on generalization and evaluation metrics. Including these factors should allow the model to move its predictions beyond the limitations of technical analysis by giving it insight into systemic patterns.

A couple of biases exist that minimally impact the datasets in this study. The first is survivorship bias in Fama-French factors, as companies may change in their market segments as defined by the Fama-French methodology; for example, if a company starts out small and grows into a giant, it would end up switching sides in calculating the small minus big factor. Although survivorship bias exists within the components of Fama-French factors, that is because the factor values aim to capture broad market dynamics. If a bunch of small companies became big but remained in their originally assigned categories, the small minus big factor would become useless in understanding market anomalies. Second, let’s acknowledge the fragmented nature of cryptocurrency markets. Several of the top 10 cryptocurrencies by market cap and trading volume are stablecoins like Tether (USDT), which is pegged to the value of the U.S. Dollar. Other stablecoins are also either pegged to the U.S. Dollar or to other assets, meaning they behave differently from conventional cryptocurrencies such as Bitcoin (BTC-USD) and

Ethereum (ETH-USD), which host frequent volatile price fluctuations. While some could argue that omitting stablecoins from analysis introduces bias into the sample of the top cryptocurrencies, including them would be akin to simply including the U.S. Dollar or whatever other underlying currencies the stablecoins are pegged to. At that point, the analysis simply turns into modeling national currencies, which are impacted by the macroeconomy and central banks to a much greater degree than cryptocurrencies.

4.4 Data Preparation

The input vectors for this LSTM model were designed with features that aimed to effectively balance computational efficiency with capturing a comprehensive view of micro-level asset dynamics and macro-level market trends. A detailed explanation is located in the Theory/Model section of this paper, but an overview of the input vector X_t is as follows:

X_t = [Open_t, Hight_t, Low_t, Volume_t, Dividends_t, Stock Splits_t, Mkt-RF_t, SMB_t, HML_t, RMW_t, CMA_t, RFt]

These input features include technical indicators as well as broader market factors from the Fama-French Five Factor dataset. Asset-specific variables include opening price, high, low, trading volume, dividends, and stock splits, and capture historical price action sequences. This tracks with the methodology of traditional technical analysis. The study moves beyond this by including Fama-French factors to capture anomalies in the broader stock market. These factors give information about market risk premium, size, value, profitability, investment style, and risk-free rates of return. Including this information allows the model to account for broader market conditions to learn beyond the scope of technical analysis, which was employed in most prior applied machine learning papers concerning the stock market. This study's holistic approach will allow for an assessment of the limitations of the purely technical approach across market information segments.

To ensure the dataset is compatible with the LSTM architecture, data normalization is necessary. The Min-Max scaler, which scales all numerical inputs to a range between zero and one, was chosen for its simplicity and widespread use in the field. This ensures that variables with larger magnitudes than others do not disproportionately impact the model optimization process. For final predictions and assessment, all model outputs are inversely scaled using the Min-Max scaler to return them to their proper magnitudes.

As significant positive alpha—returns above the benchmark buy-and-hold strategy—is both highly anomalous and coveted, the code for this study must take care to ensure that testing data remains completely separate and does not influence model predictions. To accomplish this, each asset's dataset is divided into distinct subsets. The first 81% of the data are known as the training set and are used to train the model on patterns and variable relationships. The next 9% (the final 10% of the first 90% of the data) are the validation set, used to test model accuracy during each iteration of the gradient descent process. Although the validation set is outside of the training set, it still provides information to the gradient descent algorithm and thus can influence model weights. To ensure testing on exclusively out-of-sample performance, the final 10% of the data make up the testing set, which is used to assess predictive accuracy and backtesting scenarios.

4.5 Model Implementation

This study leverages the multivariate time series forecasting capabilities of a multi-layered sequential long short-term memory (MLS-LSTM) neural network. This architecture is ideal for capturing the temporal dependencies inherent to financial time series data, which contain both short-term reactions and long-term trends. The network's memory gate mechanisms and sequential processing design make it well-suited for noisy stock and cryptocurrency data. Refer to the Theory/Model section for an in-depth explanation.

The methodology used for this paper instructs the LSTM to process sequences with 15 trading days of input data to predict the subsequent 5 trading days of closing prices. This means that the model is explicitly instructed to learn short term dependencies within 15-day sequences while retaining information about general or longer-term patterns in its memory mechanisms. After training, model predictions within the testing set timeframe are evaluated for accuracy using R^2, mean absolute error (MAE), and mean squared error (MSE) as diagnostic metrics, with alpha returns being the primary financial metric. The predictions evaluated are taken from the end of each predicted 5-day sequence; for example, if a model uses its learned weights to predict based on a sequence from January 1 to 15 (assuming all are trading days for simplicity), then its prediction sequence will run from January 16 to 20. We then use the observed value from January 20 from the test set for evaluation. This means that the first 19 days (timesteps + future horizon - 1) of values from the test set are omitted. To maximize accuracy and the extensiveness of the validation and testing steps, the model trains and tests on sliding windows. This means that, after the previously mentioned window, it would then move the window forward one day, predicting a value for January 21 based on January 2-16, then the 22nd based on the 3rd through the 17th, and so on. The sliding window approach allows the model to use as much available data as possible, enhancing robustness and reflecting practical real-world implementations that would demand daily use.

As for evaluation metrics, note that R^2 is only used as a diagnostic metric here. Due to mathematical artifacting, it is not an accurate measure of the usefulness of LSTM predictions. Designed for linear regression contexts, R^2 can produce unintuitive results, such as values below -1 when applied to this study's context. R^2's assumption that minimizing variance yields a better model causes an inherent mismatch with LSTMs, which optimize for nonlinear trends and variable relationships. Negative R^2 often arises from heavily weighing noise or being disproportionately skewed by outliers, both of which occur fairly regularly in financial time series data.

What good is a predictive model without a way to make use of it and a way to test that usefulness? That's where backtesting comes in. This study uses a fairly basic backtesting strategy over the timeframe of given testing sets to evaluate the practical implications of model predictions. A Python script computes daily returns from the end of each 5-day forecast sequence and converts those returns into trading signals based on thresholds derived from rolling 30-day volatility. Buy signals generate when the forecasted 5-day return exceeds the computed volatility for a given day, and an additional threshold ensures a buy order only executes when the price on a buy signal is at or below 0.95 times the last sale price, if applicable. Sell signals generate when the forecasted 5-day drop exceeds 1.5 times the rolling 30-day volatility, with an additional threshold to ensure sales only execute at or above 1.05 times the most recent buy price. These thresholds bias the model toward a buy-and-hold strategy. Given its lower sensitivity for sell signals, the strategy also includes a short-selling component to capitalize on the increased certainty required to predict a downturn.

In addition to the statistical metrics of MAE and MSE, the study uses financial evaluation metrics to evaluate model performance with returns from the backtesting strategy. Note that due to data constraints, dividend payments were not incorporated into return metrics, so if anything, the experimental results will show a slight underperformance compared to a real trading environment. Alpha measures the excess percentage-point return over a benchmark; in this case, a buy-and-hold strategy on a given asset. Significant alpha would imply consistent market inefficiencies. Finally, the Sharpe ratio evaluates risk-adjusted returns relative to the risk-free rate earned from compounded 1-month U.S. Treasury bills. These metrics were all selected given their widespread academic and industry use to provide a balanced picture of predictive accuracy as well as practical market implications.

5. Results & Analysis

5.1 Intro and Descriptive Statistics

This section will primarily examine the hypothesis that multi-layered sequential long short-term memory recurrent neural networks (MLS-LSTM RNNs) incorporating Fama-French factors will outperform technical-only models across market segments. The secondary hypothesis is whether LSTMs are more effective in generating returns in excess of a buy-and-hold strategy, or alpha, in low-information market segments compared to other segments. This is where the study dives into more traditional econometrics, with descriptive statistics, correlation analysis, financial return metrics, segmented performance, analysis of variance (ANOVA), and linear regression analysis with categorical predictor variables. Note that this is not an analysis of the data making up the base level of this study, which were fed into the LSTM program for training. Instead, this section analyzes performance results of the LSTM architecture across the conditions of Fama-French inclusion and market information segments. This analysis tested an LSTM-based trading strategy across a timeframe beginning around January 2023, with small variations due to data availability inconsistencies, and ending in December 2024.

Figure 2. *Graph of buy and sell signal generation based on predicted prices for 5 days into the future from a given day.*

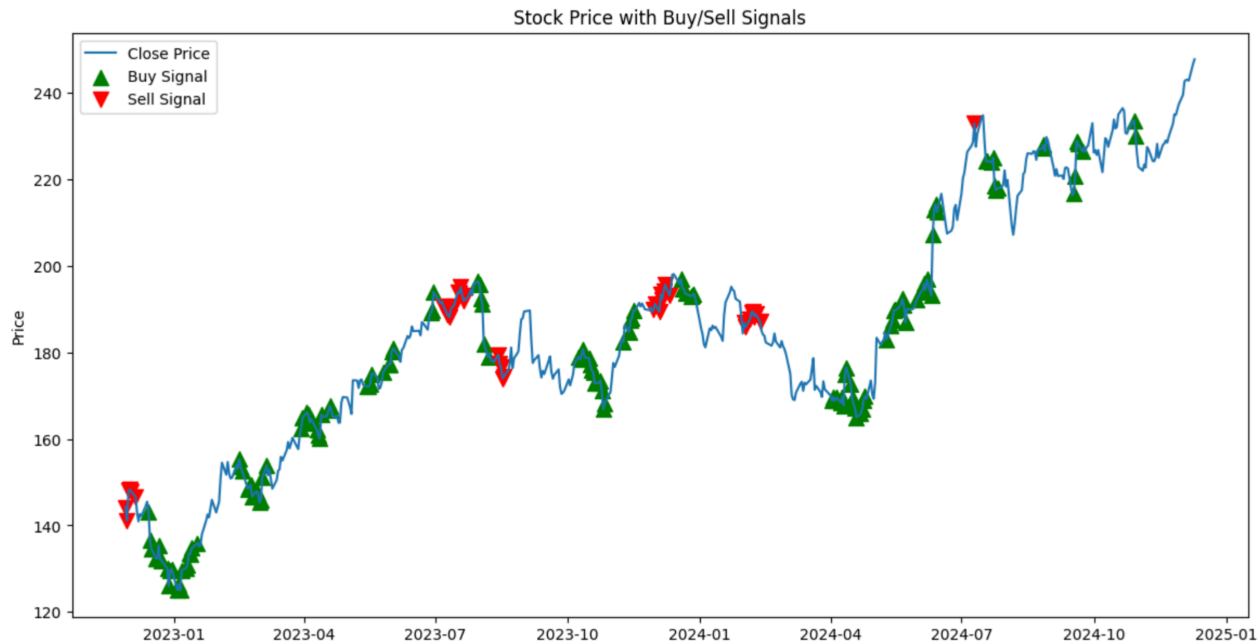
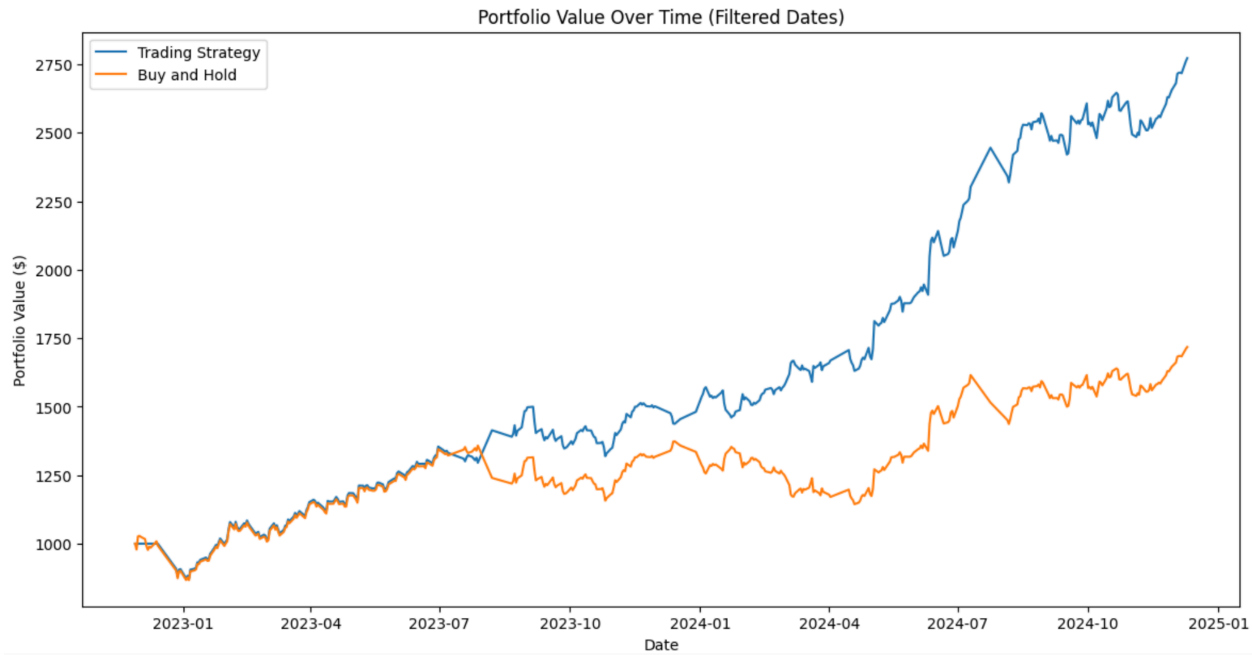


Figure 3. *Log of trading actions taken by the strategy*

```
2022-12-14: Buy signal - Bought 6.9828 shares at $143.21
2023-07-07: Sell signal - Sold holdings of 6.9828 shares at $190.68, Cash: $1331.47
2023-07-07: Sell signal - Opened short position of 6.9828 shares at $190.68, Portfolio Value: $1339.36
2023-08-07: Buy signal - Covered short position of 6.9828 shares at $178.85
2023-08-07: Buy signal - Bought 7.9065 shares at $178.85
2023-11-30: Sell signal - Sold holdings of 7.9065 shares at $189.95, Cash: $1501.84
2023-11-30: Sell signal - Opened short position of 7.9065 shares at $189.95, Portfolio Value: $1497.25
2024-04-02: Buy signal - Covered short position of 7.9065 shares at $168.84
2024-04-02: Buy signal - Bought 9.8836 shares at $168.84
2024-07-10: Sell signal - Sold holdings of 9.8836 shares at $232.98, Cash: $2302.68
2024-07-10: Sell signal - Opened short position of 9.8836 shares at $232.98, Portfolio Value: $2260.18
2024-07-24: Buy signal - Covered short position of 9.8836 shares at $218.54
2024-07-24: Buy signal - Bought 11.1897 shares at $218.54

Total Return of Trading Strategy: 177.25%
Total Return of Buy-and-Hold Strategy: 71.80%
```

Figure 4. *Performance graph of the neural network-based trading strategy versus baseline stock performance, starting with $1000.*



Before delving into the statistics, here's an example of how the model-based backtesting trading strategy works, in the case of Apple with Fama-French factor inputs. Figures 2 through 4 should give an impression of how testing works after training a model instance for each trial. Note how, in Figure 4, overperformance increases most drastically when the machine enters short positions near the end of 2023 and in June 2024.

Table 2

Descriptive Statistics of Key Metrics (N = 59)

Metric	Mean	SD	Min	25%	50%	75%	Max
MSE	6,358,919.790	43,947,791.980	6.44 × 10^-12	85.410	246.210	1,834.880	336,000,000.00
MAE	398.130	2,288.270	1.65 × 10^-6	7.280	13.890	38.920	17,098.82
Total Return	129.840	150.560	2.500	45.820	89.280	144.690	801.83
Total Return BH	123.060	135.640	-4.880	65.650	88.520	138.910	804.48
FF_YN	0.508	0.504	0.000	0.000	1.000	1.000	1.00
Low_Info	0.983	0.820	0.000	0.000	1.000	2.000	2.00
Alpha	6.779	64.814	-132.71	-27.620	6.560	26.820	255.49

Note. SD = standard deviation; Min = minimum value; 25% = 25th percentile; 50% = 50th percentile; 75% = 75th percentile; Max = maximum value; MSE = mean squared error of neural network predictions; MAE = mean absolute error of neural network predictions; R^2 = coefficient of determination of neural network predictions; Total Return = total percentage return of neural network trading strategy; Total Return BH = total percentage return of buy and hold strategy; FF_YN; whether a neural network uses Fama-French factors as input features; Low_Info = market information segmentation of tested assets; Alpha = excess percentage point return of neural network trading strategy over buy and hold strategy.

First, as seen in Table 2, both MSE and MAE have high means and variability across the whole dataset. This points to inconsistency across the dataset of model results; however, a median (50th percentile) mean absolute error of 13.89 is fairly reasonable given the magnitude of many asset prices—think stocks priced in the hundreds of dollars and cryptocurrencies like Bitcoin in the tens of thousands of dollars. Also, returns from the model-based trading strategy show a 6.78% average outperformance compared to the buy-and-hold strategy, indicating that the LSTM may be able to pick up on inefficiencies across market segments.

Figure 5. Histogram for alpha, excess returns over a buy-and-hold strategy.

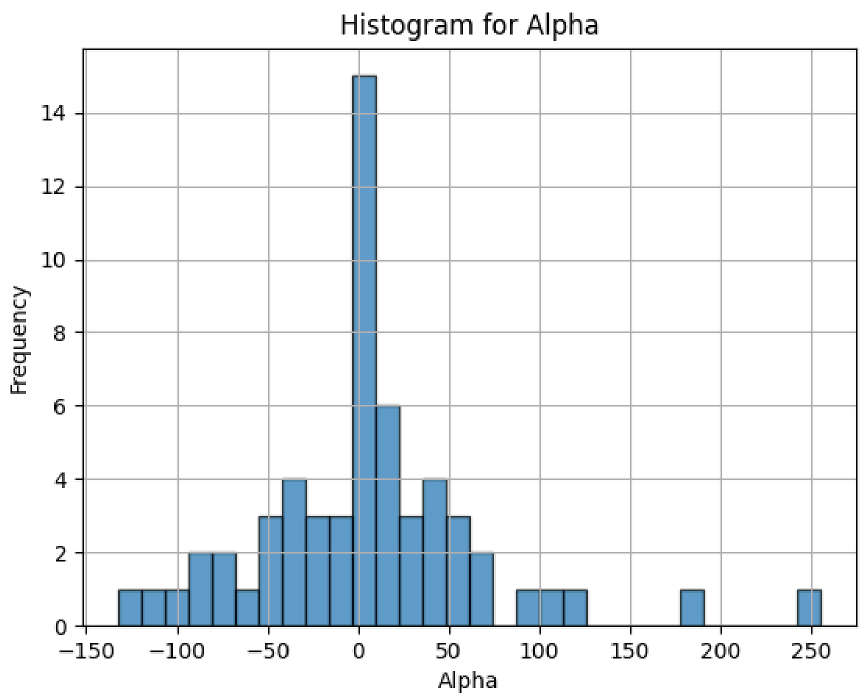


Table 3

Descriptive Statistics by Fama-French 5 factor inclusion (N = 59)

Metric	Mean	SD	Min	25%	50%	75%	Max
Technical only, no Fama-French factors (FF_YN = 0) (N = 29)							
MSE	1,345,307.780	7,218,292.920	0.0012	81.500	207.650	1,551.140	38,876,506.230
MAE	185.350	836.150	0.01860	6.400	11.210	33.840	4,525.640
R^2	-0.513	3.698	-18.4630	-0.308	0.621	0.846	0.960
Alpha	-17.850	55.870	-132.7100	-58.390	-3.310	18.200	99.260
Fama-French factors included (FF_YN = 1) (N = 30)							
MSE	11,205,411.400	61,343,907.880	6.44 × 10^-12	108.770	713.230	2,054.750	336,000,000.000
MAE	603.820	3,115.820	0.000002	8.640	21.640	43.170	17,098.820
R^2	-0.543	1.736	-5.545000	-1.115	0.369	0.561	0.861
Alpha	30.580	64.770	-53.310000	-1.570	8.760	38.900	255.490

As shown in Table 3, the average alpha—returns in excess of a benchmark strategy (buy-and-hold)—of 30.58 percentage points for models using Fama-French factors shows major outperformance compared to technical-only models at -17.85 percentage points. This suggests that the factors, which were designed to capture anomalies in the stock market, help LSTM networks to capture inefficiencies. However, this also appears to come at the cost of higher prediction errors in MSE and MAE. While it goes against intuition that models with better returns would exhibit smaller errors, it may be the case that Fama-French factors introduce broad market information that helps to generate alpha while making the models more generalizable and less sensitive to noise. The relationship between alpha and error metrics remains a compelling path for further research.

Table 4

Descriptive Statistics by information segment (N = 59)

Metric	Mean	SD	Min	25%	50%	75%	Max
Low information, small-cap stocks (Low_info = 0) (N = 20)							
MSE	3,279.16	6,065.49	43.85	210.95	864.87	1,911.96	24,314.44
MAE	38.08	38.17	5.17	11.40	23.99	40.62	147.61
Alpha	0.22	42.87	-68.79	-25.18	-0.49	14.78	123.49
High-information, large-cap stocks (Low_info = 1) (N = 20)							
MSE	1,129.93	1,967.03	27.68	94.12	201.89	1,399.04	8,394.45
MAE	22.54	20.93	4.81	8.15	11.87	34.12	85.15
Alpha	-7.71	52.06	-107.00	-43.70	0.06	17.63	99.26
Low-information, cryptocurrencies (Low_info = 2) (N = 19)							
MSE	19,741,478.20	77,101,163.65	6.44 × 10^-12	0.005	16.54	5,334.40	336,000,000.00
MAE	1,172.49	3,992.05	0.000002	0.053	3.02	62.38	17,098.82
Alpha	28.94	89.08	-132.710000	6.13	22.21	46.32	255.49

Table 4 indicates that, through the lens of information segments, the models perform the best with cryptocurrencies, at an average alpha of 28.94 percentage points. Without real financial information like that which companies publish, cryptocurrencies are a lower-information segment than small-cap stocks. Plus, this finding aligns with the hypothesis that LSTMs can better exploit inefficiencies in low-information environments. The small-cap segment exhibits a small mean overperformance of 0.22 percentage points, while the LSTM approach appears to underperform at -7.71 percentage points average alpha in high-information stocks. It's important to note the extraordinarily high MAE and MSE in the cryptocurrency sector, which is primarily influenced by outliers with very high price magnitude per unit, such as Bitcoin (BTC-USD) and Ethereum (ETH-USD). Meanwhile, large-cap stocks give the model its lowest average MAE at 22.54, as greater information availability could make it both easier to predict short-term prices but more difficult to generate alpha.

Among trials including Fama-French factors, the LSTM architecture generated the worst alphas on Stifel Financial (SF) at -53.31, Microsoft (MSFT) at -43.52, and Meta (META) at -26.36. On the other hand, overperformance on the best trials was much greater in magnitude. The neural network program showed its best performance on simulated trading of Dogecoin (DOGE-USD), a cryptocurrency token based on a popular Shiba Inu dog meme, with an alpha of 255.49 percentage points over the testing period. The next best performances were on Bitcoin (BTC-USD) at 183.81 and Apple (AAPL) at 123.49.

So far, the descriptive statistics somewhat support both hypotheses—that LSTMs gain an advantage in generating alpha when incorporating Fama-French factors as input features, and especially in the low-information segment that is cryptocurrency. However, further analysis is necessary to determine whether these differences are statistically significant and how various segments specifically impact performance.

5.2 Financial Performance

This analysis focuses on alpha, with Sharpe ratios calculated on a macro-level using the variance of segment returns. It provides specific insight into model performance based on Fama-French factor inclusion and market information segmentation.

$$Sharpe\ Ratio\ = \frac{R_p - R_f}{\sigma_p}$$

The Sharpe ratio provides a standardized way to evaluate risk-adjusted returns by comparing excess returns to the standard deviation of those returns. Here, R_p is the portfolio return of the LSTM strategy, R_f is the risk-free rate of return taken from the Fama-French factor dataset which is based on 1-month U.S. Treasury Bills, and σ_p is the standard deviation of portfolio returns. Many analyses add up periodic returns, but for the sake of simplicity and this paper's goal of strategy comparison, this study takes the standard deviation of total returns from each trial of the neural network trading strategy. In this case, the average risk-free rate of return over the testing period was 4.2%. Compounded over the 24 month testing period, this ends up at a cumulative risk-free return of 8.7%. Maverick (2024) rates Sharpe ratios as follows: A ratio below 1 is suboptimal, above 1 is good, above 2 is very good, and above 3 is excellent. The Sortino ratio is similar but is excluded from the analysis due to too low of a sample size to accurately calculate downside deviations.

Table 5

Financial return metrics by Fama-French inclusion

Fama-French inclusion	Alpha	Sharpe Ratio
No	-17.847	0.684
Yes (factors included)	30.585	0.947

Note. Alpha = average excess return over the buy-and-hold strategy across the testing set; FF_YN represents inclusion of Fama-French factors where 1 is true and 0 is false.

As discussed in the descriptive statistics subsection, models without Fama-French factors result in negative alpha at -17.847 percentage points, while those with the factors achieve positive alpha of 30.585. Table 4 shows that the Sharpe ratio also rises from 0.684 to 0.947, reflecting better performance with Fama-French inputs even when adjusted for risk. These numbers still land in the industry-defined range of below 1 as being suboptimal; however, the ratio with factors included is close enough to 1 to be acceptable.

Table 6

Financial return metrics by information segment

Information Segment	Alpha	Sharpe Ratio
Large-caps	0.219	0.722
Small-caps	-7.712	1.435
Cryptocurrencies	28.939	1.052

Note. Low_info represents the information segment, where 0 is high-information large-cap stocks, 1 is low-information small-cap stocks, and 2 is low-information cryptocurrencies.

Table 5 shows that, despite their underperformance as measured by alpha, the neural network trading strategy performs better on small-cap stocks than other segments when adjusted for risk in both the positive and negative directions. With a Sharpe ratio of 1.435, it rises into the "good" category, along with cryptocurrencies at a ratio of 1.052.

Table 7

Financial return metrics by information segment only when Fama-French factors are included (FF_YN = 1)

Information Segment	Alpha	Sharpe Ratio
Large-caps	7.417	0.759
Small-caps	8.769	1.997
Cryptocurrencies	75.568	1.286

Table 7 provides a view into a specific subset of results, taking only those that include Fama-French factors as input features for the LSTM while categorizing them by information segment. First of all, note that alpha is positive for all segments when using Fama-French inputs; this is a divergence from the entire dataset, where negative alpha values from trials without those inputs pulled the averages down. Including the factors provides a massive boost to cryptocurrency trading, as alpha rises by 46.629 percentage points to 75.568. For small-cap stocks, alpha rose by 7.198 to 7.417, and alpha on large-caps rose by 16.481 to 8.769. Both stock categories show smaller gains compared to cryptocurrencies, pointing to the hypothesis that cryptocurrency markets are especially inefficient. Small-cap stocks and cryptocurrencies, with Sharpe ratios at 1.997 and 1.286 respectively, suggest that models perform well on a risk-adjusted basis when including Fama-French factor inputs.

Fama-French factors are proving their value for LSTM performance, as categories that include these factors generate more alpha with better risk-adjusted returns across all financial metrics and market types. Within segments, cryptocurrencies consistently produce the highest returns in excess of a buy-and-hold strategy, with high downside risk-adjusted returns but lower returns than small-cap stocks when adjusting for bidirectional risk. However, with higher Sharpe ratios than the high-information segment's ratio of 0.722, both low-information segments appear to contain more inefficiencies for neural networks to exploit. Over the testing period, model simulations showed neutral to small performance gains in the large-cap segment, likely because high-information markets are more efficient due to higher scrutiny and trading volumes.

5.3 ANOVA and Pairwise t-Tests

Analysis of variance (ANOVA) will help to determine whether the differences identified in alpha returns between market segments are statistically significant.

Table 8
ANOVA for Fama-French inclusion segments

Source	Sum of Squares	df	F	p-value
Fama-French inclusion	34587.34	1	9.43**	0.003268
Residual	209064.54	57	-	-

** $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$*

Table 8 shows that the test exhibits 1 and 57 degrees of freedom; at the 0.05 significance level, the critical F value is 4.01 (University of Sussex, 2005). Since this difference between the alpha return groups exceeds this with an F-statistic of 9.43 and a p-value of 0.00327, we reject the null hypothesis that there is no difference in alpha returns between the groups with and without Fama-French factor inclusion. This already supports the primary hypothesis of this study, which is that LSTM models incorporating Fama-French factors will outperform technical-only models across market segments.

Table 9

ANOVA for market information segments

Source	Sum of Squares	df	F	p-value
Information Segment	14390.77	2	1.76	0.181843
Residual	229261.12	56	-	-

In Table 9, the F-statistic is 1.76 with a p-value of 0.182. Since this is less than the critical value of 3.16 for 2 by 56 degrees of freedom (University of Sussex, 2005), and the p-value is greater than 0.05, we fail to reject the null hypothesis that there is no difference in excess returns across market information segments. When looking at all trials, there is insufficient evidence to suggest significant differences in alpha returns.

Table 10

Pairwise t-tests for market information segments

Comparison	t-statistic	p-value
Small-caps vs Large-caps	1.56	0.130117
Small-caps vs Cryptocurrencies	1.27	0.214693
Large-caps vs Cryptocurrencies	-0.53	0.602122

For Table 10's pairwise comparisons, we fail to reject the null hypothesis that the mean alpha returns between pairs of information groups are equal. In other words, the differences in mean excess returns between groups are not statistically significant.

Table 11
ANOVA for market information segments with Fama-French factors included (FF_YN = 1)

Source	Sum of Squares	df	F	p-value
Information Segment	30361.6	2	4.49*	0.020735
Residual	91295.32	27	-	-

Now, the analysis moves to examining market segments only when including Fama-French factors. With the degrees of freedom shown in Table 11, the critical F value is 3.35 (University of Sussex, 2005). Since the observed F-statistic of 4.49 exceeds the critical value and the p-value of 0.0207 is below 0.05, we can successfully reject the null hypothesis that there is no difference in alpha returns among market information groups when only Fama-French factor models are included. This shows that there is a statistically significant difference in excess returns across market segments when examining the subset of models that incorporate Fama-French inputs.

Table 12
Pairwise t-tests for alpha by market information segments, only with Fama-French factors (FF_YN = 1)

Comparison	t-statistic	p-value
Small-caps vs Large-caps	0.08	0.940962
Small-caps vs Cryptocurrencies	-2.37*	0.036076
Large-caps vs Cryptocurrencies	-2.26*	0.040213

Analyzing the subset of model returns with Fama-French factor inputs, Table 12 shows no statistically significant difference in alpha returns between small-cap stocks and large-cap stocks; thus, we reject the null hypothesis. However, there are statistically significant differences between cryptocurrencies and small cap-stocks as well as cryptocurrencies and large-cap stocks.

This means that the mean excess return in the cryptocurrency segment differs significantly compared to both small-cap and large-cap stocks, so we can successfully reject the null hypothesis in both pairs.

The above analysis boils down to a few main insights. First, Table 8 affirms that including Fama-French factors significantly enhances returns in excess of a buy-and-hold strategy, supporting this paper's primary hypothesis. Over the entire dataset, Table 9 shows that differences in alpha returns across market information segments are not statistically significant. However, tables 11 and 12 highlight significant differences by segment when looking at the subset of only models with Fama-French factor inputs. The LSTM architecture performs significantly better with cryptocurrencies compared to other segments within the Fama-French subset.

5.4 OLS Regression

Table 12

Regression Results (Robust Standard Errors)

Dep. Variable:	Alpha	R-squared:	0.2420
Model:	OLS	Adj. R-squared:	0.1860
Method:	Least Squares	F-statistic:	5.7650
No. Observations:	59	Prob (F-statistic):	0.0006
Df Residuals:	54	Log-Likelihood:	-321.1600
Df Model:	4	AIC:	652.3000
Covariance Type:	HC1	BIC:	662.7000

	coef	std err	t	P>| t |	[0.025	0.975]
Intercept	-22.6498	12.455	-1.819	0.075	-47.620	2.321
Fama-French factors	45.2632**	15.585	2.904	0.005	14.017	75.509
Small-cap stocks	-7.8342	15.277	-0.513	0.610	-38.462	22.793
Cryptocurrencies	20.4600	20.625	0.992	0.326	-20.891	61.811
MAE	0.0062*	0.003	2.145	0.036	0.000	0.012

Omnibus:	14.617	Durbin-Watson:	0.611
Prob(Omnibus):	0.001	Jarque-Bera (JB):	24.071
Skew:	0.798	Prob(JB):	5.93e-06
Kurtosis:	5.691	Cond. No.	7.82e+03

Note. Standard Errors are heteroskedasticity robust (HC1).

The regression equation for this model is:

$$Alpha = -22.6498 + 45.2632\times\text{C(Fama-French Factors)[T.1]} + -7.8342\times\text{C(Small-cap stocks)[T.1]} + 20.4600\times\text{C(Cryptocurrencies)[T.2]} + 0.0062\times\text{MAE} + \varepsilon$$

Since this paper's LSTM was designed as a very complex regression model, Table 12 shows a regression of regressions—a meta-regression. This explores how Fama-French factor inclusion, information segmentation, and MAE predict returns in excess of a buy and hold strategy, or alpha, using heteroskedasticity-robust standard errors. This ordinary least squares (OLS) model has an R^2 value of 0.242, meaning that the independent variables explain 24.2% of the variance in alpha across LSTM model trials. The number drops to 0.186 when adjusted for the number of predictor variables, where the model still accounts for a moderate portion of the variance in alpha. As the model's F-statistic of 5.765 is greater than its corresponding critical value of 2.546, its p-value of 0.0006 shows the model is very statistically significant at the 0.05 level. This means that the predictor variables together explain a significant amount of variance in the dependent variable.

As for categorical variable effects, including Fama-French factors increases alpha by 45.26 percentage points on average. This effect is statistically significant at the 0.05 confidence level, with a p-value of 0.005. Plus, it's consistent with the ANOVA results from Table 7, which also point to a statistically significant difference in alpha returns between Fama-French factor inclusion and omission. These findings reinforce the primary hypothesis that giving time-series neural networks information about broad market dynamics boosts their predictive capabilities.

Next, small-cap stocks yield an average alpha of -7.83 compared to large-cap stocks, but this effect is not statistically significant at as $p = 0.610$. Cryptocurrencies show 20.46 percentage points higher alpha than large-cap stocks on average, but this difference is also not statistically significant at a p-value of 0.326. These findings imply that, after controlling for FF_YN and MAE, market segments do not significantly impact alpha; however, Table 11's ANOVA results conflict with this in the case of cryptocurrency.

Finally, every increase of one in MAE corresponds to a 0.0062 unit increase in alpha. Despite its small effect size and counterintuitive nature, this effect is statistically significant with a p-value of 0.036. This could reflect that models generating higher alpha benefit from greater volatility, which also tends to increase prediction error.

As a whole, the model is highly statistically significant with p-value (0.0006) under 0.001 and a moderate R^2 of 0.242 and adjusted R^2 of 0.186, showing that it contains some strong, albeit incomplete, explanatory power for the variance alpha returns. The main limitation in this experiment has been time. Simply put, more time and more trials across more assets are necessary to generate conclusive evidence about the impacts of market segmentation on alpha. While excess returns in the cryptocurrency segment are statistically significant in their difference from the other two segments, more data is needed for conclusive ANOVA and regression results about all groups. Despite the inconclusive results for the secondary hypothesis about market information segmentation's impacts on alpha, there is statistically significant evidence that incorporating broad market information through Fama-French factors boosts alpha with a remarkable effect size. This is likely due to more information allowing the LSTM model to discover more hidden relationships and more accurately create abstract interaction terms with an ability to spot and capitalize on inefficiencies in the market.

5.5 Limitations and Assumptions

An important caveat is that these results are inherently tied to the specific sample and context of this study. First, the LSTM architecture is designed to capture complex non-linear relationships and time dependencies. If simpler models such as linear regression, ARIMA, or even a vanilla neural network were used in predicting asset prices, they would likely fail to capture many relationships within variable sequences. Conversely, a more complex model such as a Transformer could uncover additional interactions and patterns, potentially hurting generalizability and certainly hurting experimental feasibility due to computational requirements. Also, the chosen simulation period impacts how the effects of time allow trading returns to compound. A longer time frame for simulated backtesting could show more drastic returns while reflecting the same trends and differences of means uncovered in this analysis. Plus, more market dynamics captured by a longer time frame, such as periods of high volatility, could amplify inefficiencies and increase alpha in the market. The high performances among cryptocurrency results point to this, as the sector was much more volatile than the other segments over the testing period.

Next, the study assumed that incorporating Fama-French factors captures broad market dynamics and that the efficient market hypothesis serves as a baseline framework to test against. Other assumptions about market behavior and what input variables to feed into the models could shift the focus of the study. For example, applying a behavioral finance perspective with information about investor behavior or sentiment could shift the focus to psychological factors rather than the informational ones with which the efficient market hypothesis is concerned.

The sample size is inherently limited by time, computational requirements, and the scope of this project. If it were expanded to include more tickers in each category, as well as more categories such as bonds and international currencies, the LSTM approach could identify other patterns that generate inefficiencies and generate a broader picture of its usefulness for investors. Plus, this experiment operates under assumptions about high availability and reliability of the information provided by Python's yfinance package about developed financial markets. Emerging markets such as developing countries, newly IPO'd stocks, or small-cap cryptocurrencies, could yield vastly different results due to information asymmetry and noisy data.

While the findings of this study are very promising for applying this approach to trading, especially in cryptocurrency markets, these findings are still context dependent. One person's decision to use further testing and deployment of time-series neural networks in real markets DOES NOT constitute financial advice. Future research should explore variations of sampling and time frames to better understand the ability of LSTMs to capture inefficiencies across asset markets.

6. Conclusion

6.1 Summary of Research Aims and Findings

The primary goal of this study was to investigate whether multi-layered sequential long short-term memory (MLS-LSTM) recurrent neural networks (RNNs) enhanced with Fama-French factor data could outperform technical-only versions of those models in generating alpha across different market information segments. In other words, do time-series neural networks perform better when given broad market and economic data, and how do those performances differ across asset groups with different levels of information surrounding them?

To answer this question, the study tested the hypothesis that incorporating macroeconomic and systemic factors would improve model generalizability and performance, particularly in low-information market segments that may be more prone to inefficiency and mispricing. The findings from this experiment reveal several valuable insights about market efficiency and the economic implications of machine learning, as the neural network architecture demonstrated high adaptability with variable performance depending on the nature of input features and assets' informational environments.

First, including Fama-French factors enhanced the LSTM architecture's potential to generate alpha—returns above a buy-and-hold strategy on the same asset—by an average of 45.26 percentage points at a statistically significant p-value of 0.005 over the two-year testing period. This confirms their ability to capture broader market patterns, as well as the impact of those patterns on asset prices, beyond the scope of technical analysis (predicting prices using only past price and volume data).

Second, the models observed the most pronounced inefficiencies and generated the most alpha in the cryptocurrency market, defined as a low-information segment due to its novelty and lack of the connection to real financials that stock prices possess. When examining the subset of models with Fama-French input factors, the differences in means between cryptocurrencies and large-cap high-information stocks as well as small-cap low-information stocks were statistically significant. No statistically significant differences between mean alphas on small-caps and large-caps were observed, likely due to a need for more trials in the future. On Fama-French models, the average alpha was 7.417 for large-caps, 8.769 for small-caps, and 75.568 for cryptocurrencies.

In small-cap stocks, the LSTM models showed moderate improvements in generating alpha compared to technical-only versions. Frequently scrutinized and traded at high volumes, large-caps stocks seemed to pose greater challenges for the models in generating and improving alpha with more input features. Despite these trends, the differences in performance between small-cap and large-cap segments were not statistically significant; however, with arguably lower information availability than either of those, differences in mean performance for cryptocurrencies were statistically significant.

While the findings align with a central tenet of the efficient market hypothesis—that information is incorporated into price action in real time—they also work against both its strong and semi-strong forms. The results for models with Fama-French inputs show a gradient of overperformance, from the highest information state at the low end of performance to the lowest information state at the high end, showing that markets are likely more efficient with more concentrated informational environments. The overperformance of models incorporating inputs about the broader market contradicts the semi-strong efficient market hypothesis, as alpha is possible with publicly available information. Thus, the findings also work against the strong-form hypothesis, which states that alpha is impossible even with insider information.

6.2 Contribution to Theory

Markets are not uniformly efficient. That is this study's main theoretical conclusion, drawn from performance differences between market information segments. In high-information segments like large-cap stocks, market prices closely reflect available data and limit the LSTM's ability to generate alpha. However, in low-information markets, particularly the volatile sector of cryptocurrencies, the study demonstrates that inefficiencies remain very exploitable. While high-information markets may approximate semi-strong efficiency, this level of the efficient market hypothesis does not hold in markets with lower transparency and scrutiny. Perhaps the hypothesis is due for a modern update that incorporates the nuances of informational segmentation.

Next, the random walk theory asserts that stock prices follow unpredictable paths. While noise certainly exists in markets and this can seem true to the observer, this study has shown that LSTM models can uncover patterns within price movements when enriched with macroeconomic financial data. With the right algorithms, it seems possible to recognize some predictability in asset prices under certain conditions.

The Fama-French five-factor model expands on Fama's efficient market hypothesis by identifying anomalies in the market that may explain excess returns. This research builds on the model with empirical evidence that incorporating the factors into a predictive neural network can improve generalizability and practical usefulness.

6.3 Practical Implications

For investors, this research highlights the utility of machine learning in guiding investment strategies, especially in specific market segments. In environments with lower information availability, such as cryptocurrencies and small-cap stocks, LSTM models excel at capturing inefficiencies. This study's findings suggest that investors can use machine learning to uncover opportunities for alpha generation while effectively navigating the higher volatility endemic to low-information assets.

In an inefficient asset class that is exploitable with cutting-edge machine learning techniques, cryptocurrencies pose a unique opportunity for investors. However, their inefficiencies also signal vulnerabilities to manipulation and speculation by celebrities and institutions. A deep understanding of how machine learning tools can exploit inefficiencies could help investors improve frameworks to mitigate the risks inherent to volatile periods and markets.

This study also provides technological insights, examining modern machine learning techniques through an approach informed by economic theory. When integrating basic technical factors with macroeconomic data, LSTMs become a scalable way to analyze complex market dynamics and break them down into simple predictions and signals. Their overperformance from an ability to adjust to changing market conditions could provide the basis for investment strategy shifts away from static models and toward flexible, data-driven approaches.

6.4 Final Thoughts

This paper's findings extend beyond academia directly into the investing world. Gaining a deeper understanding of how technology might inform financial decisions was a primary motivator for this study, and it certainly succeeded in that regard. Through this process, the potential of optimized machine learning approaches to uncover inefficiencies in financial markets has only become more apparent.

While it reveals valuable insights, this research raises some fascinating questions for future research. How robust are machine learning models in periods of macroeconomic disruption, such as geopolitical events or financial crashes? How do they perform in different market segments and over different time frames? As consumer computers constantly increase in power, what are the implications for financial markets? With the proper knowledge, one can build and run cutting-edge neural networks on just a laptop, and this democratization could empower retail investors to level the playing field against institutions. At the same time, more machine learning in markets could exacerbate volatility and uncover opportunities for insider manipulation.

For decades, investors have sought to understand the collective behavior driving financial markets. Yet, many repeatedly fail to do so. The fact that machines seem, to a degree, able to understand market movements through layers of abstract multi-dimensional vectors points to a compelling paradox: While abstract, collective human behavior may be better approximated by machines.

References

Bachelier, L. (1900). Théorie de la spéculation. *Annales scientifiques de l'É.N.S. 3*(17), 21-86. http://www.numdam.org/articles/10.24033/asens.476/

Bagrecha, C., Singh, K., Sharma, G., & Saranya, P. B. (2024). Forecasting silver prices: a univariate ARIMA approach and a proposed model for future direction. *Mineral Economics: Raw Materials Report*, 1–11. https://doi-org.coloradocollege.idm.oclc.org/10.1007/s13563-024-00461-y

Bao, W., Cao, Y., Yang, Y., Che, H., Huang, J., & Wen, S. (2025). Data-driven stock forecasting models based on neural networks: A review. *INFORMATION FUSION*, *113*, 102616. https://doi-org.coloradocollege.idm.oclc.org/10.1016/j.inffus.2024.102616

Berradi, Z., & Lazaar, M. (2019). Integration of Principal Component Analysis and Recurrent Neural Network to Forecast the Stock Price of Casablanca Stock Exchange. *Procedia Computer Science,* 148, 55-61. https://doi.org/10.1016/j.procs.2019.01.008

Crane, A. D., & Crotty, K. (2018). Passive versus Active Fund Performance: Do Index Funds Have Skill? *Journal of Financial and Quantitative Analysis*, *53*(1), 33–64. doi:10.1017/S0022109017000904

Evens, Z. (2024, May 21). *Inexpensive US small-cap exposure.* Morningstar. https://www.morningstar.com/etfs/arcx/scha/analysis

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.2307/2325486

Fama, E. F., & French, K. R. (2014). *A Five-Factor Asset Pricing Model.* University of Chicago & Tuck School of Business, Dartmouth College. https://dx.doi.org/10.2139/ssrn.4629613

Fama, E. F., & French, K. R. (2024). *Description of Fama/French 5 Factors.* Tuck School of Business, Dartmouth College. https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library/f-f_5_factors_2x3.html

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation,* 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

Kingma, D. P., & Ba, Jimmy. (2015). Adam: A Method for Stochastic Optimization. In *the 3rd International Conference for Learning Representations.* https://doi.org/10.48550/arXiv.1412.6980

Kumar, K., & Haider, M. T. U. (2021). Enhanced Prediction of Intra-day Stock Market Using Metaheuristic Optimization on RNN–LSTM Network. *New Generation Computing*, *39*(1), 231–272. https://doi-org.coloradocollege.idm.oclc.org/10.1007/s00354-020-00104-0

Kumar, M., & Thenmozhi, M. (2014). Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. *International Journal of Banking, Accounting and Finance,* 5(3), 284-308. https://www.inderscienceonline.com/doi/epdf/10.1504/IJBAAF.2014.064307

Liu, W., Suzuki, Y., & Du, S. (2024). Forecasting the Stock Price of Listed Innovative SMEs Using Machine Learning Methods Based on Bayesian optimization: Evidence from China. *Computational Economics*, *63*(5), 2035–2068. https://doi-org.coloradocollege.idm.oclc.org/10.1007/s10614-023-10393-4

Maverick, J. B. (2024, March 27). *What the Sharpe Ratio Means for Investors.* Investopedia. https://www.investopedia.com/ask/answers/010815/what-good-sharpe-ratio.asp

Mishkin, F. S.(2019). *The Economics of Money, Banking, and Financial Markets* (12th ed.). Pearson Education Limited.

Sun, X., Liu, X., & Zhou, Y. (2023). Delineating Peri-Urban Areas Using Multi-Source Geo-Data: A Neural Network Approach and SHAP Explanation. *Remote Sensing*, *15*(16), 4106. https://doi.org/10.3390/rs15164106

University of Sussex. (2005). *Table of critical values for the F distribution.* http://users.sussex.ac.uk/~grahamh/RM1web/F-ratio table 2005.pdf

Yan, X., Weihan, W., & Chang, M. (2021). Research on financial assets transaction prediction model based on LSTM neural network. *Neural Computing and Applications*, *33*(1), 257–270. https://doi-org.coloradocollege.idm.oclc.org/10.1007/s00521-020-04992-7

Zhou, Z., Gao, M., Liu, Q., & Xiao, H. (2020). Forecasting stock price movements with multiple data sources: Evidence from stock market in China. *Physica A: Statistical Mechanics and Its Applications*, *542*. https://doi.org/10.1016/j.physa.2019.123389

Appendix A: Bachelier's original French passage

"A côté des causes en quelque sorte naturelles des variations, interviennent aussi des causes factices : la Bourse agit sur elle-même et le mouvement actuel est fonction, non seulement des mouvements antérieurs, mais aussi de la position de place. La détermination de ces mouvements se subordonne à un nombre infini de facteurs : il est lors impossible d'en espérer la prévision mathématique.[...] Le Calcul des probabilités ne pourra sans doute jamais s'appliquer aux mouvements de la cote et la dynamique de la Bourse ne sera jamais une science exacte. Mais il est possible d'étudier mathématiquement l'état statique du marché à un instant donné, c'est-à-dire d'établir la loi de probabilité des variations de cours qu'admet à cet instant le marché."