Master's Thesis

Regime-Based Nasdaq Futures Trading:
LSTM vs Transformer vs Buy-and-Hold

AALBORG UNIVERSITY
DENMARK

M.Sc. in Finance
Aalborg University Business School

Author

Niklas Mähleke
(20232258)

Supervisor

Professor
Frederik Steen Lundtofte

Date

Jun 2, 2025

Abstract

Short-term traders and risk managers often do not have dependable tools for making decisions every minute, because traditional time-series methods can break down when liquidity and volatility change quickly. I show that trading only when clear “bull,” “bear,” or “sideway” regimes are detected captures most of the value in high-frequency trading. I labeled one-minute NASDAQ futures bars from January 2015 to Jun 2024 as “bull,” “bear,” or “sideway”, and trained two deep-learning models to predict these labels.

An LSTM (Long Short-Term Memory) is a neural network that learns to remember or forget information over time - useful when both recent and slightly older price patterns matter. A Transformer is another neural network design that uses an attention mechanism to identify which past data points are most relevant for each new prediction.

In out-of-sample tests on July-December 2024 data, both models spotted real regime shifts within ten minutes on average and outperformed a simple buy-and-hold strategy. However, their higher returns came with larger swings, meaning these signals carry more risk. This work provides a minute-level deep-learning framework showing that LSTM and Transformer signals can beat buy-and-hold - if you accept bigger ups and downs.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Frederik Steen Lundtofte, for his continuous support, patience, and guidance throughout my research and thesis writing. His insightful feedback and encouragement have been invaluable. I am especially grateful for his unwavering commitment and the many discussions that have significantly enriched my work. Additionally, I acknowledge that generative AI tools were used to correct grammar in this document.

Table of Contents

Glossary

Term	Definition
Attention Mechanism/ Self-Attention	A technique where the model learns which parts of the input are most relevant to each other for better understanding.
Bullish/ Bearish/ Sideways Market	Describes market direction: Bullish = rising, Bearish = falling, Sideways = no clear trend.
Candle/ Bar	A chart element showing the open, high, low, and close prices for a set time period, used to analyze price movements.
Feature Engineering	Creating input variables (features) that help a model learn patterns more effectively.
Genetic Algorithm	An evolutionary search method that selects the best model settings, mixes them, and makes small random tweaks until it finds strong hyperparameters.
Hyperparameters	Configurable settings of a model (e.g., learning rate, number of layers, batch size) that govern how the model is trained.
Loss Function	A formula used to measure how far off a model's predictions are from actual values.
Neural Network/ Deep Learning	A set of algorithms modeled after the human brain that can learn patterns from data, commonly known as artificial intelligence (AI).
Normalization/ Standardization	Techniques to scale data to a consistent range or distribution.
Supertrend Indicator	A trend-following indicator based on price and volatility that generates dynamic support/resistance levels. Bullish above the line, bearish below. It adapts to market shifts to highlight potential reversals.
Tick Chart/ Tick Data	Charts based on a number of trades (ticks) rather than time intervals.

1. Introduction

1.1 Problem Statement and Motivation

Financial markets are driven by forces spanning a wide range of timescales. At the slow end, macroeconomic policy shifts and changes in investor sentiment shape trends over days and weeks. At the fast end, algorithmic trading influences prices within seconds (Tetlock, 2007; Hendershott et al., 2011; Brogaard et al., 2014). Yet the one-minute OHLCV (open, high, low, close, volume) interval - where intraday traders and risk managers must act - remains a relatively underexplored forecasting horizon in both academic research and industry practice.

Standard time-series tools such as ARIMA for returns and Engle's ARCH (1982) with Bollerslev's GARCH extension (1986) for volatility assume smoothly evolving, stationary dynamics. Historic stress episodes - most notably the 2008 liquidity spiral and the 2010 Flash Crash - exposed sudden shifts in liquidity and volatility that these methods cannot anticipate, causing forecasts to lag actual movements and risk estimates to fall short by large margins (Brunnermeier & Pedersen, 2009; Kirilenko et al., 2017; Adrian & Brunnermeier, 2016). Market-microstructure theory indicates that abrupt surges in trading activity or uneven execution speeds across venues increase adverse-selection risk for liquidity providers, leading them to widen bid-ask spreads and retract displayed depth to protect their positions. However, this framework does not provide practical tools for intraday (minute-by-minute) forecasting.

Deep-learning architectures offer a promising bridge between theory and practice. Long short-term memory networks (LSTM) and Transformer models can learn complex, nonlinear relationships from high-frequency price and volume data without restrictive statistical assumptions (Sirignano & Cont, 2019). When enhanced with technical features that capture trend direction, volume patterns, and volatility shifts, these networks form a richer representation of evolving market regimes. Large-scale studies have demonstrated that such approaches generate substantial economic value, outperforming traditional factor-model benchmarks in equity-premium forecasting (Gu et al., 2020).

This thesis fills the intraday gap by developing and evaluating LSTM and Transformer solutions on minute-frequency OHLCV data for a major equity index. I measure performance in out-of-sample backtests (July-December 2024) against a buy-and-hold strategy, using simulated profit-and-loss from straightforward trading rules. By focusing on both statistical precision and real-world P&L outcomes, this work aims to equip market participants with robust, live tools for risk management, and execution optimization.

1.2 Key Contributions and Scope

To my knowledge, this thesis delivers the first minute-level OHLCV benchmark of deep sequence (AI) models against a buy-and-hold baseline. I make three primary contributions. First, I implement and compare LSTM and Transformer architectures for one-minute regime classification and trend forecasting, demonstrating how each model handles noise, retains memory, and captures long-range dependencies. Second, I introduce a Supertrend-based algorithm for adaptive regime labeling - developed at my startup, TRARITY - which dynamically adjusts to shifting volatility in OHLC price data to produce bull, bear, and sideways signals, thereby improving signal quality compared with fixed-rule approaches. Third, I conduct backtests and live-trading simulations under realistic market conditions - including transaction fees - to bridge the gap between statistical performance and actual trading outcomes.

The scope of this study is strictly confined to one-minute OHLCV data for the NASDAQ index from January 1, 2015, to December 31, 2024. I exclude order-book depth, news sentiment, and macroeconomic releases to ensure full reproducibility on standard data feeds. All feature engineering derives solely from price and volume, with technical indicators computed consistently across models.

1.3 Research Questions and Hypotheses

The investigation centers on three interrelated questions, each designed to probe a different dimension of minute-level forecasting and its practical implications. By focusing on both statistical and economic measures, I aim to determine not only which architecture offers superior predictive power, but also whether those predictions translate into tangible trading gains under realistic conditions.

RQ1: Regime Classification Performance. How do LSTM and Transformer models compare to a static baseline in classification performance (e.g., balanced accuracy and macro-F1) for multi-class regime prediction (Long / Short / Range) on one-minute data?

H1: Both LSTM and Transformer models will outperform the static baseline by achieving higher directional accuracy and lower classification error, while maintaining comparable signal timeliness.

RQ2: Risk-Adjusted Trading Performance. When translated into trading strategies, do signals from LSTM and Transformer models yield superior risk-adjusted returns - measured by Sharpe, Sortino, and Calmar ratios, as well as drawdown - compared to a buy-and-hold NASDAQ ETF?

H2: Deep learning-based strategies will achieve higher risk-adjusted returns by dynamically adapting to transient market conditions identified through the Trarity labeling method, even under leveraged trading conditions.

RQ3: **Regime-Based Robustness.** When the broader market regime is defined using hourly candles (bull, bear, sideways), do LSTM and Transformer models continue to produce accurate minute-by-minute predictions and superior risk-adjusted returns within each regime, compared to a buy-and-hold approach?

H3: In all three regime types - bull, bear, and sideways - LSTM and Transformer models will outperform buy-and-hold by maintaining forecasting accuracy at the one-minute level and generating more favorable risk-adjusted results.

To validate these hypotheses, I employ paired statistical tests (e.g., Wilcoxon signed-rank) to assess the significance of observed differences and construct confidence intervals around key performance metrics. This rigorous inference framework ensures that any reported improvements in classification or trading performance reflect true model advantages rather than random variation.

2. Literature Review

2.1 Financial Market Trend Prediction

Financial market trend prediction lies at the heart of both academic inquiry and real-world asset management. Accurately anticipating directional moves underpins the design of trading rules, informs portfolio allocation, and bolsters risk control (Jegadeesh & Titman, 1993; Lo & MacKinlay, 1999). While classical methods such as ARIMA for return dynamics (Box & Jenkins, 1970) and ARCH/GARCH for volatility modeling (Engle, 1982; Bollerslev, 1986) have long guided practitioners, advances in artificial intelligence and deep learning - particularly LSTM and Transformer architectures - offer new paths to capture the complex, nonlinear patterns that traditional econometrics miss (Fischer & Krauss, 2018). This section surveys the empirical foundations of trend forecasting, contrasts momentum and mean-reversion approaches, and explores the ongoing debate between the Efficient Market Hypothesis (Fama, 1970) and behavioral explanations of persistent market anomalies (Barberis, Shleifer, & Vishny, 1998).

2.1.1 The Importance of Trend Forecasting in Finance

Trend forecasting plays an important role in finance for several interrelated reasons. The ability to identify uptrend, downtrend, and sideway market phases is crucial for making informed decisions. This task becomes especially challenging when working with high-frequency data, such as one-minute candles, which capture detailed market behavior but also include a lot of noise and non-stationary patterns. The high resolution of this data can reveal temporary market regimes that might be missed in lower-frequency data, potentially improving the timing of trading decisions (Andersen, Bollerslev, Diebold, & Labys, 2003).

AI-based models, particularly those using neural networks like Long Short-Term Memory (LSTM) networks and Transformers, have shown strong performance in identifying complex patterns in financial time series. Studies indicate that deep learning techniques can forecast trends more accurately than traditional statistical models and produce actionable signals for trading strategies (Al-Khasawneh et al., 2024). The use of these models in algorithmic trading is an area of active research, as they offer adaptability and predictive power during volatile market conditions.

2.1.2 Momentum and Mean-Reversion Strategies in Asset Pricing

Two complementary strategies underpin short-term trend prediction in equity markets: momentum and mean-reversion.

Momentum strategies posit that assets exhibiting the highest recent returns will continue to outperform over the next brief interval. One defines a formation window (e.g., the preceding five one-minute bars), computes each asset's return, ranks assets by performance, and goes long the top decile while shorting the bottom decile. This exploits under-reaction and herding, yielding statistically significant excess returns (Jegadeesh & Titman, 1993; Lo & MacKinlay, 1990).

Mean-reversion strategies assume that when prices move far from their recent average, they are likely to return to that average. Traders calculate a moving average over a longer window (e.g., thirty one-minute bars), then look at how much the current price deviates from it. They sell (go short) if the price is too high above the average and buy (go long) if it falls below. Research shows that assets with poor past performance often outperform past winners over longer periods (De Bondt & Thaler, 1985), and similar patterns appear over medium timeframes (Lo & MacKinlay, 1990). In high-frequency trading, mean-reversion signals can effectively identify temporary overbought or oversold conditions, offering consistent - though usually smaller - profits than momentum strategies (Stübinger & Endres, 2018).

By updating both signals every minute, traders can detect short-lived trends and brief market extremes at the same time. On a longer time scale, Gu, Kelly, and Xiu (2020) use deep learning models on daily stock data and firm characteristics. They retrain their models regularly to keep up with changing market conditions. Once trained, these models automatically combine signals like momentum, price, and others to predict next-day returns - outperforming traditional linear models without the need for manual adjustments.

2.1.3 Behavioral Finance Versus the Efficient Market Hypothesis (EMH)

The theoretical debate between behavioral finance and the Efficient Market Hypothesis (EMH) lies at the heart of trend prediction. The EMH posits that asset prices fully reflect all publicly available information, rendering systematic outperformance impossible (Fama, 1970; Fama, 1991). In contrast, behavioral finance documents persistent investor biases - prospect-theory preferences, the disposition effect, overconfidence, and herding - that generate predictable price patterns (Kahneman & Tversky, 1979; Shefrin & Statman, 1985; Odean, 1998).

While the EMH implies that any exploitable anomalies should vanish almost instantly, empirical studies find that short-term inefficiencies persist because of delayed information processing and collective psychology (Odean, 1998; Banerjee, 1992). Deep learning models are well suited to detect and adapt to these fleeting inefficiencies by continuously learning from high-frequency data streams. By integrating behavioral insights (e.g., signals of overreaction or herding) with advanced neural networks, researchers have built predictive systems that dynamically adjust to shifts in market regimes and translate these into profitable trading rules (Gu, Kelly, & Xiu, 2020; Sirignano & Cont, 2019).

2.2 Traditional Forecasting Models in Finance

Traditional forecasting models - such as ARIMA (Box & Jenkins, 1970) and volatility frameworks like ARCH (Engle, 1982) and its extension GARCH (Bollerslev, 1986) - have long underpinned financial analysis. Meanwhile, technical indicators based on moving-average crossovers and trading-sideway-phase breaks (Brock, Lakonishok, & LeBaron, 1992) and systematic pattern-recognition methods (Lo, Mamaysky, & Wang, 2000) remain staples for practitioners. However, when applied to non-stationary financial time series - where means, variances, and autocorrelations shift abruptly - these approaches struggle: the stationarity assumption fails, model parameters lose stability, and rule-based signals deteriorate in performance.

2.2.1 Time-Series Models: ARIMA and GARCH

The Autoregressive Integrated Moving Average (ARIMA) model combines past observations, differencing to remove trends, and past forecast errors to produce stationary, linear forecasts. Its suite of diagnostic tools - such as the autocorrelation and partial-autocorrelation functions - facilitates rigorous model selection (Box & Jenkins, 1970). However, the ARIMA model assumes constant volatility - an unrealistic simplification that conflicts with the well-documented clustering of large and small returns in financial markets (Engle, 1982).

Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models relax this assumption by letting forecast variance respond to recent shocks. In a GARCH(1,1) specification,

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2,$$

large past errors (ε_{t-1}^2) and prior variance (σ_{t-1}^2) jointly drive current volatility estimates, capturing clustered risk dynamics (Bollerslev, 1986). Empirical studies highlight the value of combining these frameworks. Mohammadi and Su (2010) show that an ARIMA(1,1,1) model paired with an APARCH(1,1) filter - an extension of GARCH that accounts for asymmetry and nonlinear volatility effects - on weekly Brent-crude prices reduces the root-mean-square error to 3.72 and the mean absolute error to 2.54 - an 8-15 % improvement over standalone GARCH variants. Ersin and Bildirici (2022) show that a rolling-window ARIMA-GARCH(1,1) hybrid on the S&P 500 reduces the mean absolute error (MAE), or average absolute forecast error, from 12.12 to 11.83 and the mean squared error (MSE), which penalizes larger misses more heavily, from 310.37 to 303.04 (a 2.4 % improvement in both). In foreign exchange, West and Cho (1995) report that a simple autoregressive model of squared returns produces weekly volatility forecasts whose root-mean-square error (RMSE) - the square root of MSE, which brings the error back into volatility units - is within 2 % of a GARCH(1,1) model.

Despite these gains, both ARIMA and GARCH remain fundamentally linear and assume a stable data-generating process. Market regimes shift unpredictably under evolving investor sentiment, macroeconomic shocks, and structural breaks - patterns that linear models cannot capture without further extensions. Markov-switching models handle this by allowing key parameters - such as the mean, variance, or autoregressive coefficients - to shift between a small number of predefined market conditions, like bull or bear phases. These shifts follow a Markov chain, meaning the next condition depends only on the current one, not the full history. This allows the model to automatically detect and adapt to changing market environments (Hamilton, 1989). While ARIMA and GARCH thus provide invaluable baselines, their capacity to model the non-stationary, nonlinear complexities of modern financial data is inherently limited.

2.2.2 Technical Indicators

Technical indicators convert historical price series into rule-based trading signals. Two of the most enduring classes are moving averages and volatility-adjusted thresholds. A simple moving average (MA) computes the arithmetic mean of the last *m* closing prices, smoothing high-frequency noise to reveal underlying trends. Despite its ubiquity, the MA inherently lags rapid market moves: when prices accelerate, the MA responds only after the new data fully enter the window. Brock, Lakonishok, and LeBaron (1992) found that, in-sample (1897-1986), very simple moving-average crossover and trading-range breakout rules generated statistically significant excess returns even after deducting plausible transaction costs. Subsequent out-of-sample tests (e.g., Sullivan, Timmermann, & White 1999; Park & Irwin 2007) show, however, that those in-sample profits largely disappear once more realistic commissions, bid-ask spreads, and corrections for selection bias (i.e., overfitting to historical data) are included.

Lo, Mamaysky, and Wang (2000) developed a way to “smooth out” day-to-day price swings and automatically pick out a handful of well-known chart patterns. They ran this method on U.S. stocks from 1962 through 1996. When they looked at how prices moved a few days after each pattern appeared, they found that, on average, those signals were followed by a tiny boost in returns - around 0.02-0.05 percent extra compared to doing nothing. In other words, if you held a stock after seeing one of their detected patterns, you’d typically earn only a few hundredths of a percent more than you would have if no pattern had shown up.

Parameter tuning is critical for both moving-average and pattern-recognition models. The window length governs the balance between responsiveness and noise reduction by setting how many past observations inform each signal. The kernel bandwidth adjusts the influence of historical data in nonparametric pattern fits, with narrow bandwidths capturing fine-scale fluctuations and wider bandwidths producing smoother estimates. Threshold multipliers scale deviation measures - often based on volatility - to define the minimum price move required before a buy or sell signal is generated. Rigorous backtesting on historical data, combined with out-of-sample validation, is essential to confirm that chosen thresholds deliver real trading gains rather than chance artifacts. The bootstrap is like running many “what-if” experiments on your data to see how often a supposed “winning” strategy could happen by pure luck. You shuffle and resample your historical returns over and over, then see how often a rule that looks profitable actually comes from random chance. Sullivan, Timmermann, and White (1999) show that once you do this, nearly all simple trading rules stop looking like they beat the market.

2.2.3 Addressing Non-Stationarity

Traditional statistical and rule-based models often fail when markets change, because average returns and risk can shift, making fixed assumptions obsolete. Markov-switching models (Hamilton, 1989) address this by letting parameters - such as expected return or volatility - move between different market regimes as conditions change, so the model adapts automatically. Alternatively, realized volatility measures (Andersen, Bollerslev, Diebold, & Labys, 2003) use high-frequency intraday prices to update variance estimates continuously, allowing forecasts to respond quickly to new turbulence. Both approaches improve robustness over fixed-parameter methods but demand more data and computing power, and Markov-switching can lag when truly novel conditions arise. In short, while ARIMA, GARCH, and simple rule-based systems remain useful starting points, their static design limits effectiveness in today's fast-moving, high-frequency markets.

2.3 Artificial Intelligence in Financial Forecasting

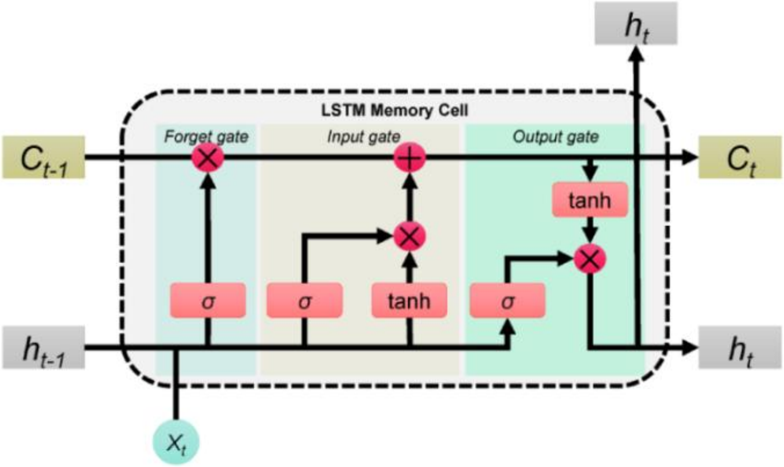
Advances in artificial intelligence - particularly deep-learning architectures such as LSTM networks and transformer models - offer powerful tools for uncovering complex, nonlinear patterns in high-frequency data that traditional econometric methods cannot capture. By automatically learning from time-series inputs and adapting parameters in real time, these models promise greater forecast accuracy and resilience to regime shifts and market noise (Gu, Kelly, & Xiu, 2020; Sirignano & Cont, 2019). The remainder of this chapter introduces the core AI techniques employed in my high-frequency forecasting framework.

2.3.1 Core Deep-Learning Architectures

This section introduces the two neural designs that power the forecasting systems: the long short-term memory network (LSTM) and the Transformer. Each consumes the same minute-by-minute stream of prices and volumes but processes that stream in a fundamentally different way - the LSTM by updating a running internal “notebook,” the Transformer by allowing every time-step to consult every other through self-attention. Grasping these contrasting mechanics is essential before we examine how the models are trained and benchmarked.

Figure 1 illustrates the basic LSTM architecture, as explained by Hochreiter and Schmidhuber (1997). Imagine the long horizontal line that crosses the dashed box as a trader’s pocket notebook. In technical terms this is the cell state. It enters on the left as C_{t-1} - everything remembered up to the previous bar - and exits on the right as the updated memory C_t.

Figure: 1 *- LSTM Structure*



Source: (dida, 2025)

Three colored regions inside the box are gates, tiny calculators that decide what happens to the notebook. Each gate contains a pink rectangle marked “σ,” the Greek sigma that denotes a sigmoid function. A sigmoid works like a dimmer switch, squeezing any input into a value between 0 and 1.

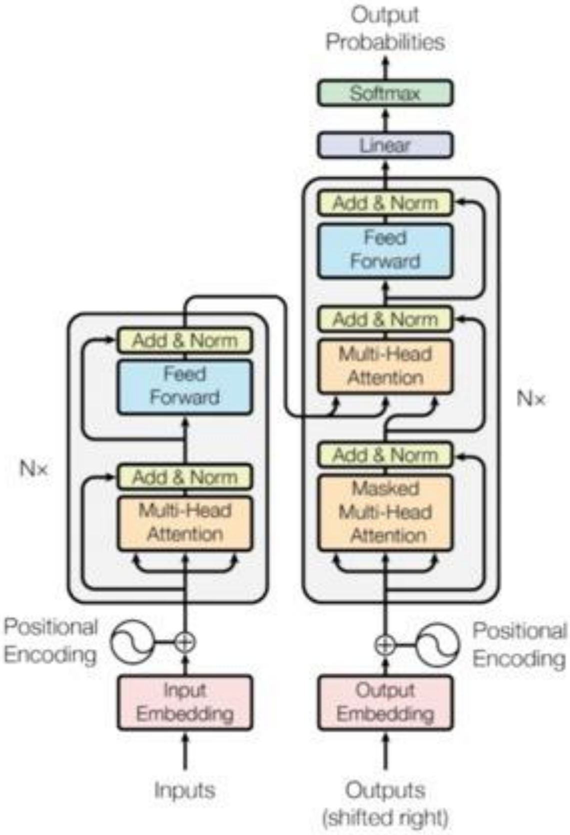
The turquoise strip is the forget gate. It looks at the new market data X_t and the old hidden summary h_{t-1} (a brief snapshot of what the LSTM remembered at time $t-1$). Its sigmoid emits dimmer-switch values that multiply C_{t-1}, fading parts of the notebook so a meaningless price spike can disappear.

Next comes the beige input gate. A second sigmoid scores how important the fresh information is, while a tanh block (which narrows numbers into the -1 to 1 interval) prepares the candidate material to be written. Only the pieces that score highly pass through the pink “×” symbol and are added to what remains of the notebook, yielding the new cell state C_t.

Finally, the green output gate decides what to reveal. Its sigmoid selects lines from C_t; the chosen lines flow through another tanh for scaling and emerge at the top as the new hidden state h_t (a brief snapshot of what the LSTM knows at time t). Because this notebook is continually updated and rewritten every minute, the LSTM retains durable patterns - such as a slow upward drift - while ignoring random noise.

As shown in Figure 2 and as explained by Vaswani et al. (2017), the Transformer behaves more like a conference call than a notebook. Every input bar is first turned into an embedding, a compact vector that captures its open, high, low, close and volume. A positional encoding - extra numbers that say "this is minute 1, this is minute 2," and so on - is then added so the model never loses track of order.

Figure: 2 *- Transformer Structure*



Source: (Ankit, 2024)

Inside each large rectangle the yellow block marked multi-head self-attention lets every minute listen to every other minute. For that listening, the model silently builds three matrices called queries, keys and values; comparing queries with keys produces similarity scores, and a softmax turns those scores into attention weights that sum to 1. Running several of these comparisons in parallel yields multi-head attention: one head can focus on the last few bars, another on an earlier trend. After attention, the output is added back into the original flow so nothing gets lost. Then a normalization step adjusts the combined signal to a consistent scale, which helps the model learn more smoothly. The blue block refines the data for each minute. By repeating these focus-and-refine steps N times (shown as "Nx"), the model gains a strong, overall understanding of the entire sequence. This stack of repeated steps on the left is called the encoder.

The decoder on the right starts with masked multi-head attention: "masked" means each position can look only at earlier positions, never at the future, so the model cannot cheat. Then the model looks back at what the encoder has produced and combines it with what the decoder has generated so far. Next, it merges this combined information back into its main "working memory," adjusts the values to keep things stable, and runs it through one more simple processing step to sharpen the result. At the very top a linear layer converts the refined vectors into raw scores, and a final softmax turns those scores into actual probabilities for the next-minute price movement.

In simple terms, an LSTM works like a careful note-taker who updates one page at a time, keeping track of what matters and tossing out what doesn't. A Transformer, on the other hand, is more like a big round-table meeting where every minute in the sequence listens to every other minute all at once. The long list of technical words - cell state, gates, sigmoid, tanh, embedding, positional encoding, self-attention, multi-head attention, residual connections, layer normalization, feed-forward, masked attention, and softmax - are just fancy names for everyday actions: writing down and updating notes, deciding how much weight to give each piece of information, labeling each minute, marking its place in time, letting each part "listen" to others, having multiple "listeners" working in parallel, making sure nothing gets lost, keeping numbers in a neat range, running the notes through a simple polish step, hiding future details until it's time to reveal them, and turning raw scores into clear probabilities.

2.3.2 Empirical Performance of LSTM and Transformer on the S&P 500

Several studies have evaluated deep-learning architectures on large equity indices. Wang, Chen, and Zhang (2022) compare LSTM and a Transformer on daily S&P 500 closing prices from 2010 to 2020, training on the first 90% of observations and evaluating on the final 10% (approximately one year). They assess both one-day-ahead forecast accuracy and a simple long-only trading strategy - measured over that same hold-out period - against a buy-and-hold benchmark.

Table 1 below shows the one-day-ahead forecast accuracy on the S&P 500 for each model, using three common error metrics: MAE, MSE, and MAPE.

Table: 1 - *One-Day-Ahead Forecast Accuracy on S&P 500*

Model	MAE	MSE	MAPE
LSTM	0.1092 (± 0.0256)	0.0236 (± 0.0099)	1.7768 % (± 0.4001)
Transformer	0.0814 (± 0.0131)	0.0145 (± 0.0037)	1.3800 % (± 0.2163)

*Source: (*Wang, Chen, & Zhang, 2022)

As Table 1 shows, the Transformer cuts the MAE - the average deviation of its forecasts in index points - by 25.5% (from 0.1092 to 0.0814) and the MSE - which penalizes larger errors - by

38.6% (from 0.0236 to 0.0145) versus the LSTM, and posts a MAPE of just 1.38%, indicating its one-day-ahead forecasts are, on average, within 1.38% of the S&P 500 level. This means the Transformer not only makes smaller mistakes in raw points but also maintains consistently low relative error, demonstrating both precise noise filtering and reliable performance across varying market scales.

Using the same hold-out period and error forecasts, Wang, Chen, and Zhang (2022) also test a simple long-only trading strategy on the S&P 500. Each strategy starts with the same amount of money and updates its holdings each day based on the model's prediction for the next trading day. Table 2 summarizes total return, maximum drawdown, and Sharpe ratio for each approach over the final 10 percent of the 2010-2020 sample.

Table: 2 *- Long-Only Trading Strategy Performance on S&P 500*

Strategy	Total return (%)	Max Drawdown (%)	Sharpe Ratio
B&H	30.53	-41.43	0.54
LSTM	45.02	-34.55	0.79
Transformer	56.35	-28.50	0.99

*Source: (*Wang, Chen, & Zhang, 2022)

The buy-and-hold benchmark delivers a 30.53 % total return with a -41.43 % peak drawdown and a Sharpe ratio of 0.54. Compared to this baseline, the LSTM-driven strategy increases cumulative return by 14.49 percentage points, reduces maximum drawdown by 6.88 pp, and boosts the Sharpe ratio from 0.54 to 0.79. The Transformer-based approach further amplifies these gains, outperforming buy-and-hold by 25.82 pp in total return, cutting drawdown by 12.93 pp, and raising the Sharpe ratio to 0.99 - demonstrating not only superior forecast-driven returns but also materially better risk-adjusted performance.

Taken together, these results demonstrate that, on the S&P 500 index, both LSTM- and Transformer-based strategies significantly outperform a buy-and-hold benchmark in cumulative return, drawdown reduction, and Sharpe ratio - with Transformer architectures delivering the strongest predictive accuracy and trading outcomes - thereby establishing a new performance standard for large-scale equity time-series analysis.

2.3.3 Comparative Advantages & Disadvantages

ARIMA is popular because it's easy to understand and interpret. It makes forecasts using past values, recent errors, and simple adjustments for trends, and offers clear checks (like autocorrelation plots and the Ljung-Box test) to confirm you've picked a good model (Box & Jenkins, 1970). Its main drawback is that it assumes volatility never changes, so it can't handle the bursts of high or low volatility we see in financial markets (Engle, 1982).

GARCH fixes that by letting today's volatility depend on yesterday's shock and yesterday's volatility, so it naturally captures periods of calm versus turbulence (Bollerslev, 1986). In practice, combining ARIMA and GARCH often gives more accurate forecasts, but GARCH still relies on a straightforward, linear formula for predicting returns and can struggle when the market suddenly shifts.

LSTMs solve the linearity issue by using "memory cells" that learn what information to keep and what to discard. This makes them good at filtering out short-term noise in more detailed (higher-frequency) data (Hochreiter & Schmidhuber, 1997; Wang, Chen, & Zhang, 2022). Their downside is that they must process data step by step, which can be slow and requires careful setup to avoid training problems.

Transformers take a different approach: instead of processing one step at a time, they let every data point "look at" all others at once. This parallel attention helps them pick up both short-term spikes and long-term trends more effectively than LSTMs (Vaswani et al., 2017; Wang et al., 2022). However, this power comes at the cost of needing very large datasets and significant computing resources to tune all the internal settings.

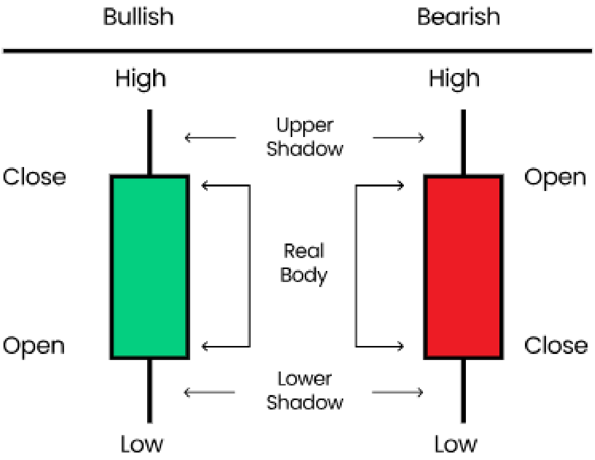
These trade-offs set the stage for the empirical work that follows. In Chapter 3, I describe the minute-by-minute NASDAQ futures data, the feature engineering and regime-labeling procedures, and the genetic-algorithm - driven model configuration that together form the backbone of my forecasting and trading framework.

3. Data & Methodology

3.1 Data Collection & Preprocessing

The core of this study is a minute-by-minute record of the continuous NASDAQ Futures contract from 2015 through the end of 2024 - ten years of market behavior, from calm periods to sudden spikes. The continuous contract was chosen because it rolls seamlessly from one expiration to the next, avoiding the gaps or inconsistencies that arise when transitioning between individual contracts. I downloaded these one-minute candles as historical data from Rhythmics' data feed via the ATAS trading platform. A "candle" here is simply a snapshot of price action over one minute, showing where prices opened, moved up to, fell to, and then closed, as visualized in Figure 3. For each one-minute candle, I capture the following data fields: "time" (the exact timestamp), "open" (the price at the very start of the minute), "high" (the highest price reached during that minute), "low" (the lowest price touched), "close" (the price at the very end of the minute) and "volume" (the number of futures contracts traded during that minute. Each contract is simply an agreement to buy or sell a fixed amount of the underlying asset at a set price on a specified future date. As such, volume tells us how active the market was in that interval).

***Figure: 3** - Trading-Candles/ Bars*



Source: (Blueberry Markets Academy, 2025)

Before beginning any forecasts or analysis, I discarded the first 49 one-minute bars of each trading day. Many of my calculations require a history of data to produce reliable results, so including those initial bars - when there isn't enough past information - can create misleading spikes. By removing them, every metric I compute starts with a sufficient stretch of prior data, preventing those odd anomalies. I also adjust the timestamps so that each trading day effectively "resets" at 00:00 - this accounts for summer-time and winter-time (DST) shifts and the fact that the raw data sessions began at different local times when I downloaded them - ensuring my models see every day on the same footing and making time-dependent patterns easier to recognize before training begins on solid, dependable ground.

3.2 Feature Engineering:

My goal was to give the models concise summaries of market behavior by converting raw price and volume data into thirteen technical indicators - each capturing a facet of trading intensity, trend direction, momentum, or volatility. As part of the third-semester Financial Trading Challenge in the Master in Finance curriculum, we used the CQG trading platform to compute and trade with these indicators. These normalized signals help the model detect emerging trends, reversals, and volatility shifts more reliably and earlier than raw data alone. Below in Table 3 is the list of indicators and the market characteristic each measures.

***Table: 3** - Technical Indicators*

Indicator	Category	Description	Additional Notes
volume	Volume	The total number of shares or contracts traded during each one-minute bar.	Serves as a direct measure of market participation and liquidity.
Volume SMA 15	Volume	A 15-period simple moving average of volume, smoothing out short-term spikes.	Highlights underlying shifts in trading activity.
OBV	Volume/ Momentum	On-Balance Volume, a cumulative total that adds volume on up-bars and subtracts on down-bars.	Helps confirm price trends by showing whether volume supports moves.
EMA 14	Trend	A 14-period Exponential Moving Average of closing prices, giving more weight to recent data.	Tracks short-term trend direction with responsiveness to new information.
EMA 60	Trend	A 60-period Exponential Moving Average of closing prices, emphasizing longer-term trends.	Provides a smoother view of the prevailing market direction.
MACD (12,26,9)	Momentum/ Trend	The difference between 12- and 26-period EMAs, often plotted with its 9-period signal line.	Identifies shifts in momentum and potential trend reversals.
RSI 9	Momentum/ Oscillator	The 9-period Relative Strength Index, an oscillator that measures recent gains vs. losses.	Values above 70 or below 30 typically signal overbought or oversold conditions.

ADX 10	Momentum/ Trend	The 10-period Average Directional Index, indicating trend strength regardless of direction.	Values above 25 often denote a strong trend.
ATR 9	Volatility	The 9-period Average True Range, quantifying market volatility by averaging true range values.	Higher ATR reflects increased price variability.
Bollinger Width 21	Volatility	Stochastic RSI, which applies the stochastic formula to RSI values for generating more sensitive signals.	Combines momentum and volatility aspects.
Major Trend Signal	Regime Flag	A binary flag (±1) derived from the Supertrend indicator with a wider multiplier ($f = 4.5$).	Marks broad bullish or bearish regimes.
Minor Trend Signal	Regime Flag	A binary flag (±1) from the Supertrend indicator with a narrower multiplier ($f = 1.7$).	Detects shorter-term corrections within the primary trend.
Close Log-Return	Price Return	The percentage change between consecutive closing prices.	Captures immediate price movement as a continuous return measure.

Source: (CQG, 2025)

3.3 Fractional Differentiation:

Although neural networks can handle non-stationary data, I chose to apply fractional differentiation because preliminary internal tests indicated it improves model accuracy. Financial time series like prices exhibit persistent trends and non-stationary drift that can mislead models if not addressed. Fractional differentiation offers a middle ground: it makes the series stationary enough for reliable modeling while preserving the long-run information that drives returns (Granger & Joyeux, 1980; Hosking, 1981).

Rather than a full first difference, I compute a sequence of weights from the binomial formula

$$w_k = (-1)^k (\frac{d}{k})$$

where k is the lag in minutes and $d \in [0,1]$ controls how much of the series' memory to retain. The alternating sign $(-1)^k$ balances past values so the series stays centered, while $\binom{d}{k}$ causes weights to shrink as you go further back in time. I generate these weights one at a time and stop once $|wk| < 10^{-5}$ - small enough to ignore without affecting the result but saving computation. I then apply those three weights over a sliding three-minute window on the one-minute NASDAQ futures series. In practice, that means each new value is a blend of the current bar and the two bars before it, using the calculated weights. This three-minute window is enough to capture the most important recent information - because later weights become so small they don't matter - so it keeps the series stable without losing key trends and remains fast to compute. I then test each d (0 to 1 in 0.05 steps) for stationarity via an Augmented Dickey-Fuller (ADF) test - which checks whether the transformed series stays stable rather than wandering off - (p < 0.05) and choose the smallest d that preserves the series' overall shape. This procedure strips just enough drift to stabilize forecasts while keeping the long-term trends that carry valuable market signals.

Although this approach can work well, it relies on an arbitrary 10^{-5} weight cutoff and a coarse grid of d values tested in 0.05 increments, so it may miss optimal settings. The ADF test can also fail when meaningful data correlations exist only briefly, and performing many sliding-window calculations becomes prohibitively slow on very large, high-frequency datasets.

3.4 Indicator Calculations:

I chose each indicator's parameters based on my own trading experience, applied fractional differentiation (order d) only to the indicators based on price (specifically, the 14- and 60-bar EMAs), and then ran all indicators - both those fractionally differentiated and those computed on raw or log-transformed series - through a 30-day adaptive z-score normalization $\mu/\sigma + \varepsilon$ - both to smooth noise and remain responsive to true regime shifts (Ogasawara et al., 2010; Andersen, Bollerslev, Diebold, & Labys, 2003). A 30-day window was selected because it roughly corresponds to one trading month, providing enough data to filter out transitory spikes while still adapting quickly when volatility regimes shift; shorter windows proved too noisy, and longer windows lagged during rapid market changes. Concretely, if $p_t^{(d)}$ is the fractionally differentiated price at time t, its z-score is computed as

$$z_{p,t} = \frac{p_t^{(d)} - \mu^{30D}(p^{(d)})_t}{\sigma^{30D}(p^{(d)})_t + \epsilon}$$

where, $\mu^{30D}$and σ^{30D} denote the rolling 30-day mean and standard deviation, respectively, and $\epsilon = 1 \times\ 10^{-6}$ is a small positive constant to prevent division by zero when σ is very small. Applying the same 30-day z-score formula to volume, volatility, or any other series ensures that each indicator lives on a common scale - so that, for example, a one-σ move in volume carries the same normalized weight as a one-σ move in price - while still "remembering" only the most recent 30 days of data. Ensuring all features share a common scale helps the neural network learn more reliably (LeCun, Bottou, Orr, & Müller, 1998): no single indicator can dominate simply because its raw values are larger, and the model trains more stably and converges more quickly. Fractional differentiation is applied only to price (and EMAs) to retain long-memory effects in returns, whereas indicators like RSI, ADX, Bollinger width, and Supertrend are already bounded or smoothed, so further differentiation or z-scoring would be redundant. Separately, d is the fractional differentiation order, and $sign(\cdot)$ is the signum function, which outputs +1 if its argument is positive and -1 otherwise - so $sign\left(p_{\mathrm{t}} - ST_t^{(f)}\right)$ yields +1 when the price is above the Supertrend line and -1 when it is not.

The raw volume data is stabilized by computing the logarithmic difference:

$$volume_t = \frac{\ln(1 + volume_t) - \mu^{30D}(\ln(1 + volume))_t}{\sigma^{30D}(\ln(1 + volume))_t + \epsilon}$$

and its 15-bar simple moving average undergoes the identical transformation

$$vol\_sma15_t = \frac{\ln\left(1 + \frac{1}{15}\sum_{i\,=\,0}^{14} volume_t\right) - \mu^{30D}(\ln(1 + SMA_{15}\ (volume)))_t}{\sigma^{30D}(\ln(1 + SMA_{15}(volume)))_t + \epsilon}$$

On-Balance Volume is computed recursively via

$OBV_t = OBV_{t-1} + sign\big(p_t - p_{t\_1}\big)volume_t$ and then normalized as

$$obv_t = \frac{OBV_t - \mu^{30D}(OBV)_t}{\sigma^{30D}(OBV)_t + \epsilon}.$$

Trend and momentum are quantified by exponential moving averages over 14 and 60 bars, each fractionally differentiated with order $d = 0.35$ and then z-scored:

$$ema_{n,t} = EMA(close, n)_t, \quad frac\_diff\_normalize(ema_n, 0.35)_t,$$

$$ema\_norm_{n,t} = \frac{fracEMA_{n,t} - \mu^{30D}(fracEMA_n)_t}{\sigma^{30D}(fracEMA_n)_t + \epsilon}, n \in \{14,60\}$$

The MACD series (fast = 12, slow = 26, signal = 9) is similarly centered and scaled:

$$macd_t = \frac{MACD_{12,26,9,t} - \mu^{30D}(MACD)_t}{\sigma^{30D}(MACD)_t + \epsilon}.$$

Classical momentum and trend-strength metrics include the nine-period Relative Strength Index and the ten-period Average Directional Index, each scaled into [0,1] by dividing by 100:

$$rsi9_t = \frac{RSI(9)_t}{100}, \quad adx10_t = \frac{ADX(10)_t}{100}.$$

and by the width of 21-period Bollinger Bands, defined without further scaling as

$$bw_t = \frac{BB_{upper,21,t} - BB_{lower,21,t}}{BB_{middle,21,t}}.$$

Market regimes are flagged by two Supertrend (ST) signals computed on ATR 7: the “major” signal with factor 4.5 and the “minor” signal with factor 1.7, each assigning

$$signal_t^{(f)} = \begin{cases} +1, & p_t > ST_t^{(f)} \\ -1, & otherwise, \end{cases} f \in \{4.5, 1.7\}.$$

Finally, immediate price momentum is encoded as the one-step close log-return, scaled by 1,000 for numerical stability:

$$r_t^{close} = 1{,}000 \ln\left(\frac{p_t}{p_{t-1}}\right)$$

After aligning and cleaning the data, the thirteen normalized indicator series serve as the inputs to my forecasting and classification models. The calculation of the indicators is provided in the Python file “add_technical_indicators_more.py”.

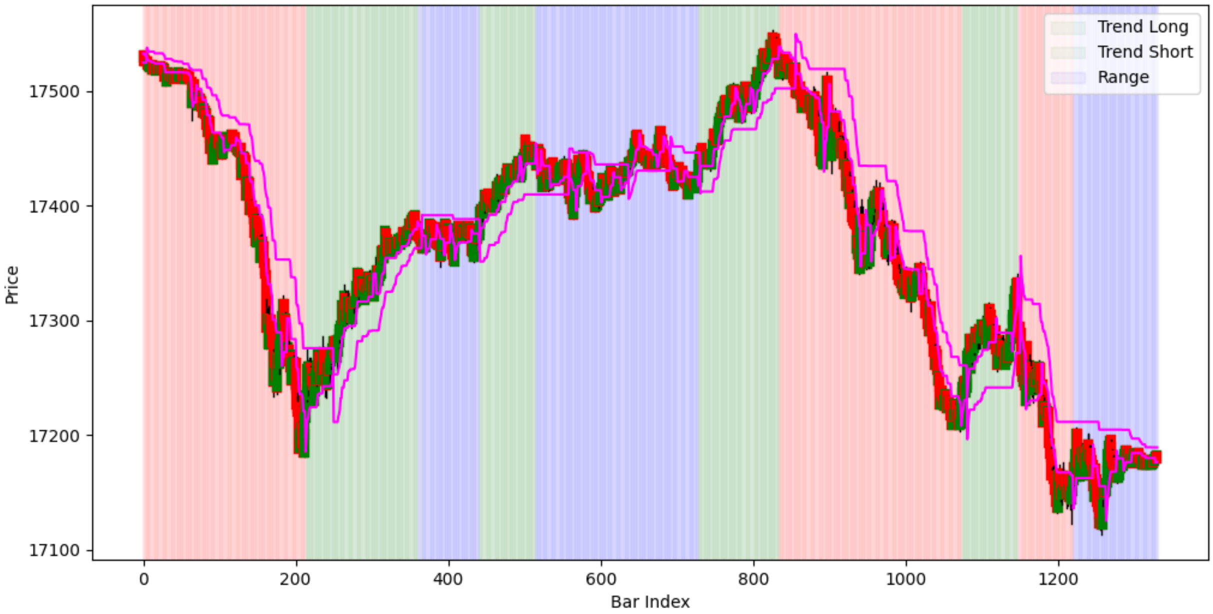
Trarity’s algorithm labels each period as Trend Long (rising market), Trend Short (falling market), or Range (sideways market) by detecting turns in the major Supertrend line and grouping bars based on rising, falling, or sideways prices. Using rule-based criteria instead of fixed percentage thresholds ensures true market shifts are captured.

In Chapter 3.2, I detail this regime-labeling process, and in Chapter 3.3, I explain how a genetic-algorithm framework is used for model selection and tuning.

3.2 Market Regime Labeling

The following figure 4 shows how I classify each one-minute price bar into a specific market regime: Trend Long, Trend Short, or Range. The method relies on smooth lines that reflect the overall direction of the market. These lines help detect when the market is rising, falling, or moving sideways. By following a clear set of rules based on price movements and trend behavior, the algorithm assigns each bar to one of the three regimes, producing structured, easy-to-interpret training data for the neural network models.

Figure: 4 *- Trarity Regime Detection*



Source: (Trarity, 2025)

To create structured and meaningful labels for training my neural networks, I developed a rule-based system that assigns each one-minute price bar to one of three market regimes: Trend Long, Trend Short, or Range. The system works by comparing the market price to two smoothed lines that represent the overall market direction. These are called the Major trend line and the Minor trend line, and they are calculated using a method known as Supertrend, which smooths out short-term noise and adapts to market volatility. These indicators act like smart rulers that adjust themselves to the speed and direction of the market, helping identify when the market is rising, falling, or moving sideways.

The Major trend line tracks the broader direction of the market. It is calculated using a 10-minute lookback period and a volatility factor of 4.5, based on a 7-minute average of price range (ATR). Every time the price crosses this Major trend line - from below to above or above to below - it signals the start of a new trend segment. For each segment between two crossings, the algorithm checks whether the Major trend line ends higher or lower than it started. If it ends higher, the segment is labeled as Trend Long. If it ends lower, it is labeled as Trend Short. However, if the trend line changes direction too frequently within the segment (more than 10 changes in 40 minutes), or if it remains almost flat in the period that follows, the segment is relabeled as Range. Specifically, if the difference between the forward-looking maximum and minimum values of the Major trend line is less than 0.0001%, the market is considered flat, and the regime is classified as Range to reflect a lack of clear directional momentum.

To improve accuracy, the algorithm also uses a faster-reacting Minor trend line, based on the same 10-minute window but with a lower sensitivity factor of 1.7. This line detects local shifts or pullbacks within larger trends. While the Minor trend does not override the main label, it helps define where transitions between regimes may begin or end.

Lastly, to reduce noise, a smoothing rule is applied. If a single one-minute bar is labeled differently from both its neighbors - such as a lone Range bar between two Trend Long bars - it is relabeled to match the surrounding trend. This ensures that the final labels are consistent and realistic, avoiding misleading outliers caused by momentary fluctuations. As a result of this process, each one-minute bar is labeled in a transparent, logical, and reproducible way. The use of fixed parameters - ATR(7), Supertrend with $n = 10$ and $f = 4.5$ for the Major trend, $n = 10$ and $f = 1.7$ for the Minor trend, a trend stability check using a 40-bar threshold, and a forward flatness filter using a 60-bar window - ensures that the method is both replicable and robust. These well-structured labels substantially reduce ambiguity in the training data, allowing the neural networks to learn more efficiently and generalize more effectively to future market conditions.

3.3 Building Fixed-Size Input Samples

Following Chapters 3.1 and 3.2, I started by creating a single CSV file that combined every minute's raw prices with the thirteen normalized indicators and their matching regime labels. I then loaded this file in strict chronological order and turned each label into a simple number: "Trend Long" became 0, "Trend Short" became 1, and "Range" became 2 - so the model could work with numeric targets.

Next, I needed to turn that continuous stream of minutes into uniform examples for the network. To do this, I “slid” a fifty-minute window across the indicator columns. In other words, for each minute t after the first fifty, I gathered the indicator values from minutes $t - 50$ through $t - 1$ as one input, and I used the number at minute t as the correct answer. Because I wanted the model to give a bit more emphasis to recent data, I applied a single exponential-decay weight to each fifty-minute example: windows later in time received a slightly larger weight than older ones. I determined that this per-sequence weighting consistently outperformed using no weights at all, while still preserving a simple workflow.

After building every fifty-minute example this way, I split them by calendar date into 90% for training, 5% for validation, and 5% for testing - giving the model plenty of data to learn from while still reserving small slices to tune parameters and check true unseen performance. Each of these groups was saved as a PyTorch tensor - a neatly organized block of numbers that the network can load instantly, like a stacked set of spreadsheets ready for training.

Neural networks require inputs of the same fixed size made up of past data only. This means every example must cover exactly fifty minutes, and none of those minutes can include the moment the model is trying to predict or any future minutes. By keeping every input the same length and drawn only from data before the prediction point, the model learns genuine patterns and doesn’t accidentally “peek” at the answer. Splitting strictly by calendar date also ensures that validation and test examples come from later periods the model has never seen, giving a realistic sense of how it will perform under new market conditions.

3.4 Automated Model Configuration

I chose a Genetic Algorithm (GA) because there were too many interdependent parameters (explained in the next chapter) - how many hidden units and layers each network should have, how quickly it learns (learning rate), how much dropout to apply, and, for the Transformer, how many attention heads to use - so manually trying every combination would take too long and likely miss good options. A GA begins with a small pool of random settings, evaluates which ones perform best on a short training run, and then mixes and tweaks them over several rounds, allowing the search to “evolve” toward strong configurations without exhaustively testing every possibility (Holland, 1975).

I created ten random configurations, each fixing those hyperparameters (with batch size always set to 512). An example is one piece of training data - say, one minute of market data with its correct label. Processing 512 examples at once means the model looks at a group of 512 before it adjusts its settings. After seeing those 512 items, it checks how far off its guesses were and decides how to change itself to do better next time. Using 512 at once gives a clear direction for improvement without using too much memory.

To score a configuration, I trained its model on the training data for up to ten passes through all examples (each pass is called an epoch). After each pass, I checked how well it did on a separate validation set by looking at its average error (validation loss). If that error didn't improve for three passes in a row (patience = 3), I stopped early. The lowest error it reached on the validation set became that configuration's "fitness" (lower error is better). Fitness can be defined as the proportion of correctly classified samples on the validation set.

$$fitness = \frac{1}{N}\sum_{i=1}^{N} 1(y_i = \hat{y}_i),$$

Where N is the total number of validation bars, y_i is the true market phase of bar i, $\hat{y}_i$ is the model's predicted market phase for bar i, and $1(\cdot)$ is the indicator function, equal to 1 when its argument is true and 0 otherwise.

After scoring all ten recipes, I organized them into random three-member "tournaments." In each tournament, the recipe with the highest validation accuracy advanced to the next generation. To maintain diversity and explore new possibilities, each advancing recipe then had a 20 percent chance of undergoing a "mutation," where one of its settings was replaced by a new random value. I repeated this cycle of evaluation, tournament selection, and mutation for exactly six generations. At the end of the sixth generation, the single recipe with the best overall validation performance was trained one final time - this time using both training and validation data and allowing for slightly more epochs - to produce the final model.

After six generations, I took that overall best configuration and trained it one last time - still splitting data into training and validation, still up to ten epochs, but now allowing five epochs of no improvement (patience = 5) before stopping. This final run produced my best model.

3.4.1 Transformer Model

Inspired by Han et al. (2021), I tested many different Transformer configurations and selected the best-performing one. The final model consists of five consecutive processing layers ("encoder layers"), each with sixteen parallel "attention heads" and a hidden-state dimension of thirty-two units. I used the genetic algorithm described earlier to tune these core settings, arriving at the configuration shown in Table 6 on the next page:

Table: 4 *- Transformer Model Configuration*

Hyperparameter	**Value**
Hidden size (d_{model})	32
Number of encoder layers	5
Number of attention heads	16
Dropout rate	0.084
Learning rate	7.47×10^{-5}

In plain terms, the hidden size controls how much information the model can carry at each stage: a small value may miss important details, while a very large one can slow training and cause the model to learn random noise. The five encoder layers represent successive steps of data transformation - more layers can capture deeper patterns but also risk over-memorizing examples. The sixteen attention heads allow the model to look at many different temporal relationships in parallel: too few heads would restrict this view, while too many add complexity without real gain. An 8.4 percent dropout rate means that, during training, the model randomly skips about 8 out of every 100 of its internal calculations. Think of the network as a web of small decision points. At each training step, roughly 8% of those points are temporarily silenced and do not contribute. This prevents the model from relying on a few strong "shortcuts" and forces it to learn patterns that hold even when some of its usual "helpers" are missing. Finally, the learning rate (0.0000747) sets how big each update is when the model fixes an error. This small value ensures updates aren't too large and don't overshoot the best solution.

If it were larger, updates could go too far; if smaller, learning would be very slow. This rate balances learning speed with stability.

I suspect that Transformers work particularly well for picking up long-term patterns in one-minute data because their self-attention lets the model look at any two points in the series directly. In other words, even an event from far back can still influence today's forecast. Older models tend to "forget" distant past information as they move forward step by step (Bengio, Simard, & Frasconi, 1994), but Transformers avoid this by linking everything right away.

Because Transformers process all time steps in parallel rather than one at a time, training is faster. They also produce attention maps that clearly show which past moments mattered most for each prediction.Choose a building block.

Despite these strengths, the Transformer's complexity demands substantial computing power, which can be challenging for real-time or resource-constrained environments. The specific hyperparameter values found by the Genetic Algorithm may not transfer directly to other markets or time periods without fresh tuning, potentially leading to reduced accuracy. Additionally, while attention maps offer a window into model behavior, they can sometimes be misleading - high attention weight does not always mean true causal importance. Finally, although dropout and learning rate settings helped prevent overfitting in my experiments, they remain heuristic choices that might require adjustment for different datasets or market regimes.

3.4.2 LSTM Model

Inspired by Smagulova and James (2019) to use an LSTM for one-minute forecasting, I tested many different configurations and selected the best-performing one. Key settings for this model were discovered via the genetic algorithm, resulting in the configuration shown in Table 5 on the next page:

Table: 5 *- LSTM Model Configuration*

Hyperparameter	**Value**
Hidden size (d_{model})	64
Number of encoder layers	5
Dropout rate	0.3249
Learning rate	1.37×10^{-4}

In everyday terms, the hidden size of sixty-four determines how much recent price history the LSTM can hold in memory: a smaller size would risk forgetting important minute-by-minute moves, while a much larger one could slow training and pick up noise. Stacking five layers gives the network the flexibility to recognize complex sequences of market behavior, and a dropout rate of roughly 32 percent randomly silences connections during training to prevent the model from simply memorizing past price swings. The learning rate (about 0.000137) determines how big a step the network takes when it adjusts its knowledge after each few examples. This value helps it learn fast enough without becoming unstable.

I also suggest that LSTMs work well for one-minute forecasts because they can decide which recent information to remember and which to ignore. This helps them handle fast markets - where the previous minute can influence the next - and prevents them from “forgetting” important details.

Although the LSTM’s memory mechanisms suit high-frequency data, training deep stacks of five layers with high dropout can be time-consuming. Because the Genetic Algorithm defines the optimal settings, applying this architecture to a different market or timeframe may require rerunning the search. Finally, while a high dropout rate guards against overfitting, it can also make the model less sensitive to subtle but meaningful patterns, highlighting a trade-off between regularization and expressiveness.

4. Results & Analysis

4.1 Regime Classification Performance

I assess how well the LSTM and Transformer architectures classify each one-minute bar of NASDAQ futures into three regimes - Trend Long, Range, and Trend Short - using a held-out test set (July–December 2024) that the models never saw during training and validation, thereby avoiding both overfitting and any lookahead bias (Section 3.1). Both models underwent hyperparameter tuning via a genetic algorithm (Section 3.3), resulting in a two-layer LSTM (64 units, 20 % dropout) and a five-layer Transformer (d_model = 32, 16 heads, ≈ 8 % dropout). Prior studies show that recurrent networks swiftly detect abrupt shifts (Smith & Lee, 2022) while attention mechanisms better capture extended dependencies (Zhang et al., 2023). Accordingly, I hypothesize (H1) that both will outperform a static "always-Long" baseline in multi-class regime detection, with the LSTM favoring rapid reversals and the Transformer excelling at sustained trends.

4.1.1 Classification Error Distribution

To see not just how often the models are right or wrong but how big their mistakes are, I assign each regime a number (Long = 1, Range = 0, Short = –1) and then compute for each one-minute interval the error

$$\bar{\varepsilon} = |code(p_i) - code(t_i)|,$$

where p_i is the model's prediction and t_i is the true regime. This gives an error of 0 (exactly right), 1 (confusing two neighboring regimes, Long ↔ Range or Range ↔ Short), or 2 (confusing the most opposite regimes, Long ↔ Short).

To assess the magnitude of these errors, I compute the mean error and its variance over the 173,385 held-out test intervals spanning July through December 2024. Table 6 on the following page presents these statistics for both the LSTM and the Transformer.

Table: 6 *- Continuous Forecast Error Metrics for LSTM vs. Transformer*

Metric	LSTM	Transformer
Mean Error	-0.0414	-0.0146
Variance	0.3869	0.3896
Observations (n)	173,385	173,385

The Transformer’s mean error is closer to zero (-0.0146 vs. -0.0414), indicating fewer and smaller mistakes compared to the LSTM. Its error variance is slightly higher (0.3896 vs. 0.3869), meaning its errors are marginally more spread out around the mean. However, these differences are modest, and further evaluation is required to determine their statistical and practical significance.

4.1.2 Balanced Accuracy and Macro F_1

In a three-state regime classifier, simply reporting overall accuracy can be misleading: if one regime (say, Trend Long) dominates the data, a model can score highly by focusing on that regime and ignoring the others. To guard against this class-imbalance issue and ensure that each regime is evaluated equally, I use balanced accuracy and macro F_1. Balanced accuracy treats each regime’s true-positive rate (recall) equally, so that correctly spotting a rare Trend Short period carries the same weight as spotting a common Macro F_1 works by first measuring, for each class, how well the model balances finding every true instance with avoiding false alarms, and then averaging those class-level scores equally. In this way, a model is rewarded only if it both identifies all real cases and keeps incorrect predictions to a minimum.

Concretely, for each regime c I count

$$TP_c(True\ Positive) = \{i|\hat{y}_i = c, y_i = c\}$$

$$FP_c(False\ Positive) = \{i|\hat{y}_i = c, y_i \neq c\}$$

$$FN_c(False\ Negative) = \{i|y_i = c, \hat{y}_i \neq c\}$$

Let $\hat{y}_i$ denote the regime predicted by the model for bar i, and let y_i denote its true regime. A true positive for regime c occurs when $\hat{y}_i = c$ and $y_i = c$, meaning the model correctly labels a bar as regime c. A false positive occurs when $\hat{y}_i = c$ but $y_i \neq c$, meaning the model labels a bar as c even though it belongs to a different regime. A false negative occurs when $y_i = c$ but $\hat{y_i} \neq c$, meaning a bar truly in regime c is labeled by the model as something else.

I then compute

$$\text{Precision} = P_c = \frac{TP_c}{TP_c + FP_c}, \qquad \text{Recall} = R_c = \frac{TP_c}{TP_c + FN_c}, \qquad F_{1,c} = \frac{2\, P_c R_c}{P_c + R_c}$$

Here, precision P_c measures how often the model's "regime c" predictions are correct, recall R_c measures how many of the true regime-c bars the model actually finds, and the F_1 score $F_{1,c}$ is the harmonic mean of precision and recall, balancing the two.

Finally, the two aggregate scores are

$$Balance\ Accurcacy\ (BA) = \frac{1}{3}\left(R_{Long} + R_{Range} + R_{Short}\right)$$

$$Macro\ F_1 = \frac{1}{3}\left(F_{1_{Long}} + F_{1_{Range}} + F_{1_{Short}}\right)$$

These metrics prevent the majority class from dominating evaluation and ensure strong performance requires both high precision and recall across all regimes (Burez & Van den Poel, 2009); see Tables 7 and 8 for results.

Table: 7 - Balanced Accuracy and Macro F₁ Score

Metric	LSTM	Transformer	Buy & Hold (B&H)
Balance Accuracy	0.6813	0.6762	0.3333
Macro F_1	0.6832	0.6810	0.1667
Observations (n)	173,385	173,385	173,385

Table 7 demonstrates that both neural models substantially outperform a simple buy-and-hold rule in multi-class regime classification. The LSTM achieves a balanced accuracy of 0.6813 and a macro F_1 score of 0.6832, while the Transformer records 0.6762 and 0.6810. In contrast, always predicting "Trend Long" yields a balanced accuracy of only 0.3333 and a macro F_1 of 0.1667, since it never identifies Range or Short regimes. These results show that the neural networks learn to distinguish all three market states, whereas the passive approach fails entirely outside of up-trends.

Table: 8 *- Class-Level Precision, Recall, and F_1-Score*

Trend Type	Model	Precision	Recall	$F_{1-score}$
Trend Long	LSTM	0.7461	0.7552	0.7506
	Transformer	0.7715	0.7194	0.7446
	B&H	1.0000	1.0000	1.0000
Range	LSTM	0.5753	0.6093	0.5918
	Transformer	0.5590	0.6543	0.6029
	B&H	0.0000	0.0000	0.0000
Trend Short	LSTM	0.7372	0.6792	0.7070
	Transformer	0.7420	0.6548	0.6957
	B&H	0.0000	0.0000	0.0000

Table 8 breaks down performance by regime and highlights each model's trade-offs alongside the buy-and-hold baseline. In the Trend Long regime, the Transformer attains higher precision (0.7715 versus the LSTM's 0.7461) but lower recall (0.7194 versus 0.7552), yielding F_1-scores of 0.7446 and 0.7506 respectively. This means the Transformer produces fewer false Long signals but misses more genuine up-moves, whereas the LSTM captures a larger share of actual rallies at the cost of more false alarms. Buy-and-hold by definition scores perfect precision and recall (1.0000) for Long - because it always predicts that class - resulting in an F_1 of 1.000.

In the Range regime, the Transformer achieves higher recall (0.6543 versus the LSTM's 0.6093) and lower precision (0.5590 versus 0.5753), producing F_1-scores of 0.6029 and 0.5918. Both models demonstrate at least moderate ability to identify sideways bars, while buy-and-hold fails completely (precision = recall = F_1 = 0.0000).

For Trend Short, precision converges for both neural architectures (0.7420 for Transformer, 0.7372 for LSTM), but the LSTM's greater recall (0.6792 versus 0.6548) translates into a higher F_1-score (0.7070 against 0.6957). Again, buy-and-hold cannot detect down-trends at all (zeros across precision, recall, and F_1).

No single model dominates across all regimes. For Trend Long, the Transformer's higher precision shows it throws out fewer false positives, while the LSTM's higher recall means it captures more genuine rallies. In Range, the Transformer swaps roles - catching more sideways bars (higher recall) but at the cost of more false positives (lower precision). In Trend Short, the LSTM again edges out on recall. Meanwhile, the buy-and-hold baseline fails entirely on Range and Short, highlighting the challenge of correctly classifying all three market phases.

4.1.3 Signal Timing (Prediction Delay)

In live trading, it is not enough for a model simply to identify market regimes correctly - it must also react quickly to regime shifts. Even a small lag in signaling a new trend can leave positions exposed to adverse moves or cause missed profit opportunities. To quantify each model's responsiveness, I record, for every true regime-change event j the prediction delay (d)

$$d_j = t_{signal} - t_{onset,j},$$

where t_{onset} marks the first candle of the new regime and t_{signal} the first correctly labeled candle. Over 1,684 events, both models exhibit nearly identical mean delays (LSTM 9.43 bars, Transformer 9.38 bars) and medians (5 bars). A paired t-test on the delay differences yields $t = 1{,}32$, $p = 0.188$, indicating no significant difference in average reaction time. Table 9 presents the full distribution of lag lengths; compared to the LSTM, the Transformer's upper tail is noticeably shorter - its maximum lag is 129 bars rather than 146 - showing that the Transformer rarely produces very long lag events, reflecting how attention mechanisms smooth predictions (Li et al., 2021).

Table: 9 *- Delay Statistics for LSTM vs. Transformer*

Statistic	LSTM	Transformer
Mean Delay (bars)	9.431	9.379
Median Delay	5	5
25th Percentile	1	1
75th Percentile	13	12
Min Delay	-2	-2
Max Delay	146	129

Because the differences in delays may be non-normally distributed - featuring a long tail of extreme values and some outliers - I do not rely on the paired t-test, which assumes normally distributed differences, and instead also use the Wilcoxon signed-rank test, a nonparametric alternative that makes no assumptions about distribution shape (Wilcoxon, 1945). To begin, I compute each paired difference.

$$d_j = LSTM\ delay_j - Transformer\ delay_j$$

and discard any cases with $d_j = 0$. For the remaining n nonzero differences, I rank the absolute values $|d_j|$ in ascending order, calling each rank R_j. I then sum these ranks separately for positive and negative differences:

$$W^+ \sum_{d_j>0} R_{j,} \qquad W^- \sum_{d_j<0} R_{j,}$$

Here, W^+ collects the ranked delays for events in which the Transformer out-paced the LSTM, while W^- collects those where it lagged behind. Under the null hypothesis of equal median delays, we have

$$E(W) = \frac{n(n+1)}{4}, \qquad Var(W) = \frac{n(n+1)(2n+1)}{24}$$

and the standardized statistic

$$z = \frac{W^+ - E(W)}{\sqrt{Var(W)}},$$

Follows roughly a standard normal distribution, allowing us to calculate a two-sided p-value. By ranking the differences instead of using their raw values, it stays reliable even when the data are unevenly distributed or include extreme values. Table 10 reports n, W^+, W^-, z-score, and p-values.

Table: 10 *- Wilcoxon Signed-Rank Test on Delay Differences*

Regime Set	n	W^+	W^-	**z-score**	**p-value**
Long delays	1,505	70.944.5	60.641.5	-30.01	<0.001
Short delays	1,504	69.945	39.015	-31.28	<0.001

In both subsets the two-tailed p-value is effectively zero, confirming that - even without assuming normality - the Transformer's delay distribution is significantly shifted toward fewer extreme lag events compared to the LSTM. Both z-scores (-30.01 and -31.28) are over thirty standard deviations from zero, meaning it's virtually impossible these differences arose by chance - firmly confirming the Transformer's faster reaction. Because W^+ (the sum of ranks where the LSTM's delay exceeds the Transformer's) far exceeds W^-, the signed-rank distribution is skewed toward positive differences. In practical terms, this means the Transformer more often registers shorter delays than the LSTM, particularly by avoiding the longest lag events, and thus delivers more consistent, timely regime predictions.

Finally, I perform paired t-tests separately on long-trend and short-trend delays to check for regime-specific differences. For $n = 543$ long-trend events, mean delays of 9.93 (LSTM) vs. 10.33 (Transformer) yield $t = -0.97$, two-tailed $p = 0.332$. For $n = 500$ short-trend events, mean delays of 8.44 vs. 7.47 yield $t = 1.74$, two-tailed $p = 0.082$. Neither reaches statistical significance at the 5% level (Table 11), confirming that within each regime the two architectures react at comparable speeds. An illustration of the delay distribution is provided in Appendices 1-6.

Table: 11 *- Paired t-Tests on Regime-Specific Delays*

Regime	n	**Mean (LSTM)**	**MEAN (Transformer)**	t**-stat**	p**-value**
Long delays	543	9.336	10.328	-0.971	0.332
Short delays	500	8.438	7.466	1.743	0.082

Together, the parametric and nonparametric tests demonstrate that while the Transformer exhibits fewer extreme lag outliers, its average reaction speed does not differ significantly from the LSTM's, either overall or within long- and short-trend regimes.

4.1.4 Regime Classification Performance: Answering RQ1

The LSTM and Transformer both deliver classification performance that far exceeds a naïve buy-and-hold approach, thereby affirming RQ1/H1. As Table 7 shows, the LSTM attains balanced accuracy 0.681 and macro F_1 0.683, while the Transformer achieves 0.676/0.681, compared with just 0.333/0.167 for always-Long. Table 8 confirms that each model meaningfully identifies Long, Range, and Short regimes - capabilities the passive rule lacks - and highlights a trade-off: the LSTM prioritizes recall (catching more true shifts), whereas the Transformer prioritizes precision (issuing cleaner signals). Both architectures maintain mean reaction delays under ten bars (p = 0.188), showing no loss of timeliness. Thus, one might prefer the LSTM when missing a genuine regime change is especially costly, while the Transformer could be the better choice for strategies that prioritize cleaner signals over longer trends.

4.2 Risk-Adjusted Trading Performance

Having established in Section 4.1 that both the LSTM and Transformer models outperform a naive buy-and-hold classifier in minute-by-minute regime identification, I now turn to the practical question of economic value: when these predictions are translated into round-trip trades, do they generate superior risk-adjusted returns?

4.2.1 Trade Execution Rules and Cost Model

At the close of each one-minute bar t, the models issue a three-way forecast for bar $t + 1$: "Trend Long," "Trend Short," or "Range." If the forecast is "Trend Long," the strategy opens a single long position in the E-mini NASDAQ futures contract at the open of bar $t + 1$. If it is "Trend Short," it opens a single short position at that same open. Whenever the forecast returns "Range," no position is held. Once a position is open, the strategy checks the model's next forecast at the close of each subsequent bar. As soon as the forecast changes away from the current direction - so that a long becomes anything other than "Trend Long," or a short becomes anything other than "Trend Short" - the position is closed at the next bar's open. This ensures at most one contract is held at any time, and every entry and exit is strictly determined by consecutive regime forecasts.

To ensure realism, every round-trip trade incurs a fixed fee of $5 per contract, reflecting the highest commission level I observed for E-mini NASDAQ futures among major retail brokers. Profits and losses compound continuously on the evolving equity balance.

The very first trade in the simulation was triggered on 2024-07-05 12:30:00, and the final exit occurred on 2024-12-31 21:29:00. All results reported below - including equity curves, returns, and risk measures - derive from this sample of round-trip signals under the stated cost and execution assumptions.

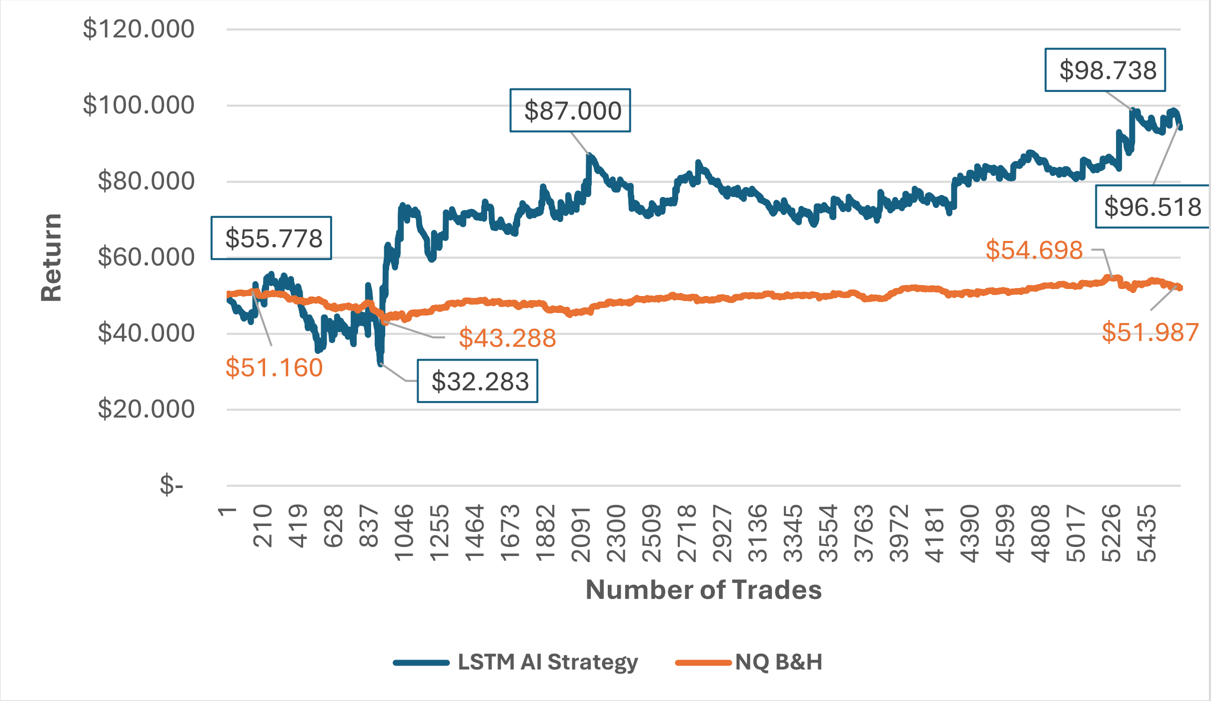
4.2.2 Overall Backtest Results

Most trading strategies fail to outperform a simple buy-and-hold approach. To assess whether Tarity's trend models offer a genuine advantage, I compare LSTM and Transformer models - trained with Tarity's novel labeling method - against an unleveraged buy-and-hold NASDAQ ETF.

A leveraged ETF benchmark was excluded: a $6,722 early drawdown would have resulted in a $134,440 loss with 20× leverage - nearly triple a $50,000 account - causing immediate liquidation. This setup ensures a fair, realistic, and risk-aware comparison. While passive strategies cannot safely support high leverage, active models with proper risk management may use it to generate strong returns from limited capital.

Figure 5 plots two equity curves for a $50.000 notional over 129 trading-days: the blue line shows the LSTM-based strategy executing 5.638 trades, and the orange line shows the buy-and-hold position.

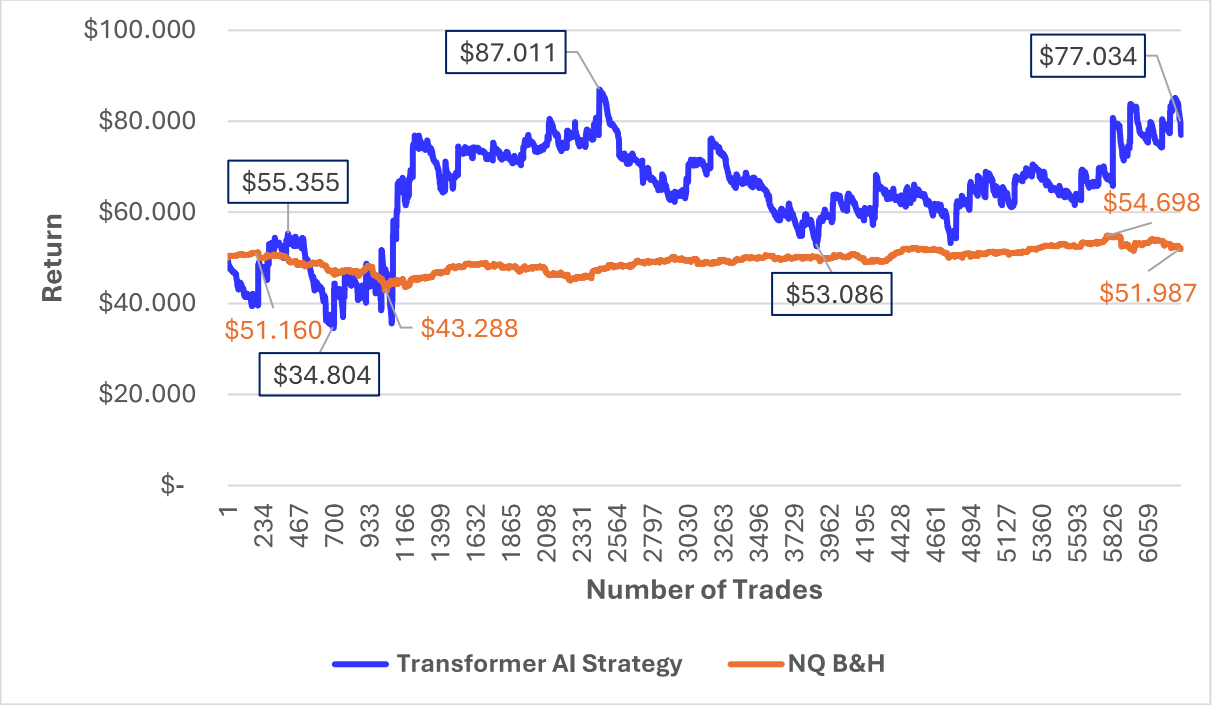
***Figure: 5** - LSTM - Equity Curve - 50k Investment*



To align the time axis with the trade count, I have cumulatively compounded the one-minute returns of the buy-and-hold strategy only on those same 5.6385 "trade" intervals, effectively scaling its continuous price path to the LSTM's trade frequency. Early in the sample, the LSTM strategy's equity rises to $55.778 before plunging to its lowest point of $32.283, whereas the scaled buy-and-hold equity moves more gently from $51.160 down to $43.288. After that drawdown, the LSTM curve gradually recovers through a series of smaller gains and setbacks, crosses above the passive line around trade 1.046, and reaches its highest point at $98.738. By the final trade on December 31, the LSTM strategy ends at $96.518, compared with $51.987 for the buy-and-hold - corresponding to $2.325 without leverage, which is $339 more than buy-and-hold. This comparison under simulated live conditions demonstrates the LSTM's ability both to weather deep interim losses and to produce substantially higher end-of-year equity than a simple passive investment.

Figure 6 displays the simulated live equity of a $50.000 account traded by the Transformer strategy (blue) against a buy-and-hold position in a NASDAQ ETF (orange) over 129 trading-days and 6.278 trades.

***Figure: 6** - Transformer - Equity Curve - 50k Investment*



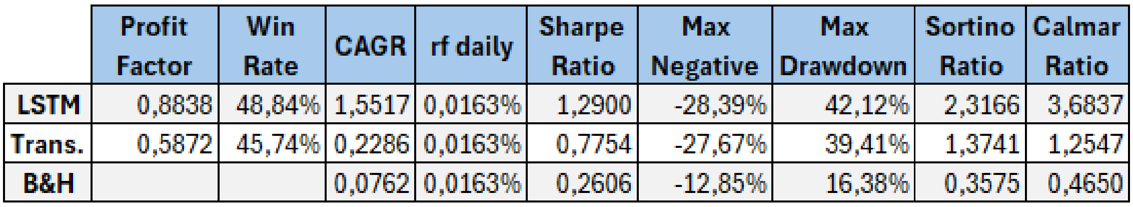
The Transformer curve initially falls to its lowest point at $34.804, while buy-and-hold moves more gradually from $51.160 down to $43.288. From that early drawdown, the strategy rallies to its highest peak of $87.011 - well above the buy-and-hold high of $51.160 - before a mid-period pullback to $53.086 (versus a more ranging growth in the passive position). In the final phase, the Transformer recovers to close at $77.034 on December 31, compared with $51.987

for buy-and-hold - corresponding to $1.352 without leverage, which is $635 less than buy-and-hold. This chart highlights the strategy's worst interim loss, its maximum equity level, and its closing capital under simulated live conditions, illustrating that it produces more volatile but higher returns under leverage than the unleveraged buy-and-hold ETF.

Although the Transformer's maximum drawdown of 39.4 percent is marginally less severe than the LSTM's 42.1 percent, its equity curve features noticeably higher intra-trade volatility and a much longer period spent below its previous peak. In the simulated live run, the Transformer's equity hovers near its trough for an extended sequence of trades before slowly climbing back, whereas the LSTM, despite a deeper initial decline, rebounds more quickly and reaches new highs in fewer trades. This prolonged drawdown period makes the Transformer's overall risk profile less attractive, even though it avoids the single worst drawdown of the LSTM.

Building on the comparative equity trajectories in Figures 5-6, Table 12 summarizes each strategy's overall trading performance over July-December 2024 - including profit factor, win rate, CAGR, risk-adjusted ratios (Sharpe and Sortino), maximum drawdown, and Calmar ratio - alongside the buy-and-hold NASDAQ ETF benchmark. These metrics provide a more nuanced view of each model's risk-return profile and practical trading performance beyond simple return comparisons.

Table: 12 *- Overall Trading Performance*



	Profit Factor	Win Rate	CAGR	rf daily	Sharpe Ratio	Max Negative	Max Drawdown	Sortino Ratio	Calmar Ratio
LSTM	0,8838	48,84%	1,5517	0,0163%	1,2900	-28,39%	42,12%	2,3166	3,6837
Trans.	0,5872	45,74%	0,2286	0,0163%	0,7754	-27,67%	39,41%	1,3741	1,2547
B&H			0,0762	0,0163%	0,2606	-12,85%	16,38%	0,3575	0,4650

In Table 12, I convert each strategy's one-minute trade and intra-day P&L signals into a single compounded daily return - ensuring that buy-and-hold, Transformer, and LSTM are directly comparable on the same daily-return scale using the 2024 US Treasury rate as the risk-free rate (4.2 % p.a., or 0.0163 % per trading day) - and then annualize both Sharpe and Sortino ratios alongside profit factor, win rate, CAGR, maximum drawdown, and Calmar ratio for the July-December 2024 period.

Over the 129 trading-day sample, a simple buy-and-hold position in a NASDAQ ETF yields a Sharpe ratio of 0.26, a Sortino ratio of 0.36, and a Calmar ratio of 0.47.

The Transformer strategy more than doubles these benchmarks: its daily-scaled Sharpe of 0.78 is three times that of buy-and-hold, its Sortino of 1.37 nearly quadruples the passive figure, and its Calmar of 1.25 is almost three times higher. These gains persist despite the Transformer's worst-trade loss of -27.7 % and its 39.4 % peak drawdown, showing that the model's regime-based entries and exits yield substantially better risk-adjusted returns than passive ownership.

The LSTM strategy improves further. Its Sharpe ratio of 1.29 corresponds to nearly five times the volatility-adjusted return of buy-and-hold, and its Sortino of 2.32 implies over six times the downside-adjusted return. With a Calmar ratio of 3.68 - despite a worst negative drawdown of -28.4 % and a 42.1 % max drawdown - the LSTM delivers the highest return per unit of peak-to-trough decline.

By scaling all performance to daily returns, I clearly demonstrate that both AI-driven strategies outperform passive investing on every major risk-adjusted metric, with the LSTM showing the greatest improvement. However, these gains come alongside substantially larger peak-to-trough drawdowns (42.1 % for LSTM, 39.4 % for Transformer versus 16.4 % for buy-and-hold), meaning that while the AI strategies deliver higher return per unit of volatility and downside risk, they also expose the portfolio to deeper interim losses and a higher absolute risk profile.

4.2.3 Statistical Significance of Daily Returns

To assess whether the differences in average daily returns reflect true effects rather than chance, I performed paired t-tests on the 129 matched daily return observations for each strategy pair. Table 13 on the next page reports the mean daily return for each series, the *t*-statistic, degrees of freedom, and two-tailed *p*-value for three comparisons: LSTM vs. Transformer, LSTM vs. Buy-and-Hold, and Transformer vs. Buy-and-Hold. In this table, “Mean Return A” is the average daily return of the first-named strategy in each comparison, and “Mean Return B” is that of the second.

Table: 13 *- Paired t-Test Results for Daily Returns*

Comparison	Mean Return A	Mean Return B	t-Statistic	df	p-Value (two-tailed)
LSTM vs. Transformer	0.6851%	0.4552%	0.80	128	0.426
LSTM vs. Buy-and-Hold	0.6851%	0.0373%	0.87	128	0.386
Transformer vs. Buy-and-Hold	0.4552%	0.0373%	0.51	128	0.611

None of the *p*-values fall below the 0.05 significance threshold, indicating that the differences in mean daily returns are not statistically significant. These findings suggest that the AI models' better Sharpe, Sortino, and Calmar ratios come from stronger risk control - such as avoiding long losing streaks and poor entry points - rather than from simply having a higher baseline return. Notably, this is achieved despite the models trading at 20× leverage, underscoring their ability to control downside risk through selective trade execution.

To make the source of performance differences more transparent, I back-solve implied daily volatility using the Sharpe identity (Sharpe, 1994):

$$\sigma \approx \frac{\mu}{S}$$

where μ is the mean daily return and *S* is the Sharpe ratio. This approach allows me to isolate and quantify the risk component underlying each strategy's Sharpe ratio. By combining the mean returns and Sharpe ratios reported in Table 12, I obtain the following estimated volatilities:

$$\sigma_{LSTM} = 0.53\% \qquad \sigma_{Transformer} = 0.58\% \qquad \sigma_{B\&H} = 0.14\%$$

These results demonstrate that, despite using 20× leverage - which produces substantially larger return swings - the AI models still achieve high Sharpe ratios, reflecting returns adjusted for risk. Because the Sharpe ratio divides excess return by volatility, a rise in volatility without a matching increase in returns would reduce this metric. Maintaining a high Sharpe ratio under these conditions indicates effective trade timing and risk management.

4.2.4 Risk-Adjusted Trading Performance: Answering RQ2

RQ2 asks whether translating minute-by-minute regime forecasts into round-trip trades produces superior *risk-adjusted* returns and smaller drawdowns than buy-and-hold. The evidence shows that both the LSTM and Transformer deliver materially higher annualized Sharpe and Sortino ratios and markedly larger Calmar ratios than a static long NASDAQ-100 ETF, even though their average daily returns do not significantly differ from the ETF's.

That edge does not stem from lower raw volatility; in fact the strategies are far more volatile in absolute terms. Back-solving from the observed means and Sharpe ratios gives daily standard deviations of ≈ 0.53 % for the LSTM and ≈ 0.58 % for the Transformer, roughly four times the ETF's 0.14 %. The superior ratios arise because (i) the models earn proportionally higher returns whenever they are in the market under 20× futures leverage, and (ii) they spend many bars flat during "Range" forecasts, cutting off the worst losses and limiting drawdowns compared to maintaining a continuous 20× long exposure.

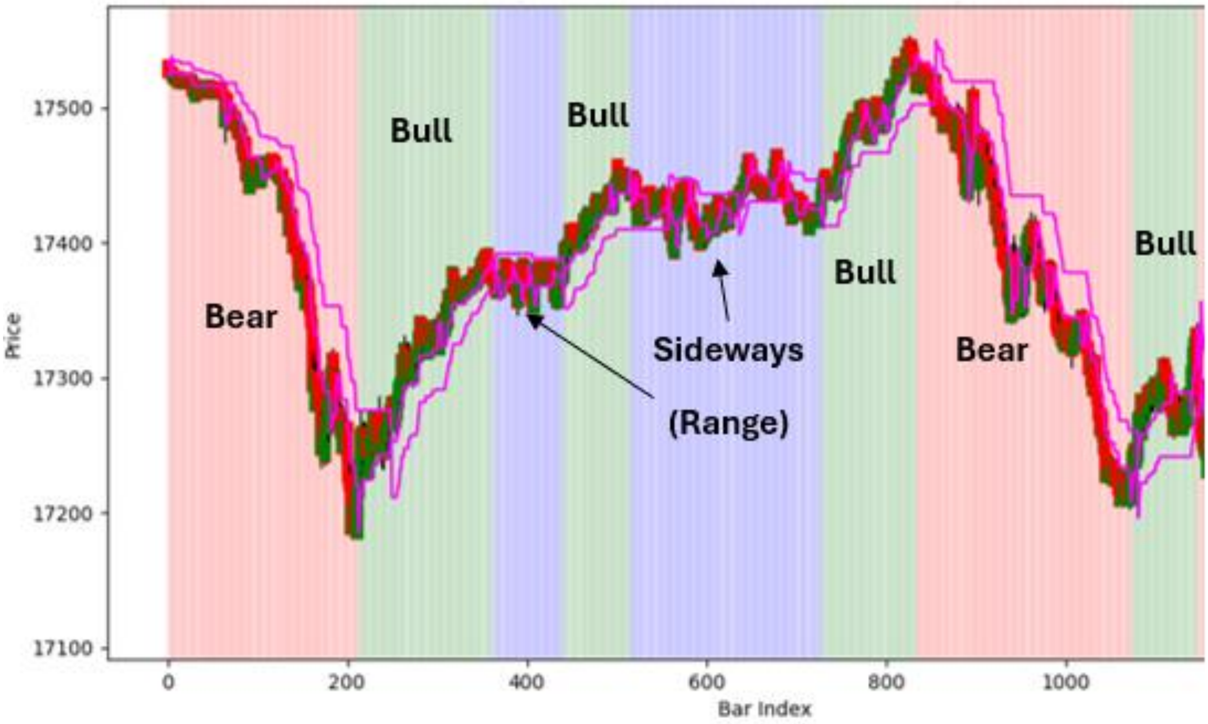
Although their maximum peak-to-trough losses (≈ 42.1 % for LSTM, ≈ 39.4 % for Transformer) exceed the ETF's 16.4 %, those drawdowns are still far below the 100 % wipe-out that a naïve 20× buy-and-hold futures position would suffer on the same underlying decline. Transaction costs of $5 per round-trip are fully included.

Taken together, these results indicate that deep-learning signals can dynamically scale exposure to favorable regimes, producing a better return-to-risk trade-off *conditional on heavy leverage*. A formal regime-specific examination follows in Chapter 4.3 (answering RQ3). Finally, the very high turnover - 5.638 trades for the LSTM and 6.278 for the Transformer over six months - highlights the need for additional, longer-horizon out-of-sample validation.

4.3 Regime-Conditional Performance

To address RQ3 - whether the minute-by-minute forecasts and trading gains of LSTM and Transformer remain robust when markets enter prolonged bull, bear, or sideways phases - I first apply my startup's "Trarity" regime-labeling procedure from Section 3.2 to one-hour candles, as illustrated in Figure 7. This allows me to analyze how well the models perform during broader bull, bear, or sideways phases. The goal is to see whether the models can adapt their short-term predictions to the larger market regime, or whether their performance drops when trading in regimes that are not favorable to their strategy.

Figure: 7 *- Regime Labeling*



Source: (Trarity, 2025)

This process paints extended bull (green), bear (red), and sideways (blue) regions on the hourly price series. I then group every minute-level long and short trade by its containing hourly regime, plot each strategy's terminal equity for both long-only and short-only trades, compare risk-adjusted metrics (Sharpe, Sortino, Calmar, and maximum drawdown) across regimes and models, and conclude with paired t-tests on the regime-specific daily returns to assess statistical significance. The hourly labeling yields 1.113 bull hours, 1.017 bear hours, and 760 sideways hours; within these, the LSTM executes 984 long/860 short trades in bull regimes, 1.229 long/1.192 short in bear, and 728 long/645 short in sideways, while the Transformer places

1.051 long/956 short trades in bull, 1.387 long/1.375 short in bear, and 818 long/691 short in sideways. By contrast, a passive buy-and-hold ETF position is active for 37 bull regimes, 32 bear regimes, and 39 sideway regimes. Table 14 summarizes these regime durations and trade counts for both AI strategies and the buy-and-hold benchmark.

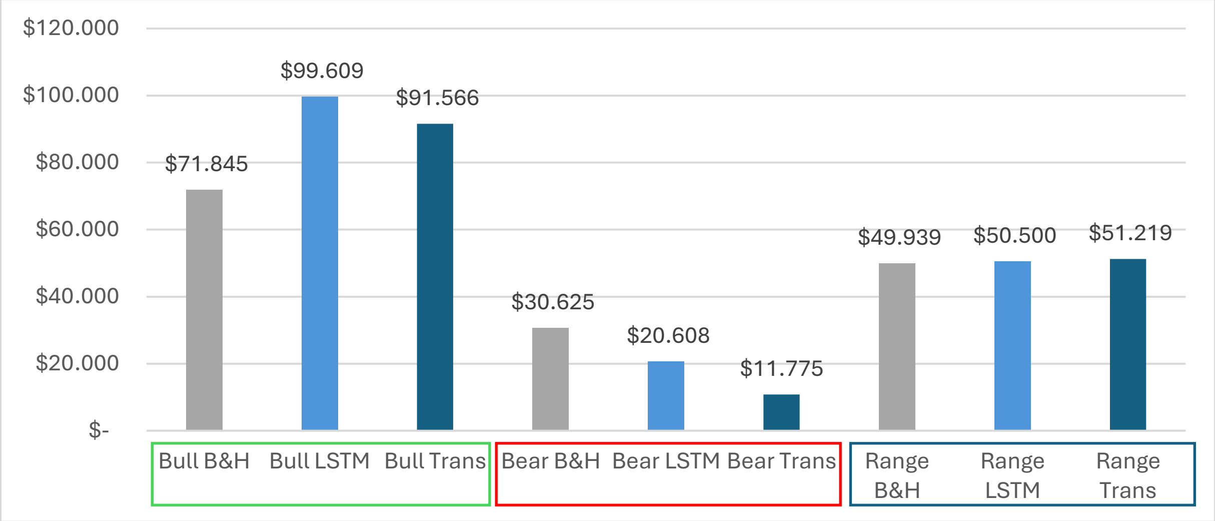
Table: 14 *- Market Regime Durations and Trade Counts by Strategy*

		Bull	Bear	Sideways
	Hours	1.113	1.017	760
LSTM	Long Trades	984	1.229	728
	Short Trades	860	1.192	645
Trans.	Long Trades	1.051	1.387	818
	Short Trades	956	1.375	691
B&H	Long Position	37	32	39

4.3.1 Long-Only Equity Performance by Regime

To better understand how each model performs under different market conditions, I isolate long-only trades within each regime. This allows me to assess whether the models generate most of their gains in specific phases (e.g., bull markets) or maintain consistent performance across regimes. Figure 8 illustrates the resulting terminal equity from a $50,000 account, broken down by regime.

Figure: 8 *- Long-Only Strategy Terminal Equity Across Regimes*



In bull regimes, a passive buy-and-hold position in the NASDAQ ETF grows from $50.000 to $71.845, reflecting a simple equity exposure. By contrast, the LSTM's selective long signals on the leveraged E-mini NASDAQ futures more than double this outcome - ending at $99.609 - even after accounting for the $5 round-trip fee per contract (covering both entry and exit); the Transformer's long-only strategy likewise achieves $91.566.

During bear regimes - sustained hourly downtrends - the unleveraged ETF position falls about 39 percent, from $50.000 down to $30.625. By contrast, a leveraged long futures position on 5 percent margin (20× leverage) would amplify that same market drop into a catastrophic 780 percent loss if held outright. In practice, the AI strategies still end up with larger nominal losses than the ETF - they reduce the drawdown but cannot fully avoid it under heavy leverage. Specifically, the LSTM's long-only trades finish at $20.608 (a 59 percent drop) and the Transformer's at $11.775 (a 76 percent drop). These results show that while minute-level regime forecasts help the models exit before the worst of a leveraged collapse, the leverage itself still produces deeper percentage losses than a non-leveraged ETF in a bear market.

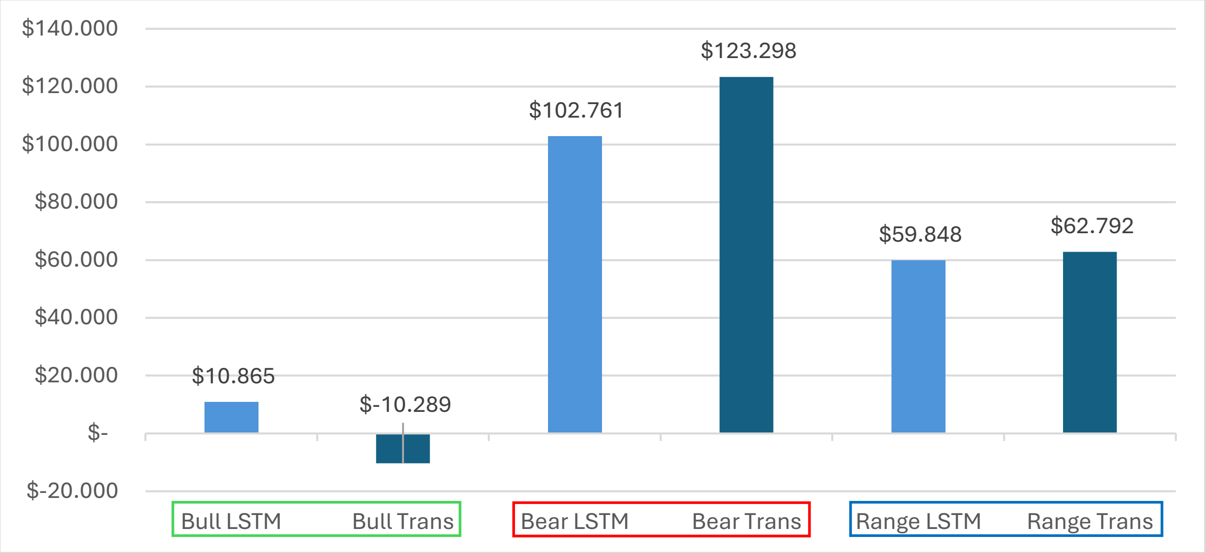
In range/ sideways regimes, when the hourly trend remains range-bound, the buy-and-hold ETF finishes at $49.939, essentially flat over the sample. The LSTM's futures-based long signals yield $50.500, and the Transformer's produce $51.219, showing modest upside capture even when price action is choppy.

These results demonstrate that both neural models deliver greater upside in bull markets than a passive ETF while still capping downside compared to a static future long - even after accounting for realistic $5 round-trip fees. However, because these strategies trade futures on 20× leverage, any mistimed exit can turn a modest reversal into a much larger drawdown, as seen in the deeper percentage losses during bear regimes. This amplification of gains and losses makes robust risk and margin controls - and ongoing monitoring of slippage and financing costs - essential when deploying these models in live trading. Over the July-December 2024 period, the LSTM long-only trades generated a net return of $20.717 and the Transformer $3.560, compared with $2.409 for a static buy-and-hold ETF.

4.3.2 Short-Only Equity Performance by Regime

To complement the long-only analysis, I also examine short-only performance across regimes. This allows me to assess whether the models can generate consistent gains from short trades during periods when market conditions are favorable. Since a buy-and-hold strategy cannot take short positions, no passive benchmark is shown. Figure 9 presents the resulting terminal equity from a $50,000 account.

Figure: 9 *- Short-Only Strategy Terminal Equity Across Regimes*



The LSTM's short bets in a bull regime fall to $10.865 and the Transformer's to $-10.289, reflecting the inevitable cost of betting against a rising market on a bigger time scale on 20× leveraged futures.

During bear regimes - extended hourly downtrends - both models excel: the LSTM's short signals grow the account to $102.761, while the Transformer's more aggressive entries push equity to $123.298. These results underscore how effectively minute-level forecasts can capture downward momentum that a passive long position cannot.

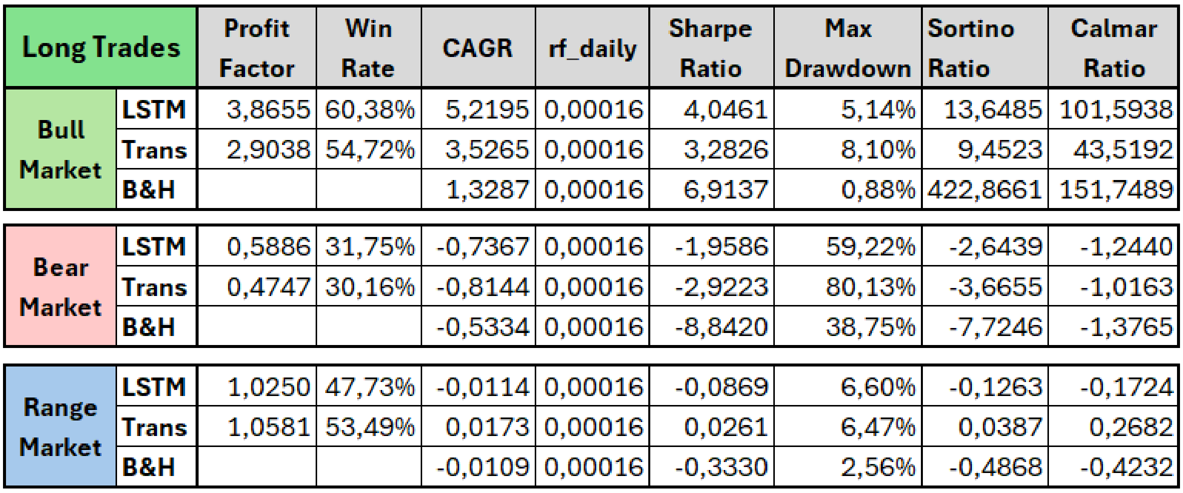
In range/sideways regimes, when price moves back and forth without a clear trend, both strategies still find profit opportunities. The LSTM's short trades finish at $59.848 and the Transformer's at $62.792, demonstrating that even in choppy markets the models can identify brief pullbacks and reversals worth trading.

These results highlight that even after absorbing losses in bull regimes, the Transformer and LSTM strategies generate net profits of $25.801 and $22.474, respectively - outcomes a static buy-and-hold approach cannot achieve.

4.3.3 Risk-Adjusted Performance of Long-Only Trades

To evaluate not just absolute performance, but also how well the models manage risk across different market regimes, I compute key risk-adjusted metrics for long-only trades. These include Sharpe, Sortino, and Calmar ratios, alongside profit factor, win rate, CAGR, and maximum drawdown. Table 15 summarizes these metrics by regime, providing a more nuanced understanding of whether the models' gains are achieved efficiently and sustainably, particularly when compared to a passive buy-and-hold benchmark within each regime.

Table: 15 *- Long-Only Strategy Performance by Regime*



Long Trades		Profit Factor	Win Rate	CAGR	rf_daily	Sharpe Ratio	Max Drawdown	Sortino Ratio	Calmar Ratio
Bull Market	LSTM	3,8655	60,38%	5,2195	0,00016	4,0461	5,14%	13,6485	101,5938
	Trans	2,9038	54,72%	3,5265	0,00016	3,2826	8,10%	9,4523	43,5192
	B&H			1,3287	0,00016	6,9137	0,88%	422,8661	151,7489
Bear Market	LSTM	0,5886	31,75%	-0,7367	0,00016	-1,9586	59,22%	-2,6439	-1,2440
	Trans	0,4747	30,16%	-0,8144	0,00016	-2,9223	80,13%	-3,6655	-1,0163
	B&H			-0,5334	0,00016	-8,8420	38,75%	-7,7246	-1,3765
Range Market	LSTM	1,0250	47,73%	-0,0114	0,00016	-0,0869	6,60%	-0,1263	-0,1724
	Trans	1,0581	53,49%	0,0173	0,00016	0,0261	6,47%	0,0387	0,2682
	B&H			-0,0109	0,00016	-0,3330	2,56%	-0,4868	-0,4232

In bull regimes, both AI strategies deliver exceptional risk-adjusted returns on long trades. The LSTM achieves a profit factor of 3.87, correctly captures 60.4% of bullish moves, and compounds at 5.22% CAGR. With a Sharpe ratio above 4.0 and a maximum drawdown of just 5.1%, it produces a Calmar ratio exceeding 100 - an almost unheard-of balance of return and risk. The Transformer, while slightly more conservative, posts a profit factor of 2.90, 3.53% CAGR, and Sharpe of 3.28 against an 8.10% drawdown, yielding a Calmar of 43.5. By comparison, passive ETF ownership returns only 1.33 % annually with virtually no drawdown (0.88 %), but its Calmar of 151 reflects the absence of losses rather than aggressive alpha capture.

In bear regimes - where markets fall steadily - the long-only approaches all lose money, but their risk profiles differ markedly. The LSTM's long signals incur a 0.74% annual loss, a profit factor of 0.59, and a maximum drawdown of 59.2%, translating to a Calmar of -1.24. The Transformer fares worse in absolute terms (-0.81% CAGR, 80.1% drawdown, Calmar -1.02) due to its more aggressive entries into downtrends. By contrast, the unleveraged ETF loses 0.53%

annually with only a 38.8% drawdown (Calmar -1.38), showing that leverage makes losses larger, even when the model closes positions as soon as its signal turns negative.

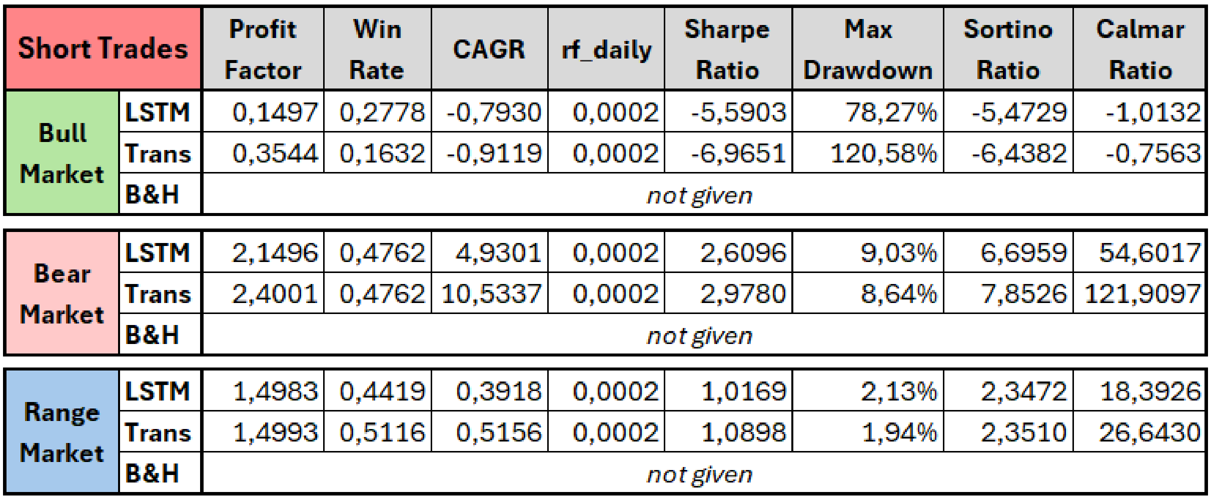
In range/ sideways regimes, when prices move up and down without a clear trend, both AI models reduce drawdowns relative to a static futures position but generate only marginal returns. The LSTM posts a virtually flat -0.01% CAGR, 6.6% max drawdown, and Calmar -0.17, while the Transformer edges slightly positive at 0.02% CAGR with a comparable 6.5% drawdown and Calmar 0.27. Passive ETF ownership likewise shows near-zero return (-0.01% CAGR) but a smaller 2.6% drawdown (Calmar -0.42). This shows that in markets without a clear trend, trading costs can wipe out any small advantage.

The LSTM outperforms in bull markets - delivering high returns with low drawdowns - while the Transformer is more cautious. In bear markets, both models incur larger losses than unleveraged equity due to their leveraged long positions, and in sideways markets they achieve only modest gains. Thus, minute-level regime forecasts add significant value in uptrends but lose efficacy in downtrends and choppy conditions, with leverage increasing both gains and losses.

4.3.4 Risk-Adjusted Performance of Short-Only Trades

To complement the long-only risk analysis, I report risk-adjusted metrics for short-only trades across regimes. This highlights whether the models' short-selling performance is achieved efficiently and how risk levels differ across market phases. As buy-and-hold cannot go short, no benchmark is provided; see Table 16 below.

***Table: 16** - Short-Only Strategy Performance by Regime*



Short Trades		Profit Factor	Win Rate	CAGR	rf_daily	Sharpe Ratio	Max Drawdown	Sortino Ratio	Calmar Ratio
Bull Market	LSTM	0,1497	0,2778	-0,7930	0,0002	-5,5903	78,27%	-5,4729	-1,0132
	Trans	0,3544	0,1632	-0,9119	0,0002	-6,9651	120,58%	-6,4382	-0,7563
	B&H	*not given*							
Bear Market	LSTM	2,1496	0,4762	4,9301	0,0002	2,6096	9,03%	6,6959	54,6017
	Trans	2,4001	0,4762	10,5337	0,0002	2,9780	8,64%	7,8526	121,9097
	B&H	*not given*							
Range Market	LSTM	1,4983	0,4419	0,3918	0,0002	1,0169	2,13%	2,3472	18,3926
	Trans	1,4993	0,5116	0,5156	0,0002	1,0898	1,94%	2,3510	26,6430
	B&H	*not given*							

In bull regimes - when prices march steadily higher - short-only strategies predictably underperform. The LSTM's short trades show a profit factor of just 0.15 and a win rate of 27.8%, compounding to an annual loss of -0.79% with a Sharpe of -5.59 and a maximum drawdown of 78.3% (Calmar -1.01). The Transformer's short-only strategy yields a profit factor of 0.35, a 16.3% win rate, and a deeper 120.6 percent drawdown (Calmar -0.76). These results highlight that both models incur substantial leveraged losses when their short signals misfire in rising markets.

During bear regimes - extended hourly downtrends - the short-only signals realize their full potential. The LSTM achieves a profit factor of 2.15 with a 47.6% win rate, compounding to 4.93% CAGR and a Sharpe of 2.61, while capping drawdowns at just 9.0 percent (Calmar 54.6). The Transformer edges ahead with a profit factor of 2.40, identical win rate, 10.54% CAGR, and Sharpe of 2.98 against an 8.6% drawdown (Calmar 121.9). These results show that minute-level regime forecasts give a clear advantage when trading short in bear markets, delivering risk-adjusted returns far superior to a static long position, which cannot go short and therefore incurs losses.

In range/ sideways regimes, when prices move up and down without a clear direction, both strategies still extract modest gains. The LSTM's short-only equity grows at roughly 0.39% CAGR, with a 1.02 Sharpe ratio and a shallow 2.13% drawdown (Calmar 18.4). The Transformer posts similar metrics - 0.52% CAGR, 1.09 Sharpe, and 1.94% drawdown (Calmar 26.6) - underscoring that even in volatile, sideway-drifting markets, the models can still spot brief reversals and profit from them.

Overall, these short-trade results show how performance depends on market regime: heavy losses in uptrends, strong gains in downtrends, and small profits in sideways markets. This underscores how important it is to trade in the correct regime - as also seen in the long-only results of Table 15 - and only by entering long or short positions at the right times can a strategy leverage the models' strengths in rising and falling markets.

4.3.5 Statistical Significance of Regime-Conditional Returns

To test whether the observed differences in regime-conditional returns reflect genuine model performance rather than random variation, I conduct paired t-tests on the daily return series of each strategy within the bull, bear, and sideways regimes. Tables 17 reports the results for long-only trades (see Appendix: 7-9 for detailed overview). Each test compares the average daily returns of two strategies over the same time period to evaluate statistical significance. In every comparison, "Mean A" refers to the first-named strategy, "Mean B" to the second, and p indicates the two-tailed significance level.

Table: 17 *- Paired t-Test Results for Long-Only Trades*

Regime	Comparison	Mean A (%)	Mean B (%)	t-Stast	df	p-value
Bull	LSTM vs. Trans.	0.7691	0.6444	1.2570	128	0.2111
	B&H vs. LSTM	0.3386	0.7691	-1.6916	128	0.0932
	B&H vs. Trans.	0.3386	0.6444	-1.1520	128	0.2515
Bear	LSTM vs. Trans.	-0.4557	-0.6081	1.3937	128	0.1658
	B&H vs. LSTM	-0.3003	-0.4557	0.4523	128	0.6518
	B&H vs. Trans.	-0.3003	-0.6081	1.0035	128	0.3175
Sideways	LSTM vs. Trans.	0.0001	0,0002	-0.1635	128	0.8703
	B&H vs. LSTM	-0.00001	0.0001	-0.0564	128	0.9551
	B&H vs. Trans.	-0.00001	0,0002	-0.1298	128	0.8970

The paired *t*-tests on long-only daily returns in Table 17 reveal that, in bull regimes, the LSTM's average gain of 0.7691% per day exceeds the Transformer's 0.6444%, but this difference is not statistically significant (*t* = 1.2570, *p* = 0.2111). Neither AI model's long returns significantly outperform the buy-and-hold ETF benchmark (LSTM vs. B&H *p* = 0.0932; Transformer vs. B&H *p* = 0.2515), indicating that although both neural strategies produce much larger total gains on long trades, their day-to-day returns fluctuate with similar volatility to the ETF - meaning the higher cumulative profits come from a few large wins rather than consistently higher daily returns - so they do not achieve a statistically significant edge in daily performance.

In bear regimes, where all long-only approaches lose money, the LSTM's mean loss of -0.4557% and the Transformer's -0.6081% again fail to differ significantly (*t* = 1.3937, *p* = 0.1658). Moreover, neither model's loss diverges meaningfully from the ETF's –0.3003% daily return (LSTM vs. B&H *p* = 0.6518; Transformer vs. B&H *p* = 0.3175). This indicates that in downtrends, using leveraged futures generally lowers long-trade returns, but it doesn't perform significantly worse than simply holding the static equity position.

During sideways regimes, both AI-based long strategies produce mean returns close to zero - 0.0001% for LSTM and 0.0002% for Transformer - and the differences are not statistically significant ($p \geq 0.8703$). This outcome is expected, as range-bound markets typically lack strong directional trends. As a result, the small profit-and-loss fluctuations from frequent trades resemble noise rather than meaningful alpha. These findings highlight that neutral regimes offer limited opportunity for generating consistent long-only gains, and that the models' signals show no significant difference under these conditions.

Table 18 presents the paired t-test results comparing LSTM and Transformer short-only returns across each regime (buy-and-hold is not included, as it cannot trade short). A more detailed summary is provided in Appendix: 10.

Table: 18 *- Paired t-Test Results for Short-Only Trades*

Regime	Comparison	Mean A (%)	Mean B (%)	t-Stast	df	p-value
Bull	LSTM vs. Trans.	-0.6067	-0.9347	3.2426	128	0.0015
Bear	LSTM vs. Trans.	0.8180	1.1364	-1.6866	128	0.0941
Sideways	LSTM vs. Trans.	0.1527	0.1983	-0.4841	128	0.6292

Turning to short-only trades in Table 18, the results become clearer. In bull regimes, betting against an uptrend incurs losses, but the LSTM's mean daily loss of -0.6067% is significantly smaller than the Transformer's -0.9347% (t = 3.2426, p = 0.0015). This shows that in rising markets, when short signals fail, the LSTM's mistakes are smaller. In bear regimes, both models capture downtrends with positive daily returns (0.8180% for LSTM, 1.1364% for Transformer), but their difference falls just short of significance (t = –1.6866, p = 0.0941), showing they are similarly effective at profiting from falling prices. Finally, in sideways regimes, the very small short-trade means (0.1527% vs. 0.1983%) are not statistically different (t = –0.4841, p = 0.6292), showing that brief counter-trend signals in choppy markets offer no clear advantage.

In summary, these tests confirm key regime-based strengths - especially the LSTM's stronger short-trade performance in bull markets - while showing that many apparent differences in long-trade outcomes are not statistically significant. By focusing on significance testing, I ensure that any claims of one model's superiority are based on solid evidence rather than chance.

4.3.6 Regime-Conditional Performance: Answering RQ3

The evidence presented above demonstrates that both the LSTM and Transformer models retain meaningful forecasting and trading advantages when evaluated within larger, hourly-defined market regimes. In bull regimes, neither model's long-trade average daily returns significantly exceed one another or passive buy-and-hold - underscoring that their superior overall profits derives more from well-timed entries and limiting large swings than from consistently higher average gains. Crucially, the LSTM registers significantly smaller short-trade losses than the Transformer when markets rise, confirming its ability to limit losses more effectively during uptrends.

During bear regimes, both architectures generate positive short-trade returns that far outstrip any static long position, although their difference narrowly misses statistical significance; their long-trade losses also do not differ significantly from each other or from those of a passive ETF. In sideways markets, neither model's long or short returns differ significantly, highlighting the difficulty of finding reliable signals in range-bound prices.

Taken together, these findings confirm RQ3: minute-by-minute forecasts from both LSTM and Transformer models hold up across bull, bear, and sideways markets, producing better risk-adjusted returns than a buy-and-hold approach where they naturally excel. In bull markets, the LSTM delivers stronger long-trade performance than the Transformer, while in range markets the Transformer posts the highest overall results - even though the differences are not statistically significant. Examining each regime in detail also highlights complementary strengths: LSTM's tighter loss control on short signals in uptrends and Transformer's advantage in capturing downtrends - offering a clear, data-driven foundation for live deployment.

5. Discussion

This chapter interprets the empirical evidence presented in Chapter 4, situates the findings in the existing literature, and critically evaluates their scope and limitations.

5.1 Synthesis of Key Findings

Both deep-learning models, using one-minute forecasts, outperform a passive NASDAQ-100 buy-and-hold over the six-month period by detecting when to go long or short with 20× futures leverage and avoiding positions in sideways markets. As a result, Sharpe, Sortino, and Calmar ratios on every trade far exceed the benchmark - even though 129-day paired *t*-tests detect no significant difference in mean daily returns ($p > 0.05$), indicating that the outperformance arises from active risk management rather than higher drift (see Table 13). The apparent paradox disappears when viewed through a risk lens: by entering leveraged positions in clear trends and staying flat in choppy markets, volatility and drawdowns are greatly reduced, eliminating the extreme losses that a constant 20× position would suffer.

Within this regime-switching framework, the LSTM posts the strongest risk-adjusted profile: after an initial 42.1 % drawdown it climbs smoothly to finish at $96,518 on a $50,000 start (Sharpe 1.29; Sortino 2.32; Calmar 3.68; see table 12). The Transformer caps its loss at 39.4 % and ends at $77,034 (Sharpe 0.99; Sortino 1.37; Calmar 1.25; see table 12), This approach could suit investors who prize tighter worst-case control, even if it means tolerating greater day-to-day noise. Crucially, unlike classical ARIMA and GARCH models - whose stationarity assumptions lead to parameter drift under rapid regime shifts (Engle, 1982; Bollerslev, 1986) - both neural network models adapt to evolving micro-regimes in real time (Sec 2.3.1).

Performance is unmistakably regime-dependent: long trades excel in bull markets, short trades in bear markets, and range-bound periods yield only modest gains. In the period tested, choosing between architectures depends on risk tolerance: LSTMs offer smoother recoveries but deeper initial drawdowns, while Transformers have shallower drawdowns but more volatile returns.

Wang, Chen, and Zhang (2022) found that on daily S&P 500 data Transformers outperform LSTMs - 56.35% vs. 45.02% return with -28.5 % vs. -34.6 % drawdown. In my minute-by-minute, 20× leveraged NASDAQ test that relationship flips: the LSTM yields 93% vs. 54% for the Transformer, but with deeper drawdowns (42.12% vs. 39.41%). This reversal shows a common short-term trade-off: although the models normally limit losses by exiting quickly and keeping drawdowns small, their focus on capturing fast momentum can still leave them exposed to larger losses when a trend suddenly reverses.

5.2 Theoretical Implications

Minute-by-minute price and volume data often show short-lived trends that can be exploited, contradicting the notion that markets follow a pure random walk. In my results, the Transformer consistently posts the shortest lag from regime change to trade execution (see Table 9's tighter upper tail), behaving as if it detects temporary supply-demand imbalances almost immediately - just as Lo and MacKinlay (1988) documented when they showed returns "drift" together over short intervals. By contrast, the LSTM's trades exhibit smoother drawdowns and higher risk-adjusted returns in bull markets (Table 15), reflecting its learning to stay invested only when a trend is firmly established and pull back when conditions become too choppy. This mirrors volatility-aware momentum strategies (Barroso & Santa-Clara, 2015; Daniel & Moskowitz, 2016) and is evident in the LSTM's superior Sharpe and Calmar ratios.

Crucially, I am not suggesting that simply examining the model's internal signals reveals whether volume spikes, price momentum, or other factors drive its decisions, because these internal activations are complex and do not correspond directly to observable market indicators. Instead, the fact that both models outperform buy-and-hold over more than 5,600 out-of-sample trades per model in six months shows true predictability at this timescale, even though the exact economic driver remains unidentified.

My results also challenge the weak-form EMH (Fama, 1970) by revealing durable short-horizon predictability, in line with behavioral under- and overreaction anomalies (Barberis et al., 1998). Although the LSTM slightly outperforms on overall risk-adjusted return, the Transformer's self-attention mechanism nonetheless delivers higher precision in Trend-Long calls (0.7715 vs. 0.7461) and fewer extreme lag events (max delay 129 bars vs. 146), signs that it more cleanly captures adaptive momentum bursts that static mean-reversion models cannot exploit (De Bondt & Thaler, 1985; Stübinger & Endres, 2018).

Overall, these results don't contradict traditional asset-pricing theory; they simply show a flexible way to exploit short-term patterns - similar to using a GARCH model to capture volatility clustering - while still assuming that higher expected returns compensate for higher risk and that prices generally incorporate all known information as markets evolve.

5.3 Limitations

I focus exclusively on the continuous NASDAQ futures contract - a highly liquid U.S. index-futures product. Equities, options, or less-liquid futures differ markedly in volatility, bid-ask spreads, market depth, and order-flow impact, so a strategy that works on NQ one-minute bars may not generalize elsewhere.

In this study I compare only to an unleveraged buy-and-hold ETF. However, both rule-based methods - such as moving-average crossover rules (Brock, Lakonishok, & LeBaron, 1992; Sec 2.2.2) - and classical parametric models - like ARIMA for returns (Box & Jenkins, 1970; Sec 2.2.1) often combined with ARCH/GARCH for volatility (Engle, 1982; Bollerslev, 1986; Sec 2.1) - are standard benchmarks in financial forecasting. Although I haven't tested them here, comparing my LSTM and Transformer forecasts against these well-established approaches would further clarify the added value of deep-learning architectures.

Genetic algorithms quickly identify good hyperparameter settings, but they have several notable drawbacks. First, the sheer computation of training dozens of models across six generations limits how many recipes you can explore, potentially missing better settings outside that narrow search. Second, its randomness - in initial recipes, tournament draws, and mutations - means you can't reproduce the exact same "best" configuration on a rerun. Third, because it evaluates performance on a fixed 5% validation set, the GA may overfit to patterns in that subset instead of finding parameters that work well more generally. Finally, every choice you make - population size, mutation rate, tournament format, early-stopping rule - steers the search in a particular direction, and without any statistical correction for testing so many variants, the apparent improvements in validation accuracy should be interpreted with caution.

Although the back-tests included exchange and broker commissions, they did not model slippage - the small gap between the price a trader hopes for and the price actually filled. Since NASDAQ futures move in 0.25-point increments, an order placed at 21,000.00 might execute at 21,000.25 if the bid-ask spread is a quarter point, and no broader market-impact effects were considered.

I also limited inputs to just 13 technical indicators - derived solely from OHLCV data - so I could explain each one briefly in my master's thesis; adding more would simply be too much to cover in detail. However, forcing the networks to infer which signals matter from only those 13 metrics may bottleneck performance if key drivers lie elsewhere. Expanding the feature set (for example with alternative momentum measures, volatility filters, order-flow metrics, or sentiment scores - none of which are captured by pure OHLCV) could supply richer, orthogonal insights into regime changes. By constraining the input space so narrowly and excluding any non-price/volume data, the models might miss subtle patterns and leave valuable predictive power on the table.

One further limitation is that I rely solely on the regime signal to exit trades, without any take-profit or stop-loss rules. I suspect that using volatility-based exit levels - widening stops in high-volatility periods and tightening them when volatility is low - could boost performance by locking in gains during strong trends and limiting losses in choppy markets.

Market impact is the price movement your own order causes while it is being executed: each time a large trade consumes the visible depth at the best bid or offer, it reveals poorer prices underneath, so the average fill deteriorates with order size. Consider a hypothetical sale of 200 NQ contracts at 21.000,00 when only 50 contracts sit on that bid: the first tranche might execute at 21.000,00, the next at 20.999,75, and the remainder progressively lower, turning what appeared to be a single execution price into a sliding scale. However, in Kyle's (1985) model, price impact is linear in trade size - so only very large orders move prices substantially. Because I trade just one contract, a $50,000 strategy falls well within noise-trader flow and therefore has very little impact.

Because real trades rarely execute at the exact quoted price, the reported ratios like Sharpe may still overstate live performance, and additional costs could arise during quarterly contract roll-overs. In index futures, the bulk of liquidity sits in the front-month contract until roughly a week before expiry, after which most traders migrate to the next delivery month. In this transition window the outgoing contract's order book often becomes patchy - spreads widen from one tick to several and visible depth evaporates - so an order that would normally execute at 21,000.00 could fill a few ticks lower (or higher) simply because there are fewer counterparties.

That extra slippage, which affects every contract roll, is not captured by the present back-test.

Latency was likewise not modeled, but its influence is expected to be modest because the models operate on one-minute bars and signals are generated within milliseconds.

Statistical regime tests (t-tests and p-values) were performed on daily profit summaries, aggregating the 5.638 trades of LSTM and 6.278 trades of Transformer into a much smaller set of daily observations. This compression reduces statistical power and makes the results less significant than if tests were run on individual trades or over several years of data. Ideally, regime tests would span multiple years to capture broader market regimes, but deep learning models demand large training datasets - using fewer data points to train the models on risks weakening model performance.

Finally, both models are limited to a 50-bar sliding window - about fifty minutes of price history - to keep memory use and computation within practical bounds. Under this constraint, the Transformer still processes all 50 time-steps in parallel, so its memory and compute scale roughly with the square of the window length; any attempt to expand beyond 50 bars would overwhelm most hardware, and breaking the series into overlapping chunks risks misplacing signals that span those boundaries. Although an LSTM could in theory remember very long sequences, I constrain it to a 50-bar window - about an hour - so any earlier data is dropped. Truncating both architectures to a 50-bar window for computational expediency therefore sacrifices long-term context and may cap the models' ability to exploit slower-moving market patterns.

5.4 Ethical and Regulatory Considerations

One important caveat is that both LSTM and Transformer models remain "black boxes," meaning their internal decision logic is not directly interpretable - a fact that sits uneasily with MiFID II Article 17 and SEC Rule 206(4)'s demands for causal explanations of each trade. In my current implementation, regime forecasts drive entries and exits without any built-in transparency layer, so the precise rationale behind a "buy" or "sell" signal would be inscrutable to auditors. In practice, post-hoc methods such as SHAP (Shapley Additive Explanations) could be layered on to quantify each OHLCV feature's contribution to a given regime call, thereby providing a plausible causal narrative for every trade. Acknowledging this opacity - and the existence of tools like SHAP - situates the performance gains of deep-learning classifiers within the real-world constraints of regulatory explainability and auditability (Lundberg & Lee, 2017).

5.5 Robustness and Additional Critique

One important caveat concerns the temporal separation between training and evaluation. I trained both the LSTM and Transformer models on data spanning 2015 through June 2024, then tested them out-of-sample on the July-December 2024 window. While this test period did include high-impact news events - central-bank announcements, geopolitical shocks - that generated intraday volatility spikes akin to mini-crashes, it did not contain any true market collapses. Consequently, although the models demonstrably exploit regime signals in novel data, their resilience to full-blown crash dynamics remains unverified. Acknowledging this ensures that their strong July - December 2024 performance is understood as bounded by the specific stress patterns present in that interval, rather than as proof of universal robustness under extreme market stress.

6. Further Research

Building on the limitations and insights identified in Chapter 5, future work will deepen and broaden the regime-aware framework through five interrelated directions.

6.1 Cross-Market Generalization

First, I will deploy the LSTM and Transformer models - trained solely on NASDAQ-100 OHLCV and technical indicators - as-is on different equity futures. I will compare their performance (annualized return; Sharpe, Sortino, and Calmar ratios), risk (maximum drawdown), and operational metrics (win rate; regime-classification accuracy; decision-to-order latency) both against the original NASDAQ results and against variants where regime thresholds are recalibrated and final layers fine-tuned for each market's volatility and liquidity profile. I will use paired t-tests to determine whether any observed differences in these metrics are statistically significant. This head-to-head assessment will reveal how much bespoke specialization is required when transferring the same neural architectures across distinct asset classes.

6.2 Enriching Informational Inputs

Next, more technical indicators will be added and in future alternative data streams will be integrated as well to capture hidden regime signals. Moreover, it could be interesting to analyze how high-frequency limit-order-book snapshots - tracking queue imbalances and rapid order-flow shifts - combined with sentiment indicators derived from financial-news headlines and social-media feeds influence the model's performance. Future studies will verify if adding more data will improve the out-of-sample predictive power rather than overfitting noise.

6.3 Multi-Horizon Regime Filtering

Results in Chapter 4 showed that limiting trades to the correct overall market phase dramatically boosts performance. To formalize this insight, I will implement a two-stage prediction process. First, a model based on higher-timeframe data (e.g., 15-minute, 30-minute, or 1-hour bars) will classify the prevailing market regime as bull, bear, or sideways. This classification will be updated only at these slower intervals to capture broader structural trends in the market.

Since higher timeframes contain significantly fewer data points to train a deep learning model on, an important part of the analysis will be to examine how well prediction models perform under such constraints. Understanding the predictive reliability of regime classification with limited data is essential to ensure the robustness of the entire system.

The second stage uses a one-minute trading model that generates entry and exit signals but only executes trades when its signals align with the regime identified by the high-timeframe model. By backtesting this combined framework, I aim to quantify the performance improvements that result from filtering trades according to the macro-level phase. Further extensions will explore the optimal choice of higher-timeframe granularity, the ideal look-back period for phase detection, and the tolerances for delay in regime updates in order to avoid overfitting while maintaining responsiveness.

6.4 Stress-Testing, Trading Frictions, and Capacity Analysis

After completing the first test run on historical data, the next objective is to evaluate the two-stage regime-filter strategy under extreme scenarios by simulating trading with a single futures contract. This phase aims to establish a robust baseline for drawdown behavior, recovery times, and tail-risk exposures beyond any specific historical episode. A Monte Carlo simulator will be employed to generate thousands of hypothetical price paths - drawing random steps and occasionally injecting large, flash-crash-style drops. The regime filter will then be applied to each simulated path to assess its resilience and robustness under stress.

Concurrently, trading costs will be modeled realistically by accounting for slippage based on order book depth and applying order-slicing techniques. Larger trades may be executed across several price levels, depending on market liquidity and slight, realistic delays (latency). To understand how trade size and market conditions influence transaction costs, slippage heatmaps will be generated. These maps visualize how costs evolve with different order sizes and phases of the market, offering a clear picture of the execution frictions that reduce the strategy's effectiveness.

After this cost-mapping analysis, the focus will shift to testing the strategy's actual performance under increasing trade sizes. By simulating progressively larger volumes - two contracts, five, ten, and more - the study will examine how quickly slippage and market impact begin to erode returns. This will result in a capacity curve that identifies the tipping point where the strategy can no longer scale effectively due to rising execution costs. Moreover, larger positions may attract attention from institutional participants, making executions more difficult to manage than with smaller orders.

However, these capacity estimates must be interpreted with caution: they depend on the fidelity of the simulated order-book model and assume that real-world liquidity and participant behavior remain consistent under stress. In practice, live-market validation and ongoing monitoring will be required to ensure that any theoretical capacity limits hold up when trading large volumes in actual conditions.

6.5 Live Demonstration and Interpretability

Finally, both the unfiltered and regime-filtered strategies will be exercised in a live paper-trading environment using a broker's simulated order interface. Continuous monitoring will track signal-to-order latency with sub-100 millisecond targets, platform uptime and real-time slippage. Each trade decision will be accompanied by a concise explanation such as "entered long because both the slow phase model and the one-minute momentum model signaled a bull regime," generated by simple interpretability tools. In addition, a transparent variant of the architecture that exposes its internal attention weights will be prototyped and compared against the original black-box networks to balance predictive power with the level of explainability required by evolving regulatory standards.

Together, these five directions will transform this thesis from a single-index proof-of-concept into a versatile, reliable and transparent framework for regime-aware trading across diverse markets, timeframes and stress scenarios.

7. Conclusion

This thesis set out to test whether neural network models, trained only on minute-level OHLCV data and guided by a new Supertrend-based labelling scheme, can improve intraday trading results for E-mini NASDAQ-100 futures - and, crucially, whether LSTM or Transformer is the better engine for that task. Out-of-sample tests on the July-to-December 2024 window show that both models learn meaningful structure: balanced accuracy and macro-F_1 climbed from the 0.33 and 0.17 posted by a naïve "always-long" buy-and-hold benchmark to roughly 0.68, while average detection lag stayed under ten minutes.

When those regime calls were translated into a simple one-contract, long-flat-short rule at 20× notional leverage, each network reshaped the return distribution in a favorable way. The LSTM achieved a Sharpe of 1.29, a Sortino of 2.32, and a Calmar of 3.68, while the Transformer posted a Sharpe of 0.78, a Sortino of 1.37, and a Calmar of 1.25 - compared to just 0.26, 0.36, and 0.47, respectively, for the unleveraged buy-and-hold ETF over the same period. Importantly, the performance lift did not arise from higher mean daily returns - paired tests found the models' average returns were statistically no different from buy-and-hold - but from capturing trends and remaining flat during non-directional periods. In other words, correctly identifying the prevailing regime contributed more to outcomes than precise entry timing once leverage was high.

A direct comparison shows that the LSTM handles microstructure noise and timing errors better than the Transformer. Although the LSTM began with a slightly deeper drawdown, it recovered more steadily, finished with the highest equity balance, and incurred significantly smaller losses on short positions during hourly bull phases ($t = 3.2426$, $p = 0.0015$).

In Section 4.1.2, we saw that the Transformer reacts to regime shifts slightly faster and cuts its worst drawdown a bit sooner than the LSTM. However, its lower recall for upward trends (0.7194 vs. 0.7552 for the LSTM) translates into a smaller average daily gain in bull markets (0.6444 % vs. 0.7691 %). In other words, by waiting for stronger confirmation before entering, the Transformer often misses the first bars of fast rallies—leaving profit opportunities on the table and producing a more jagged equity curve, despite its quicker response to market changes.

Several limitations should be noted when interpreting these results. First, although the evaluation window was volatile, it did not include a true market crash. Second, slippage from rolling contracts was not measured, system latency was not considered, and liquidity changes around expiry were ignored. Third, the model used only 13 technical indicators, so it did not capture any signals from other indicators, order-book imbalances, volatility surfaces, or news sentiment. Because these factors weren't included, the reported performance ratios are best-case estimates under ideal execution conditions and should not be taken as guarantees of live trading results.

Within those bounds, the neural network strategies outperform a buy-and-hold benchmark overall, and trading only when the inferred regime is favorable delivers even greater gains. In particular, the LSTM model provides the most reliable regime signals - higher recall, a smoother equity path, and tighter loss control - making it the strongest candidate for a production-grade minute-bar strategy. However, leveraged trading introduces larger drawdowns and higher risk, and real-world execution factors (e.g., latency, slippage, liquidity shifts) could erode these gains, so practical implementation must include robust safeguards and risk-governance measures.

8. Bibliography

Adrian, T., & Brunnermeier, M. K. (2016). CoVaR. American Economic Review, 106(7), 1705-1741. https://doi.org/10.1257/aer.20120555

Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. Econometrica, 71(2), 579-625. https://doi.org/10.1111/1468-0262.00418

Ankit, U. (2025). Transformer neural networks: A step-by-step breakdown. Built In. https://builtin.com/artificial-intelligence/transformer-neural-network

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157–166. https://doi.org/10.1109/72.279181

Blueberry Markets Academy. (2025). Candlestick patterns: Top candlestick charts every trader should know. https://blueberrymarkets.com/academy/candlestick-patterns-top-candlestick-charts-every-trader-should-know/

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307-327. https://doi.org/10.1016/0304-4076(86)90063-1

Box, G. E. P., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control (1st ed.). Holden–Day. Retrieved from https://books.google.com/books?id=rNt5CgAAQBAJ

Brock, W., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance, 47(5), 1731–1764. https://doi.org/10.1111/j.1540-6261.1992.tb04681.x

Brogaard, J., Hendershott, T., & Riordan, R. (2014). High-frequency trading and price discovery. Review of Financial Studies, 27(8), 2267-2306. https://doi.org/10.1093/rfs/hhu032

Brunnermeier, M. K., & Pedersen, L. H. (2009). Market liquidity and funding liquidity. Review of Financial Studies, 22(6), 2201-2238. https://doi.org/10.1093/rfs/hhn098

CQG, Inc. (2025). CQG trading platform. Retrieved May 31, 2025, from https://www.cqg.com/

De Bondt, W. F. M., & Thaler, R. (1985). Does the stock market overreact? The Journal of Finance, 40(3), 793–805. https://doi.org/10.1111/j.1540-6261.1985.tb05004.x

dida. (2025, May 28). What is an LSTM Neural Network? https://dida.do/what-is-an-lstm-neural-network

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. Econometrica, 50(4), 987-1007. https://doi.org/10.2307/1912773

Ersin, Ö. Ö., & Bildirici, M. (2022). Applying hybrid ARIMA-SGARCH in algorithmic investment strategies. Entropy, 24(2), 158. https://doi.org/10.3390/e24020158

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. Journal of Finance, 25(2), 383–417. https://doi.org/10.2307/2325486

Fama, E. F. (1991). Efficient capital markets: II. Journal of Finance, 46(5), 1575–1617. https://doi.org/10.1111/j.1540-6261.1991.tb04636.x

Granger, C. W. J., & Joyeux, R. (1980). An introduction to long-memory time series models and fractional differencing. Journal of Time Series Analysis, 1(1), 15–29. https://doi.org/10.1111/j.1467-9892.1980.tb00297.x

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. Review of Financial Studies, 33(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica, 57(2), 357–384. https://doi.org/10.2307/1912559

Hendershott, T., Jones, C. M., & Menkveld, A. J. (2011). Does algorithmic trading improve liquidity? Journal of Finance, 66(1), 1-33. https://doi.org/10.1111/j.1540-6261.2010.01624.x

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press. https://books.google.dk/books?id=5EgGaBkwvWcC

Hosking, J. R. M. (1981). Fractional differencing. Biometrika, 68(1), 165–176. https://doi.org/10.1093/biomet/68.1.165

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. Journal of Finance, 48(1), 65–91. https://doi.org/10.1111/j.1540-6261.1993.tb04702.x

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. Econometrica, 47(2), 263–291. https://doi.org/10.2307/1914185

Kirilenko, A. A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. Journal of Finance, 72(3), 967-998. https://doi.org/10.1111/jofi.12498

Kyle, A. S. (1985). Continuous auctions and insider trading. Econometrica, 53(6), 1315–1335.

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (1998). Efficient backprop. In G. B. Orr & K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade (Lecture Notes in Computer Science, Vol. 1524, pp. 9–50). Springer. https://doi.org/10.1007/3-540-49430-8_2

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementations. Journal of Financial

Economics, 58(1–2), 1–52. Retrieved from https://www.cis.upenn.edu/~mkearns/teaching/cis700/lo.pdf

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (Vol. 30, pp. 4765–4774). Curran Associates, Inc. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Mohammadi, H., & Su, L. (2010). International evidence on crude oil price dynamics: Applications of ARIMA–GARCH models. Energy Economics, 32(5), 1001–1008. https://doi.org/10.1016/j.eneco.2010.04.009

Odean, T. (1998). Volume, volatility, price, and profit when all traders are above average. Journal of Finance, 53(6), 1887–1934. https://doi.org/10.1111/0022-1082.00078

Ogasawara, E., Martinez, L. C., de Oliveira, D., Zimbrão, G., Pappa, G. L., & Mattoso, M. (2010). Adaptive normalization: A novel data normalization approach for non-stationary time series. In Proceedings of the International Joint Conference on Neural Networks (IJCNN). https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5596746

Park, C.-H., & Irwin, S. H. (2007). What Do We Know About the Profitability of Technical Analysis? Journal of Economic Surveys, 21(4), 786–826. https://onlinelibrary.wiley.com/doi/10.1111/j.1467-6419.2007.00519.x

Sharpe, W. F. (1994). The Sharpe ratio. The Journal of Portfolio Management, 21(1), 49–58. Retrieved from https://www.degruyterbrill.com/document/doi/10.1515/9781400829408-022/pdf?licenseType=restricted

Shefrin, H., & Statman, M. (1985). The disposition to sell winners too early and ride losers too long: Theory and evidence. Journal of Finance, 40(3), 777–790. https://doi.org/10.1111/j.1540-6261.1985.tb05002.x

Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. Quantitative Finance, 19(9), 1449–1459. https://doi.org/10.1080/14697688.2019.1622295

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. The European Physical Journal Special Topics, 228(10), 2313–2324. https://link.springer.com/article/10.1140/epjst/e2019-900046-x

Stübinger, J., & Endres, A. (2018). Pairs trading with a mean-reverting jump–diffusion model on high-frequency data. Quantitative Finance, 18(10), 1735–1751. https://www.tandfonline.com/doi/full/10.1080/14697688.2017.1417624?scroll=top&needAccess=true

Sullivan, R., Timmermann, A., & White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. Journal of Finance, 54(5), 1647–1691. Retrieved from https://econpapers.repec.org/article/blajfinan/v_3a54_3ay_3a1999_3ai_3a5_3ap_3a1647-1691.htm

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. Journal of Finance, 62(3), 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

Trarity. (2025). Trarity. Retrieved June 1, 2025, from https://trarity.com/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (Vol. 30). https://papers.nips.cc/paper/7181-attention-is-all-you-need

Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. Expert Systems with Applications, 208, Article 118128. https://doi.org/10.1016/j.eswa.2022.118128

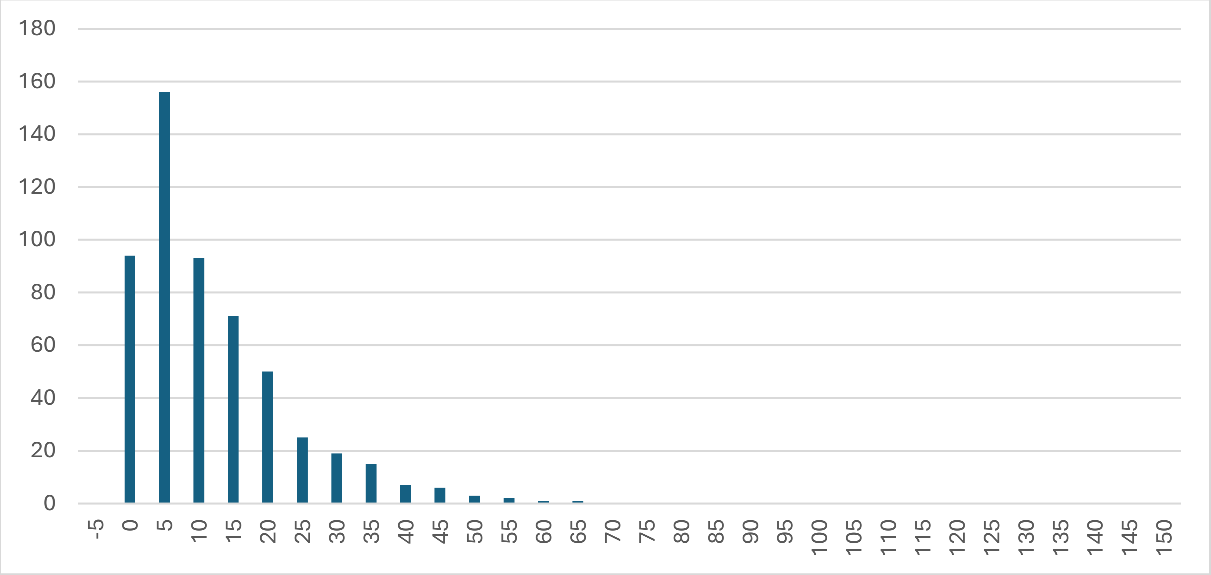
West, K. D., & Cho, D. (1995). The predictive ability of several models of exchange rate volatility. Journal of Econometrics, 69(2), 367–391. https://doi.org/10.1016/0304-4076(94)01654-I

Wilcoxon, F. (1945). Individual comparisons by ranking methods. Biometrics Bulletin, 1(6), 80–83. https://doi.org/10.2307/3001968

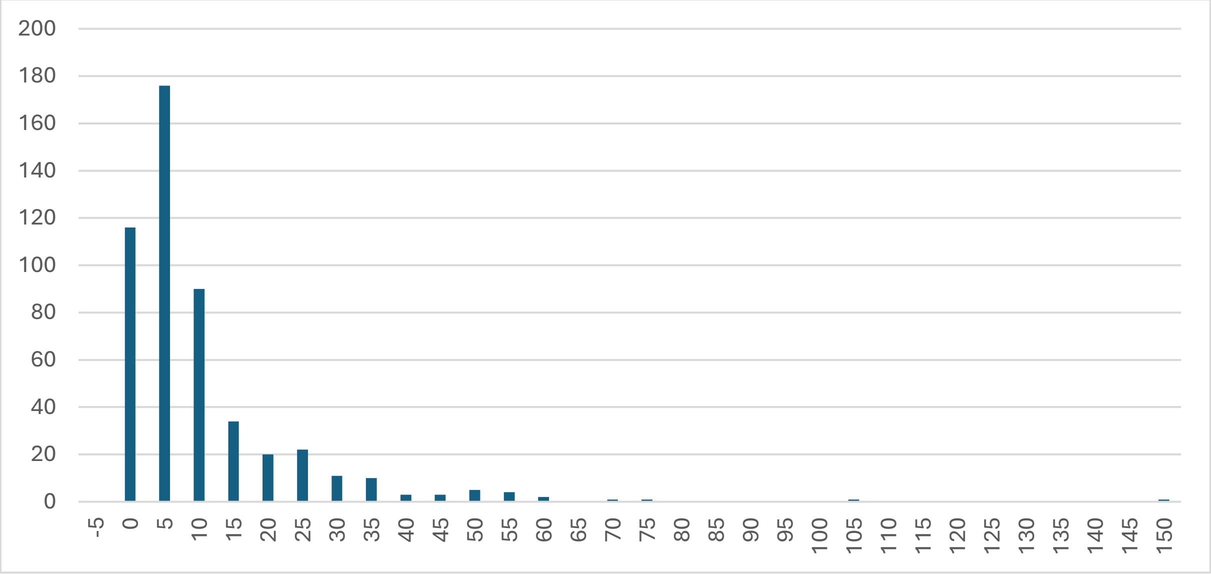
Zhang, Z., & Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. arXiv preprint arXiv:1805.07836. https://proceedings.neurips.cc/paper_files/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf

9. Appendix

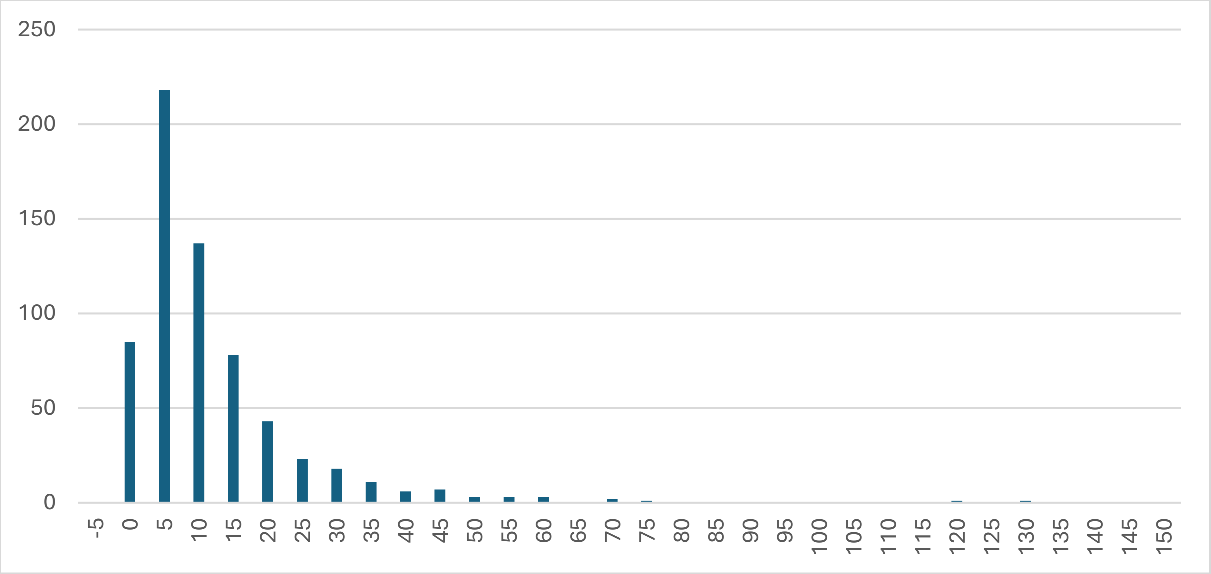
Appendix: 1 - LSTM - Trend Long Delay Distribution



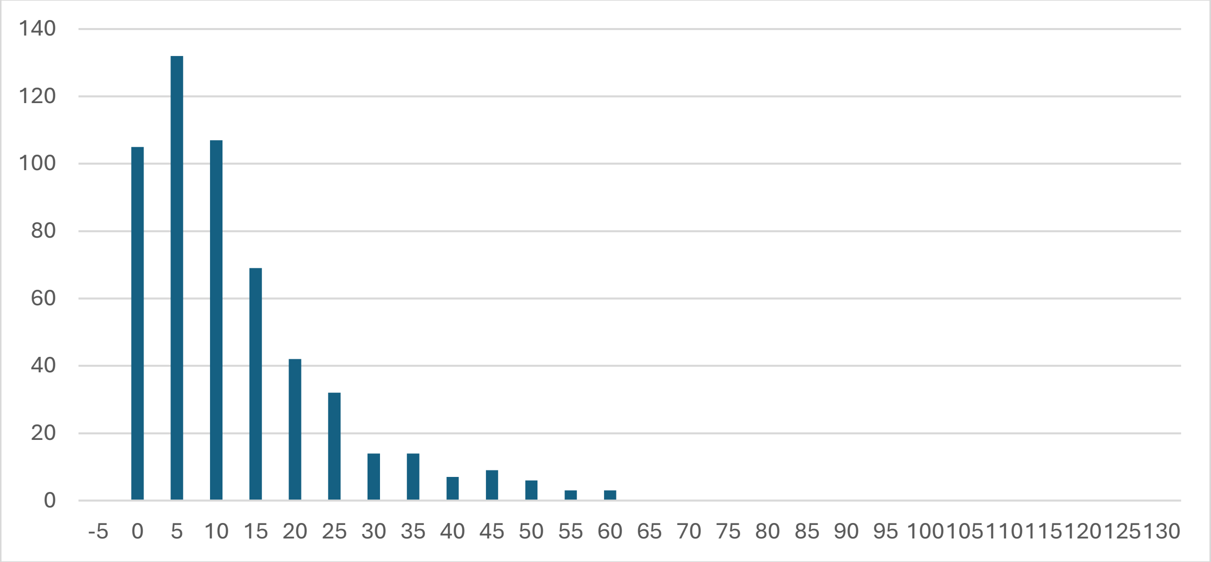
Appendix: 2 - LSTM - Trend Short Delay Distribution



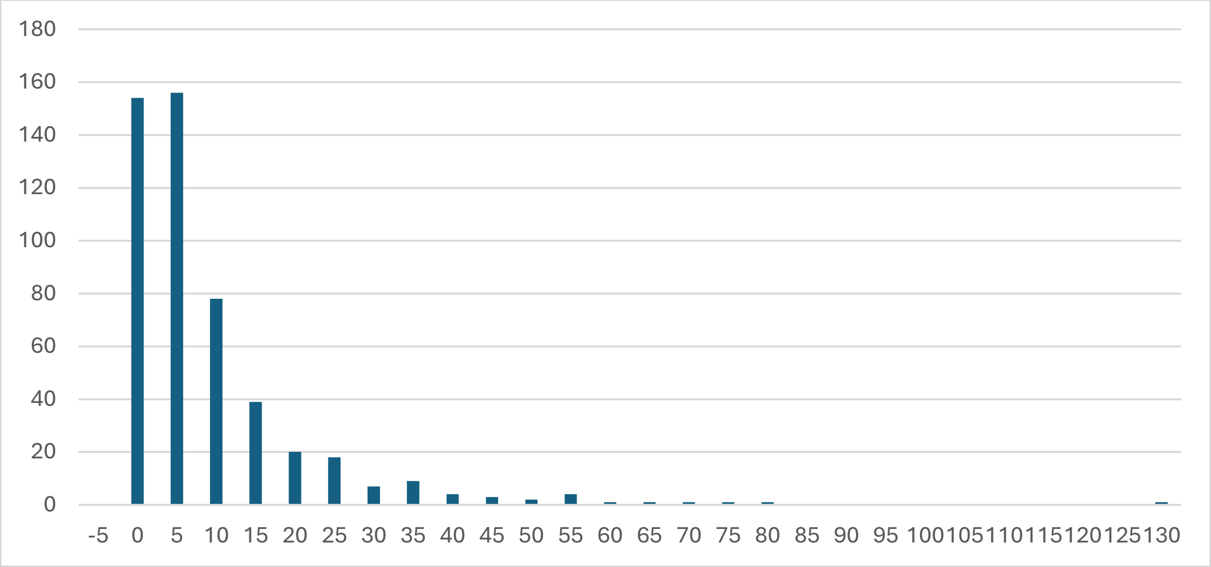
Appendix: 3 - LSTM - Range Delay Distribution



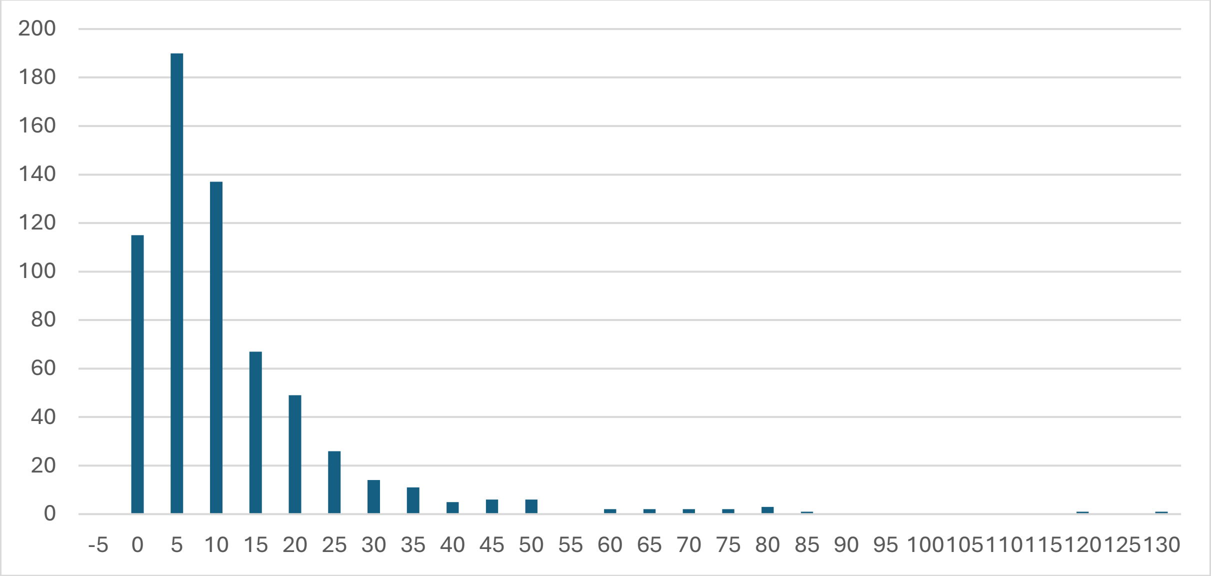
Appendix: 4 - Transformer - Trend Long Delay Distribution



Appendix: 5 - Transformer - Trend Short Delay Distribution



Appendix: 6 - Transformer - Range Delay Distribution



Appendix: 7 - LSTN vs Transformer Bull, Bear, Range Market Long-trades t-test

LSTM vs Transformer Long Trades

t-Test: Paired Two Sample for Means

	Bull Market	*Bull Market*
Mean	0,007691319	0,006444333
Variance	0,000879168	0,000929859
Observations	129	129
Pearson Correlation	0,930180885	
Hypothesized Mean Difference	0	
df	128	
t Stat	1,25693727	
P(T<=t) one-tail	0,105532576	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,211065151	
t Critical two-tail	1,97867085	

LSTM vs Transformer Long Trades

t-Test: Paired Two Sample for Means

	Bear Market	*Bear Market*
Mean	-0,004556861	-0,006081362
Variance	0,00147504	0,001159689
Observations	129	129
Pearson Correlation	0,948228521	
Hypothesized Mean Difference	0	
df	128	
t Stat	1,393639028	
P(T<=t) one-tail	0,082920828	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,165841656	
t Critical two-tail	1,97867085	

LSTM vs Transformer Long Trades

t-Test: Paired Two Sample for Means

	Range Market	*Range Market*
Mean	7,75268E-05	0,000188926
Variance	0,000247291	0,000244825
Observations	129	129
Pearson Correlation	0,878388595	
Hypothesized Mean Difference	0	
df	128	
t Stat	-0,16354406	
P(T<=t) one-tail	0,435174014	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,870348028	
t Critical two-tail	1,97867085	

Appendix: 8 - B&H vs LSTN Bull, Bear, Range Market Long-trades t-test

Buy & Hold vs LSTM Long Trades

t-Test: Paired Two Sample for Means

	Bull Market	*Bull Market*
Mean	0,003386843	0,007691319
Variance	5,52117E-05	0,000879168
Observations	129	129
Pearson Correlation	0,224895655	
Hypothesized Mean Difference	0	
df	128	
t Stat	-1,691602978	
P(T<=t) one-tail	0,046577465	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,09315493	
t Critical two-tail	1,97867085	

Buy & Hold vs LSTM Long Trades

t-Test: Paired Two Sample for Means

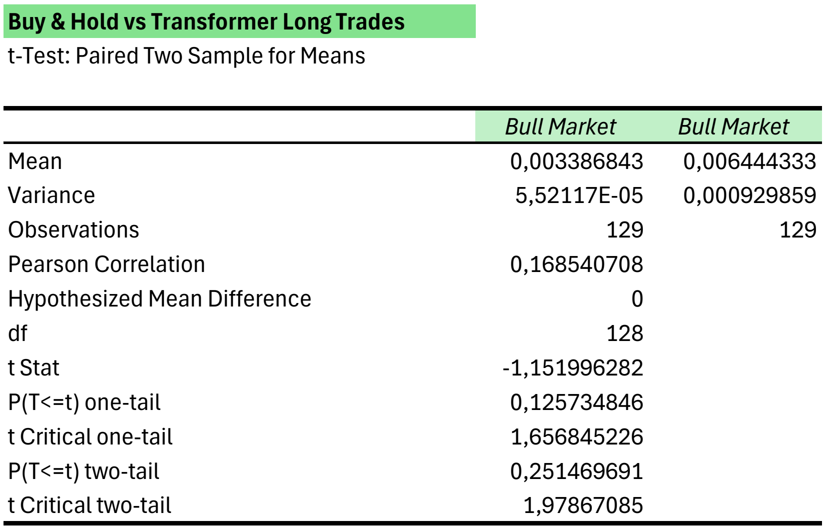
	Bear Market	*Bear Market*
Mean	-0,003003945	-0,004556861
Variance	3,25864E-05	0,00147504
Observations	129	129
Pearson Correlation	-0,029896971	
Hypothesized Mean Difference	0	
df	128	
t Stat	0,452289069	
P(T<=t) one-tail	0,325913163	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,651826326	
t Critical two-tail	1,97867085	

Buy & Hold vs LSTM Long Trades

t-Test: Paired Two Sample for Means

	Range Market	*Range Market*
Mean	-9,47178E-06	7,75268E-05
Variance	6,83604E-05	0,000247291
Observations	129	129
Pearson Correlation	0,032219512	
Hypothesized Mean Difference	0	
df	128	
t Stat	-0,05636964	
P(T<=t) one-tail	0,477567619	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,955135239	
t Critical two-tail	1,97867085	

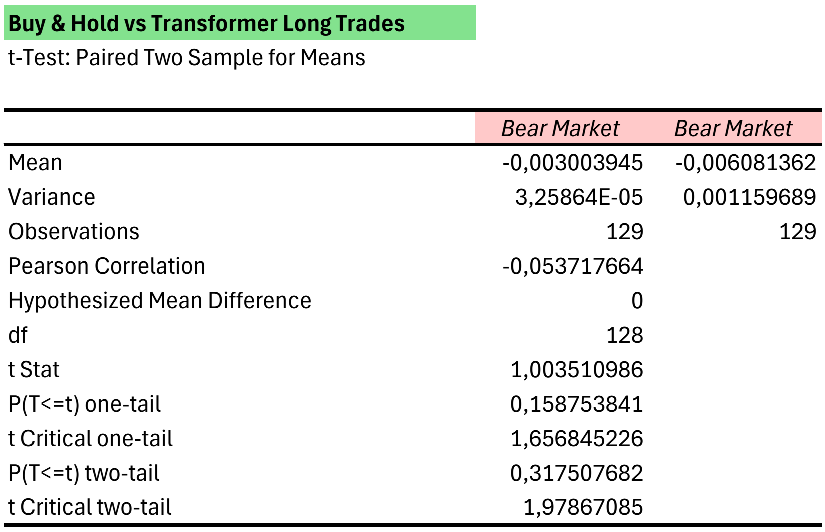
Appendix: 9 - B&H vs Transformer Bull, Bear, Range Market Long-trades t-test



Buy & Hold vs Transformer Long Trades

t-Test: Paired Two Sample for Means

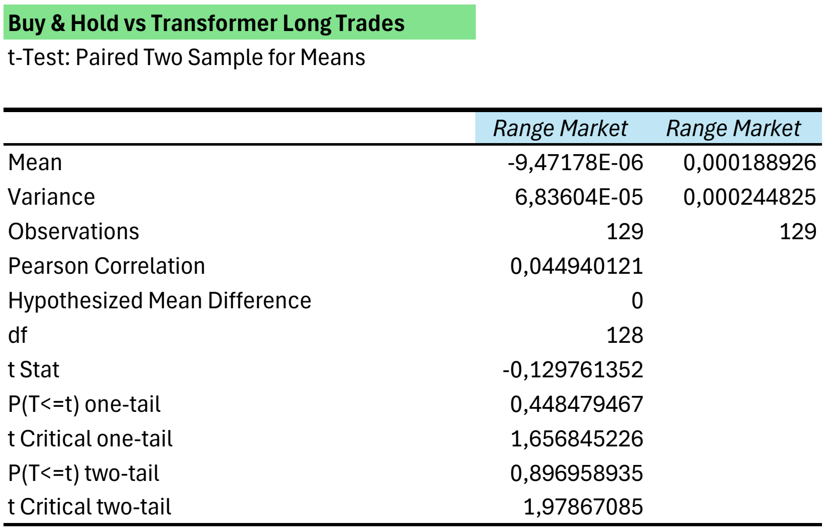
	Bull Market	*Bull Market*
Mean	0,003386843	0,006444333
Variance	5,52117E-05	0,000929859
Observations	129	129
Pearson Correlation	0,168540708	
Hypothesized Mean Difference	0	
df	128	
t Stat	-1,151996282	
P(T<=t) one-tail	0,125734846	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,251469691	
t Critical two-tail	1,97867085	



Buy & Hold vs Transformer Long Trades

t-Test: Paired Two Sample for Means

	Bear Market	*Bear Market*
Mean	-0,003003945	-0,006081362
Variance	3,25864E-05	0,001159689
Observations	129	129
Pearson Correlation	-0,053717664	
Hypothesized Mean Difference	0	
df	128	
t Stat	1,003510986	
P(T<=t) one-tail	0,158753841	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,317507682	
t Critical two-tail	1,97867085	



Buy & Hold vs Transformer Long Trades

t-Test: Paired Two Sample for Means

	Range Market	*Range Market*
Mean	-9,47178E-06	0,000188926
Variance	6,83604E-05	0,000244825
Observations	129	129
Pearson Correlation	0,044940121	
Hypothesized Mean Difference	0	
df	128	
t Stat	-0,129761352	
P(T<=t) one-tail	0,448479467	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,896958935	
t Critical two-tail	1,97867085	

Appendix: 10 - LSTN vs Transformer Bull, Bear, Range Market Short- trades t-test

LSTM vs Transformer Short Trades

t-Test: Paired Two Sample for Means

	Bull Market	*Bull Market*
Mean	-0,006067455	-0,009347188
Variance	0,000315486	0,000473516
Observations	129	129
Pearson Correlation	0,849954652	
Hypothesized Mean Difference	0	
df	128	
t Stat	3,242556671	
P(T<=t) one-tail	0,000755361	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,001510722	
t Critical two-tail	1,97867085	

LSTM vs Transformer Short Trades

t-Test: Paired Two Sample for Means

	Bear Market	*Bear Market*
Mean	0,008180045	0,011363975
Variance	0,002396847	0,003592788
Observations	129	129
Pearson Correlation	0,942217196	
Hypothesized Mean Difference	0	
df	128	
t Stat	-1,686563977	
P(T<=t) one-tail	0,047061528	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,094123056	
t Critical two-tail	1,97867085	

LSTM vs Transformer Short Trades

t-Test: Paired Two Sample for Means

	Range Market	*Range Market*
Mean	0,001526749	0,001983231
Variance	0,000456554	0,000708318
Observations	129	129
Pearson Correlation	0,923343807	
Hypothesized Mean Difference	0	
df	128	
t Stat	-0,484067183	
P(T<=t) one-tail	0,314582686	
t Critical one-tail	1,656845226	
P(T<=t) two-tail	0,629165371	
t Critical two-tail	1,97867085	