SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 2.1, (2026), pp. 699-716
Open Access. Online & Print

DOI: 10.5281/zenodo.122.12660

ARTIFICIAL INTELLIGENCE FOR UNCERTAINTY FORECASTING IN FINANCIAL MARKETS

Azzam Alroomi[1*]

Assistant Professor, Arab Open University, Kuwait, Email: aalroomi@aou.edu.kw

Received: 29/11/2025
Accepted: 09/12/2025

Corresponding Author: Azzam Alroomi (aalroomi@aou.edu.kw)

ABSTRACT

The paper reveals that while no single model consistently outperforms across different conditions, the integration of trading volume and uncertainty measures significantly enhances forecasting accuracy. These findings not only challenge traditional forecasting paradigms but also demonstrate the robust potential of combining AI with traditional statistical methods. The paper emphasizes the critical role of adaptability and innovation in algorithmic forecasting, offering substantial implications for both financial market theory and practice. Building on insights from market microstructure theory, i introduce volatility- and volume-based exogenous variables into neural network architectures to capture the interplay of risk, liquidity, and potential investor sentiment effects. Our exhaustive modeling suite includes ARIMA-type approaches, exponential smoothing, and hybrid AI–statistical ensembles. Forecast performance is assessed through established error metrics (ME, MAE, RMSE, MPE, MAPE, MASE) and further evaluated by a simple trading simulation to gauge economic significance. Results show that no single model dominates uniformly: while simpler statistical methods (e.g., Naïve, SES) match or surpass more complex AI on very short horizons, neural networks incorporating exogenous volume and high–low prices often outperform over weekly to quarterly periods. This suggests mild market inefficiencies or delayed information assimilation at intermediate frequencies particularly evident when volatility clustering or liquidity shifts are high. However, on very long horizons, forecast advantages narrow, aligning with the notion of semi-strong market efficiency.

KEYWORDS: Forecasting, Financial Markets, Predictive Analytics, Time Series, Neural Networks, Artificial Intelligence.

1. INTRODUCTION

This paper compares four main forecasting approaches using both statistical techniques and AI, testing their accuracy on FTSE 100 companies. Data collected includes opening share price, closing share price, high and low price, adjusted closing price, and trading volume, though the closing share price is the central variable in the tests. This paper investigates the longstanding debate on whether artificial intelligence (AI) methods or traditional statistical techniques offer superior predictive power in financial markets, particularly under varying market efficiency conditions. I employ an extensive dataset of 18 FTSE 100 companies spanning 20 years, testing 30 distinct forecasting models across multiple frequencies (daily, weekly, monthly, quarterly, yearly) and horizons (short- through long-term). In terms of economic relevance, selected AI-driven strategies yield notable improvements in risk-adjusted returns (5–10% annualized), underscoring the value of integrating microstructure-informed signals. These findings advance forecasting research by demonstrating the nuanced interactions between frequency, horizon, and exogenous risk factors, and offer practical guidance to traders seeking robust yet flexible predictive frameworks.

Four unique tests address horizon forecasting:

1. Test 1: Applies the 80/20 rule (80% training data, 20% testing).
2. Test 2: Uses the first 100 data points for training, then continuously adds one new data point before re-forecasting.
3. Test 3: Same as Test 2 but includes new High and Low prices inputted into the neural network.
4. Test 4: Same as Test 2 but introduces trading volume as an exogenous variable in the neural network.

Hybrid functions (combining two methods) were also examined to see if they improved forecasting accuracy. With horizon analysis, the paper explores how forecasting performance changes at different time spans, testing daily, weekly, monthly and yearly frequencies with multiple horizons. These horizons align with trading intervals, such as daily, weekly, monthly, quarterly and yearly cycles. Understanding how horizon lengths affect accuracy is crucial, as some methods perform better in the short term while others excel in the long term. Examining input variables in neural networks is similarly important appropriate selection and training can significantly enhance forecasting outputs. Furthermore, uncertainty and volatility, combined with varying data ranges, shape forecasting accuracy. GARCH-type models are often used to study volatility, while global economic policy uncertainties may also spill over locally. This understanding of volatility, uncertainty, and data ranges demonstrates that multiple factors affect forecasting accuracy.

The overarching aim was to determine which forecasting methods produced the most accurate predictions based on a range of error measures (ME, RMSE, MAE, MPE, MAPE, MASE) and different modeling approaches. A special focus was placed on several variants of neural network-based approaches (collectively referred to as "Nymphy"), including those incorporating exogenous variables such as the High and Low prices alongside the Close price. These are contrasted with more conventional statistical forecasting methods such as ARIMA variants, Naïve forecasts, Exponential Smoothing (SES and Holt-Winters), TBATS, BATS, and the THETAF method. In many cases, comparisons revolve around how well these methods perform across short and long horizons (ranging from 1-step-ahead up to 22-steps-ahead for daily data, and similarly up to 12-steps-ahead for weekly, monthly, and quarterly data). Closing prices, High and Low prices are perspective on the period of data, if the training data was yearly price movement the CP, HP and LP would be the assigned price for yearly period thus the closing price of the financial year, highest price for the financial year and low price of the financial year and that was implemented for all periods.

2. BACKGROUND, RELEVANT LITERATURE AND HYPOTHESIS DEVELOPMENT

Financial market forecasting has been a research focus for decades resulting in the development of many methodologies to study price movements and volatility and investor sentiment dynamics. Traditional statistical approaches consisting of ARIMA, GARCH together with exponential smoothing methods continue to dominate forecasting because of their straightforward nature and easy interpretation. The forecasting models work with static linear conditions despite falling short for accurately predicting non-linear patterns seen in financial data. Regression function is a basic analysis to find the relationship between target and inputs, and multiple linear regressions (MLR) are created as benchmarks for other models Li, et al, (2024)

Powerful artificial intelligent technologies with machine learning and especially neural networks have become popular alternatives to analyze complex nonlinear patterns in large financial datasets because of their ability to sign intelligent predictions

Ampountolas, (2023). According to Tealab et al. (2017) hybrid forecasting models demonstrate potential in resolving data debates and boosting their predictive identification across extensive periods of time.

The literature now incorporates additional explanatory variables as an advancement towards better forecasting outcomes. These trading variables jointly with high/low price movements and external economic indicators supply supplemental data which standard analytical models typically neglect. Joseph et al. (2011) established internet search intensity as an indicator of investor behavior which forecasted trading volume and abnormal returns while Wei et al. (2017) utilized GARCH-MIDAS models to predict volatility based on global policy uncertainty metrics. Shortages in datasets affect forecasting models results in increased uncertainty Khan, et al (2022)

Numerous gaps persist in being addressed within the existing literature. Systematic comparison of traditional methods to AI-based models consisting of neural networks does not exist for a comprehensive evaluation of multiple forecasting durations alongside horizons. Researchers have neglected to investigate the effects of incorporating exogenous variables such as price and trading data into synthetic intelligent forecasting models. Neural network applications in financial forecasting remain unclear because custom-built models lack explicit documentation of design approaches when researchers develop their own solutions instead of using commercially available packages.

Multiple gaps in the literature are addressed through systematic model comparisons using 30 forecasting techniques including ARIMA variants and neural network-based approaches with exponential smoothing methods across several time horizons and frequencies. This paper adds new knowledge to hybrid forecasting methodology literature when it incorporates volatility-based along with volume-based exogenous variables into neural networks(Yu, 2020). The research explores both practical and comparative aspects of artificial intelligence models along with classic approaches through an extensive evaluation of their real-world financial market forecasting accuracy. Subset models in AI such as machine learning and deep learning algorithms have great potential in improving forecasting Alroomi (2024).

2.1. AI Perspective & Implementation In Financial Markets

Trading in financial markets encompassing stocks, bonds, and currency exchanges has attracted considerable attention from scholars. Various models and algorithms, such as support vector regression (SVR), artificial neural networks (ANN), GARCH models, and hybrid approaches, have been employed to predict and improve trading performance.

- **SVR Models**

Sermpinis et al. (2015) proposed a hybrid Rolling Genetic Algorithm–Support Vector Regression (RG-SVR) for trading the EUR/USD, EUR/JPY, and EUR/GBP exchange rates, finding it outperformed other established models in both trading efficiency and statistical accuracy. Similarly, wSVR models (weighted SVR) tested by Sermpinis et al. (2017) showed better performance than traditional SVR models, illustrating that nonlinear, non-stationary financial markets often require more advanced variants of SVR.

- **GARCH Models & Pair Trading**

Chen et al. (2017) applied smooth transition GARCH models to a pair trading strategy in the U.S. stock market and achieved significant annualized returns. Pair trading is grounded in mean reversion, allowing investors to exploit pricing disparities between paired stocks.

- **Multivariate Adaptive Regression & Linear Regression Splines**

Kurek (2014) studied equity block trades on the Warsaw Stock Exchange using these techniques, finding that block trades signal important information to investors, resulting in positive or negative abnormal returns.

- **Neural Networks**

Neural networks have been used to forecast GDP growth rates Sokolov et al, (2016) emphasize that well-selected inputs, technical or fundamental can enhance the performance of neural network models, often outperforming simpler statistical methods.

- **Investor Sentiment**

Investor behavior, as measured by internet search intensity Joseph et al, (2011) or specific sentiment indices Li, et al, (2014), can forecast trading volume and abnormal stock returns. Online searches may signal interest from less sophisticated investors, influencing short-term trading patterns.

- **Trading Activities & Macroeconomic Forecasts**

Chatterjee (2016) and Erdogan et al. (2014) examined how stock market liquidity, volatility, and returns predict recessions. Lower liquidity often precedes recessions. Meanwhile, Arevalo et al. (2017) studied a filtered flag pattern strategy in the Dow Jones Industrial Average, finding dynamic technical trading rules can outperform simple buy & hold strategies.

- Financial Networks & Trading Performance

Booth et al. (2014) showed that global financial institutions, with larger networks, trade more efficiently due to better access to order flows, although local institutions learn and eventually reduce this gap over time.

Overall, research confirms that AI and statistics-based models, coupled with carefully selected input variables, can substantially improve trading forecasts in different market settings. Combining advanced techniques (e.g., SVR, GARCH, neural networks) with considerations of volatility, horizon length and investor sentiment often produces more robust results. Forecasting financial markets is a daunting yet essential endeavor, as investors, analysts, and policymakers seek to predict price trends, minimize risks and optimize returns. Artificial Intelligence undeniably has a significant impact on society across various domains. Given the sheer complexity and dynamism of the market, myriad methods have been proposed: from statistical techniques such as autoregressive models (AR, MA, ARMA, ARIMA, GARCH, and their variants) to more recent artificial intelligence (AI) methods including neural networks, fuzzy logic, support vector machines (SVMs), and deep learning architectures. Researchers invariably grapple with the question: "Is there a perfect model for forecasting financial markets? "Despite the laudable advances in forecasting techniques, it remains clear that financial markets have several unique characteristics. First, their time series data often exhibit no stationarities, heavy tailed distributions, jumps, and abrupt changes. Second, market complexities are magnified by the interplay of human psychology, as investor sentiment and policy uncertainties strongly influence price trends and volatility. Third, there is a fundamental tension between short-term horizon and long-term horizon forecasting. Consequently, continuous research and development efforts are imperative to enhance the precision and dependability of distinguishing human-generated content from artificially generated content. The short-term horizon approach (e.g., daily or intraday) is typically pursued by high-frequency traders, while long-term horizon models (e.g., quarterly or annual) matter greatly for macroeconomic policy or corporate decision-making. Finally, risk management through robust measurement of volatility and tail risks remains integral to the practical success of any trading or forecasting model. This paper endeavors to explore these overlapping themes. First, I consider the significance of risk, uncertainty, and trading practices in financial markets. Second, I review findings regarding neural networks and statistical benchmarks for predicting market indices. Third, I address the growing trend of big data analytics and deep neural networks, offering insights into how new algorithms tackle nonlinearity. I then answer the question of whether forecasting necessarily optimizes returns. Finally, the discussion covers horizon testing, accurate metrics, and the ways in which uncertainty influences predictive performance. In weaving these threads together, I aim to provide a thorough overview of the current state of knowledge, while also highlighting the challenges inherent in financial market forecasting Gajamannage, et al (2023). Thus, I need to do much more on understanding how to produce forecasts capturing uncertainty Alroomi, et, al (2022)

2.2. Risk And Trading In Financial Markets

Understanding risk is pivotal in trading. Risk in financial markets arises from uncertain price fluctuations, abrupt regime shifts, volatility clusters, and myriad exogenous shocks. This section explores how scholars have approached the concept of risk, the metrics used to measure it, and the trading strategies that either mitigate or exploit it.

According to Pham, et al, (2014), stock assessment and risk management form two core strategies used by practitioners to guide trading decisions. Financial markets, especially equities, are subject to rapidly changing dynamics, including uptrends, downtrends, or sideways moves. Integrating an explicit risk management plan into a trading system significantly enhances the probability of achieving above-average returns while limiting possible downside. Pham et al. (2014) developed a novel stock trading system by integrating Kansei evaluation methodology originally used to assess affective responses in design with a self-organizing map model. Their approach was tested on daily stock data from exchanges in the U.S. (NYSE and NASDAQ) as well as in Vietnam, with encouraging results in terms of reduced losses and improved risk-adjusted returns.

In a similar line of research, Vella, et al (2016) explored the possibility of mitigating risk and handling uncertainty in high-frequency trading contexts. They emphasized that market microstructure noise at very high trading frequencies aggravates the uncertainty embedded in price and volatility dynamics. The researchers proposed an interval type-2 model based on a generalization of a type-1 ANFIS (Adaptive Neuro-Fuzzy Inference System). Known as ANFIS/T2, this model not only improves risk-adjusted performance but also contains computational complexities. Their results confirmed that more sophisticated fuzzy and neuro-fuzzy systems can provide valuable tools for coping with market risk, ultimately aiding regulators, practitioners, and researchers in designing risk management protocols.

Another angle in risk management involves identifying time windows or market segments wherein risk is more pronounced. Riedel, et al (2015) investigated tail risk, especially lower tail downside risk, by employing GARCH models for returns of stock markets in the United States, Japan, Germany, and France. Their surprising finding was that overnight return innovations displayed a significant tail risk while intraday innovations did not.

In contrast to these studies focusing on exogenous and microstructure factors, Shoji, et al (2016) delved into the realm of behavioral finance. Relying on prospect theory, they used numerical simulations to demonstrate that "risk-seeking in losses" is a key driver in generating the disposition effect (the observed tendency of investors to hold onto losing positions too long and sell winning ones too soon). Indeed, Barberis, et al (2009) had stressed that the value function in prospect theory makes investors risk-averse in gains while being risk-seeking in losses. Thus, risk extends beyond purely quantitative definitions, intertwining with investor psychology to shape actual market outcomes.

2.3. Nonlinearity Versus Linearity And The Emergence Of Hybrid Approaches

Nonlinearity remains a central theme: many market time series exhibit structural breaks, abrupt shifts, cyclicality, and strong interactions across multiple time scales. Traditional linear ARIMA or ARMA models struggle to capture dynamic behaviors unless augmented with regime-switching or threshold components. In this sense, neural networks, SVMs, or GARCH variants frequently provide better fits.

Tealab, et al (2017) classify time series by their linearity behavior, maintaining that linear time series forecasting might be sufficient for well-behaved data sets, but real-world financial markets rarely remain stable or linear. Tellingly, the authors note that "common neural networks" often are not sufficient for dynamic behavior with moving average terms. Deep learning or hybrid combinations of fuzzy logic, wavelets, or evolutionary optimization might be necessary.

Deep convolutional networks have had particular success in pattern recognition tasks such as handwriting verification or image-based algorithmic trading signals Hafemann, et al (2017). Support vector machines (SVMs) also remain popular, especially for classification tasks or stock direction-of-change predictions Tay, et al (2001).

Studies show that shallow MLPs or feedforward neural networks occasionally fail to find global minima Kuremoto, et al (2014). Deep belief networks comprising stacks of restricted Boltzmann machines can mitigate some local optima issues. Moreover, big data analytics provide new frontiers for deep learning in tasks such as semantic indexing of large text corpora, unstructured data mining, or anomaly detection in real-time trading. Najafabadi, et al (2015) stressed that unsupervised deep learning methods can parse massive volumes of unlabeled data, revealing hidden correlations or latent features.

Nonetheless, as pointed out by Tealab et al. (2017), any forecasting success with deep networks is dependent on how well the architecture is suited to the problem's time scale, the stationarity or nonstationary of data, and the richness of the available features. For certain tasks especially short-horizon forecasting a simpler ARIMAmight outperform large-scale deep networks if data are limited or the series is stable. By contrast, in contexts of big data, complex patterns, and high noise, deeper architectures often shine.

2.4. Inking Forecast Accuracy And Trading Returns

An enduring question in finance is: even if a model has strong forecasting accuracy, does that necessarily translate into higher trading profitability? To address this, many scholars have designed trading simulations with transaction costs, slippage, and real-time constraints.

Choudhry, et al (2012) utilized artificial neural networks to predict foreign exchange rates (USD-EUR, DEM-USD, JPY-USD) at high-frequency. They discovered that an active trading strategy using the ANN's buy/sell signals was profitable net of transaction costs. This implies that certain microstructure variables (like bid-ask spreads and last trade price) can be harnessed for short-horizon profitability, a reflection of the real-time informational advantage of the model.

On the equity side, Bekiros, et al (2008) studied recurrent neural networks to forecast direction-of-change in the NASDAQ index. They integrated measures of volatility changes into the RNN-based trading rule, finding that it not only outperformed a standard buy-and-hold approach but also remained profitable after accounting for transaction costs.

RNNs, in capturing short-term memory and dynamic patterns, seemed adept at timing the market.

Dunis, et al (2002) aimed to forecast volatility for GBP/USD and USD/JPY exchange rates. They then constructed a trading strategy around options straddles, capitalizing on mispriced implied volatility. Similarly, they reported positive net returns from the nonparametric, RNN-based approach. Meanwhile, Sokolov, et al (2016) observed that ANN-based strategies for macroeconomic forecasting (e.g., using trade parameters) can yield improved portfolio decisions in practice.

Other research explored SVM-based trading rules. Dunis, et al, (2013) examined whether SVM predictions of weekly change in the Spanish IBEX-35 outperformed MLP or buy-and-hold strategies, concluding that SVM accuracy was high but sensitive to the length of training data. Qu, et al (2016) tackled high-frequency returns of the Chinese CSI 300 index with support vector regression. By proposing a novel kernel function to capture cyclicality and decaying influences of past returns, they achieved superior directional accuracy and higher capital gains.

These studies illustrate that while accuracy alone does not guarantee profits, many advanced AI models particularly neural networks and SVMs demonstrate real-world potential when carefully adapted. Yet, each must be tested rigorously under robust simulation conditions that incorporate realistic costs and constraints.

2.5. Methodological Considerations For Testing Multiple Models

In a practical sense, designing an empirical study to identify a "best model" typically involves the following steps:

1. Data Collection and Frequency Setting

Researchers decide which markets (e.g., equities, currencies, commodities) and which instruments or indices (e.g., S&P 500, DJIA, IBEX-35, CSI 300) to include. They also choose frequency daily, weekly, monthly, or intraday (15-minute, 30-minute bars) in alignment with the horizons they wish to forecast.

2. Multiple Methods Selection

As the overarching research question suggests, it is vital to test both widely used statistical models (ARIMA, GARCH variants, logistic regressions) and popular AI methods (MLP, RNN, SVM, deep learning). Then, additional hybrid or ensemble approaches might be added, e.g., GARCH-MIDAS with exogenous variables or ANFIS with fuzzy logic.

3. Exogenous Regressors and External Inputs

Including fundamental data (e.g., interest rates, GDP, inflation), technical indicators (RSI, MACD, moving averages), or sentiment measures (investor sentiment, web search intensity, news analytics) can significantly affect model performance. The success of these inputs depends on their correlation with future returns or volatility.

4. Implementation and Training

Each model is trained on historical data (in-sample). Hyperparameters for neural networks might be tuned using cross-validation or grid/random search. The final model is then tested on out-of-sample data to assess generalization.

5. Accuracy and Profitability Checks

Common error metrics (RMSE, MAE, MAPE, Theil's U,) are calculated. If a trading strategy is tested, performance metrics might include net returns, Sharpe ratios, maximum drawdown, and other risk-adjusted performance measures.

6. Comparison and Robustness Analysis

The next step is to check whether one model outperforms the others consistently or only in certain market regimes or time windows. Statistical significance tests like Diebold–Mariano or Model Confidence Set add rigor, while scenario analyses (e.g., sub-periods, crisis vs. non-crisis years) gauge robustness.

In practice, the variety of possible combinations (algorithm × horizon × frequency × input set) often leads to an enormous search space. It is not uncommon for academic studies to limit their scope due to computational or data constraints, which helps explain why no single "perfect model" emerges.

The literature suggests that while certain models consistently outperform others in specific scenarios, no single approach reigns supreme across all conditions. Neural networks, especially deep learning methods, have demonstrated capacity for capturing highly nonlinear patterns and gleaning hidden features from big data. However, they can be susceptible to overfitting, require extensive computational resources, and may be difficult to interpret. Statistical techniques such as ARIMA or GARCH are simpler and remain powerful when the data exhibits certain stationarity or structure, but they often struggle with abrupt regime changes.

Risk management remains paramount. Models that track or predict tail risk, volatility jumps, or macroeconomic instability can be especially valuable. The synergy between advanced volatility modeling (GARCH-MIDAS) and external explanatory variables (like policy uncertainty indices) has yielded consistent improvements, particularly in daily or monthly volatility forecasts.

Research on forecast horizons underscores that predictive accuracy for longer horizons frequently declines, but not always. In some contexts, horizon-specific calibration or combining forecast horizons can preserve accuracy. The practical question for financial market participants is whether short-term or long-term predictions better align with their investment strategies or policy objectives.

Finally, the question "Is there a perfect model for forecasting financial markets?" persists. Given that markets evolve, conditions change, and investor psychology can shift unpredictably, it seems improbable that any single approach is universally optimal. Nonetheless, the ever-growing sophistication of AI, big data analytics, and ensemble methods continues to push the boundary, making forecasting more robust, and perhaps in the future, bridging the gap between theoretical accuracy and real-world profitability.

A broad spectrum of models were tested:

NNET_THETAF (Neural Networks with Theta Model)

$$(y_t)=f(WX+b)+\theta\cdot y_t+(1-\theta)\cdot T_t,(1)$$

Where:

$\hat{(y_t)}$ predicted value at time t.

f Neural network function capturing nonlinear relationships.

W Weight matrix for the neural network.

X Input features vector.

b Bias term in the neural network.

θ Parameter balancing actual and trend components.

y_t Actual observed value at time t.

T_t Trend component from the Theta model.

· Multiplication operator.

(1-θ) Complementary fraction to θ.

This equation merges a neural network's NNET (Neural Networks for Time Series) $y_t=f(WX+b)$ capacity to capture nonlinear patterns with the Theta model's (THETAF (Theta Model)$(\hat{y_t})=\theta\cdot y_t+(1-\theta)\cdot T_t$ robust trend estimation, enhancing forecast accuracy for time series data. By incorporating both the raw series y_t and its trend component T_t. The model adeptly balances short-term fluctuations with long-term growth or decline. The parameter θ dictates the weight of each component, enabling flexibility across various market or environmental scenarios. Widely adopted in financial analytics, demand forecasting, and climate modeling, it leverages deep learning pattern extraction alongside the Theta method's proven reliability, delivering improved prediction performance and interpretability across numerous real-world applications in diverse industries.

NYMPHY_EXOGONOUS_CLOSE (Neural Network with Exogenous Variables)

$$(\hat{y_t})=f(WX+b+Z_t^{exogenous}) \quad (2)$$

Where:

$\hat{(y_t)}$ predicted value at time t.

f Neural network function capturing nonlinear relationships.

W Weight matrix for the neural network.

X Input features vector.

b Bias term.

Z_t Exogenous variable.

This model integrates exogenous variables Z_t into a neural network, capturing external factors that traditional endogenous-only approaches overlook. By including an Exogenous variable, the network identifies correlations that enrich prediction accuracy. Weight parameters W and bias b optimize how these inputs interact with the primary features X. This approach is particularly valuable in financial forecasting, where influences like 10% increase on decrease in closing price changed significantly affect market trends

NYMPHY_CLOSE_HIGH (Neural Network with Close and High Prices)

$$(\hat{y_t})=f(WX+b+Z_t^{\text{"close"}}+Z_t^{\text{"high"}}) \quad (3)$$

Where:

$\hat{(y_t)}$ predicted value at time t.

f Neural network function capturing nonlinear relationships.

W Weight matrix for the neural network.

X Input features vector.

b Bias term.

$Z_t^{\text{"close"}}$ Closing price input.

$Z_t^{\text{"high"}}$ High price input.

By incorporating both the closing and high prices as inputs, this neural network model captures a broader range of market dynamics. The weights W and bias b adapt to how these features combine with the core input X, enabling the function f to detect patterns associated with price volatility and momentum shifts. In financial forecasting, high prices often signal peaks or intraday sentiment, while closing prices reflect consolidated market conditions. This dual-price approach empowers traders, risk analysts, and automated systems with richer insights for better decision-making. Its predictive power significantly extends to algorithmic trading, asset valuation, and proactive risk mitigation.

NYMPHY_CLOSE_LOW (Neural Network with Close and Low Prices)

$$\hat{y}_t = f(WX+b+Z_t^{\text{close}}+Z_t^{\text{low}}) \quad (4)$$

Where:

$\hat{y}_t$ predicted value at time t.

f Neural network function capturing nonlinear relationships.

W Weight matrix for the neural network.

X Input features vector.

b Bias term.

Z_t^{close} Closing price input.

Z_t^{low} High price input.

This neural network architecture incorporates closing and low-price data, offering enhanced sensitivity to market fluctuations and risk signals. Low prices can reveal intraday troughs or support levels, while closing prices summarize the day's final sentiment. By combining these inputs with the network's weight matrix W and bias b, the function f can more accurately predict future values, especially under volatile conditions. Common in financial risk management and algorithmic trading, this approach identifies significant downside trends and highlights potential entry points.

NYMPHY_CLOSE_HIGH_LOW (Neural Network with Close, High, and Low Prices)

Equation:

$$\hat{y}_t = f(WX+b+Z_t^{\text{close}}+Z_t^{\text{high}}+Z_t^{\text{low}}) \quad (5)$$

Where:

$\hat{y}_t$ predicted value at time t.

f Neural network function capturing nonlinear relationships.

W Weight matrix for the neural network.

X Input features vector.

b Bias term.

Z_t^{close} Closing price input.

Z_t^{high} High price input.

Z_t^{low} High price input.

This comprehensive neural network model enriches time series forecasting by integrating closing, high, and low prices. It reveals a detailed picture of price movements, capturing both maximum peaks and potential support levels alongside final daily sentiments. These inputs feed into the weighted function f, allowing for nuanced detection of emerging trends, market volatility, and sentiment shifts. In financial contexts, such multi-price integration boosts the model's responsiveness to intraday volatility and end-of-day market stance.

3. Data, methodology and output

The data for this paper was collected from 18 companies listed on the FTSE 100 index. The selection criteria prioritized the most capitalized companies within the index, ensuring that the dataset represented the largest and most influential firms. To maintain consistency and comparability across the dataset, the unit measure was standardized across all companies, rather than aligning solely with the dates of share prices. Consequently, the dataset comprised daily (4000 units), weekly (825 units), monthly (191 units), quarterly (65 units), and yearly (17 units) observations, covering the period from 2000 to 2016. The data collected included the high, low, close, adjusted close, volume, and % of change for the selected companies' share prices.

Here, I measure the forecasting accuracy using two different approaches according to the forecasting period. For monthly data I forecast 18 months ahead using all the methods used. After the forecast, I test our six different error matrices for each method and each frequency. After the error has been developed, I analyze the matrices according to a set of ranking points. Here, fewer points are awarded for fewer errors, thus the method with the least points is the more accurate.

The second approach tests the accuracy between methods and horizons. The forecasting here uses different training data to that used in the first approach where I used all data points as training data from the beginning. However, in the second approach I used progressive continuous addition of training data. In the first approach, I used the first 100 data points to forecast 18 horizons; the next data point and training data was re-computed to produce a sequence of 18 horizons. This procedure applies to all of the remaining datasets.

The methodology tested in this paper employs a quantitative approach to compare a diverse range of forecasting methods, spanning both statistical techniques and artificial intelligence (AI). A total of 30 methods were tested, comprising 23 statistical techniques and 7 AI models. The remaining two methods were excluded from the analysis due to their inability to produce outputs or adapt to the time-series data. The adaptability of certain models was contingent on the presence of seasonality in the data. In cases where the time series lacked seasonality, these models failed to compute forecasts and yielded no results. All statistical methods were implemented using R software, while Python was employed for the AI method. The tested methods represent widely used techniques in the forecasting field. However, their performance often depends on the characteristics of raw data. As noted, models reliant on seasonality may perform poorly or result in non-applicable (NA) outputs when such patterns are absent in the dataset. This highlights the importance of aligning the chosen methodology with the underlying properties of the data to ensure reliable and meaningful results. Complete optimization is impossible when it comes to forecasting different academics debate what to optimize and how to implement their models. However, there are more widespread theories out there which many scholars do use and agree on. In our model under error testing, the more commonly used errors matrices: Mean Error (ME)

$$ME=\frac{1}{N}\sum_{i=1}^{N}(f_i-k_i) \tag{6}$$

Where:

N = # observations

f_i = represents the predicted value for the i^{th}observation

k_i = represents the actual value for the i^{th} observation

(f_i-k_i) = represents the error for the i^{th} observation

Mean Absolute Percentage Error (MAPE)

$$MAPE=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{|f_i|-|k_i|}{|k_i|}\right) \times 100 \tag{7}$$

Where:

$\left(\frac{|f_i|-|k_i|}{|k_i|}\right)$= represents the absolute error for the i^{th} observation

Mean Percentage Error (MPE)

$$MPE=\frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i-k_i}{k_i}\right) \times 100 \tag{8}$$

Where:

$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{|f_i|-|k_i|}{|k_i|}\right) \times 100$ = with the exclusion of absolute value

Mean Squared Error (MSE)

$$MSE=\frac{1}{N}\sum_{i=1}^{N}(f_i-k_i)^2 \tag{9}$$

$(f_i-k_i)^2$ = represents the squared for the i^{th} observation

Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i-k_i)^2} \tag{10}$$

Where:

The square root of $\frac{1}{N}\sum_{i=1}^{N}(f_i-k_i)^2$ is calculated

Mean Absolute Error (MAE)

$$\frac{1}{N}\sum_{i=1}^{N}\left(|f_i|-|k_i|\right) \tag{11}$$

Mean Absolute Scaled Error (MASE)

$$q_t=\frac{e_t}{\frac{1}{n-1}\sum_{i=1}^{n}|Y_i-Y_{i-1}|} \tag{12}$$

Where:

$e_t = f_i-k_i$ which indicates the error at time t

Y_i = represents the actual value for the i^{th} observation

Y_{i-1} = represents the actual value -1 indicating previous for the i^{th} observation

Log returns were also determined by the model, with the same capacity as the error matrices calculation. Furthermore, as six different error matrices were tested, the scale of the test itself increases and develops a more determined and detailed conclusion as I can conclude which function is more accurate under the different circumstances mentioned. The results that are presented are a mere fraction of the results that were produced. The total output will be freely available upon request. Due to the size of the total output, I were unable to showcase all the results. Furthermore, the vast output that was computed presented us with the dilemma of how to be fair and consistent when presenting the results from our horizon testing. Therefore, in a commanding rule, the paper presents the absolute percentage error (APE) & (MSE) accuracy tests.

Table 1: Title caption.

Methods	Description
1. NYMPHY_EXOGONOUS_CLOSE	Neural Network with exogenous variables
2. NYMPHY_CLOSE_HIGH	Neural Network with Close and High share price
3. NYMPHY_CLOSE_LOW	Neural Network with Close and Low share price
4. NYMPHY_CLOSE_HIGH_LOW	Neural Network with Close, High and Low share price

The tested models here are the neural networks testing uncertainty. Nymphy was the function tested in this paper. The algorithm was constructed to test whether the volatility of the share price could predict the future for that share price. The table below presents the results from our test for each method. The methods were tested under the six error matrices above. This shows us which method performed better under which error. A detailed table on the classification of accuracy testing shows an overview of how the methods are performed. The table below reveals strength in the random walk forecast (RWF) and the mean in the mean error (ME) test, but this strength diminishes along with the error matrices. Specifically, the mean became the least accurate test in all the other error testing apart from the mentioned ME test. The naïve model showed better accuracy than the other models in three of the six accuracy tests, performing better in MAE, MAPE and MASE. Neural networks and the Holt-Winters performed better in RMSE and MPE, respectively. The Nnet function would take advantage of the RMSE high values and variance thus here being uncertainty and thus meaning nonlinearity of the test would eventually perform better.

Table: 2: Title caption.

Models	*Weekly*		*Values*				
	ME	RMSE	MAE	MPE	MAPE	MASE	AVE
AUTO	0.577	85.463	57.664	-0.407	5.961	0.963	25.037
SES	0.577	85.463	57.664	-0.407	5.961	0.963	26.547
HOLT	0.267	86.895	58.387	***-0.313***	6.047	0.983	25.378
BATS & TBATS	-0.308	86.332	58.239	-0.750	5.952	0.980	25.074
DSHW	1.063	95.586	64.763	1.092	7.916	1.091	28.585
NAIVE & SNAIVE	5.738	88.257	59.434	-0.317	6.042	1.000	26.692
NNET	-0.051	***82.118***	***56.562***	-0.739	***5.714***	***0.939***	***24.091***
SPLINE	898.464	1252.456	1073.246	73.509	80.697	NA	NA
THETA F	5.880	87.759	58.980	-0.335	6.004	0.992	26.547
RWF	***0.000***	87.679	58.843	-0.322	6.074	0.992	25.544
MEAN	0.000	452.212	395.048	-57.127	79.138	6.289	145.927
AUTO % SES	188.543	530.967	438.400	15.892	31.776	6.290	201.978
AUTO % TBATS	94.809	449.495	364.327	12.970	29.857	5.713	159.528
AUTO % NNET	193.753	549.685	459.314	15.377	32.748	6.507	209.564
SES & THETA F	209.285	564.813	470.709	19.446	34.611	6.290	217.526
SES % MEAN	493.431	668.433	599.344	-11.466	57.966	8.685	302.732
TBATS & THETA	115.550	466.547	381.223	16.524	32.233	6.313	169.732
NNET & THETA	188.543	530.967	438.400	15.892	31.776	6.290	201.978

In our weekly tests, the neural network model performed significantly better than other models in four out of the six error matrices, beating the naïve model in the weekly test where the random walk model had performed better in one out of the six error matrices. The neural network performed better in the following errors: RMSE, MAE, MAPE, and MASE. The high performance in four errors shows that there is more nonlinearity in the weekly test where Nnet recorded better performance under the mentioned conditions. The lowest performing model was produced by the spline model, which had the least accurate results in four out of the six error measurements. RMSE, MAE, MPE, and MAPE were the measurements where the spline model showed its weakness.

Table 3: Analysis & Evaluation.

	Classifications of Accuracy Testing					
	ME	RMSE	MAE	MPE	MAPE	MASE
Daily 1st	!RWF&MEAN	NNET	NAÏVE	HOLT	NAÏVE	NAÏVE
Daily LE	SES%MEAN^	MEAN	MEAN	MEAN	MEAN	MEAN
Weekly 1st	!RWF/MEAN	NNET	NNET	HOLT	NNET	NNET
Weekly LE	SES%MEAN^	SPLINE	SPLINE	SPLINE	SPLINE	SES/MEAN^

* (LE) here meaning least accurate model. ^ Here represents a hybrid model combining two functions. & here meaning a joint accurate best accurate model. So, both methods have the same result of error. ! Net was very close behind with -0.0043.

As seen from the tables above, both neural networks and the naïve methods have performed very well in the test run. As the RWD enforced its will upon other models, Nnet performed better in the weekly frequency and almost did so in the daily frequency. If an average of the errors was taken for each method, neural networks performed better in both daily and weekly frequencies. This also takes into consideration that the neural network test was implemented with a single input and no alternative inputs were chosen for the network to learn, work, and develop.

Table: 4:

	Classifications of Accuracy Test					
	Daily	Weekly	Monthly	Quarterly	Yearly	Total
ME	48.78	4.12	1.199	0.221	0.018	54.338
MAE	48.78	4.12	1.199	0.221	0.018	54.338
MAPE	48.78	4.12	1.199	0.221	0.018	54.338
MPE	48.78	4.12	1.199	0.221	0.018	54.338
MSE	48.78	4.12	1.199	0.221	0.018	54.338
RMSE	48.78	4.12	1.199	0.221	0.018	54.338
TOTAL	292.68	24.72	7.194	1.326	0.108	326.028

The table above shows the errors that were produced, calculated and tested. Due to the significant number of errors that were tested, the implemented test was carried out on the HPC Wales supercomputer (cloud). The implementation of the test in this paper would have taken over a year on a normal everyday computer. At first, the implementation was carried out on a Dell XPS I7 laptop but after a few weeks of running the test was stopped. At this point, the notion of using a supercomputer arose, while there were also some clouds and clusters to consider. Finally, a decision was taken to implement the test on the HPC Wales supercomputer cloud. The test took 44.5 hours on 45 cores. This was due to the high computation being run. At the start of our code being run on the HPC cloud was going to result in a significant difference time wise to run the code. However, after introducing parameters to the test and implementing dynamic arrays, the time was reduced. However, there was more effort taken to reduce the time that it would take to reduce the time the test was taken, in effort to do that, the methods were allocated to individual cores and the more computationally intensive functions were allocated more cores. The forecasts produced according to the actuals of the company's stock price are publicly available data.

Table: 5:

DAILY	APE								
METHOD					**HORIZON**				
	1	2	3	4	5	10	22	1-10	1-22
AUTO_ARIMA	0.014	0.020	0.024	0.028	0.031	0.045	0.067	0.031	0.045
AUTOARIMA_FOURIER	**0.013** *0.013*	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
AUTOARIMA_NNET	1.106	1.105	1.103	1.103	1.103	1.102	1.102	1.103	1.102
AUTOARIMA_SEASDUMMY	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
AUTOARIMA_SES	1.106	1.106	1.106	1.106	1.107	1.108	1.110	1.107	1.108
AUTOARIMA_TBATS	1.106	1.107	1.107	1.107	1.107	1.109	1.112	1.108	1.109
BATS	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
DSHW_DAILY	0.016	0.022	0.026	0.030	0.033	0.045	0.068	0.033	0.047
HOLT	0.013	0.019	0.023	0.026	0.030	0.042	0.062	0.030	0.043
HOLT_WINTERS	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
MEANF	0.801	0.802	0.803	0.803	0.804	0.806	0.813	0.804	0.807
naïve	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
NNET	0.017	0.025	0.032	0.038	0.044	0.067	0.104	0.045	0.068
NNET_THETAF	1.105	1.103	1.102	1.101	1.100	1.096	1.088	1.100	1.095
RWF	0.013	0.019	0.023	0.027	0.030	0.042	0.062	0.030	0.043
SES	0.013	**0.019** *0.019*	**0.023** *0.023*	**0.026** *0.026*	**0.029** *0.030*	**0.041** *0.042*	**0.059** *0.062*	**0.029** *0.031*	**0.041** *0.045*
SES_MEAN	0.923	0.923	0.922	0.922	0.922	0.921	0.917	0.922	0.920
SES_THETAF	1.105	1.105	1.105	1.104	1.104	1.101	1.096	1.103	1.101
SINDEX	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
SNAIVE	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
STL	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
TBATS	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041
TBATS_THETAF	1.106	1.106	1.105	1.105	1.104	1.102	1.098	1.104	1.102
THETAF	0.013	0.019	0.023	0.026	0.029	0.041	0.060	0.029	0.042
TSLM	0.013	0.019	0.023	0.026	0.029	0.041	0.059	0.029	0.041

The first table of our results section shows the output results for our tested APE and the table shows specifically the daily output. The winning method from each horizon is presented in bold and the median for the same horizon is presented in italics (this is carried over through all results).In our daily APE results, the simple exponential smoothing (SES) takes over the accuracy testing from the second horizon on, becoming more accurate even when the horizons were averaged, demonstrating its near monopoly of the daily APE. However, the Autoarima Fourier had its say when it came to the first horizon. Whereas both SES and Autoarima_Fourier both show 0.13 in the first horizon Autoarima Fourier performed better on the 4th decimal.

Table: 6:

daily	ape								
METHOD					**HORIZON**				
	1	2	3	4	5	10	22	1-10	1-22
close	0.024248	0.024247	0.024245	0.024244	0.024243	0.024237	0.024219	0.024242	0.024234
close_high	0.024895	0.024894	0.024894	0.024894	0.024893	0.024889	0.024873	0.024892	0.024886
close_high_low	**0.016707** *0.021847*	**0.016708** *0.021847*	**0.016708** *0.021847*	**0.016708** *0.021848*	**0.016708** *0.021848*	**0.016708** *0.021847*	**0.016702** *0.021843*	**0.016708** *0.021848*	**0.016706** *0.021847*
close_low	0.019447	0.019448	0.019449	0.019451	0.019452	0.019458	0.019467	0.019453	0.019459

ses	**0.013289** *0.013383* **(AUTOARIMA_FOURIER)**	**0.018832** *0.018959*	**0.022836** *0.023048*	**0.026175** *0.026486*	**0.029162** *0.029591*	**0.040821** *0.042036*	**0.059134** *0.062275*	**0.029275** *0.031007*	**0.041303** *0.045043*

The daily APE results for the Nymphy model and the winning method from TEST 2 are shown above, with very interesting results. Produced from 4 thousand data points, the method close_high_low shows very consistent results going from horizon 1 through to horizon 22, however that is not the case for the best method carried over from TEST 2, only beating the close_high_low method on the first horizon only where it was not the SES method but rather the Autoarima_Fourier method.

Table:7.

monthly	ape							
METHOD					**HORIZON**			
	1	2	3	4	9	18	1-9	1-18
close	0.110653	0.10991	0.108936	0.108032	0.102802	0.095695	0.106916	0.102559
close_high	0.048861	0.04867	0.048108	0.047595	0.045819	0.044107	0.047275	0.045935
close_high_low	**0.043350** *0.048346*	**0.043355** *0.048161*	**0.043174** *0.047784*	**0.042933** *0.047416*	**0.042019** *0.045705*	**0.041146** *0.043429*	**0.042735** *0.047049*	**0.042062** *0.045658*
close_low	0.047832	0.047652	0.047460	0.047237	0.045591	0.042753	0.046823	0.045381
autoarima_fourier	**0.056734** *0.059492*	**0.081133** *0.086376*	**0.100575** *0.108120* **(NAÏVE)**	**0.118678** *0.129480*	**0.199235** *0.230741*	**0.349945** *0.442083*	**0.133154** *0.150454*	**0.206680** *0.242198*

Producing the same outcome as the daily APE, the weekly results show us how uncertainty can produce better results. Furthermore, where in the daily results the winning method for horizon 1 did perform better than the close_high_low method, here the close_high_low method performed stronger even one horizon ahead.

Table: 7.

quarterly	ape							
METHOD					**HORIZON**			
	1	2	3	4	8	12	1-6	1-12
close	0.496267	0.495893	0.495791	0.497656	0.508113	0.533274	0.497914	0.507351
close_high	0.265978	0.266474	0.267439	0.269942	0.278473	0.291858	0.269350	0.276379
close_high_low	**0.164642** *0.285519*	**0.165393** *0.286098*	**0.166231** *0.286737*	**0.167947** *0.289040*	**0.174909** *0.298342*	**0.183577** *0.312349*	**0.167552** *0.288737*	**0.172817** *0.296064*
close_low	0.305061	0.305721	0.306035	0.308139	0.318211	0.332839	0.308124	0.315750
autoarima_fourier	**0.108245** *0.120869*	**0.164416** *0.190705*	**0.213086** *0.252326*	**0.262497** *0.321027*	**0.538682** *0.628008*	**0.508976** *0.832367* **(SES_THETAF)**	**0.239539** *0.286677*	**0.441059** *0.565188*

The quarterly APE outcome also shows us how the close_high_low method can perform better than all other neural network methods tested here and can also perform stronger and more consistently than the strongest methods from TEST 2. However, in this case, the most accurate method from TEST 2 did perform better one horizon and two horizons ahead, however it lost accuracy after that. Meanwhile, to compare the close_high_low showed that it was consistent moving from one horizon to the next, this is because it did not lose accuracy as quickly as the methods from TEST 2.

Table: 8.

yearly	ape					
METHOD					**HORIZON**	
	1	2	3	4	1-2	1-4
close	0.647111	0.651285	0.676842	0.715215	0.649198	0.672613
close_high	0.217690	0.218180	0.229277	0.248240	0.217935	0.228347
close_high_low	**0.164094** *0.252530*	**0.166871** *0.250952*	**0.179122** *0.262187*	**0.196443** *0.282098*	**0.165483** *0.251741*	**0.176632** *0.261942*
close_low	0.287370	0.283723	0.295097	0.315955	0.285546	0.295536
thetaf_yearly	**0.238219** *0.451838*	**0.324738** *0.558452*	**0.428134** *0.936784* **(NNET_THETAF)**	**0.422913** *1.294401* **(NNET_THETAF)**	**0.281478** *0.505145*	**0.400210** *0.747618*

The same occurs in the yearly APE results, where the close_high_low performs better than all the other methods including the methods from TEST 2, and in this case even in horizon 1 and 2 where in the quarterly results the methods from TEST 2 performed better.

Table: 9.

daily	mse								
METHOD					**HORIZON**				
	1	2	3	4	5	10	22	1-10	1-22
close	1825	1828	1832	1836	1840	1866	1914	1844	1871
close_high	**1788** *2020*	**1791** *2024*	**1794** *2029*	**1798** *2035*	**1801** 2041	**1821** *2073*	**1858** 2130	**1804** *2045*	**1825** *2080*
close_high_low	2215	2220	2227	2234	2241	2280	2354	2246	2289
close_low	2453	2459	2466	2474	2483	2528	2614	2488	2538
ses	**1058**	**1977**	**2839**	**3673**	**4526** *4545*	**8521**	**16861**	**4882**	**9405**

	1061	*1985*	*2851*	*3687*		*8566*	17010	*4904*	*9469*

For the daily MSE results, the close_high method performed better than all other methods, however the SES method from TEST 2 was stronger one horizon ahead but then lost strength and lost strength faster than the close_high after horizon 1. Compared to the APE daily results, the close_high_low method was the stronger model, but this was not the case for the MSE daily.

Table: 10.

weekly	mse							
METHOD					**HORIZON**			
	1	2	3	4	6	12	1-6	1-12
close	**1137** *1968*	**1149** *1979*	**1161** *1986*	**1175** *1991*	**1200** *2011*	**1278** *2052*	**1169** *1989*	**1206** *2012*
close_high	2277	2292	2301	2306	2325	2375	2302	2328
close_high_low	1817	1827	1832	1835	1850	1881	1833	1851
close_low	2119	2130	2139	2147	2172	2223	2144	2174
naive	**4503** *4538*	**8556** *8599*	**12343** *12419*	**15871** *15959*	**22293** *22510*	**42137** *42850*	**13771** *13875* **(SES)**	**23799** *24094* **(SES)**

The MSE weekly results compare well with the APE weekly results, showing us that volatility is not a suitable approach when it comes to the weekly frequency.

4. RESULTS

4.1. Daily Forecasting Results

Traditional Methods (TEST 2 Results)

From the earlier set of tables (referred to here as "TEST 2" results), methods such as

Naïve model $y_t = y_{t+1}$ (13)

Where:

y_t Actual observed value at time t + 1 data point.

Simple Exponential smoothing (SES)

$$\hat{y}_{t+1|t} = a y_t + (1-a)\,\hat{y}_{t|t-1} \quad (14)$$

Where:

$\hat{y}_{t+1|t}$ is the forecast for the next period

a is the smoothing parameter between 0 and 1

$\hat{y}_{t|t-1}$ is the forecasted value at time t made at time t-1

and especially the neural network approach (NNET) showed interesting performance patterns. For instance, the Naïve model was particularly strong in terms of MAE, MAPE, and MASE, whereas the neural network excelled at RMSE in the daily tests, indicative of nonlinearity and higher variance handling capability.

The daily APE table revealed that SES took over the accuracy testing from the second horizon onward, while AUTOARIMA_FOURIER Hyndman (2014)

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{J=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{k=1}^{k} \left(a_k \cos(2\pi k\epsilon/m) + b_k \sin(2\pi kt/m) \right) + \varepsilon_t \quad (15)$$

Where:

μ is the Overall Mean Level: The average level or intercept of the series. Sets the baseline around which the series fluctuates.

$\sum_{i=1}^{p} \phi_i y_{t-i}$ is the autoregressive (AR) term, captures momentum/inertia in the time series

Where:

p is the order of the AR terms

ϕ_i is the AR coefficients, Measure the influence of past values on the current value. High ϕ_i means past values heavily influence y_t

$\sum_{J=1}^{q} \theta_j \varepsilon_{t-j}$ the moving average terms

Where:

q is order of the MA term number of past forecast included

θ_j is the MA coefficient which measure the influence of past errors on the current value

ε_{t-j} is the past error terms, thus the difference between previous observations and their forecasts, hereby adjusting on past mistakes

$\sum_{k=1}^{k} \left(a_k \cos(2\pi k\epsilon/m) + b_k \sin(2\pi kt/m) \right)$

Where:

k is the number of fourier frequencies: Determines how many sine and cosine pairs are included,

a_k and b_k are the fourier coefficients weights for the cosine and sine terms at frequency k,

m is the seasonal period = Number of observations that complete a full seasonal cycle,

Then: ε_t error term at time t: thus equating for random shock or noise at time t,

performing slightly better at horizon 1. The daily MSE (Mean Squared Error) table highlighted that SES tended to be strong over multiple horizons. However, the naive approach and random walk (RWF) occasionally performed competitively, sometimes ranking as top methods at certain short horizons (e.g., 1-step-ahead or 2-step-ahead).

Nymphy (Close-High-Low) vs. TEST 2

When introducing the "Nymphy" methods that leverage the Close price plus volatility aspects (High and Low prices), the results shifted in interesting ways. For example, in the daily APE comparison, NYMPHY_CLOSE_HIGH_LOW (referred to as close_high_low in the tables) demonstrated remarkable consistency across horizons. Meanwhile, the best method from the original set (AUTOARIMA_FOURIER) only outperformed close_high_low at the first horizon. Over longer horizons, the neural network method with Close, High, and Low retained a steady advantage, suggesting that capturing volatility helps the model generalize better when forecasting daily price changes.

When looking at the daily MSE results for these new neural network variants, a slight contrast emerges: NYMPHY_CLOSE_HIGH turned out to be more accurate than close_high_low under the MSE criteria. This discrepancy highlights the importance of distinguishing between absolute percentage errors (APE) and squared errors (MSE). Methods that do well on percentage-based metrics do not necessarily top the rankings on squared error measures, and vice versa.

4.2. Weekly Forecasting Results

Traditional Methods

For weekly data, the neural network approach from the earlier tests (NNET) was a standout, winning four out of six error metrics (RMSE, MAE, MAPE, and MASE). However, the spline model performed poorly, ranking last in four out of six metrics. The weekly MSE table also showed that

Naive (Close-High-Low) vs. TEST 2

In the weekly APE context, the introduction of close_high_low did not meaningfully surpass the best methods from TEST 2 for short horizons. Yet, the overall performance often skewed in favor of neural network approaches that integrate volatility when the horizon was extended. The weekly MSE results could be very strong at specific short horizons. Nonetheless, the table suggests that SES provides robust performance over extended horizons, thus making it appealing for those prioritizing lower long-term risk.

Nymphy

reinforced the impression that volatility-based neural networks may not always dominate every horizon but still present a reliable choice once horizon lengths increase.

4.3. Monthly Forecasting Results

Traditional Methods

The monthly APE and MSE tables show a departure from the daily and weekly scenarios. In the original sets (TEST 2 methods), no single model dominates all horizons outright. Instead, there is a rotation between models like AUTOARIMA_FOURIER, Naïve, SES, and occasionally NNET.

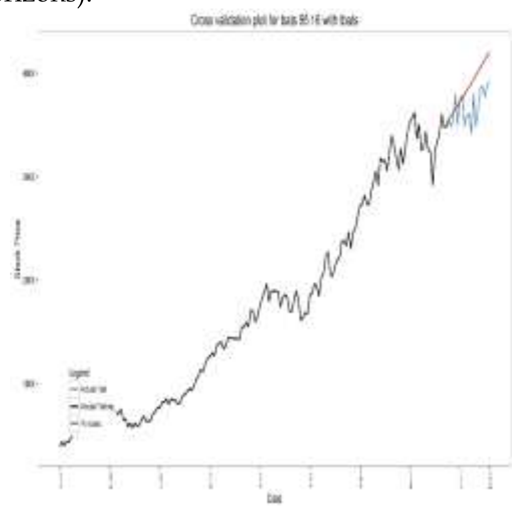
Nymphy (Close-High-Low) vs. TEST 2

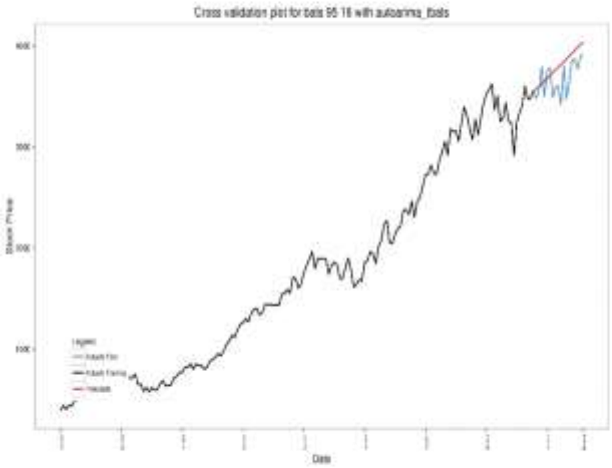
Once I incorporate the new neural network models, NYMPHY_CLOSE_HIGH and NYMPHY_CLOSE_HIGH_LOW frequently emerge as top contenders across multiple horizons. The monthly MSE results confirm that these close-price-plus-volatility approaches beat out older methods by a notable margin, underscoring the advantage of including both High and Low-price parameters. This advantage suggests that monthly data, with moderate frequencies of volatility, benefits from capturing both the amplitude and range of price movements.

4.4. Quarterly Forecasting Results

Traditional Methods

At the quarterly level, some classical methods such as THETAF or Naïve demonstrate particular strengths at certain horizons, especially shorter ones. However, the performance is more mixed as horizon lengthens. The BATS or TBATS models can show large error values if they fail to capture complex seasonality's or produce overfitting problems (as indicated by exceptionally high MSE for certain horizons).





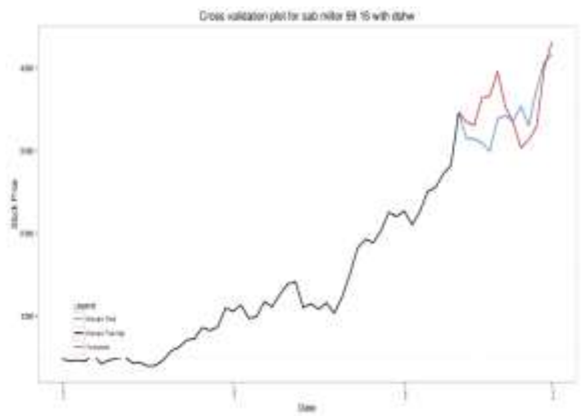
Nymphy (Close-High-Low) vs. TEST 2.

A pronounced pattern emerges in the quarterly results: NYMPHY_CLOSE_HIGH_LOW continues to dominate many horizons under both APE and MSE. It appears that, as the time horizon gets longer, capturing the full range of price movement (Close, High, Low) yields more stable and accurate forecasts. The synergy among these price dimensions helps the neural network anticipate the price direction and magnitude more effectively than single-variable or simpler multi-variable methods.

4.5. Yearly Forecasting Results

- **Traditional Methods**

For yearly data, the prior tests showed that THETAF Assimakopoulos et al (2000) frequently outperforms many classical methods, particularly for the first year or two. Meanwhile, double seasonal Holt-Winters (DSHW) can excel one year ahead but then loses strength for further horizons. SES_MEAN occasionally emerges as a surprise winner at longer horizons (years three and four). Hence, the best yearly approach can vary dramatically, depending on whether the user focuses on a 1-year or multi-year forecast horizon.



Nymphy (Close-High-Low) vs. TEST 2.

When factoring in the new neural network methods that use High and Low, NYMPHY_CLOSE_HIGH_LOW becomes particularly robust as horizons extend to multiple years. This is evidenced by lower APE and MSE values in the tables, suggesting that capturing volatility is a key advantage when predicting over longer horizons. Interestingly, while THETAF can be strong in short yearly horizons, the neural network that integrates the Close, High, and Low data tends to maintain more consistent accuracy beyond the first forecast year.

4.6. Method Performance

At higher frequencies (daily, weekly), the outcome depends on the error metric. For instance, close_high_low dominated daily APE, but close_high led in daily MSE.

At lower frequencies (monthly, quarterly, yearly), close_high_low more consistently led on both APE and MSE, indicating that incorporating volatility becomes increasingly valuable for longer-term forecasts.

Importance of volatility

Methods that integrate exogenous parameters particularly High and Low prices often outperform those relying on the Close price alone. This suggests that volatility dynamics carry predictive power, allowing the neural networks to gather additional patterns and thereby reduce error.

Performance of classical methods

Naive and RWF methods can be surprisingly competitive in short horizons, as they essentially capture very recent trends.

SES, Holt-Winters, and THETAF remain reputable contenders, demonstrating that classical methods are not necessarily inferior but can excel under particular data characteristics.

5. STRATEGIC TRADING, FORECASTING, EVALUATION AND CONCLUDING REMARKS

Strategic trading follows the most accurate trading tool. The trading tool in this essence, I mean the most accurate percentage of error thus the least error computed. This will be incorporated into future work on how to merge the forecasting that was tested in this paper and strategic trading.

From the APE testing, the following is apparent:

Autoarima, simple exponential smoothing (SES) and THETAF performed better than other algorithms. However, frequency does have a significant effect on which models work better and where. Similarly, the horizon also has a noteworthy effect on the models tested, where Autoarima in most cases beat all other models on the first horizon.

The models are competitive. There are somewhat diverse results as I move along the horizon. SES shows complete dominance from the daily APE table where from horizon 2 onwards it performs best in terms of accurate testing. However, Autoarima shows overall dominance.

Despite Autoarima' s overall dominance there are horizons where it showed weakness, especially the third horizon and, in some instances, on the second. It also showed greater strength on medium frequencies and progressively lost strength as the frequencies became greater with less and less training data. Autoarima lost out to other models, which is observable from the yearly accuracy test where it did not win on a single horizon. In the yearly results, other models emerged, for instance THETAF _YEARLY and NNET_THETAF_YEARLY performed significantly better than Autoarima.

While Autoarima showed great dominance overall, SES showed great dominance on the daily accuracy exclusively, in this test, the latter won on all of the horizons except the first. Similar dominance was only shown by Autoarima in the quarterly and monthly APE results. Thus, I can say with great confidence that the best trading model for daily testing is SES and for monthly and quarterly testing is Autoarima. From these observations I can easily say that, with more training data, SES is the best method, however with medium training data Autoarima is a better option.

Despite constraints in place preventing the collection of larger amounts of data, our test shows that THETAF and NNET_THETAF are the most suitable approaches.

To conclude, for each frequency there is a specific model that prevails. I now take our test to the next level; in the next chapter I introduce neural networks with volatility variables with the introduction of high frequency trading. The aim here is to determine the perfect model, and to test if more complexity is better than less in models. It has been suggested that the simplest models prevail and produce greater accuracy than more complicated models.

The addition of High and Low prices to neural network forecasting models generates improved accuracy levels for different time intervals. When applied over medium- to long-term periods the NYMPHY_CLOSE_HIGH_LOW approach delivers notable effectiveness as no strategy achieves complete optimality. SES together with Naïve along with classical models demonstrate effective performance when predicting short-term results.

Performance drops as forecast timeframes elongates but Naïve and SES maintain short-run accuracy along with THETAF and AUTOARIMA demonstrating superiority in the long term despite a decrease in reliability. Neural networks achieve robust forecast performance when used with external regressors to monitor market behavior patterns.

Results show that the CLOSE_HIGH_LOW method delivery sustained superior performance compared to alternative methods but showed limited exceptions where additional horizon testing would enhance selection refinement. The development of real-world trading applications must remain a priority since CLOSE_HIGH_LOW demonstrates the optimal combination of predictive accuracy and operational adaptability and consistency.

REFERENCES

Alroomi, A. (2024). Forecasting analytics for industry 4. Edelweiss Applied Science and Technology, 8(6), 9778–9783.

Alroomi, A., et al. (2022). Fathoming empirical forecasting competitions' winners. International Journal of Forecasting, 38(4), 1519–1525.

Ampountolas, A. (2023). Comparative analysis of machine learning, hybrid, and deep learning forecasting models: evidence from European financial markets and bitcoins. Forecasting, 5(2), 472–486.

Arévalo, R, García, J, Guijarro, F, Peris, A, (2017) A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting, Expert Systems with Applications, 81,177-192,

Assimakopoulos, V, Nikolopoulos, K. (2000) The theta model: a decomposition approach to forecasting, International Journal of Forecasting,16 (4), 521-530,

Barberis, N., & Xiong, W. (2009). What drives the disposition effect? An analysis. The Journal of Finance, 64(2), 751–784.

Bekiros, S., & Georgoutsos, D. A. (2008). Direction-of-change forecasting using a volatility-based recurrent neural network. Journal of Forecasting, 27, 407–417.

Booth, G, Gurun, U, Zhang, H, Financial networks and trading in bond markets, (2014) Journal of Financial Markets,18,126-157.

Chatterjee, U, Do stock market trading activities forecast recessions? (2016) Economic Modelling, 59, 370-386.

Chen, C, Wang, Z, Sriboonchitta, S, Lee, S, (2017) Pair trading based on quantile forecasting of smooth transition

GARCH models, The North American Journal of Economics and Finance, 39,38-55.

Choudhry, T., McGroarty, F., Peng, K., & Wang, S. (2012). Can high-frequency data improve the forecasting performance of popular volatility models? Journal of Forecasting, 31, 565–587.

Dunis, C. L., Rosillo, R., de la Fuente, D., & Pino, R. (2013). Forecasting IBEX-35 moves using support vector machines. Neural Comput & Applic, 23, 229–236.

Dunis, C., & Huang, X. (2002). Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. Journal of Forecasting, 21(5), 317–354.

Erdogan, O, Bennett, P, Ozyildirim, C, (2014) Recession Prediction Using Yield Curve and Stock Market Liquidity Deviation Measures, Review of Finance,9,1572-3097

Gajamannage, K., Park, Y., & Jayathilake, D. I. (2023). Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs. Expert Systems with Applications, 223, 119879.

Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognition, 70, 163–176.

Hyndman, R. J. (2014). Forecasting: principles and practice.

Joseph, K, M. Babajide Wintoki, Zelin Zhang, (2011), Forecasting abnormal stock returns and trading volume using investor sentiment: Evidence from online search, International Journal of Forecasting, 27, 4,1116-1127

Khan, M, Alroomi, A., & Nikolopoulos, K. (2022). Supply chain disruptions and consumer behavior change from COVID-19: Empirical evidence and long-term implications for supermarkets in the UK. J. Humanit. Arts Soc. Sci, 6, 28–42.

Kurek, B, The Information Content of Equity Block Trades on the Warsaw Stock Exchange: An Estimation of Shares' Returns with the Usage of Simple Linear Regression and Multivariate Adaptive Regression Splines, Journal of Forecasting, 33(6),433-454

Kuremoto, T., Kimura, S., Kobayashi, K., & Obayashi, M. (2014). Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing, 137, 47–56.

Li, J., Alroomi, A., & Nikolopoulos, K. (2024). Forecasting crude oil markets. Journal of Econometrics and Statistics, 4(1).

Li, Y, Bu, H, Li, J, Wu, J, (2020),The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning, International Journal of Forecasting, 36, 4, 1541-1562,

Moghaddam, A., Moghaddam, M., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. Journal of Economics, Finance, and Administrative Science, 21, 89–93.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., & Wald, R. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1–21.

Pham, H, Cooper, E, Cao, T, Kamei, K, (2014) Hybrid Kansei-SOM model using risk management and company assessment for stock trading, Information Sciences, 256, 8-24.

Qu, H., & Zhang, Y. (2016). A new kernel of support vector regression for forecasting high-frequency stock returns. Mathematical Problems in Engineering, 22, 1–9.

Riedel, C., & Wagner, N. (2015). Is risk higher during non-trading periods? The risk trade-off for intraday versus overnight market returns. Int. Fin. Markets, Inst. and Money, 39, 53–64.

Sermpinis, G, Charalampos, Stasinakis, C, Theofilatos, K, Karathanasopoulos, A, (2015)Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms Support vector regression forecast combinations, European Journal of Operational Research, 247, 3, 831-846.

Sermpinis, G., Stasinakis, C., Rosillo, R., and de la Fuente, D. (2017) European exchange trading funds trading with locally weighted supportvector regression. European Journal of Operational Research, 258(1), 372-384.

Shoji, I., & Kanehiro, S. (2016). Disposition effect as a behavioral trading activity elicited by investors' different risk preferences. International Review of Financial Analysis, 46, 104–112.

Sokolov-Mladenovic, S., Milovancevic, M., Mladenovic, I., & Alizamir, M. (2016). Economic growth forecasting by artificial neural network with extreme learning machine based on trade parameters. Computers in Human Behavior, 65, 43–45.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. Omega, 29, 309–317.

Tealab, A., Hefny, H., & Badr, A. (2017). Forecasting of nonlinear time series using ANN. Future Computing and Informatics Journal, 20, 1–9.

Vella, V., & Ng, W. L. (2016). Improving risk-adjusted performance in high frequency trading using interval type-2 fuzzy logic. Expert Systems With Applications, 55, 70–86.

Wei, Y., Liu, J., Lai, X., & Hu, Y. (2017). Which determinant is the most informative in forecasting crude oil market volatility: Fundamental, speculation, or uncertainty? Energy Economics, 68, 141–150.

Yu, H., Fang, L., & Sun, W. (2018). Forecasting performance of global economic policy uncertainty for volatility of Chinese stock market. Physica A, 505, 931–940.

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32(6), 1609–1628.

Zeng, N., Zhang, H., Liu, W., Liang, J., & Alsaadi, F. E. (2017). A switching delayed PSO optimized extreme learning machine for short-term load forecasting. Neurocomputing, 240, 175–182