NHITS for Forecasting Stock Realized Volatility

Hugo Gobato Souto[1]

[1] HAN University of Applied Sciences, the Netherlands
Ruitenberglaan 31, 6826 CC Arnhem, the Netherlands
Hugo.GobatoSouto@han.nl. https://orcid.org/0000-0002-7039-0572

(current draft) December 2023

Abstract

Disclaimer: although the research part of this paper has already been completed and the results and findings of this paper can already be used by the scientific community, readers must realized that this paper is still a preprint (i.e., not been published yet) and should be cited accordingly

1 Introduction

The accurate prediction of realized volatility is a pivotal element in the field of financial economics, bearing significant implications for portfolio management, risk assessment, and strategic investment decisions (Atkins et al., 2018; Bašta & Molnár, 2018a; Bonato et al., 2021; Bouri et al., 2021; M. Liu et al., 2022; Mesquita et al., 2023; Souto, 2023c; Tang et al., 2023). The evolution of this domain has been marked by the continual development of advanced statistical and machine learning models, each aiming to enhance the precision and reliability of volatility forecasts (Atkins et al., 2018; Bouri et al., 2021; Souto, 2023a). This study is situated within this evolving landscape, focusing on the potential of Neural Hierarchical Interpolation for Time Series Forecasting (NHITS) (Challu et al., 2023) in predicting stock market volatility.

Volatility forecasting is integral to financial market analysis, providing essential insights for risk management, derivative pricing, and strategic asset allocation (Andersen & Teräsvirta, 2009; Bašta & Molnár, 2018b; Bauwens et al., 2006; Degiannakis et al., 2022; Poon & Granger, 2003; Souto, 2023b; Todorova & Souček, 2014). Traditional models, such as the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model (Bollerslev, 1986) and the Heterogeneous Autoregressive (HAR) model of realized volatility (Corsi, 2009), have been the cornerstone in this arena. In recent years, however, there has been an expanding interest in applying machine learning techniques, such as Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997) and advanced architectures like the Temporal Fusion Transformer

(TFT) (Lim et al., 2021) and NBEATSx (Olivares et al., 2023), to enhance forecasting accuracy.

The NHITS model, a relatively new entrant in the field, has shown promise in other time series forecasting applications (Challu et al., 2023; Chen et al., 2023; Hewamalage et al., 2023; Y. Liu et al., 2022; Woo et al., 2023; Zheng et al., 2023) but remains underexplored in the context of stock market volatility. This research aims to bridge this gap by rigorously evaluating the NHITS model's effectiveness in forecasting stock realized volatility and comparing its performance with both traditional and contemporary models.

While extensive research has been conducted on volatility prediction using various models (Andersen et al., 2003; Bucci, 2020; Deo et al., 2006; Louzis et al., 2014; McAleer & Medeiros, 2008; Miura et al., 2019; Qu et al., 2018; Souto & Moradi, 2023c; Vortelinos, 2017) , the application of NHITS in this domain is not well-established. This study adds to the realized volatility literature by:

1. Introducing the NHITS model to the realm of stock market volatility forecasting.
2. Comparing the performance of NHITS with established models (GARCH, HAR) and recent machine learning approaches (LSTM, TFT, NBEATSx) in forecasting stock realized volatility.
3. Introducing a thorough and robust methodology framework for properly evaluating novel forecasting models .
4. Introducing a dynamic variation of the famous Model Confidence Set (MCS) (Hansen et al., 2011) for financial time series data that allows for statistically temporal evaluation of forecasting models.

The study adopts a methodical approach, integrating several key components crucial for comprehensive research. In terms of **Data Collection**, the research utilizes a dataset that includes daily realized volatility values of 80 stocks from the S&P 100 index, covering the time-frame from July 1, 2007, to June 30, 2021. This dataset provides a rich ground for analysis, encompassing a broad spectrum of market conditions.

For **Model Implementation**, a detailed and meticulous process is employed. This involves the implementation and optimization of the NHITS model, along with a comparison to established models such as GARCH and HAR, and other recent machine learning approaches including LSTM, TFT, and NBEATSx. This comparative framework allows for a nuanced understanding of the NHITS model's capabilities within the broader context of existing volatility forecasting methodologies.

In the realm of **Performance Evaluation**, the study rigorously assesses the performance of these models using various error measurements, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Quasi-Likelihood (QLIKE). These metrics provide a multi-dimensional view of model accuracy and effectiveness, enabling a comprehensive evaluation of their predictive capabilities.

Additionally, the study incorporates robust **Statistical Tests** to further validate the findings. This includes the application of MCS (Hansen et al., 2011) and Diebold-

Mariano (DM) tests (Diebold & Mariano, 1995) to evaluate the models' forecast performance as well as T-tests and F-tests to evaluate the neural network models' robustness to the change of random seed choice. These tests are instrumental in evaluating and comparing the predictive accuracy and robustness of the models, offering a statistical basis for the assessment of their performance.

Finally, the research conducts extensive **Robustness Checks** to ensure the reliability of the results. These checks include alternative data splitting strategies, using reduced training samples, and performing sensitivity analysis with varying random seeds. These robustness tests are critical in affirming the models' stability and effectiveness across different scenarios and data conditions.

This research holds significant implications for both academic inquiry and practical application in financial markets. By exploring the utility of the NHITS model in volatility forecasting, the study not only expands the methodological repertoire for financial analysts but also offers insights into the adaptability and effectiveness of machine learning techniques in capturing the complex dynamics of stock market volatility. The findings of this study are anticipated to contribute valuable perspectives to the ongoing discourse on the integration of advanced machine learning models in financial forecasting.

The remainder of this paper is organized as follows: Section 2 thoroughly explains the novel model. Section 3 details the methodology, including data collection, model implementation, and evaluation metrics. Section 4 presents the empirical results for both the main sample and the robustness tests. Finally, Section 5 concludes the paper with a summary of the key findings and their implications.

2 NHITS

2.1 Conceptual Explanation

Conceptually, the NHITS model is a generalization and extension of the NBEATS model (Oreshkin et al., 2020). The structure of NHITS allows it to to perform a hierarchial construction of forecast for time series data, making the model more accurate and computationally effcient, especially in the context of long-horizon forecasting (Challu et al., 2023). Thanks to the multi-scale synthesis of the forecast and multi-rate sampling of the input times series, NHITS greatly reduces computational costs while achieving a higher forecasting power (Challu et al., 2023).

The NHITS model employs a block-based neural architecture, where stacks are employed, each possessing blocks that are responsible for capturing different components of the time series data. These components can be trend, seasonality, or other cyclical patterns that are common in time series analysis. In the context of stock realized volatility, the stacks will be presumably focused on the monthly and weekly cyclical patterns, similarly to the HAR model. Figure 1 illustrates such a structure. The model's architecture allows for the direct forecasting of the target time series at various aggregation levels, which is crucial for generating coherent and reconciled forecasts across the hierarchy (Challu et al., 2023).

One of the novel features of the NHITS model is its ability to do temporal aggrega-

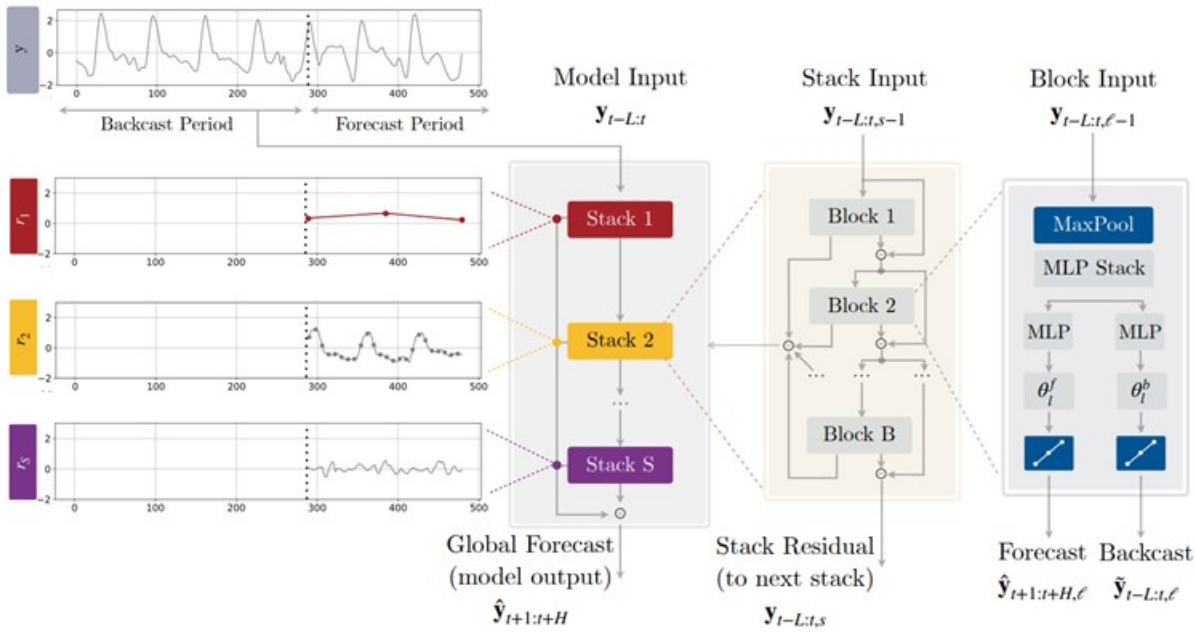

Figure 1: NHITS Architecture (Challu et al., 2023). The model consists of various stacks, each focused on a certain time series patterns. Each stack, on the other hand, possesses multiple interconnected Multi-Layer Perceptrons (MLPs), with ReLU as their activation function, using the doubly residual stacking basis.

tion directly within the neural network. This is achieved through the use of hierarchical interpolation blocks, which learn to interpolate the time series data at different temporal resolutions. By doing so, the NHITS model can effectively capture the underlying patterns at various frequencies, leading to more accurate and robust forecasts (Challu et al., 2023).

The NHITS model primarily involves a series of feed-forward neural networks, each corresponding to a block in the model's architecture (as shown in Figure 1). Blocks are clustered per stack, where each stack focus on a certain temporal component. Moreover, the input to each stack is a window of past observations, and the output is a forecast for the corresponding future period. The blocks, on the other hand, are connected in a hierarchical manner, with the outputs of lower-level blocks serving as inputs to higher-level blocks, allowing the model to integrate information across different levels of the time series hierarchy.

In conclusion, the NHITS model is a powerful neural network model for time series forecasting, offering a high degree of flexibility, accuracy, and consistency. Its potential application to financial forecasting demonstrates the potential of advanced machine learning techniques to further revolutionize the field of econometrics and financial analysis.

2.2 Mathematical Explanation

For this mathematical explanation, we will use Figure 1 as a reference. Let y_t be the realized volatility at time t of a certain stock, called stock A for the of example, and $\mathbf{y_{t-L:t}}$ be a vector with L lags of stock A realized volatility. Also, let $\tilde{\mathbf{y}}_{\mathbf{t-L:t,b}}$ and $\hat{\mathbf{y}}_{\mathbf{t-L:t+H,b}}$ respectively be the backcast and forecast of the b-th block. Finally, let $\tilde{\mathbf{y}}_{\mathbf{t-L:t,s}}$

and $\hat{\mathbf{y}}_{\mathbf{t-L:t+H,s}}$ respectively be the residual and forecast of the s-th stack and let $\hat{\mathbf{y}}_{\mathbf{t-L:t+H}}$ be the final forecast of the NHITS model.

As shown in Figure 1, NHITS receives the input $\mathbf{y}_{\mathbf{t-L:t}}$, which firstly goes to the first block of the first stack. In the first and all blocks, a MaxPool layer with kernel size of k_b is used to select the important components of the input. While a smaller kernel size allows more high-frequency components from the realized volatility data to be used, a bigger kernel size cuts more high-frequency elements from the time series data. As a result, the MLP from block b will be either forced to focus on learning low- or high-frequency patterns depending on the chosen kernel size. This is called Multi-Rate Signal Sampling (MRSS) by Challu et al. (2023). MRSS also allows NHITS to perform training more rapidly since the input size of the MLP from block b decreases, which itself also decreases the number of parameters for the MLP, reducing overfitting risks. MRSS is mathematically defined as:

$$\mathbf{y}^{(\mathbf{p})}_{\mathbf{t-L:t,b}} = \mathbf{MaxPool}(\mathbf{y}_{\mathbf{t-L:t,b}}, k_b) \tag{1}$$

Subsequent to MRSS, block b uses the reduced input to non-linearly regress forward $\theta^{\mathbf{f}}_{\mathbf{b}}$ and backward $\theta^{\mathbf{b}}_{\mathbf{b}}$ interpolation MLP parameters that estimates the hidden vector $\mathbf{h_b}$. This is mathematically translated into:

$$\begin{aligned} \mathbf{h_b} &= \mathbf{MLP_b}(\mathbf{y}^{(\mathbf{p})}_{\mathbf{t-L:t,b}}), \\ \theta^{\mathbf{f}}_{\mathbf{b}} &= \mathbf{LINEAR^f}(\mathbf{h_b}), \\ \theta^{\mathbf{b}}_{\mathbf{b}} &= \mathbf{LINEAR^b}(\mathbf{h_b}) \end{aligned} \tag{2}$$

The learned parameters are subsequently utilized in order to estimate $\tilde{\mathbf{y}}_{\mathbf{t-L:t,b}}$ and $\hat{\mathbf{y}}_{\mathbf{t-L:t+H,b}}$ via hierarchical interpolation (HI), explained in Sub-section 2.2.1. Then $\tilde{\mathbf{y}}_{\mathbf{t-L:t,b}}$ is subtracted from the input $\mathbf{y}^{(\mathbf{p})}_{\mathbf{t-L:t,b}}$, creating the new input for the $b+1$-th block, which itself will estimate $\tilde{\mathbf{y}}_{\mathbf{t-L:t,b+1}}$ and $\hat{\mathbf{y}}_{\mathbf{t-L:t+H,b+1}}$. Finally, all forecasts will be summed up to create the forecast of the s-th stack and after subtracting the last backcast residual from the input signal, the input for the $s+1$-th stack is created. This process can be mathematically seen below:

$$\begin{aligned} \mathbf{y}_{\mathbf{t+1:t+H,s}} &= \sum_{b=1}^{B} \hat{\mathbf{y}}_{\mathbf{t+1:t+H,b}} \\ \mathbf{y}_{\mathbf{t-L:t,s}} &= \mathbf{y}_{\mathbf{t-L:t,B}} - \tilde{\mathbf{y}}_{\mathbf{t-L:t,B}} \end{aligned} \tag{3}$$

Lastly, the forecasts of all stacks are summed up to create the final model forecast $\hat{\mathbf{y}}_{\mathbf{t-L:t+H}}$.

2.2.1 Hierarchical Interpolation

To avoid the exploding amount of computational power required by other neural network multi-horizon forecasting models as the forecast horizon H increases, NHITS employs a technique coined Temporal Interpolation (TI) by Challu et al. (2023). Let r_b be the dimensionality of the interpolation parameters, which governs the parameter count per unit output time, with $|\theta_{\mathbf{f_b}}| = \lceil r_b H \rceil$. For the purpose of reinstating the initial

sampling rate and forecasting all H points within the horizon, TI is employed through the interpolation function g:

$$\begin{aligned} \hat{y}_{\tau,b} &= g(\tau, \theta_{\mathbf{f_b}}), \forall \tau \in \{t+1, \ldots, t+H\}, \\ \tilde{y}_{\tau,b} &= g(\tau, \theta_{\mathbf{b_b}}), \forall \tau \in \{t-L, \ldots, t\}. \end{aligned} \tag{4}$$

where, g is an interpolation function that can be in the form of nearest neighbor, piecewise linear, and cubic. This mathematical explanation limits itself to the linear interpolator for the sake of simplicity, yet readers who are interested in the other types of interpolators are invited to read Challu et al. (2023). g, along with the time partition $T = \{t+1, t+1+\frac{1}{r_b}, \ldots, t+H-\frac{1}{r_b}, t+H\}$, is defined as:

$$g(\tau, \theta) = \theta[t_1] + (\frac{\theta[t_2] - \theta[t_1]}{t_2 - t_1})(\tau - t_1) \tag{5}$$

where $t_1 = \text{argmin}_{t \in T: t \leq \tau}(\tau - t)$ and $t_2 = t_1 + \frac{1}{r_b}$. With the HI approach, MRSS is complemented by the strategic allocation of expressiveness ratios across various blocks. Blocks situated nearer to the input exhibit a smaller r_b but a larger k_b. This indicates that such so-called input blocks are tasked with producing signals of lower granularity, achieved through more pronounced interpolation. Additionally, they are compelled to engage with signals that have undergone more intense sub-sampling and smoothing. The hierarchical forecast $\hat{\mathbf{y}}_{\mathbf{t+1:t+H}}$ is constructed by aggregating the outputs from all blocks and stacks, thereby effectively synthesizing it from interpolations conducted at diverse levels of the time-scale hierarchy. Figure 2 depicts the HI technique. In Figure 2, it can be seen how the employment of exponentially increasing expressive ratios can deal with diverse frequency bands within the constraints of parameter quantity. Additionally, each stack might be dedicated to replicating distinct, recognized cycles (e.g., monthly, weekly, daily, etc.) of the realized volatility data, employing a corresponding r_b. Last but not least, Challu et al. (2023) prove the theoretical guarantees of the HI technique, showing that this method is not only empirically powerful and robust, but also theoretically.

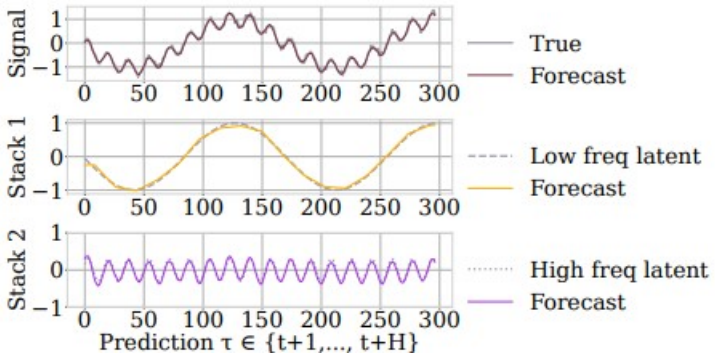

Figure 2: Neural Hierarchical Interpolation (Challu et al., 2023).

3 Methodology

3.1 Data

3.1.1 Comprehensive Analysis of the S&P 100 Stocks Dataset

This investigation utilizes an extensive dataset encompassing equities listed in the Standard & Poor's 100 (S&P 100) index, meticulously selected for their uninterrupted trading activity spanning from July 1, 2007, to June 30, 2021. This timeframe is strategically chosen to encapsulate a spectrum of financial epochs, notably the 2008 global financial crisis and the market perturbations caused by the COVID-19 pandemic. Such a range presents a formidable testing ground for examining the efficacy of various volatility prediction models.

A corpus of 80 stocks was curated, each furnishing a dataset of 3,409 daily realized volatility (RV) values, thus offering a nuanced panorama of market dynamics throughout the chosen period. The dataset's RV calculations are grounded in high-frequency intraday data obtained from the LOBSTER database, a repository acclaimed for its precision in limit order book data. This selection guarantees the integrity and uniformity of the RV computations.

The RV values were derived utilizing the method delineated by L. Y. Liu et al. (2015), a sophisticated approach that meticulously gauges volatility through high-frequency intraday data. For more details about this method, please see L. Y. Liu et al. (2015).

Furthermore, a comprehensive catalog of these 80 stocks, alongside a statistical compendium of their daily 5-minute RV values, is meticulously presented in Appendix A. This appendix provides a constellation of essential statistics including mean, median, standard deviation, and other pivotal statistical indices, thereby offering an initial exploration into the volatility profile of each constituent stock. The dataset's expansive

scope and depth, traversing varied market conditions and encompassing a diverse array of S&P 100 equities, render it an ideal substrate for scrutinizing the performance of disparate volatility prediction models, prominently featuring the innovative NHITS model.

3.1.2 Data Partitioning Strategy

In the realm of this study, the dataset is partitioned into a training sample, constituting 70% of the total, and a testing sample, comprising the remaining 30%. This division adheres to a well-established norm within the domains of machine learning and statistical modeling, reflecting a balanced approach to both training the models effectively and appraising their predictive performance with precision. The adoption of a 70%/30% training/testing split is underpinned by its prevalent application in the literature of forecasting (Joseph, 2022), albeit the 80%/20% training/testing split is considered the most common one (Joseph, 2022). Nonetheless, the 80%/20% training/testing split is used in the **Robustness Test 4** of this paper.

Moreover, the selections of a 70%/30% split for the main sample and a 80%/20% split for the **Robustness Test 4** represent a deliberate effort to harmonize bias and variance trade-off in the model's predictions. A larger training set is advantageous in diminishing bias, as it allows the model to glean a more comprehensive understanding from the dataset. Conversely, ensuring that the testing set is adequately large is pivotal for evaluating the variance in the model's predictions, thereby affirming the model's capability to generalize effectively to novel data scenarios.

3.2 Benchmark Models

3.2.1 The HAR Model

The HAR model, introduced by Corsi (2009), represents a significant advancement in the modeling of financial time series, particularly in the context of realized volatility forecasting. The HAR model is conceptually grounded in the recognition of heterogeneous market components that operate across various time horizons (Corsi, 2009). This heterogeneity is a fundamental characteristic of financial markets, where traders and investors exhibit diverse trading behaviors and react to information over different time scales.

Mathematically, the HAR model can be expressed as:

$$\hat{RV}_{t+1} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} \tag{6}$$

where $\hat{RV}_{t+1}$ denotes the estimated realized volatility at time $t+1$ and β_0 is a constant term. The terms $RV_t^{(d)}$, $RV_t^{(w)}$, and $RV_t^{(m)}$ represent the daily, weekly, and monthly realized volatilities, respectively, capturing the volatility dynamics at different time scales. The coefficients β_d, β_w, and β_m quantify the impact of these time-varying volatilities on the future volatility.

The HAR model's ability to encapsulate volatility dynamics across multiple time horizons renders it particularly adept at capturing the complex temporal structures inherent in financial time series (Corsi, 2009). It acknowledges that market participants

operate under different time frames, from intraday traders to long-term investors, and each group's actions contribute to the overall volatility dynamics.

3.2.2 Rationale for the Choice of the HAR Model as a Benchmark

The selection of the HAR model as a benchmark in this study is motivated by several factors. Firstly, the HAR model's simplicity and interpretability make it a standard reference point in volatility forecasting literature (Audrino & Knaus, 2015; Audrino et al., 2018; Corsi et al., 2012; Y. Wang et al., 2016; Yao et al., 2019). Its ability to model long memory and capture the persistence of volatility over different time horizons aligns well with the characteristics of financial time series data. Moreover, the HAR model has been empirically validated in numerous studies, proving its efficacy in forecasting market volatility (Audrino & Knaus, 2015; Audrino et al., 2018; Corsi, 2009; Corsi et al., 2012; Y. Wang et al., 2016; Yao et al., 2019).

3.2.3 The GARCH Model

The GARCH model, initially conceptualized by Bollerslev (1986), is a cornerstone in the quantitative analysis of financial markets, particularly in volatility modeling. The GARCH model extends the ARCH (Autoregressive Conditional Heteroskedasticity) model, introduced by Engle (1982), by incorporating lagged conditional variances, thereby enhancing its ability to model the time-varying volatility commonly observed in financial time series.

The standard GARCH(1,1) model is mathematically represented as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{7}$$

where σ_t^2 denotes the conditional variance (volatility) at time t, α_0 is a constant term, ϵ_{t-1} is the lagged error term, and σ_{t-1}^2 is the lagged conditional variance. The parameters α_1 and β_1 capture the effects of short-term shocks and the persistence of volatility, respectively.

The GARCH model is renowned for its proficiency in capturing the "volatility clustering" phenomenon, a ubiquitous feature in financial markets where periods of high volatility tend to be followed by high volatility and vice versa (Bollerslev, 1986). This ability to model the persistence and mean-reverting nature of volatility makes the GARCH model particularly suitable for financial time series analysis (Bollerslev, 1986).

3.2.4 Rationale for the Choice of the GARCH Model as a Benchmark

The inclusion of the GARCH model as a benchmark in this study is underpinned by the GARCH model's widespread acceptance and usage in both academic research and industry practice establish it as a fundamental benchmark for volatility forecasting (BUCCI, 2018; Corsi et al., 2008; Hansen et al., 2011; Hansen et al., 2014; Kambouroudis et al., 2016; Sharma & Vipul, 2016). Its methodological rigor and proven track record in capturing the dynamics of financial market volatility make it an ideal point of comparison for evaluating newer and more complex models.

3.2.5 The LSTM Model

The LSTM model, a model in the family of of recurrent neural networks (RNN) architecture, devised by Hochreiter and Schmidhuber (1997), was specifically designed to attempt to solve the limitations of RNNs in capturing long-term dependencies. LSTMs have since become a pivotal tool in the realm of sequential data analysis, particularly in fields where understanding time-based dynamics is crucial, such as financial time series forecasting.

The core concept of the LSTM model is its ability to maintain a 'memory' of past information, enabling it to capture temporal dependencies over extended periods. This is achieved through its unique structure, consisting of various types of gates: the input gates, the output gates, and the forget gates. These gates collectively regulate the transfer of information in and out of the cell state, allowing the model to retain or discard information based on its relevance.

The mathematical representation of an LSTM unit can be articulated as follows:

$$p_t = \sigma(V_p \cdot [h_{t-1}, x_t] + c_p) \tag{8}$$

$$q_t = \sigma(V_q \cdot [h_{t-1}, x_t] + c_q) \tag{9}$$

$$\tilde{D}_t = \tanh(V_D \cdot [h_{t-1}, x_t] + c_D) \tag{10}$$

$$D_t = p_t * D_{t-1} + q_t * \tilde{D}_t \tag{11}$$

$$r_t = \sigma(V_r \cdot [h_{t-1}, x_t] + c_r) \tag{12}$$

$$h_t = r_t * \tanh(D_t) \tag{13}$$

Where σ denotes the sigmoid function, tanh is the hyperbolic tangent function, p_t, q_t, and r_t represent the forget, input, and output gates, respectively, D_t is the cell state, h_t is the hidden state, and V and c are the weights and biases of the respective gates.

3.2.6 Rationale for the Choice of the LSTM Model as a Benchmark

The decision to incorporate the LSTM model as a benchmark in this study is grounded in its demonstrated capabilities in handling time series data, especially in financial contexts (Cao et al., 2019; Mehtab & Sen, 2022; Siami-Namini et al., 2018). The LSTM model's architecture is inherently suited for capturing the intricacies and nonlinear patterns observed in stock market volatility. This suitability is attributed to its design, which allows it to remember important information over long time horizons and forget non-essential information, a critical feature for effective volatility forecasting.

Furthermore, the LSTM's versatility in handling different types of data and its ability to be tailored for specific forecasting tasks make it an exemplary benchmark. Its comparison with traditional models like GARCH and HAR, as well as with other advanced machine learning models, provides valuable insights into the evolution and efficacy of forecasting methodologies in finance. Empirical studies, such as those by Fischer and Krauss (2018), Bucci (2020), and Souto and Moradi (2023b), have underscored the LSTM model's superior performance in capturing complex temporal dependencies in financial time series, further justifying its inclusion as a benchmark in this study.

3.2.7 The TFT Model

The TFT model, a novel architecture in the domain of time series forecasting, integrates the advantages of attention mechanisms as seen in transformers with components specifically tailored for temporal data. Introduced by Lim et al. (2021), the TFT model is designed to effectively handle the complexities inherent in multivariate time series forecasting tasks. It excels in capturing both long-term dependencies and time-varying relationships within the data, making it particularly suitable for financial market analysis.

Conceptually, the TFT model distinguishes itself through its ability to process and interpret multiple input features with varying temporal relevance. This is accomplished through its unique architecture that combines recurrent layers, attention mechanisms, and temporal processing layers. The model's design allows it to discern important features, adaptively focusing on relevant time steps and disregarding irrelevant information. This capability is crucial in financial markets, where the relevance of information can vary significantly over time.

Given the mathematical complexity of the TFT model, a detailed of its architecture is not covered in this paper to ensure sparsity. Nonetheless, for a complete explanation of the TFT model, please see Lim et al. (2021).

3.2.8 Rationale for the Choice of the TFT Model as a Benchmark

The inclusion of the TFT model as a benchmark in this research is predicated on several key considerations. Firstly, the TFT model represents the cutting-edge in machine learning-based time series forecasting. Its advanced architecture, which adeptly combines attention mechanisms and recurrent neural networks, positions it at the forefront of forecasting methodologies capable of handling complex, multivariate time series data (Lim et al., 2021).

The model's proficiency in discerning and leveraging time-dependent relationships within data makes it particularly relevant for financial market volatility forecasting (Frank, 2023; Hu, 2021; Olorunnimbe & Viktor, 2022; Wu et al., 2022). This relevance is underscored by the ever-evolving and dynamic nature of financial markets, where the ability to adapt to changing conditions and extract meaningful patterns from a multitude of variables is imperative.

Moreover, the TFT model's recent adoption in financial applications and its reported success in various empirical studies highlight its potential as a powerful tool for volatility prediction (Frank, 2023; Hu, 2021; Lim et al., 2021; Olorunnimbe & Viktor, 2022; Wu et al., 2022). Its comparison with other established models will not only validate its effectiveness in a financial context but also provide insights into the evolving landscape of financial time series forecasting.

3.2.9 The NBEATSx Model

The NBEATSx model, an extension of the original NBEATS model, represents a significant innovation in the field of neural network-based time series forecasting. Initially introduced by Olivares et al. (2023) and further developed in subsequent iterations

(for example the Realized Covariance Matrix NBEATSx (Souto & Moradi, 2023a)), NBEATSx enhances the original architecture to better accommodate external variables and complex time series patterns, making it highly applicable to multifaceted datasets, such as those found in financial markets.

At its core, NBEATSx is grounded in a deep learning framework that eschews the conventional reliance on recurrent or convolutional layers, common in many neural network models. Instead, it utilizes a fully connected network architecture with stacked blocks, each designed to forecast a forward-looking window of the time series. These blocks employ a combination of backward and forward residual links, enabling the model to effectively capture both historical trends and future patterns in the data.

Due to the compex nature of the NBEATSx model's mathematical structure, a thorough mathematical explanation of the model is not given in this paper. Yet, interested readers can see such an explanation in Olivares et al. (2023).

3.2.10 Rationale for the Choice of the NBEATSx Model as a Benchmark

The decision to incorporate the NBEATSx model as a benchmark in this study stems from several compelling factors. Firstly, the NBEATSx model represents a forefront development in neural forecasting, offering a novel approach that diverges from traditional recurrent neural network designs (Olivares et al., 2023). Its architecture is specifically tailored to capture complex patterns in time series, an attribute crucial for accurately modeling the nuanced dynamics of financial markets.

Moreover, the NBEATSx model's ability to integrate external variables into its forecasting framework makes it uniquely suited for financial time series analysis (Souto & Moradi, 2023a, 2023d). This feature is specially relevant in the context of stock market volatility forecasting, where external economic indicators and market sentiments play a pivotal role in shaping volatility trends.

Another key factor in choosing NBEATSx as a benchmark is its empirical performance. Recent studies have demonstrated the model's superior forecasting abilities in various domains (Han et al., 2023; Iftikhar et al., 2023; Marcjasz et al., 2023; Mathonsi & van Zyl, 2021; X. Wang et al., 2022), including realized volatility and correlation (Souto & Moradi, 2023a, 2023d).

3.3 Error Metrics

For an effective evaluation of the forecasting performance of the models, this study employs four error metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Quasi-Likelihood (QLIKE). Each metric provides a unique perspective on the models' accuracy and reliability.

3.3.1 RMSE

RMSE is a famous metric in regression analysis, quantifying the average magnitude of the prediction errors. The formula for RMSE is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{14}$$

where y_i represents the actual observation, $\hat{y}_i$ is the corresponding prediction, and n denotes the total number of trading days in the testing sample. RMSE is particularly sensitive to large errors, thereby emphasizing significant discrepancies in the model's predictions (Naser & Alavi, 2021).

3.3.2 MAE

MAE is another key measurement in assessing prediction accuracy. Its is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{15}$$

MAE is robust to outliers, as it treats all errors uniformly without giving undue weight to larger errors, thus offering a more balanced measure of average prediction error (Naser & Alavi, 2021).

3.3.3 MAPE

MAPE is a measure that expresses prediction accuracy as a percentage. It is given by:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{16}$$

MAPE is particularly beneficial for comparing the accuracy across different models or datasets, as it normalizes the errors, making it an excellent metric for scenarios where the target variable's magnitudes vary widely (Naser & Alavi, 2021).

3.3.4 QLIKE

QLIKE error metric is specially tailored for evaluating forecasts of quantities that are inherently positive, such as volatility. QLIKE is defined as follows:

$$\text{QLIKE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i}{\hat{y}_i} - \log\left(\frac{y_i}{\hat{y}_i}\right) - 1\right) \tag{17}$$

Here, y_i is the actual volatility, and $\hat{y}_i$ is the predicted volatility. QLIKE effectively penalizes both overestimations and underestimations, thereby being particularly relevant for the precise assessment of volatility forecasting models (Souto & Moradi, 2023c; Zhang et al., 2023).

Collectively, these error metrics offer a multifaceted evaluation of the models' predictive performance, providing distinct perspectives on the accuracy and characteristics of prediction errors.

3.4 Statistical Tests

For a rigorous comparison of the forecasting models, the study employs four statistical tests: the Model Confidence Set (MCS) (Hansen et al., 2011) and Diebold-Mariano (DM) tests (Diebold & Mariano, 1995) to evaluate the models' forecast performance as well as T-tests and F-tests to evaluate the neural network models' robustness to the change of random seed choice. These tests are instrumental in assessing the relative performance of the models and establishing statistical significance.

3.4.1 Model Confidence Set (MCS)

The Model Confidence Set (MCS) procedure, introduced by Hansen et al. (2011), is a statistical method designed to identify a subset of models that are statistically indistinguishable in terms of their predictive performance. The MCS allows for a rigorous comparison of multiple models by constructing a set of 'superior' models with a certain level of confidence.

Description of the MCS Procedure The MCS method involves several steps. Initially, all models under consideration are included in the confidence set. Then, through a process of sequential testing, models that are statistically significantly worse than the best-performing model are eliminated. The procedure continues until the final set of models cannot be differentiated in terms of their predictive accuracy at a chosen confidence level. The mathematical formulation and detailed implementation of the MCS test can be found in (Hansen et al., 2011).

Rationale Behind the Choice of MCS The rationale for utilizing the MCS procedure in this study is multifaceted:

1. **Comparative Model Evaluation**: MCS provides a robust framework for comparing multiple models simultaneously, which is essential in studies like ours where several models are being assessed.

2. **Statistical Rigor**: The MCS method offers a statistically rigorous approach to model comparison, ensuring that differences in performance are not due to random chance.

3. **Identification of Top Performers**: By identifying a set of models that are statistically similar in performance, MCS helps in pinpointing the models that are most effective in forecasting volatility.

Dynamic Application of MCS In this study, the MCS procedure is applied dynamically to assess how the models' forecast performance evolves over time. This dynamic approach involves estimating the MCS for each trading month in the testing sample, allowing for the observation of changes in model performance across different market conditions, gathering the MCS results of each model, which range from 0 and 1 where 1 means on broad terms that the model should be added to the set of the best models with 100% of confidence and 0 that the model should be added to the set of the

best models with 0% of confidence. The MCS results of each model is them plotted to analyze how the relative performance and even superiority of models change over time.

The reasons for this approach instead of the standard MCS are stated below:

1. **Temporal Variability in Model Performance**: The dynamic application of MCS is particularly relevant for financial time series, where model effectiveness can vary over time due to changing market conditions.

2. **Insights into Model Stability**: By assessing the MCS over different periods, we gain insights into the stability and consistency of each model's performance.

3. **Adaptability to Market Changes**: This approach helps in understanding the adaptability of models to different phases of market cycles, an important aspect in volatility forecasting.

The dynamic MCS test thus provides a comprehensive view of the models' performance, highlighting their effectiveness and stability over time, which is crucial in the context of stock market volatility prediction.

3.4.2 Diebold-Mariano (DM) Test

The Diebold-Mariano (DM) test, initially proposed by Diebold and Mariano (1995) and later improved by Harvey et al. (1997), is a statistical test widely used for comparing the predictive power of two forecasting models. This test is particularly suited for assessing whether there exists a statistically significant difference in forecasting performance between two models.

Description of the DM Test The DM test compares the forecast errors of two models to determine if one model has a statistically significantly lower forecasting error than the other. The null hypothesis of the DM test states that the two models have equal predictive accuracy. The test statistic is computed based on the differences in the forecast errors of the models over the evaluation period. The DM test accommodates various loss functions, allowing flexibility in how forecast accuracy is measured. The statistical formulation and implementation details of the DM test are thoroughly described in Diebold and Mariano (1995).

Rationale Behind the Choice of the DM Test The rationale for incorporating the DM test in this research includes:

1. **Focused Model Comparison**: The DM test allows for a direct, pairwise comparison of models, making it ideal for evaluating specific hypotheses about relative forecasting performance.

2. **Flexibility in Loss Function**: The ability of the DM test to work with different loss functions makes it versatile and applicable to various forecasting contexts and error metrics.

3. **Statistical Significance**: The DM test provides a formal statistical framework to assess the significance of differences in model performance, adding rigor to the model evaluation process.

Application in the Study In this study, the DM tests are performed for each stock separately. This individualized approach ensures that the specific characteristics and dynamics of each stock are taken into account when assessing model performance. The common threshold of 0.05 for the *p-values* is used to determine whether the two models have statistically significantly different predictive power or not. Then, the number of stocks where NHITS yields statistically significantly more accurate and less accurate are counted and graphically displayed for analysis. By applying the DM test to each stock, the study aims to:

1. **Capture Individual Stock Behaviors**: This method recognizes that different stocks may exhibit unique volatility patterns and responses to market conditions, which could influence model performance.

2. **Comprehensive Model Assessment**: Conducting separate DM tests for each stock allows for a more detailed and nuanced understanding of model effectiveness across the spectrum of stocks in the study.

3. **Enhanced Insights into Model Performance**: The stock-specific application of the DM test helps in identifying which models perform best for particular types of stocks or market conditions.

The use of the DM test thus contributes significantly to the robustness and depth of the model evaluation process, providing valuable insights into the relative forecasting abilities of the models across different stocks in the S&P 100 index.

3.4.3 T-tests and F-tests

Application of T-tests The T-test statistically compares the means of two groups. In Robustness Test 3, T-tests are utilized to determine whether the mean forecasting errors of the neural network models differ significantly when different random seeds are used during the training phase. A significant T-test result would indicate that the selection of random seed has a substantial impact on the model's forecasting accuracy, suggesting that on average the models' performance is statistically significantly different.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_{\bar{X}_1 - \bar{X}_2}} \tag{18}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the sample means of the forecasting errors from two sets of random seeds, and $s_{\bar{X}_1 - \bar{X}_2}$ is the standard error of the difference between the sample means.

Application of F-tests The F-test, on the other hand, is used to compare the variances of two populations and is especially useful for assessing the consistency of model performance across different initializations. In the context of this robustness test, F-tests are applied to evaluate whether there is a statistically significant discrepancy in the variability of forecasting errors between two models.

$$F = \frac{s_1^2}{s_2^2} \tag{19}$$

where s_1^2 and s_2^2 are the sample variances of the forecasting errors for models trained with two distinct sets of random seeds. A significant F-test result would imply that the choice of random seed affects not only the accuracy but also the consistency of the model's forecasts.

The combined use of T-tests and F-tests in Robustness Test 3 provides a dual perspective on model stability: the T-tests assess whether there is a statistically significant difference in the mean of the error measurements between NHITS and the other models, while the F-tests determine there is a statistically significant difference in the variance of the error measurements between NHITS and the other models.

3.5 Robustness Tests

Ensuring the robustness of the findings is crucial in empirical research, particularly in financial modeling. This study employs four robustness tests: multiple-step-ahead forecast, reducing the training sample size, performing a sensitivity analysis, varying data split ratios. The rationale behind each of these tests is discussed below.

3.5.1 Robustness Test 1

In Robustness Test 1, the forecasting models are evaluated based on their ability to predict market volatility over longer horizons, specifically at 5 and 10 forecast steps ahead. This test is designed to assess the models' long-term forecasting performance, with a particular focus on the hierarchical structure of NHITS which is hypothesized to provide superior predictions for such extended periods.

Description of the Robustness Test The test involves generating forecasts from each model for 5 and 10 steps ahead, as opposed to the single-step forecasts evaluated in previous tests. The error metrics used in earlier tests (RMSE, MAE, MAPE, QLIKE) are recalculated for these multi-step predictions to determine the models' performance over a longer-term. The NHITS model's performance is compared against that of NBEATSx, HAR, TFT, LSTM, and GARCH. Unfortunately, since this research employs the Python library neuralforecast of Nixtla (2023) for the neural network models, the use of the LSTM model for multiple-step-ahead forecast is not possible as the current version of the Python library does not handle this task. Nevertheless, given the poor results of the LSTM model for multiple-step-ahead forecast in the task of forecasting realized volatility (Souto & Moradi, 2023d), this is not a problem.

Rationale Behind the Robustness Test The rationale for Robustness Test 1 stems from the inner structure of the NHITS model, which is designed to capture hierarchical relationships within the data. Such relationships are particularly relevant when considering forecasts that extend beyond the immediate future. NHITS's architecture allows it to effectively integrate information across multiple time scales, potentially enabling it to outperform other models that might not capture these dynamics as effectively over longer horizons. This test, therefore, serves as a critical evaluation of NHITS's capability to leverage its structural advantages for long-term forecasting in the volatile domain of the stock market.

3.5.2 Robustness Test 2

The second robustness test in this research aims to evaluate the forecasting models' performance when trained with a reduced dataset. This test is essential to understand the models' efficiency and effectiveness under conditions of limited data availability.

Description of the Robustness Test In this test, the models are retrained using only 50% of the original training sample while keeping the testing sample size unchanged. This reduction is implemented by selecting the last half of the data points from the original training set. The chosen reduced sample is intended to simulate scenarios where less historical data is available for model training, a situation often encountered in practical financial analysis, particularly when dealing with newly listed stocks or less frequently traded assets (Souto et al., 2023).

Rationale Behind the Robustness Test The rationale for conducting Robustness Test 2 encompasses several key considerations:

1. **Data Efficiency**: By training the models on a smaller dataset, this test assesses their data efficiency. It is crucial to determine how well the models can utilize and generalize from a small amount of data, a common challenge in financial forecasting.

2. **Practical Applicability**: This test evaluates the models' applicability in real-world scenarios where the amount of available training data is often constrained. It is particularly relevant for stocks with shorter historical data or in rapidly evolving market segments.

3. **Model Robustness**: Understanding how model performance is impacted by the volume of training data provides insights into their robustness. A robust model should maintain reasonable performance levels even with reduced training data.

4. **Comparative Analysis**: This test allows for a comparative analysis of different models' abilities to adapt to reduced data availability, offering insights into their relative strengths and weaknesses in data-constrained environments.

Overall, Robustness Test 2 plays a critical role in assessing the models' capabilities in less-than-ideal conditions, reflecting the realities of varying data availability in

financial markets. The insights gleaned from this test are instrumental in gauging the practicality and resilience of the forecasting models in diverse operational contexts.

3.5.3 Robustness Test 3

The third robustness test in this research is designed to assess the stability and reliability of the forecasting neural network models under varying the randoms seed. This test involves conducting a sensitivity analysis based on different choices of random seeds used in the neural network model training process.

Description of the Robustness Test In this robustness test, each model is retrained 20 times with different random seed values. The choice of 20 times is motivated since when taking 20 samples of the error metrics, each error metric distribution always follows a Gaussian distribution (this is confirmed by the employment of the Anderson-Darling test). By varying the random seeds, the test introduces a range of initial weights to evaluate the neural network models' performance consistency. The selection of random seeds is done to cover a representative range, ensuring a comprehensive assessment of model stability.

Rationale Behind the Robustness Test The rationale for conducting Robustness Test 3 includes several important aspects:

1. **Model Stability**: This test probes the stability of the models against variations in initial training conditions. Stability is a critical attribute, indicating the models' reliability and predictability in practical applications.

2. **Reproducibility of Results**: Sensitivity analysis with different random seeds helps to verify the reproducibility of the models' results. In financial modeling, reproducibility is essential to ensure confidence in the models' forecasts.

3. **Comparative Analysis**: Comparing the performance variations across different models under changing random seeds provides insights into their relative robustness. This comparison helps in identifying models that consistently perform well, regardless of the initial conditions.

Overall, Robustness Test 3 is integral to establishing the forecasting models' credibility and effectiveness. It offers a thorough examination of how varying initial conditions impact model performance, contributing to a more nuanced understanding of each model's reliability in the dynamic landscape of financial market forecasting.

3.5.4 Robustness Test 4

The last robustness test in this study involves employing alternative data splitting strategies to assess the stability and reliability of the forecasting models. This test is pivotal in determining whether the models' performances are sensitive to changes in the training and testing dataset proportions.

Description of the Robustness Test In the primary analysis, the data is split into a 70%/30% training/testing ratio. Robustness Test 4 explores the effects of varying this split, specifically examining the 80%/20% split. This alternative split is chosen to reflect a wider range of potential scenarios in model training and evaluation. This approach emphasizes a more extensive training set, potentially allowing models to learn more comprehensive patterns from the data. The rationale here is to examine the models' ability to generalize from a richer training experience, albeit at the cost of a smaller testing set.

Rationale Behind the Robustness Test The rationale for conducting Robustness Test 4 is multi-fold:

1. **Assessing Model Sensitivity**: This test evaluates the sensitivity of the models to the amount of data available for training and testing. It is crucial to ascertain whether the models' forecasting abilities are robust to variations in the training/testing split.

2. **Generalizability**: By varying the data split ratios, the study can explore the generalizability of the models under different data availability scenarios. This is particularly important in financial time series forecasting, where the model's ability to adapt to different market conditions is key.

3. **Benchmark Comparison**: Comparing the performance of different models across various data splits provides a more comprehensive understanding of their relative strengths and weaknesses. This is valuable in assessing the practical applicability of these models in real-world scenarios.

Overall, Robustness Test 4 is designed to provide a deeper insight into the models' performance dynamics, offering a more nuanced understanding of their predictive capabilities in the context of stock market volatility forecasting.

3.6 Hyperparameter Search Space

The tables found in Appendix B show the hyperparameter search space used to determine the optimal parameters for each neural network model. For the validation stage, within the training sample, 28.5% of the data was reserved for the determination of optimal hyperparameters for the considered neural network models. A total of 40 trials are performed for each stock index to explore the hyperparameter search space, of which 20 trials are specially dedicated to exploring the random seed range. Finally, the optimal hyperparameters found after the hyperparameter search can be found in Appendix C.

4 Empirical Results

4.1 Main Sample

4.1.1 Error Metrics

Table 1 presents the error metrics results for the main sample.

Table 1: Error Metrics for Forecasting Models

Model	RMSE (%)	MAE (%)	MAPE (%)	QLIKE (%)
NHITS	0.401%	0.246%	18.412%	3.300%
GARCH	0.508%	0.337%	28.669%	4.992%
HAR	0.376%	0.248%	19.881%	3.248%
LSTM	0.420%	0.260%	20.077%	3.489%
TFT	0.406%	0.268%	21.755%	3.293%
NBEATSx	0.390%	0.241%	18.048%	3.112%

The RMSE values are lowest for the HAR model (0.376%), followed closely by NBEATSx (0.390%) and NHITS (0.401%). These values suggest that the HAR and NBEATSx models, in particular, are more effective at capturing the volatility patterns in the main sample without being heavily impacted by large individual errors. The GARCH model, with an RMSE of 0.508%, is, on the other hand, less adept at this, potentially indicating that it may not capture extreme market movements as effectively as the other models.

Similarly, the MAE values are also lowest for the HAR and NBEATSx models, with NHITS closely following. This suggests a consistent performance across these models in terms of handling average errors, with the HAR model slightly outperforming the others.

When considering the MAPE values, there is larger discrepancy with the GARCH model registering the highest error at 28.669%. The lower MAPE values for NHITS (18.412%), HAR (19.881%), and NBEATSx (18.048%) indicate that these models are relatively better at dealing with the scale of the data and are less affected by the magnitude of the actual values, which is particularly important when dealing with financial data that can vary widely in scale.

The lowest QLIKE value is observed for NBEATSx (3.112%), suggesting that it provides the best balance between underestimating and overestimating volatility. This is closely followed by HAR and TFT, indicating their relevance in volatility forecasting. NHITS, on the other hand, had a poor performance regarding this error measure.

In conclusion, the performance of the forecasting models across different error metrics suggests that HAR and NBEATSx have a slight edge, with NHITS also showing promising results for potentially being a novel benchmark model that could replace the commonly used benchmark models (i.e., GARCH and LSTM) and newer benchmark models (e.g., TFT). Lastly, for industry use, the HAR and NBEATSx models would be preferred over the NHITS model given their better performance in the main sample.

4.1.2 Diebold-Mariano Tests

Figure 3 shows the DM tests results for RMSE in the main sample

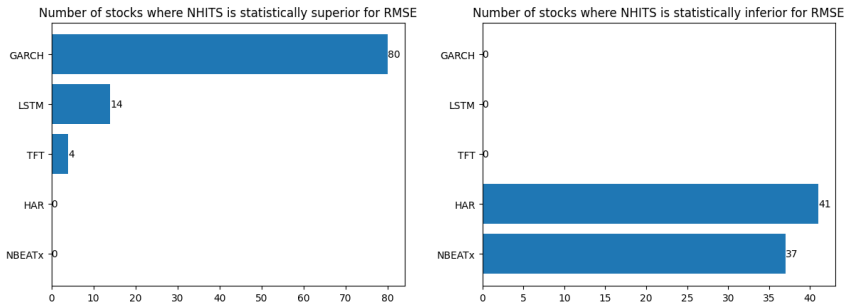

Figure 3: DM Tests Results for RMSE (Main Sample)

A striking outcome is observed in the comparison between NHITS and NBEATSx. The DM test indicates that NHITS did not yield statistically significant better forecasts than NBEATSx for any of the stocks. Conversely, NBEATSx yields statistically significant better forecasts than NHITS for 37 out of 80 stocks. This suggests a substantial advantage for NBEATSx over NHITS in terms of RMSE across a significant portion of the stocks considered.

Similar to the NBEATSx comparison, NHITS does not outperform HAR in terms of statistically significant better forecasts for any stock. In contrast, HAR outperforms NHITS for 41 stocks, further emphasizing the relative strength of the HAR model in forecasting stock market volatility as measured by RMSE.

In contrast to the comparisons with NBEATSx and HAR, NHITS shows better performance than TFT for 4 stocks, while TFT does not outperform NHITS for any stock. This result indicates that NHITS has some advantages over TFT, albeit limited to a small subset of stocks.

The comparison with LSTM reveals a notable advantage for NHITS, as it yields statistically significant better forecasts than LSTM for 14 stocks, with LSTM not outperforming NHITS for any stock. This outcome highlights the effectiveness of NHITS over LSTM in certain cases.

As already expected, the most pronounced result comes from the comparison with GARCH. NHITS yields statistically significant better forecasts than GARCH for all 80 stocks in the sample, demonstrating a clear and comprehensive superiority over the traditional GARCH model in terms of RMSE.

These results collectively suggest that while NHITS has clear advantages over LSTM and GARCH models and a bit of superiority over the TFT model, it falls short when compared to NBEATSx and HAR models. This confirms the conclusion of the error metrics results, which suggests that NHITS could be used a novel and better benchmark model in the realized volatility literature, yet not in the industry as the main or sole model.

Figure 4 shows the DM tests results for MAE in the main sample

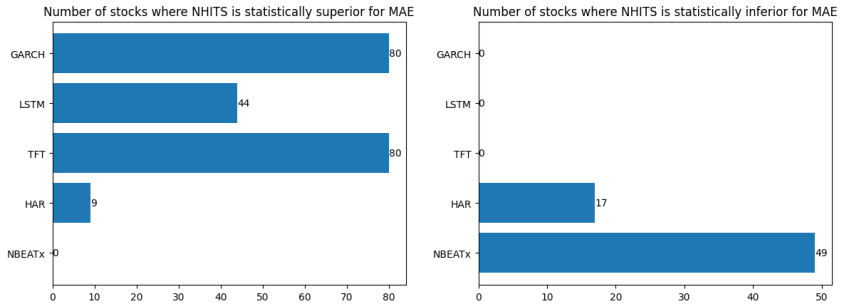

Figure 4: DM Tests Results for MAE (Main Sample)

Anew, NHITS does not outperform NBEATSx in terms of statistically significant better forecasts for any stock when evaluating MAE. On the other hand, NBEATSx demonstrates a clear advantage, with statistically significant better forecasts than NHITS for 49 stocks. This disparity underscores the superior performance of NBEATSx.

The comparison between NHITS and HAR presents a more balanced outcome. NHITS yields statistically significant better forecasts than HAR for 9 stocks, while HAR outperforms NHITS for 17 stocks. This indicates that while HAR has an edge in certain cases, NHITS remains competitive in its forecasting capability for a number of stocks.

In a stark contrast, NHITS significantly outperforms TFT in all 80 stocks, as per the DM test results for MAE. The results also reveal a notable advantage for NHITS over LSTM, with NHITS providing statistically significant better forecasts for 44 stocks. LSTM does not show any superiority over NHITS in terms of MAE, highlighting the effectiveness of NHITS in this context. Similarly, NHITS significantly outperforms the GARCH model for all 80 stocks.

The DM test results for MAE paint a picture of NHITS as a highly competent model, particularly in comparison to TFT, LSTM, and GARCH, where it consistently demonstrates superior forecasting accuracy. However, NBEATSx appears to be a stronger model in most cases, and HAR shows mixed results.

Figure 5 shows the DM tests results for QLIKE in the main sample

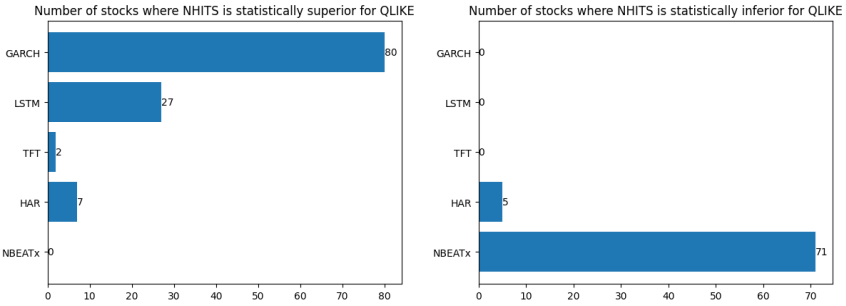

Figure 5: DM Tests Results for QLIKE (Main Sample)

Again, NHITS does not exhibit statistically significant better forecasts compared to NBEATSx for any of the stocks. Conversely, NBEATSx shows statistically significant better forecasts than NHITS for a considerable number of stocks (71 out of 80), indicating a strong advantage of the NBEATSx model in terms of QLIKE across the majority of the stocks.

The comparison between NHITS and HAR yields a more nuanced result. NHITS outperforms HAR in terms of statistically significant better forecasts for 7 stocks, whereas HAR is superior for 5 stocks. This suggests a relatively balanced performance between the two models when assessed using the QLIKE metric.

NHITS shows a marginally better performance than TFT, with statistically significant better forecasts for 2 stocks, while TFT does not outperform NHITS for any stock. On the other hand, NHITS demonstrates a noticeable advantage over LSTM, providing statistically significant better forecasts for 27 stocks. LSTM does not achieve better forecasts than NHITS for any stock. Anew, NHITS significantly outperforms GARCH for all 80 stocks.

The DM test results for QLIKE reveal that while NHITS has considerable advantages over LSTM and GARCH models, it generally underperforms compared to NBEATSx. The comparison with HAR and TFT shows mixed results, indicating areas where NHITS either slightly outperforms or is on par with these models.

Figure 6 shows the DM tests results for MAPE in the main sample

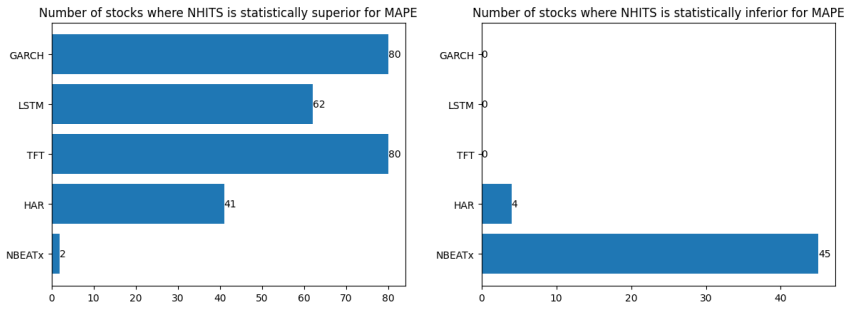

Figure 6: DM Tests Results for MAPE (Main Sample)

NHITS shows a limited advantage over NBEATSx, with statistically significant better forecasts for only 2 stocks. In contrast, NBEATSx outperforms NHITS significantly in 45 stocks, indicating a more consistent and accurate performance by NBEATSx in terms of percentage errors.

The comparison between NHITS and HAR reveals a substantial advantage for NHITS, with significantly better forecasts for 41 stocks. Conversely, HAR shows better performance than NHITS for only 4 stocks. This result suggests that NHITS is generally more accurate than HAR in terms of MAPE across a wide range of stocks.

In the case of TFT and GARCH, NHITS demonstrates a clear superiority, providing statistically significant better forecasts for all 80 stocks in the sample. Against LSTM, NHITS again shows strong performance, yielding significantly better forecasts for 62 stocks, while LSTM does not outperform NHITS for any stock

The DM test results for MAPE demonstrate the strengths of NHITS in providing accurate percentage error forecasts, particularly when compared to TFT, LSTM, and GARCH models. While NHITS shows some limitations against NBEATSx, its overall performance is commendable, especially considering its substantial advantage over HAR.

The findings of the DM tests demonstrate that NHITS ought to be a novel benchmark model used in the realized volatility literature that could replace the commonly used benchmark models (i.e., GARCH and LSTM) and newer benchmark models (e.g., TFT). However, for industry use, the HAR model and especially the NBEATSx model would be preferred over the NHITS model.

4.1.3 Dynamic Model Confidence Set

Figure 7 shows the Dynamic MCS results for all error metrics in the main sample.

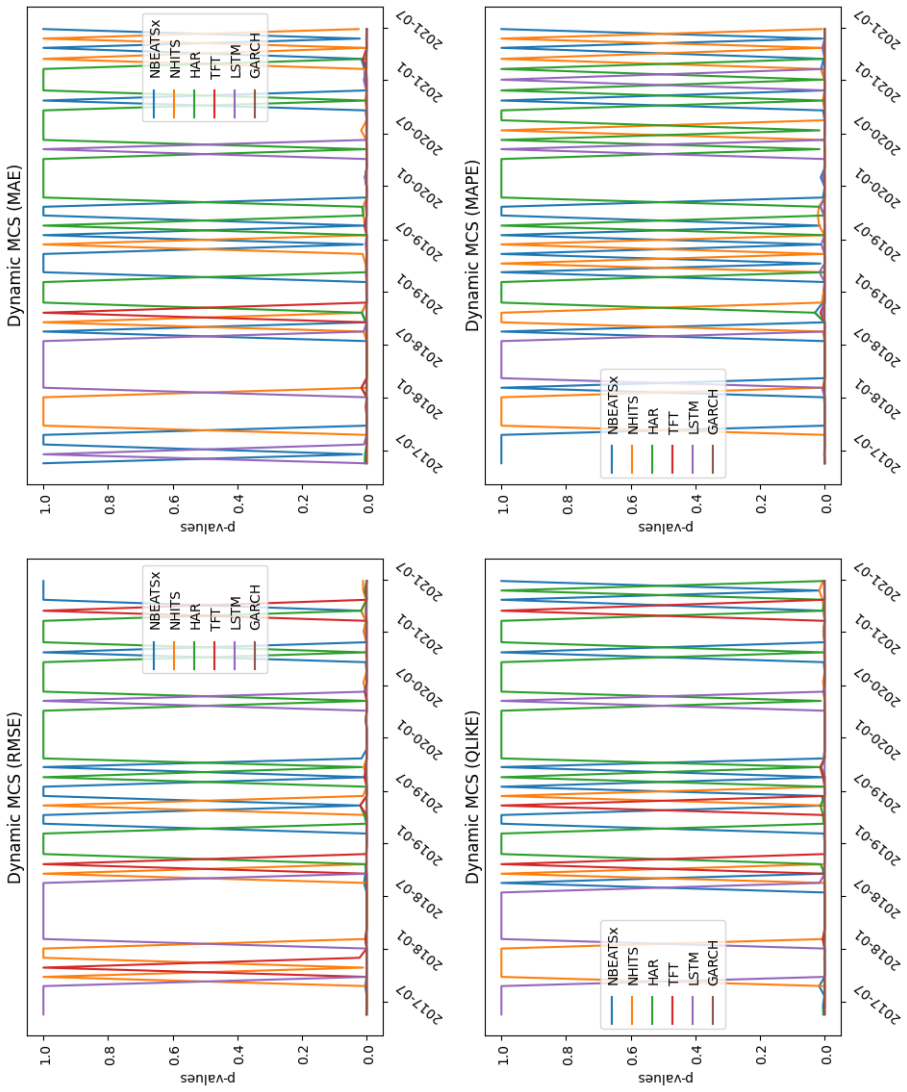

Figure 7: MCS Results (Main Sample)

Interestingly, the LSTM model is the best model in the period from January 2018 until approximately July 2018, while the HAR model is the top model during the first corona crisis in the first semester of 2020, showing its robustness during turbulent periods. Even though NBEATSx is the best model in the main sample considering all error metrics together, it is not very often the best model concerning the Dynamic MCS results. This shows that NBEATSx performance is considerably stable over time, while the other models's performance can vary over time (sometimes even significantly such as the LSTM model). Finally, NHITS is at times the best model, specially considering the MAPE metric, yet there is no considerable periods of dominance of the NHITS model like there is with the LSTM and HAR models.

4.1.4 Conclusion

In conclusion, it can be affirmed that although NHITS showed some clear promising results in the main sample in becoming an important benchmark model in the realized volatility literature and possibly replacing the GARCH, LSTM and TFT models, it fails to surpass HAR and NBEATSx forecast performance. As a result, the use of NHITS as the main or sole model in the industry for one-step-ahead forecasting would not be recommended

4.2 Robustness Test 1

4.2.1 Error Metrics

The results from Robustness Test 1 (H=5), which examines the models' forecasting performance over a horizon of 5 steps, provide insight into the models' long-term predictive capabilities. Table 2 presents the error metrics results for this robustness test.

Table 2: Error Metrics for Forecasting Models

Model	RMSE	MAE	MAPE	QLIKE
NHITS	0.480%	0.290%	21.546%	4.569%
GARCH	0.548%	0.363%	30.754%	5.759%
HAR	0.503%	0.315%	24.814%	5.049%
TFT	0.485%	0.300%	23.047%	4.397%
NBEATSx	0.476%	0.286%	21.192%	4.507%

In terms of RMSE, NHITS performs quite well with a score of 0.480%, indicating a high level of accuracy in its predictions for a 5-step horizon. This success can be attributed to NHITS' architecture, which is specifically designed to capture hierarchical relationships within the data for long-term forecasting. As expected, GARCH lags behind with an RMSE of 0.548%. In the middle ground, HAR and TFT exhibit intermediate RMSE values of 0.503% and 0.485%, respectively. While they show some predictive capability over multiple steps, they may not fully capture the patterns that become more pronounced over longer horizons. Similar to the findings of Souto and Moradi (2023d), NBEATSx stands out with the lowest RMSE of 0.476%, implying

that it might be slightly better at forecasting volatility over a 5-step horizon compared to NHITS.

Moving on to MAE, NHITS has an MAE of 0.290%, which, although a bit higher than NBEATSx's MAE of 0.286%, still reflects a relatively accurate forecast. In contrast, GARCH, HAR and TFT have MAE values suggesting they are less appropriate for long-term forecasting.

When considering MAPE, NHITS achieves a MAPE of 21.546%, indicating that the percentage errors in its forecasts are slightly higher than NBEATSx's but lower than TFT, HAR and GARCH. Finally, when looking at QLIKE, NHITS and NBEATSx have comparable values of 4.569% and 4.507%, respectively, showing that both models are effective in forecasting volatility for one business week ahead. Anew, GARCH and HAR have MAE values suggesting they are less appropriate for long-term forecasting. In contrast, TFT surprisingly achieves the lowest QLIKE value.

The results from Robustness Test 4 for H=5 suggest that while NHITS is effective for long-term forecasting, it faces strong competition from NBEATSx, which slightly outperforms NHITS across several metrics. However, statistical tests are still needed to determine whether the superiority of NBEATSx is statistically significant or due to randomness. On the other hand, GARCH, HAR and TFT seem less suited for longer-term forecasts. Possible reasons for NHITS's performance could include its hierarchical structure, which is designed for multi-scale data but may not always capture the nuances that become important over a longer forecasting horizon. Meanwhile, NBEATSx's slight edge could stem from its flexible architecture that might be better at handling the complexities of predicting further into the future.

The results from Robustness Test 4 (H=10) provide a detailed picture of the models' performance when forecasting 10 steps ahead. These longer-term forecasts test the models' ability to capture and predict the underlying trends and patterns in the data over an extended period. Table 3 presents the error metrics results for this robustness test.

Table 3: Error Metrics for Forecasting Models

Model	RMSE	MAE	MAPE	QLIKE
NHITS	0.563%	0.327%	23.499%	6.017%
GARCH	0.679%	0.461%	40.835%	9.346%
HAR	0.638%	0.414%	34.248%	8.829%
TFT	0.592%	0.363%	28.020%	5.764%
NBEATSx	0.564%	0.329%	23.713%	5.936%

Concerning RMSE, NHITS records an RMSE of 0.563%, making NHITS the best model in this robustness test and indicating a good understanding of volatility trends in the data for an extended forecast horizon. Besides NBEATSx, NHITS considerably outperforms the other models. Switching to MAE, NHITS anew displays the best result with a value of 0.327%, indicating that for such long-term forecasting tasks, NHITS is presumably the best choice, albeit NBEATSx's result is quite close to NHITS's result. Examining MAPE, NHITS shows a MAPE of 23.499%, indicating a moderate relative error size, which can be crucial for investors interested in percentage accuracy of

forecasts. Again, HITS considerably outperforms the other models besides NBEATSx. Finally, regarding QLIKE, NHITS records a QLIKE of 6.017%, which is higher than the results of TFT and NBEATSx. This indicates that considering QLIKE, NHITS is possibly not the best choice for long-term forecasting tasks similar to the one of Robustness Test 4 (H=10).

The relatively strong performance of NHITS and NBEATSx can be attributed to their neural network architectures, which are well-suited to capturing complex temporal dependencies in the data. Their ability to maintain forecasting performance over an extended horizon suggests an effective capture of long-term patterns. As a result, it can be concluded that for long-term forecasting, NHITS is as powerful as NBEATSx, and thus could be used in the industry for such a forecasting task.

4.2.2 Diebold-Mariano Tests

Figure 8 shows the DM tests results for RMSE in the robustness test 1 (H=5). Similar to the main sample, NHITS is considerably better than the GARCH model and worse than NBEATSx. However, NHITS now yields statistically significantly more accurate forecasts than the HAR model, and the slight superiority of NHITS over TFT is not anymore present.

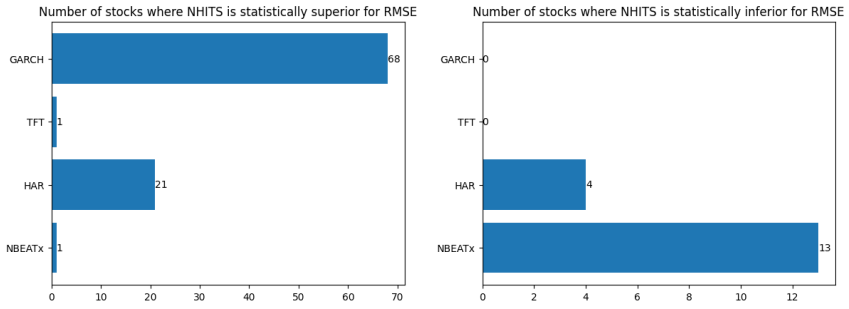

Figure 8: DM Tests Results for RMSE (Robustness Test 1 H=5)

Figure 9 depicts the DM tests results for MAE in the robustness test 1 (H=5). Like in the main sample, NHITS is statistically superior than the GARCH and TFT models, yet this is the opposite in comparison to NBEATSx. Something that is new is the superiority of NHITS over the HAR model.

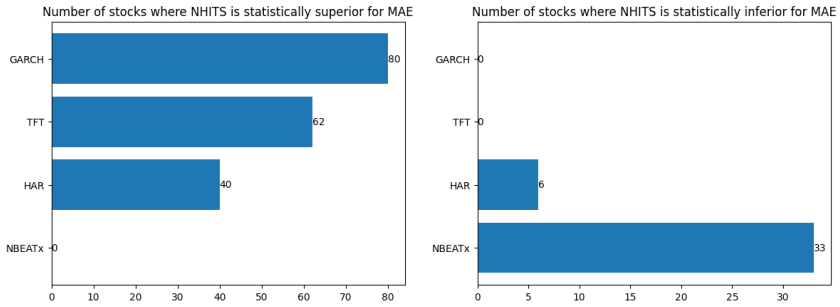

Figure 9: DM Tests Results for MAE (Robustness Test 1 H=5)

Figure 10 presents the DM tests results for QLIKE in the robustness test 1 (H=5). Resembling the main sample, there is a clear superiority of NHITS over GARCH and of NBEATSx over NHITS. Nonetheless, different to the main sample, NHITS yields statistically significantly more precise forecasts than the HAR model and statistically significantly less precise forecasts than the TFT model.

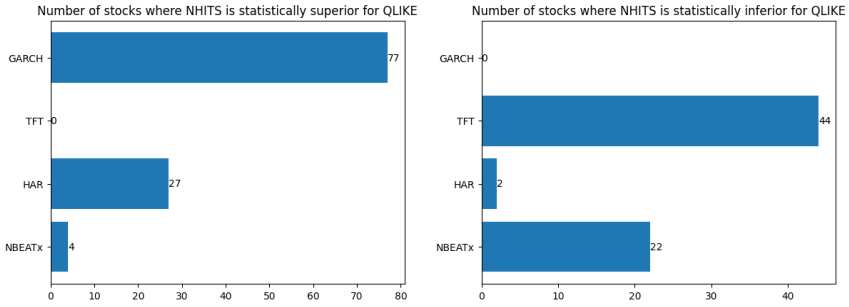

Figure 10: DM Tests Results for QLIKE (Robustness Test 1 H=5)

Figure 11 shows the DM tests results for MAPE in the robustness test 1 (H=5). Similar to the main sample, there is a clear superiority of NHITS over the GARCH, TFT and HAR models, and there is a clear superiority of NBEATSx over the NHITS model.

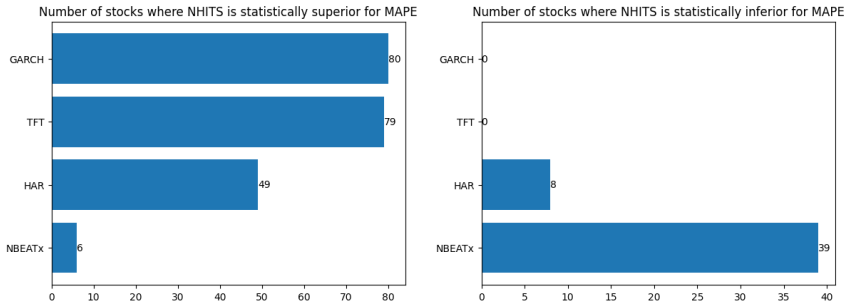

Figure 11: DM Tests Results for MAPE (Robustness Test 1 H=5)

Given the DM tests results for H=5, it can be concluded that for long-term forecasting (i.e., five-steps-ahead forecasting), NHITS is a powerful model and ought to start being used in the realized volatility literature as a benchmark model. Nevertheless, NBEATSx is a better model than NHITS and thus should be preferred in the industry for such a forecasting task.

Figure 12 illustrates the DM tests results for RMSE in the robustness test 1 (H=10). It can be seen that NHITS is this time statistically superior to the other considered models, albeit this superiority over the NBEATSx model is not as clear as for the other models.

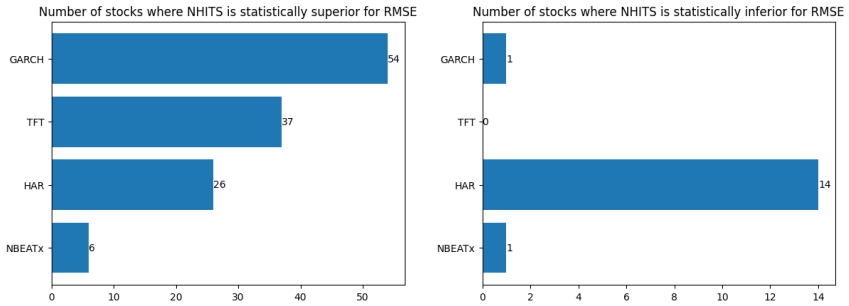

Figure 12: DM Tests Results for RMSE (Robustness Test 1 H=10)

Figure 13 presents the DM tests results for MAE in the robustness test 1 (H=10). Similar to the DM tests results for RMSE, NHITS yields statistically significantly more accurate forecasts than all the other models, and now the superiority over the NBEATSx model is clear.

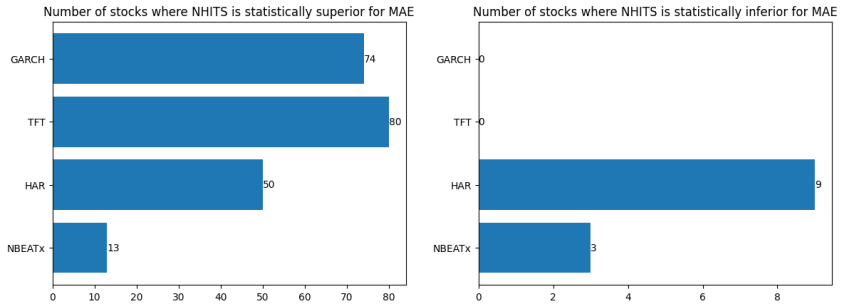

Figure 13: DM Tests Results for MAE (Robustness Test 1 H=10)

Figure 14 presents the DM tests results for QLIKE in the robustness test 1 (H=10). Resembling the DM tests results for QLIKE in the robustness test (H=5), NHITS is superior to the GARCH and HAR models and inferior to TFT and NBEATSx models.

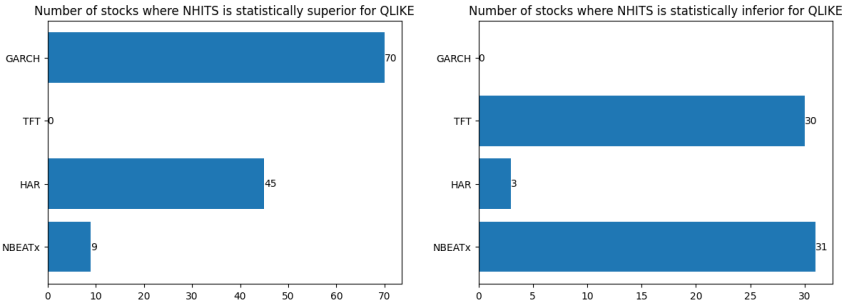

Figure 14: DM Tests Results for QLIKE (Robustness Test 1 H=10)

Figure 15 presents the DM tests results for MAPE in the robustness test 1 (H=10). Similar to the DM tests results for MAPE in the robustness test (H=5), NHITS is clealry superior to the GARCH, HAR and TFT models and is now superior to the NBEATSx model.

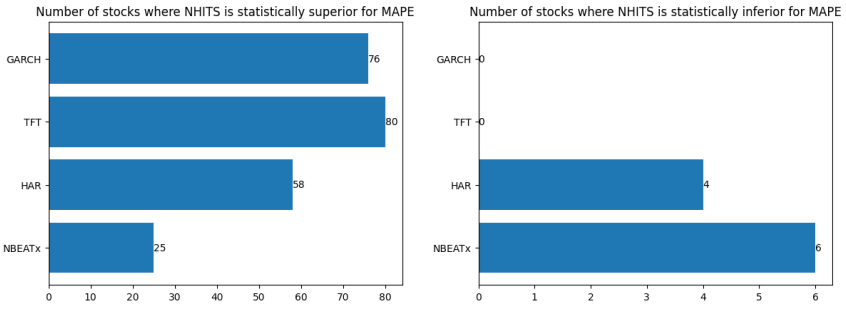

Figure 15: DM Tests Results for MAPE (Robustness Test 1 H=10)

Given the results above, it can be affirmed that for long-term forecasting (i.e., ten-steps-ahead forecasting), NHITS is not only a powerful model that should be used in the realized volatility literature as a benchmark model, but it is also a model that ought to be preferred in the industry for such a forecasting task.

4.2.3 Dynamic Model Confidence Set

Figure 16 and Figure 17 present the Dynamic MCS results for all error metrics in the robustness test 1. Two striking results can be seen. The first one is that the model that is occupying the best model position changes much more often than in the main sample, showing that as the number of forecasted steps ahead increases, the dynamic relative forecast performance differences among the models become less clear. The second one is that NHITS is more regularly the best model, specially for H=10, which agrees with the error metrics and DM results in the robustness test 1.

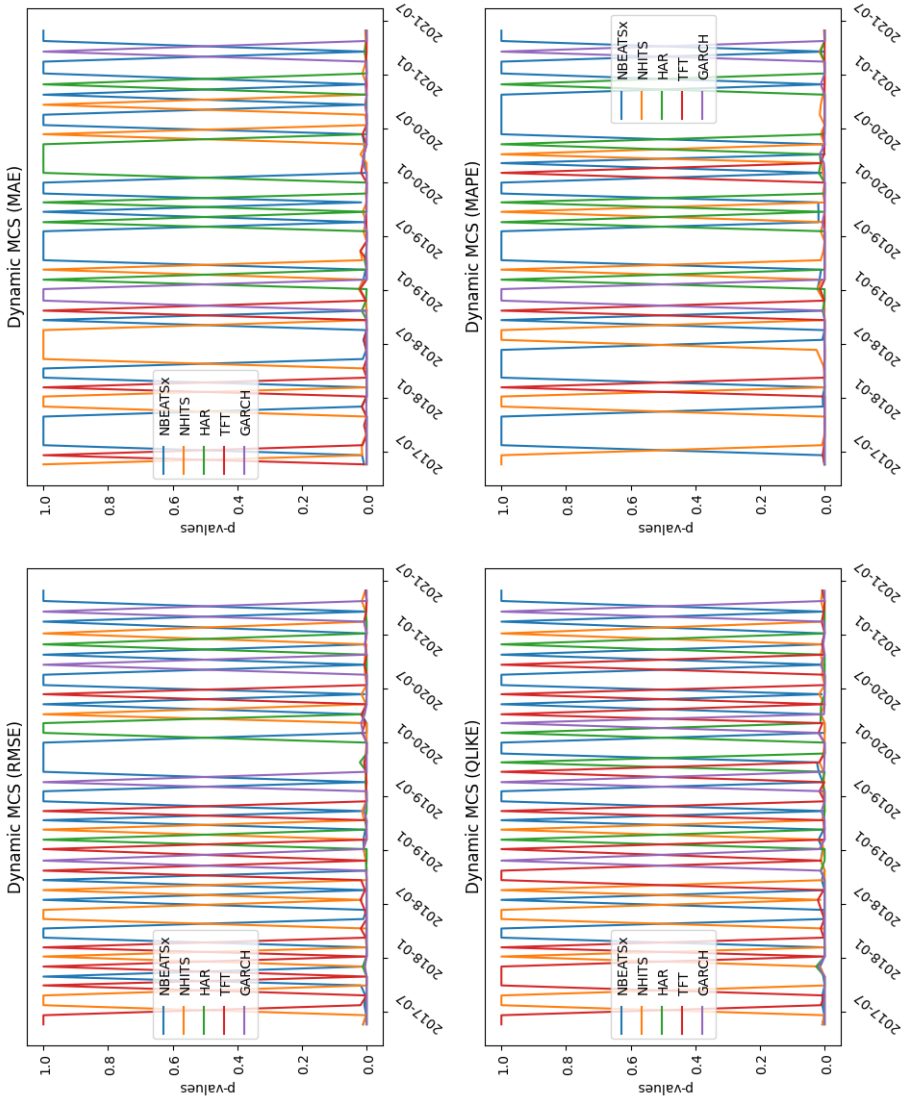

Figure 16: MCS Results (Robustness Test 1 H=5)

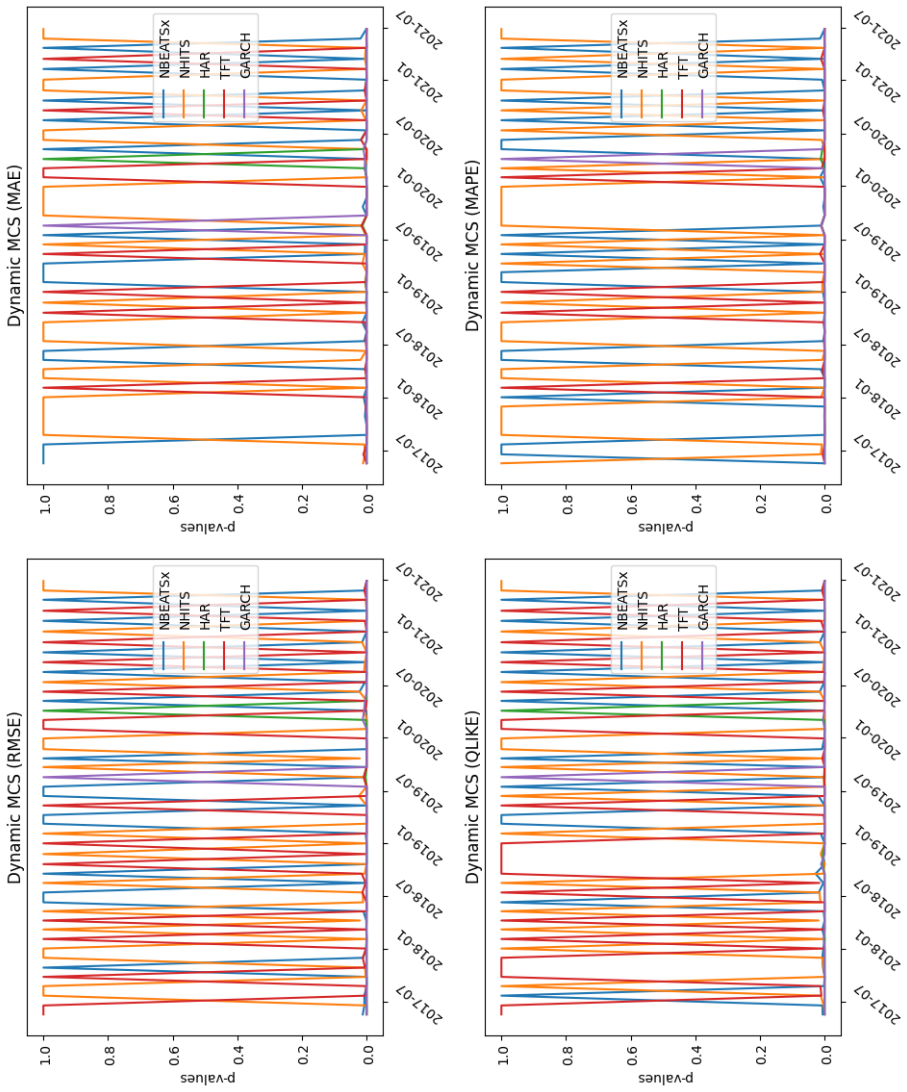

Figure 17: MCS Results (Robustness Test 1 H=10)

4.2.4 Conclusion

In summary, when H=5, it is evident that NHITS emerges as a robust model, making it a suitable candidate for adoption as a benchmark model in realized volatility research. However, NBEATSx outperforms NHITS, making it the preferred choice for forecasting tasks in the industry. On the other hand, when H=10, NHITS not only proves its strength as a benchmark model for realized volatility research but also establishes itself as the top choice for industry forecasting tasks.

4.3 Robustness Test 2

4.3.1 Error Metrics

Table 4 shows the error metrics results for the robustness test 2. These results offer a distinct perspective on the models' performance. In this test, the models were trained with less data to evaluate their performance in situations where historical data is limited.

Table 4: Error Metrics for Forecasting Models

Model	RMSE	MAE	MAPE	QLIKE
NHITS	0.413%	0.250%	18.511%	3.426%
GARCH	0.500%	0.316%	26.161%	4.844%
HAR	0.393%	0.250%	19.582%	3.377%
LSTM	0.580%	0.295%	19.337%	5.398%
TFT	0.397%	0.246%	18.707%	3.114%
NBEATSx	0.399%	0.244%	18.180%	3.207%

Regarding RMSE, in Table 4, NHITS was the fourth-best model, behind the HAR, TFT, and NBEATSx models. This is a slight drop in performance relative to Table 1 from the main sample, where NHITS was the third-best model, outperforming TFT and closely following HAR and NBEATSx. In terms of MAE, NHITS dropped from the second-best model position to the third-best model, performing better than GARCH and LSTM, as well as HAR, but worse than TFT and NBEATSx.

For MAPE, NHITS maintained its relative performance of the main sample, remaining the second-best model. In both the main sample and robustness test 2, NHITS was surpassed by NBEATSx but showed better performance than the other models. With respect to QLIKE, NHITS was ranked fourth in Table 4, like in Table 1. In both tables, NHITS performed better than GARCH and LSTM but was outperformed by HAR, TFT, and NBEATSx.

Overall, while NHITS's performance in Table 4 shows some variations compared to the main sample, it consistently remains in the middle tier among the models analyzed. Notably, NHITS's performance relative to the other models shows that it has maintained a competitive position, particularly in MAE and MAPE, despite changes in the training dataset size. However, its relative decline in RMSE, MAE and QLIKE rankings suggests that NHITS may be sensitive to the volume of data available for

training, making it not suitable for contexts wherein historical data is limited. However, statistical tests are still needed to confirm the aforementioned results.

4.3.2 Diebold-Mariano Tests

Figures 18, 19, 20, and 21 depict the DM tests results for all error metrics in the robustness test 2. It is clear that besides MAPE, where NHITS is statistically superior to all models apart from NBEATSx, NHITS yields statistically significantly less accurate forecasts than TFT, HAR and NBEATSx when confronted with limited historical data. This shows that NHITS is likely sensitive to the volume of data available for training, making it not appropriate for situation wherein historical data is limited.

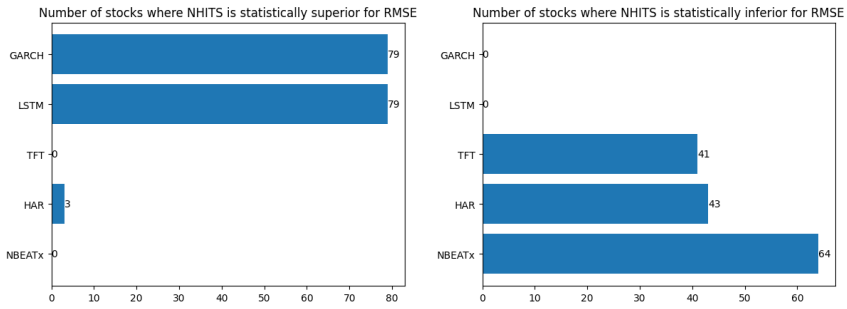

Figure 18: DM Tests Results for RMSE (Robustness Test 2)

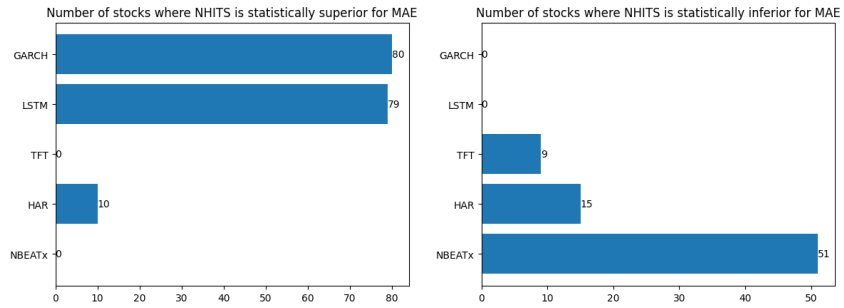

Figure 19: DM Tests Results for MAE (Robustness Test 2)

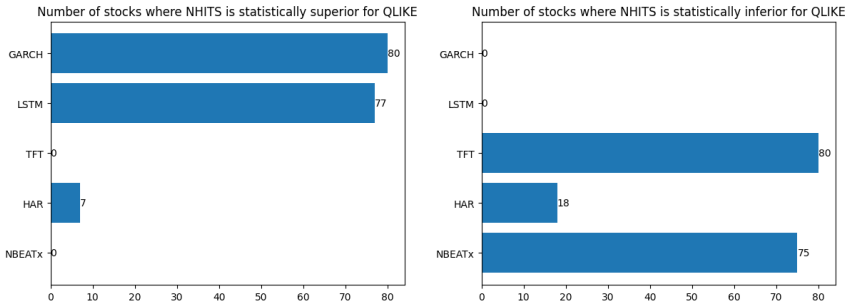

Figure 20: DM Tests Results for QLIKE (Robustness Test 2)

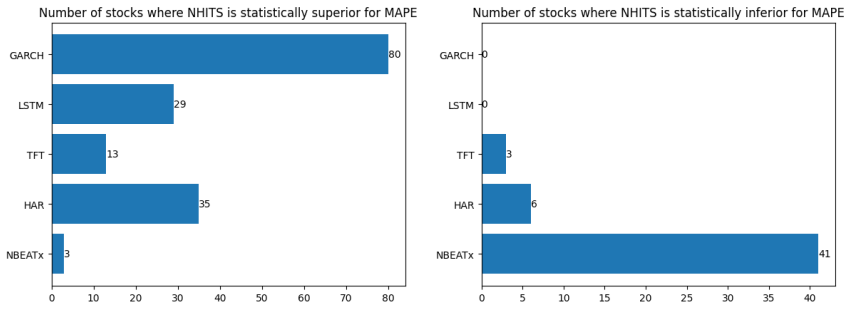

Figure 21: DM Tests Results for MAPE (Robustness Test 2)

4.3.3 Dynamic Model Confidence Set

Figure 22 shows the Dynamic MCS results for all error metrics in the robustness test 2. It can be seen that for all error measures apart from MAPE, NHITS is less often in the position of the best model, further proving the conclusions from the error metrics and DM tests results. It is worth mentioning that the HAR model is again the best model for the the best model during the first corona crisis in the first semester of 2020, showing its robustness during turbulent periods.

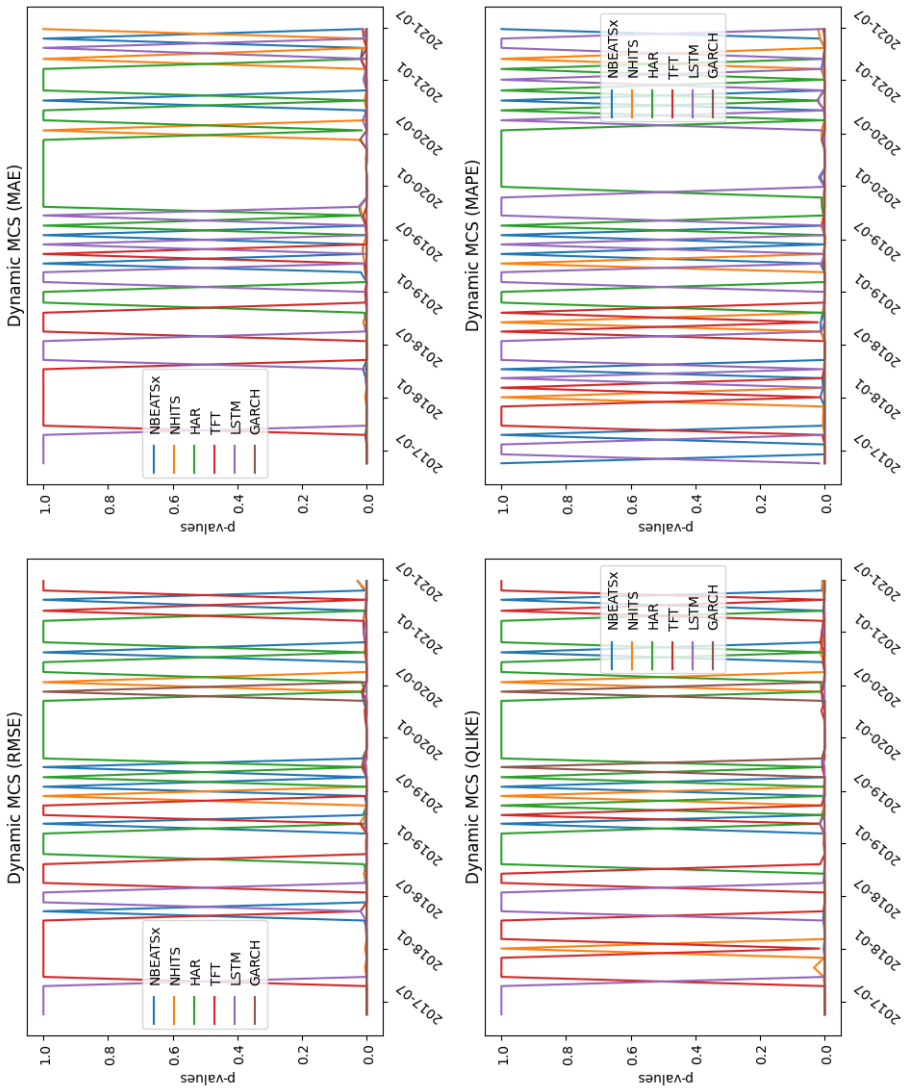

Figure 22: MCS Results (Robustness Test 2)

4.3.4 Conclusion

Given the results of robustness test 2, it can be concluded that NHITS is likely sensitive to the volume of data available for training, making it not appropriate for situation wherein historical data is limited. However, even in the case of limited historical data available, NHITS is still a better benchmark model to be used instead of the GARCH and/or LSTM models. Lastly, in such a situation, NBEATSx is presumably the best model to choose form

4.4 Robustness Test 3

4.4.1 Error Metrics

Table 5 shows the mean and standard deviation of the error metrics for the robustness test 3. Robustness test 3 provides an assessment of the forecasting models' sensitivity to different choices of random seed in the training process. This test is crucial for understanding the stability and consistency of model performance.

Table 5: Sensitivity Analysis Results for Error Metrics

Model	Metric	Mean	Std Dev
NHITS	RMSE	0.402%	0.004%
	QLIKE	3.284%	0.057%
	MAE	0.247%	0.001%
	MAPE	18.519%	0.283%
NBEATSx	RMSE	0.390%	0.002%
	QLIKE	3.127%	0.025%
	MAE	0.241%	0.001%
	MAPE	18.058%	0.126%
TFT	RMSE	0.397%	0.005%
	QLIKE	3.123%	0.101%
	MAE	0.252%	0.008%
	MAPE	19.721%	1.024%
LSTM	RMSE	0.431%	0.019%
	QLIKE	3.662%	0.312%
	MAE	0.261%	0.009%
	MAPE	19.631%	0.686%

When considering the RMSE metric, NHITS exhibits a mean RMSE of 0.402% with a low standard deviation (Std Dev) of 0.004%, indicating a consistent performance across different initializations. NBEATSx, on the other hand, outperforms other models with the lowest mean RMSE of 0.390% and a minimal Std Dev of 0.002%, showcasing strong stability. TFT's RMSE mean is comparable to NHITS and NBEATSx, but it has a higher Std Dev of 0.005%, suggesting more variability in performance. In contrast, LSTM shows the highest mean RMSE at 0.431% and the highest Std Dev of 0.019%, indicating substantial performance fluctuations across different seeds, which suggests lack of robustness.

In terms of MAE, NHITS and NBEATSx both have relatively low mean values at 0.247% and 0.241%, respectively, with very tight Std Devs (0.001% for NHITS and 0.001% for NBEATSx), indicating their resilience to changes in the random seed choice. TFT's MAE mean is slightly higher at 0.252%, and its Std Dev of 0.008% indicates variability in average error. LSTM, on the other hand, has the highest MAE mean at 0.261% and a significant Std Dev of 0.019%, confirming the variability in its performance across different seeds.

When analyzing the QLIKE metric, NHITS exhibits a mean of 3.284% with a Std Dev of 0.057%, suggesting stable performance in forecasting volatility. NBEATSx also demonstrates strong performance with the lowest QLIKE mean of 3.127% and a minimal Std Dev of 0.025%. TFT, while having the best mean, shows a remarkly higher Std Dev of 0.101% than NHITS and NBEATSx, indicating more variability in its ability to capture volatility. In contrast, LSTM has the highest QLIKE mean at 3.662% and a large Std Dev of 0.312%, indicating potential inconsistencies due to lack of robustness.

Finally, considering MAPE, NHITS has a mean MAPE of 18.519% with a Std Dev of 0.283%, demonstrating consistency. NBEATSx also maintains a low MAPE mean of 18.058% and low Std Dev, emphasizing its consistent performance. TFT's MAPE mean is slightly higher at 19.721% than NHITS and NBEATSx, with a Std Dev of 1.024%, suggesting a considerably greater sensitivity to random seed choice than NHITS and NBEATSx. LSTM anew exhibits the highest MAPE mean at 19.631% and higher associated Std Devs, further confirming the variability in its performance.

In conclusion, NHITS and NBEATSx demonstrate stability across different initializations, suggesting a robust architecture less affected by the randomness in the training process. Robustness test 3 highlights the importance of stability in forecasting models, with NHITS and NBEATSx outperforming TFT and LSTM in terms of consistency and reliability under varying random seed choice. This robustness is particularly important in financial forecasting, where the ability to perform reliably across different market conditions and data samples is valued, and practitioners usually do not possess a great amount of time to perform numerous trials for the optimal random seed search stage.

4.4.2 Statistical Tests

Table 6 shows the T-tests and F-tests results for the robustness test 3. The results from robustness test 3, using statistical tests to compare the NHITS model with NBEATSx, TFT, and LSTM, give us valuable insights into each model's performance. In the context of these results, the T-Statistics and F-Statistics provide measures of the statistical significance of the differences in performance between models. Asterisks denote the level of significance, with three asterisks (*) indicating a significance level of 0.10, two asterisks (**) indicating a significance level of 0.05, and three asterisks (***) indicating a significance level of 0.01.

Table 6: Statistical Tests Results

Model in comparison with NHITS	Metric	T-Statistics	F-Statistics
NBEATSx	RMSE	-12.34***	0.29
	QLIKE	-11.28***	0.19
	MAE	-17.92***	0.61
	MAPE	-6.64***	0.19
TFT	RMSE	-4.11***	1.47
	QLIKE	-6.22***	3.15***
	MAE	3.02***	39.72***
	MAPE	5.05***	13.09***
LSTM	RMSE	6.77***	25.35***
	QLIKE	5.32***	29.85***
	MAE	6.79***	59.27***
	MAPE	6.70***	5.87***

Across all metrics (RMSE, QLIKE, MAE, and MAPE), NHITS is outperformed by NBEATSx, as indicated by the negative T-Statistics, which are highly significant. The magnitude of these statistics suggests a robust and consistent outperformance of NBEATSx over NHITS. The F-Statistics for the comparisons between NHITS and NBEATSx are relatively low, and the null-hypothesis is not rejected for any error measure. This indicates that the variances of the forecast errors for NHITS and NBEATSx are not significantly different from each other.

For RMSE and QLIKE, NHITS is outperformed by TFT with a significant negative T-Statistic. However, the F-Statistic for QLIKE is highly significant, suggesting that the forecast error variance of TFT is larger and hence less consistent than that of NHITS. In contrast, when comparing MAE and MAPE, NHITS significantly outperforms TFT as indicated by the positive T-Statistics. The F-Statistics are also highly significant, implying that NHITS's forecast errors are not only smaller on average but also more consistent than TFT's.

NHITS significantly outperforms LSTM for all error metrics. The highly significant F-Statistics for these metrics also suggest that LSTM's forecast errors have a higher variance than NHITS, which may indicate less consistency.

In summary, these results illustrate the complex nature of forecasting model performance. While NBEATSx appears to provide more accurate and consistent forecasts than NHITS, the variance of forecast errors between the two models is not significantly different. TFT shows a mixed set of results, with NHITS having smaller and more consistent errors for MAE and MAPE, but not for RMSE and QLIKE, albeit NHITS presents more consistent errors for QLIKE. Lastly, NHITS outperforms LSTM across all metrics in terms of average forecast error.

4.4.3 Conclusion

In conclusion, it can be affirmed that NHITS is a statistically significantly more robust model than TFT and LSTM, yet the same cannot be said about NBEATSx. Consequently, opting for NHITS over TFT and LSTM as a benchmark model for forecasting

realized volatility research would be logical.

4.5 Robustness Test 4

4.5.1 Error Metrics

Table 4 presents the error metrics results for the robustness test 4.

Table 7: Error Metrics for Forecasting Models

Model	RMSE	MAE	MAPE	QLIKE
NHITS	0.467%	0.296%	20.552%	3.797%
GARCH	0.559%	0.363%	27.155%	4.846%
HAR	0.404%	0.268%	19.500%	3.112%
LSTM	0.562%	0.326%	21.118%	5.039%
TFT	0.452%	0.298%	21.408%	3.625%
NBEATSx	0.446%	0.286%	19.924%	3.669%

In Table 7, with respect to RMSE, NHITS ranks as the fourth-best model, lagging behind HAR, NBEATSx, and TFT, which is a change from its position in the main sample where it was the third-best, outperforming GARCH, LSTM, and TFT but behind HAR and NBEATSx. For MAE, NHITS in Table 7 is positioned as the third-best model, which is a drop from the second-best in the main sample.

Regarding MAPE, NHITS maintained its relative performance of the main sample, remaining the second-best model. In both the main sample and robustness test 4, NHITS was surpassed by NBEATSx but showed better performance than the other models. Lastly, looking at QLIKE, NHITS is observed to be the fourth-best model in Table 7, similar to the main sample.

Overall, the relative performance of NHITS in Table 7 compared to Table 1 indicates that NHITS is more sensitive to changes in the training data proportion. The larger testing set in robustness test 4 seems to challenge NHITS more than some other models, particularly HAR and NBEATSx, which consistently outperform NHITS across all metrics. Therefore, this suggests that NHITS could be a novel benchmark model to be used in the realized volatility literature instead of the GARCH and LSTM models, yet this is not necessarily the case for the TFT, HAR and NBEATSx model. Nonetheless, statistical tests are still needed to confirm the results of the robustness test 4

4.5.2 Diebold-Mariano Tests

Figures 23, 24, 25, and 26 depict the DM tests results for all error metrics in the robustness test 4. The results are similar to the main sample, with the exception that NHITS is now not statistically superior nor inferior to TFT MAE, and TFT now yields statistically significantly more precise forecasts than NHITS regarding QLIKE. Hence, it can be inferred from these results that NHITS is a better option for benchmark models than the GARCH and LSTM models, has a similar forecast performance to TFT given the (semi-)optimal random seed, and is inferior to the HAR and NBEATSx models.

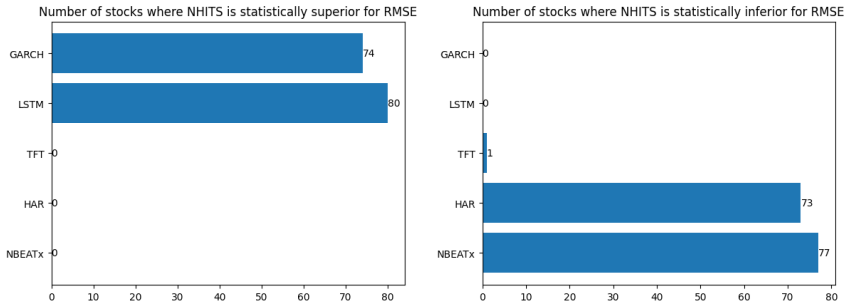

Figure 23: DM Tests Results for RMSE (Robustness Test 4)

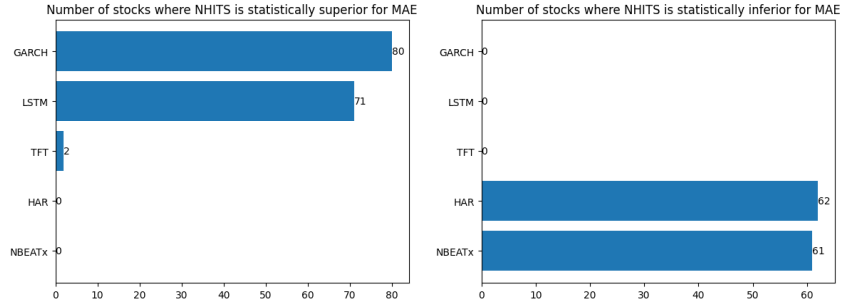

Figure 24: DM Tests Results for MAE (Robustness Test 4)

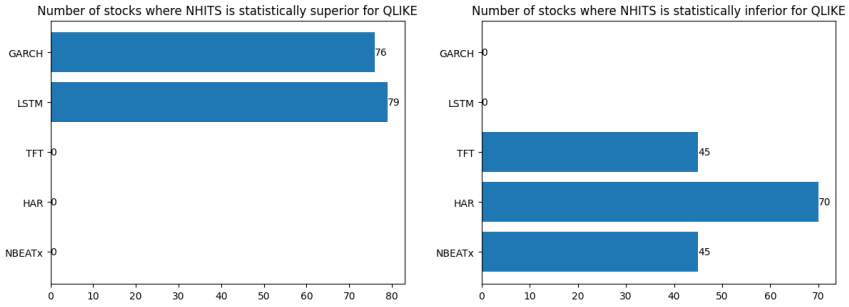

Figure 25: DM Tests Results for QLIKE (Robustness Test 4)

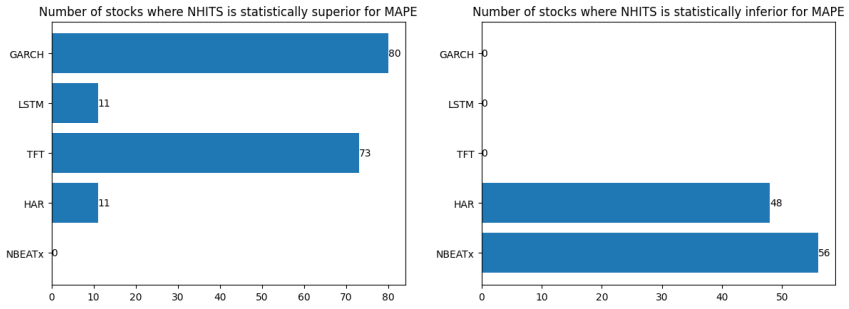

Figure 26: DM Tests Results for MAPE (Robustness Test 4)

4.5.3 Dynamic Model Confidence Set

Figure 27 depicts the Dynamic MCS results for all error metrics in the robustness test 4. It can be seen that for all error measures, NHITS is less often in the position of the best model, especially considering MAE where it is not a single trading month as the best model. It is worth mentioning that the HAR model is again the best model for the the best model during the first corona crisis in the first semester of 2020 and is more regularly the best model, specially regarding MAE and QLIKE. The superioriy of the HAR model is also reflected in the error metrics results.

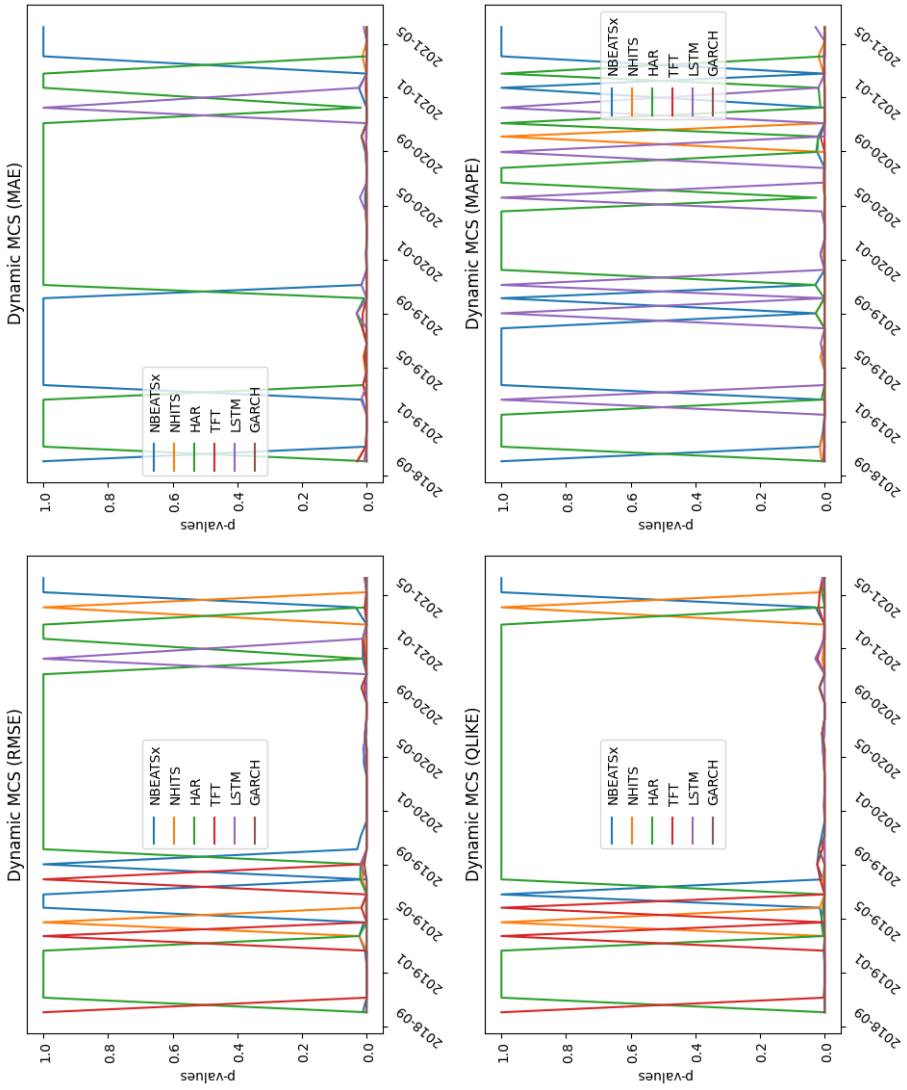

Figure 27: MCS Results (Robustness Test 4)

4.5.4 Conclusion

In conclusion, it is asserted that NHITS presents a superior choice for benchmarking models when compared to GARCH and LSTM models. Furthermore, NHITS demonstrates comparable forecasting performance to TFT when employing (semi-)optimal random seed selection, yet it exhibits a lower level of efficacy in comparison to the HAR and NBEATSx models.

Data and Code Availability

Due to the nature of the data used in this study, the author of this paper is unfortunately not allowed to share it. Regarding the code used in this study, the link to the GitHub repository where all Jupyter Notebooks used in this research can be found will be shared upon the publication of this study.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used ChatGPT in order to check the use of language in the manuscript and point out any spelling or grammatical errors. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

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

Andersen, T. G., & Teräsvirta, T. (2009). Realized volatility. In *Handbook of financial time series* (pp. 555–575). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-71297-8_24

Atkins, A., Niranjan, M., & Gerding, E. (2018). Financial news predicts stock market volatility better than close price. *The Journal of Finance and Data Science*, *4*(2), 120–137. https://doi.org/10.1016/j.jfds.2018.02.002

Audrino, F., Huang, C., & Okhrin, O. (2018). Flexible har model for realized volatility. *Studies in Nonlinear Dynamics & Econometrics*, *23*(3). https://doi.org/10.1515/snde-2017-0080

Audrino, F., & Knaus, S. D. (2015). Lassoing the har model: A model selection perspective on realized volatility dynamics. *Econometric Reviews*, *35*(8–10), 1485–1521. https://doi.org/10.1080/07474938.2015.1092801

Bašta, M., & Molnár, P. (2018a). Oil market volatility and stock market volatility. *Finance Research Letters*, *26*, 204–214. https://doi.org/10.1016/j.frl.2018.02.001

Bašta, M., & Molnár, P. (2018b). Oil market volatility and stock market volatility. *Fin. Res. Lett.*, *26*, 204–214. https://doi.org/10.1016/j.frl.2018.02.001

Bauwens, L., Laurent, S., & Rombouts, J. V. K. (2006). Multivariate garch models: A survey. *Journal of Applied Econometrics*, *21*(1), 79–109. https://doi.org/10.1002/jae.842

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

Bonato, M., Cepni, O., Gupta, R., & Pierdzioch, C. (2021). Forecasting realized volatility of international reits: The role of realized skewness and realized kurtosis. *Journal of Forecasting*, *41*(2), 303–315. https://doi.org/10.1002/for.2813

Bouri, E., Gkillas, K., Gupta, R., & Pierdzioch, C. (2021). Forecasting realized volatility of bitcoin: The role of the trade war. *Comput. Econ.*, *57*(1), 29–53. https://doi.org/10.1007/s10614-020-10022-4

BUCCI, A. (2018). Forecasting realized volatility: A review. *Journal of Advanced Studies in Finance*, *8*(2), 94–138. https://doi.org/10.14505//jasf.v8.2(16).02

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, *18*(3), 502–531. https://doi.org/10.1093/jjfinec/nbaa008

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical Mechanics and its Applications*, *519*, 127–139. https://doi.org/10.1016/j.physa.2018.11.061

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza Ramirez, F., Mergenthaler Canseco, M., & Dubrawski, A. (2023). NHITS: Neural hierarchical interpolation for time series forecasting. *Proc. Conf. AAAI Artif. Intell.*, *37*(6), 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854

Chen, Z., Ma, M., Li, T., Wang, H., & Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. *Inf. Fusion*, *97*(101819), 101819. https://doi.org/10.1016/j.inffus.2023.101819

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196. https://doi.org/10.1093/jjfinec/nbp001

Corsi, F., Audrino, F., & Renò, R. (2012, March). Har modeling for realized volatility forecasting. https://doi.org/10.1002/9781118272039.ch15

Corsi, F., Mittnik, S., Pigorsch, C., & Pigorsch, U. (2008). The volatility of realized volatility. *Econometric Reviews*, *27*(1–3), 46–78. https://doi.org/10.1080/07474930701853616

Degiannakis, S., Filis, G., Klein, T., & Walther, T. (2022). Forecasting realized volatility of agricultural commodities. *Int. J. Forecast.*, *38*(1), 74–96. https://doi.org/10.1016/j.ijforecast.2019.08.011

Deo, R., Hurvich, C., & Lu, Y. (2006). Forecasting realized volatility using a long-memory stochastic volatility model: Estimation, prediction and seasonal adjustment. *Journal of Econometrics*, *131*(1–2), 29–58. https://doi.org/10.1016/j.jeconom.2005.01.003

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, *13*(3), 253. https://doi.org/10.2307/1392185

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, *50*(4), 987. https://doi.org/10.2307/1912773

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Frank, J. (2023). *Forecasting realized volatility in turbulent times using temporal fusion transformers* (FAU Discussion Papers in Economics No. 03/2023). Nürnberg, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics. http://hdl.handle.net/10419/268951

Han, M., Su, Z., & Na, X. (2023). Predict water quality using an improved deep learning method based on spatiotemporal feature correlated: A case study of the tanghe reservoir in china. *Stochastic Environmental Research and Risk Assessment*, *37*(7), 2563–2575. https://doi.org/10.1007/s00477-023-02405-4

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497. https://doi.org/10.3982/ecta5771

Hansen, P. R., Lunde, A., & Voev, V. (2014). Realized beta garch: A multivariate garch model with realized measures of volatility. *Journal of Applied Econometrics*, *29*(5), 774–799. https://doi.org/10.1002/jae.2389

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*(2), 281–291. https://doi.org/10.1016/s0169-2070(96)00719-4

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Min. Knowl. Discov.*, *37*(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5

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

Hu, X. (2021). Stock price prediction based on temporal fusion transformer. *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. https://doi.org/10.1109/mlbdbi54094.2021.00019

Iftikhar, H., Turpo-Chaparro, J. E., Canas Rodrigues, P., & López-Gonzales, J. L. (2023). Forecasting day-ahead electricity prices for the italian electricity market using a new decomposition—combination technique. *Energies*, *16*(18), 6669. https://doi.org/10.3390/en16186669

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Kambouroudis, D. S., McMillan, D. G., & Tsakou, K. (2016). Forecasting stock return volatility: A comparison of garch, implied volatility, and realized volatility models. *Journal of Futures Markets*, *36*(12), 1127–1163. https://doi.org/10.1002/fut.21783

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of*

Forecasting, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Liu, L. Y., Patton, A. J., & Sheppard, K. (2015). Does anything beat 5-minute rv? a comparison of realized measures across multiple asset classes. *Journal of Econometrics*, *187*(1), 293–311. https://doi.org/10.1016/j.jeconom.2015.02.008

Liu, M., Choo, W.-C., Lee, C.-C., & Lee, C.-C. (2022). Trading volume and realized volatility forecasting: Evidence from the china stock market. *Journal of Forecasting*, *42*(1), 76–100. https://doi.org/10.1002/for.2897

Liu, Y., Wu, H., Wang, J., & Long, M. (2022). Non-stationary transformers: Exploring the stationarity in time series forecasting. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (pp. 9881–9893, Vol. 35). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/4054556fcaa934b0bf76da52cf4f92cb-Paper-Conference.pdf

Louzis, D. P., Xanthopoulos-Sisinis, S., & Refenes, A. P. (2014). Realized volatility models and alternative value-at-risk prediction strategies. *Economic Modelling*, *40*, 101–116. https://doi.org/10.1016/j.econmod.2014.03.025

Marcjasz, G., Narajewski, M., Weron, R., & Ziel, F. (2023). Distributional neural networks for electricity price forecasting. *Energy Economics*, *125*, 106843. https://doi.org/10.1016/j.eneco.2023.106843

Mathonsi, T., & van Zyl, T. L. (2021). A statistics and deep learning hybrid method for multivariate time series forecasting and mortality modeling. *Forecasting*, *4*(1), 1–25. https://doi.org/10.3390/forecast4010001

McAleer, M., & Medeiros, M. C. (2008). Realized volatility: A review. *Econom. Rev.*, *27*(1-3), 10–45. https://doi.org/10.1080/07474930701853509

Mehtab, S., & Sen, J. (2022). Analysis and forecasting of financial time series using cnn and lstm-based deep learning models. In *Advances in distributed computing and machine learning* (pp. 405–423). Springer Singapore. https://doi.org/10.1007/978-981-16-4807-6_39

Mesquita, C. M., Valle, C. A., & Pereira, A. C. M. (2023). Scenario generation for financial data with a machine learning approach based on realized volatility and copulas. *Comput. Econ.* https://doi.org/10.1007/s10614-023-10387-2

Miura, R., Pichl, L., & Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In *Lecture notes in computer science* (pp. 165–172). Springer International Publishing. https://doi.org/10.1007/978-3-030-22796-8_18

Naser, M. Z., & Alavi, A. H. (2021). Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. *Architecture, Structures and Construction*. https://doi.org/10.1007/s44150-021-00015-8

Nixtla. (2023). Neuralforecast.

Olivares, K. G., Challu, C., Marcjasz, G., Weron, R., & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx. *International Journal of Forecasting*, *39*(2), 884–900. https://doi.org/10.1016/j.ijforecast.2022.03.001

Olorunnimbe, K., & Viktor, H. (2022). Similarity embedded temporal transformers: Enhancing stock predictions with historically similar trends. In *Foundations of intelligent systems* (pp. 388–398). Springer International Publishing. https://doi.org/10.1007/978-3-031-16564-1_37

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting. https://doi.org/10.48550/ARXIV.1905.10437

Poon, S.-H., & Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. *J. Econ. Lit.*, *41*(2), 478–539. https://doi.org/10.1257/jel.41.2.478

Qu, H., Duan, Q., & Niu, M. (2018). Modeling the volatility of realized volatility to improve volatility forecasts in electricity markets. *Energy Economics*, *74*, 767–776. https://doi.org/10.1016/j.eneco.2018.07.033

Sharma, P., & Vipul. (2016). Forecasting stock market volatility using realized garch model: International evidence. *The Quarterly Review of Economics and Finance*, *59*, 222–230. https://doi.org/10.1016/j.qref.2015.07.005

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of arima and lstm in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. https://doi.org/10.1109/icmla.2018.00227

Souto, H. G. (2023a). Application of persistent homology in forecasting realized volatility. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4471531

Souto, H. G. (2023b). Time series forecasting models for s&p 500 financial turbulence. *Journal of Mathematical Finance*, *13*(01), 112–129. https://doi.org/10.4236/jmf.2023.131007

Souto, H. G. (2023c). Topological tail dependence: Evidence from forecasting realized volatility. *The Journal of Finance and Data Science*, *9*, 100107. https://doi.org/10.1016/j.jfds.2023.100107

Souto, H. G., Blackmon, J., & Moradi, A. (2023). Augmented har. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4516177

Souto, H. G., & Moradi, A. (2023a). Realized covariance matrix nbeatsx. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4529219

Souto, H. G., & Moradi, A. (2023b). Forecasting realized volatility through financial turbulence and neural networks. *Economics and Business Review*, *9*(2). https://doi.org/10.18559/ebr.2023.2.737

Souto, H. G., & Moradi, A. (2023c). A novel loss function for neural network models exploring stock realized volatility using wasserstein distance. *Decision Analytics Journal*, 100369. https://doi.org/10.1016/j.dajour.2023.100369

Souto, H. G., & Moradi, A. (2023d). Introducing nbeatsx to realized volatility forecasting. *Expert Systems with Applications*, 122802. https://doi.org/10.1016/j.eswa.2023.122802

Tang, X., Song, Y., Jiao, X., & Sun, Y. (2023). On forecasting realized volatility for bitcoin based on deep learning PSO–GRU model. *Comput. Econ.* https://doi.org/10.1007/s10614-023-10392-5

Todorova, N., & Souček, M. (2014). Overnight information flow and realized volatility forecasting. *Fin. Res. Lett.*, *11*(4), 420–428. https://doi.org/10.1016/j.frl.2014.07.001

Vortelinos, D. I. (2017). Forecasting realized volatility: Har against principal components combining, neural networks and garch. *Research in International Business and Finance*, *39*, 824–839. https://doi.org/10.1016/j.ribaf.2015.01.004

Wang, X., Li, C., Yi, C., Xu, X., Wang, J., & Zhang, Y. (2022). Ecoforecast: An interpretable data-driven approach for short-term macroeconomic forecasting using n-beats neural network. *Engineering Applications of Artificial Intelligence*, *114*, 105072. https://doi.org/10.1016/j.engappai.2022.105072

Wang, Y., Ma, F., Wei, Y., & Wu, C. (2016). Forecasting realized volatility in a changing world: A dynamic model averaging approach. *Journal of Banking & Finance*, *64*, 136–149. https://doi.org/10.1016/j.jbankfin.2015.12.010

Woo, G., Liu, C., Sahoo, D., Kumar, A., & Hoi, S. (2023, 23–29 Jul). Learning deep time-index models for time series forecasting. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 37217–37237, Vol. 202). PMLR. https://proceedings.mlr.press/v202/woo23b.html

Wu, B., Wang, L., & Zeng, Y.-R. (2022). Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy*, *252*, 123990. https://doi.org/10.1016/j.energy.2022.123990

Yao, X., Izzeldin, M., & Li, Z. (2019). A novel cluster har-type model for forecasting realized volatility. *International Journal of Forecasting*, *35*(4), 1318–1331. https://doi.org/10.1016/j.ijforecast.2019.04.017

Zhang, C., Pu, X., Cucuringu, M., & Dong, X. (2023). Graph neural networks for forecasting realized volatility with nonlinear spillover effects. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4375165

Zheng, P., Zhou, H., Liu, J., & Nakanishi, Y. (2023). Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture. *Appl. Energy*, *349*(121607), 121607. https://doi.org/10.1016/j.apenergy.2023.121607

Appendices

A Summary Statistics of Dataset

Table 8: Summary Statistics of 5-minute Realized Volatility Daily Values

Tickers	Mean	Std	Min	25%	50%	75%	Max
AAPL	1.32%	0.74%	0.27%	0.84%	1.12%	1.57%	6.08%
ABT	1.07%	0.51%	0.35%	0.75%	0.94%	1.22%	5.70%
ACN	1.15%	0.63%	0.37%	0.76%	0.96%	1.33%	7.28%
ADBE	1.42%	0.70%	0.40%	0.96%	1.24%	1.66%	6.64%
ADP	1.04%	0.57%	0.32%	0.70%	0.89%	1.18%	6.56%

Continued on next page

Table 8 – *Continued from previous page*

Tickers	Mean	Std	Min	25%	50%	75%	Max
AMGN	1.26%	0.55%	0.40%	0.90%	1.13%	1.46%	5.71%
AMT	1.27%	0.74%	0.43%	0.83%	1.05%	1.45%	7.17%
AMZN	1.57%	0.86%	0.33%	1.01%	1.36%	1.89%	7.71%
AXP	1.43%	1.05%	0.35%	0.80%	1.07%	1.63%	9.31%
BA	1.40%	0.85%	0.36%	0.88%	1.16%	1.61%	9.37%
BAC	1.75%	1.36%	0.32%	1.01%	1.34%	1.92%	11.45%
BDX	1.06%	0.50%	0.36%	0.74%	0.93%	1.22%	5.27%
BMY	1.20%	0.55%	0.29%	0.85%	1.07%	1.39%	5.43%
BSX	1.57%	0.81%	0.45%	1.07%	1.38%	1.83%	7.31%
C	1.80%	1.48%	0.39%	1.00%	1.35%	1.98%	15.64%
CAT	1.47%	0.79%	0.39%	0.97%	1.26%	1.70%	6.62%
CB	1.11%	0.76%	0.27%	0.66%	0.86%	1.27%	7.59%
CI	1.61%	1.01%	0.43%	1.01%	1.32%	1.81%	12.53%
CMCSA	1.34%	0.73%	0.34%	0.88%	1.15%	1.57%	6.84%
COST	1.07%	0.57%	0.31%	0.73%	0.92%	1.21%	5.32%
CVS	1.29%	0.65%	0.39%	0.87%	1.10%	1.47%	6.02%
CVX	1.35%	0.79%	0.38%	0.85%	1.13%	1.57%	6.77%
D	0.94%	0.52%	0.27%	0.63%	0.80%	1.07%	5.11%
DD	1.37%	0.71%	0.37%	0.90%	1.17%	1.59%	6.46%
DHR	1.16%	0.60%	0.35%	0.78%	0.99%	1.32%	5.96%
DIS	1.25%	0.73%	0.34%	0.83%	1.07%	1.44%	7.38%
DUK	0.94%	0.49%	0.30%	0.63%	0.79%	1.06%	5.24%
EMR	1.18%	0.64%	0.36%	0.80%	1.01%	1.36%	6.29%
EXC	1.10%	0.64%	0.31%	0.70%	0.91%	1.25%	6.83%
F	1.62%	0.89%	0.40%	1.03%	1.35%	1.89%	8.23%
FB	1.52%	0.84%	0.41%	0.98%	1.27%	1.78%	7.88%
FDX	1.28%	0.70%	0.38%	0.85%	1.09%	1.47%	6.93%
GD	1.15%	0.60%	0.36%	0.78%	0.99%	1.32%	6.11%
GE	1.49%	0.85%	0.36%	0.93%	1.23%	1.73%	7.54%
GILD	1.37%	0.70%	0.39%	0.90%	1.16%	1.59%	6.75%
GM	1.60%	0.86%	0.42%	1.04%	1.35%	1.88%	7.92%
GOOGL	1.37%	0.72%	0.39%	0.90%	1.17%	1.60%	7.02%
GS	1.65%	1.03%	0.44%	1.02%	1.35%	1.90%	9.45%
HCA	1.49%	0.83%	0.44%	0.97%	1.25%	1.73%	7.68%
HD	1.25%	0.67%	0.38%	0.84%	1.07%	1.43%	6.93%
HON	1.17%	0.63%	0.38%	0.80%	1.00%	1.31%	6.52%
IBM	1.21%	0.66%	0.36%	0.82%	1.04%	1.38%	6.89%
INTC	1.39%	0.73%	0.39%	0.90%	1.17%	1.62%	7.13%
JNJ	0.97%	0.49%	0.31%	0.68%	0.83%	1.08%	5.44%
JPM	1.59%	1.09%	0.38%	0.93%	1.23%	1.79%	10.12%
KMB	0.97%	0.50%	0.30%	0.68%	0.84%	1.08%	5.60%

Continued on next page

Table 8 – *Continued from previous page*

Tickers	Mean	Std	Min	25%	50%	75%	Max
KO	0.90%	0.48%	0.30%	0.63%	0.78%	1.00%	5.33%
LLY	1.23%	0.65%	0.38%	0.84%	1.06%	1.39%	6.90%
LMT	1.12%	0.61%	0.35%	0.76%	0.95%	1.26%	6.33%
LOW	1.30%	0.72%	0.35%	0.89%	1.11%	1.48%	7.23%
MA	1.39%	0.77%	0.38%	0.91%	1.17%	1.62%	7.41%
MCD	0.98%	0.54%	0.30%	0.67%	0.83%	1.10%	5.66%
MRK	1.13%	0.60%	0.35%	0.76%	0.96%	1.32%	5.49%
MS	1.90%	1.42%	0.44%	1.12%	1.48%	2.08%	15.72%
MSFT	1.20%	0.61%	0.34%	0.82%	1.04%	1.38%	5.44%
NFLX	2.15%	0.95%	0.60%	1.46%	1.94%	2.60%	8.48%
NKE	1.26%	0.65%	0.38%	0.86%	1.07%	1.42%	6.87%
NVDA	2.04%	0.98%	0.63%	1.35%	1.79%	2.44%	8.42%
ORCL	1.22%	0.64%	0.27%	0.81%	1.07%	1.43%	6.52%
PEP	0.89%	0.48%	0.25%	0.61%	0.76%	1.00%	5.17%
PFE	1.12%	0.54%	0.37%	0.76%	0.97%	1.29%	5.09%
PG	0.88%	0.47%	0.30%	0.62%	0.76%	0.99%	5.50%
PNC	1.54%	1.11%	0.40%	0.89%	1.17%	1.74%	11.58%
QCOM	1.39%	0.71%	0.32%	0.90%	1.22%	1.66%	6.38%
SBUX	1.35%	0.79%	0.42%	0.84%	1.11%	1.57%	7.84%
SO	0.97%	0.49%	0.34%	0.68%	0.85%	1.10%	5.86%
SYK	1.15%	0.59%	0.29%	0.78%	0.99%	1.32%	6.94%
T	1.05%	0.61%	0.29%	0.69%	0.87%	1.18%	5.53%
TGT	1.35%	0.78%	0.34%	0.87%	1.11%	1.53%	7.18%
TJX	1.34%	0.72%	0.40%	0.87%	1.11%	1.59%	7.32%
TMO	1.23%	0.61%	0.39%	0.84%	1.07%	1.41%	6.25%
TXN	1.36%	0.68%	0.41%	0.92%	1.19%	1.60%	6.89%
UNH	1.41%	0.83%	0.40%	0.88%	1.16%	1.60%	7.13%
UNP	1.39%	0.77%	0.37%	0.91%	1.18%	1.58%	6.66%
UPS	1.10%	0.60%	0.31%	0.71%	0.94%	1.31%	5.51%
USB	1.42%	1.07%	0.37%	0.79%	1.08%	1.63%	9.41%
VZ	1.03%	0.57%	0.31%	0.70%	0.88%	1.15%	5.70%
WFC	1.58%	1.23%	0.33%	0.85%	1.18%	1.80%	10.07%
WMT	0.96%	0.50%	0.34%	0.67%	0.82%	1.08%	5.09%

B Hyperparameters Search Space for Each Model

Table 9: LSTM Hyperparameters Search Space

Hyperparameters	**Options**
n_inputs	[3, 5, 10, 15, 21, 42]
encoder_layers	[1, 2, 4, 6]
decoder_layers	[1, 2, 4, 6]
encoder_hidden_size	[50, 100, 200, 300, 400, 500]
decoder_hidden_size	[50, 100, 200, 300, 400, 500]
encoder_dropout	[0, 0.2, 0.3, 0.4]
epochs	[50, 100, 150, 250, 350, 450, 550, 650, 750]
learning_rate	[0.0005, 0.0001, 0.00005, 0.00001]
num_lr_decays	[5, 3, 2, 1]
scaler_type	[”robust”, ”standard”, ”minmax”]
context_size	[5, 10, 15, 20, 30, 40]
losses	[MSE(), MAE(), MQLoss(level=[80, 90]), DistributionLoss(distribution=’StudentT’, level=[80, 90])]

Table 10: TFT Hyperparameters Search Space

Hyperparameters	**Options**
n_inputs	[3, 5, 10, 15, 21, 42, 84]
hidden_size	[50, 100, 150, 200, 300, 500]
epochs	[50, 100, 150, 250, 350, 450, 550, 650, 750]
dropout	[0, 0.2, 0.3, 0.4]
n_head	[2, 4, 5, 8, 10, 20]
attn_dropout	[0, 0.2, 0.3, 0.4]
learning_rate	[0.0005, 0.0001, 0.00005, 0.00001]
losses	[MSE(), MAE(), MQLoss(level=[80, 90]), DistributionLoss(distribution='StudentT', level=[80, 90])]
num_lr_decays	[5, 3, 2, 1]
scaler_type	["robust", "standard", "minmax"]

Table 11: NBEATSx Hyperparameters Search Space

Hyperparameters	**Options**
n_inputs	[3, 5, 10, 15, 21, 42, 84]
mlp_units	[[[712, 712], [712, 712]], [[512, 512], [512, 512]], [[250, 250], [250, 250]], [[100, 100], [100, 100]]]
epochs	[50, 100, 150, 250, 350, 450, 550, 650, 750]
learning_rate	[0.0005, 0.0001, 0.00005, 0.00001]
num_lr_decays	[5, 3, 2, 1]
scaler_type	["robust", "standard", "minmax"]
losses	[MSE(), MAE(), MQLoss(level=[80, 90]), DistributionLoss(distribution='StudentT', level=[80, 90])]
n_harmonics	[0, 0, 1, 1]
n_blocks	[[1, 1], [2, 2], [3, 3], [5, 5]]
n_polynomials	[0, 1, 0, 1]

Table 12: NHITS Hyperparameters Search Space

Hyperparameters	**Options**
n_inputs	[3, 5, 10, 15, 21, 42, 84]
mlp_units	[[[712, 712], [712, 712]], [[512, 512], [512, 512]], [[250, 250], [250, 250]], [[100, 100], [100, 100]]]
epochs	[50, 100, 150, 250, 350, 450, 550, 650, 750]
learning_rate	[0.0005, 0.0001, 0.00005, 0.00001]
num_lr_decays	[5, 3, 2, 1]
scaler_type	[”robust”, ”standard”, ”minmax”]
losses	[MSE(), MAE(), MQLoss(level=[80, 90]), DistributionLoss(distribution=’StudentT’, level=[80, 90])]
n_blocks	[[1, 1], [2, 2], [3, 3], [5, 5]]
dropout_prob_theta	[0, 0.2, 0.3, 0.4]
n_pool_kernel_size	[[2, 2, 1], [4, 2, 1], [4, 4, 2]]
pooling_mode	[”MaxPool1d”, ”AvgPool1d”]
interpolation_mode	[’linear’, ’nearest’]

C Optimal Hyperparameters for Each Model

Table 13: LSTM Optimal Hyperparameters

Hyperparameters	**Options**
n_inputs	15
encoder_layers	4
decoder_layers	4
encoder_hidden_size	200
decoder_hidden_size	300
encoder_dropout	0.2
epochs	350
learning_rate	0.0005
num_lr_decays	5
scaler_type	”robust”
context_size	30
losses	DistributionLoss(distribution=’StudentT’, level=[80, 90])

Table 14: TFT Optimal Hyperparameters

Hyperparameters	**Options**
n_inputs	15
hidden_size	500
epochs	350
dropout	0.2
n_head	20
attn_dropout	0.4
learning_rate	0.00001
losses	MSE()
num_lr_decays	1
scaler_type	”minmax”

Table 15: NBEATSx Optimal Hyperparameters

Hyperparameters	**Options**
n_inputs	15
mlp_units	[[512, 512], [512, 512]]
epochs	550
learning_rate	0.0005
num_lr_decays	5
scaler_type	”standard”
losses	DistributionLoss(distribution=’StudentT’, level=[80, 90])
n_harmonics	0
n_blocks	[3, 3]
n_polynomials	0

Table 16: NHITS Optimal Hyperparameters

Hyperparameters	**Options**
n_inputs	21
mlp_units	[[712, 712], [712, 712]]
epochs	150
learning_rate	0.0005
num_lr_decays	5
scaler_type	”standard”
losses	DistributionLoss(distribution=’StudentT’, level=[80, 90])
n_blocks	[2, 2]
dropout_prob_theta	0
n_pool_kernel_size	[4, 2, 1]
pooling_mode	”MaxPool1d”
interpolation_mode	’nearest’