The Journal of Finance and Data Science 10 (2024) 100143

Contents lists available at ScienceDirect

The Journal of Finance and Data Science

journal homepage: www.keaipublishing.com/en/journals/jfds

Research Article

Time-mixing and feature-mixing modelling for realized volatility forecast: Evidence from TSMixer model

Hugo Gobato Souto [a,*], Storm Koert Heuvel [b], Francisco Louzada Neto [a]

[a] University of São Paulo (USP), Avenida Trabalhador São-carlense, 400, São Carlos, 13566-590, Brazil
[b] Leiden University, Mathematical Institute, NL-2300 RA Leiden, the Netherlands

ARTICLE INFO

Keywords:
TSMixer
Financial time series
Stock realized volatility
S&P 100
Neural networks

ABSTRACT

This study evaluates the effectiveness of the TSMixer neural network model in forecasting stock realized volatility, comparing it with traditional and contemporary benchmark models. Using data from S&P 100 index stocks and three other datasets containing various financial securities, extensive analyses, including robustness tests, were conducted. Results show that TSMixer outperforms benchmark models in predicting individual stock volatility when applied to datasets with a large number of securities, leveraging its feature-mixing MLP techniques, which can properly model the financial tail dependence phenomenon. However, its superiority diminishes in datasets with fewer securities, such as stock indexes, foreign exchange rates, and commodities, where models like NBEATSx and NHITS often perform better. This indicates that TSMixer's performance is context-dependent, excelling when feature interdependencies can be fully exploited. The findings suggest that simplified neural network architectures like TSMixer can enhance forecasting accuracy in appropriate contexts but may have limitations in datasets with fewer securities.

1. Introduction

Soft computing techniques have increasingly become a cornerstone in solving complex, high-level problems where traditional algorithms fall short. Their ability to produce approximate solutions to otherwise intractable issues has led to a significant presence in scientific literature. The flexibility and adaptability of these techniques make them particularly suited for real-world applications where exact solutions are either unattainable or impractical (Das et al., 2013; Denaï et al., 2007; Ibrahim, 2016; O. Omolaye, 2017; Subasi, 2012; Yardimci, 2009).

Among the various soft computing methodologies, neural networks have emerged as one of the most, if not the most, influential techniques (Bahrammirzaee, 2010; Deb, 2011; de Campos Souza, 2020; Kecman, 2001; Khan et al., 2021; Looney, 1993; Özbakır et al., 2010; Sahin et al., 2012). Their versatility and powerful computational abilities have enabled applications across a diverse array of domains, particularly in forecasting tasks. The literature is rich with instances where neural networks have been successfully applied to predict weather patterns (Abhishek et al., 2012; Baboo and Shereef, 2010; Fente and Kumar Singh, 2018; Maqsood et al., 2004; Roesch and Günther, 2018), financial data (Bucci, 2020; S. Gu et al., 2020; Kim, 2006; Kim and Ahn, 2011; Malliaris and Salchenberger, 1993; Pavlidis et al., 2006; Souto, 2023c; Souto and Moradi, 2023c; C.Zheng et al., 2023,a, b), energy consumption (Aydin et al., 2016; Azadeh et al., 2008, 2013; Javeed Nizami and Al-Garni, 1995; Ruiz et al., 2018; Runge and Zmeureanu, 2019; Yan et al., 2019), and more. This success is attributed mainly to the ongoing development and refinement of novel neural network models, each surpassing its predecessors in accuracy and efficiency.

* Corresponding author.
E-mail addresses: hgsouto@usp.br (H.G. Souto), s.k.heuvel@umail.leidenuniv.nl (S.K. Heuvel), louzada@icmc.usp.br (F.L. Neto).
Peer review under the responsibility of KeAi Communications Co., Ltd.

https://doi.org/10.1016/j.jfds.2024.100143
Received 8 May 2024; Received in revised form 20 September 2024; Accepted 21 October 2024
Available online 26 October 2024

In the realm of financial forecasting, neural networks have gained considerable attention (Chen et al., 2023a; Miura et al., 2019; Ruf and Wang, 2019; Vortelinos, 2017; Weigand, 2019). Their integration into financial models has significantly advanced the field, particularly in the challenging task of stock realized volatility forecasting (Arnerić et al., 2018; Q. Chen and Robert, 2022; D'Ecclesia and Clementi, 2019; Hamid and Iqbal, 2004; J. Li, 2022; Y. Liu, 2019; Reisenhofer et al., 2022; Vidal and Kristjanpoller, 2020; Wing-Yi Chio et al., 2021). The complexity and unpredictability of financial markets (R. Engle, 2004; Sewell, 2011; Sezer et al., 2020; Souto, 2023b; Tsay, 2005) make them an ideal candidate for the application of neural networks.

Recent advancements in forecasting neural network models, such as Temporal Fusion Transformer (TFT) (Lim et al., 2021), neural basis expansion analysis with exogenous variables (NBEATSx) (Olivares et al., 2023), Neural Hierarchical Interpolation for Time Series Forecasting (NHITS) (Challu et al., 2023), TimesNet (H. Wu et al., 2022), and Time-Series Mixer (TSMixer) (Chen et al., 2023b), have demonstrated remarkable forecasting capabilities in various contexts, including stock realized volatility (Lim et al., 2021; Souto, 2023a, 2024; Souto and Moradi, 2023a, 2023d). These models have set new benchmarks in the field, offering more accurate and reliable predictions than traditional methods. However, one model, TSMixer, while having shown promising results in other forecasting domains (S.-A. Chen et al., 2023; Z. Huang and He, 2024; Z. Li et al., 2023; Z. Liu et al., 2024; Ye et al., 2023), remains unexplored in the context of stock realized volatility, presenting a notable gap in the current literature.

This research investigates the applicability and performance of the TSMixer model in forecasting stock realized volatility. The scientific contributions of this paper are manifold:

1. **Introduction of a novel neural network forecasting model for financial forecasting.**
2. **Provision of a thorough forecast power comparison between the TSMixer and benchmark models.**
3. **Provision of open-source code for easy implementation by practitioners and researchers of the TSMixer model in the context of realized volatility prediction.**
4. **Provision of novel insights about the efficiency of relatively simple neural network architectures for forecasting tasks**
5. **Provision of empirical evidence for the potential of time-mixing and feature mixing multi-layer perceptrons approach for modelling stock realized volatility**

Concerning the 4th contribution, the traditional view in financial forecasting has often been that more complex neural network architectures yield better performance due to their ability to capture intricate patterns and nonlinear relationships inherent in financial data. This belief stems from the success of deep learning models in various domains, where increasing model depth and/or complexity often leads to improved performance (Dunis et al., 2011; Goodfellow, 2016; Zimmermann et al., 2012). In the context of financial markets, studies have utilized complex architectures like Long Short-Term Memory (LSTM) networks and deep neural networks to model and predict financial time series, operating under the assumption that greater complexity allows the model to better capture market dynamics (K. Chen et al., 2015; Fischer and Krauss, 2018; G. P. Zhang and Berardi, 2001).

However, recent literature has begun to challenge this notion, suggesting that increased complexity does not necessarily guarantee better forecasting accuracy (Lara-Benítez et al., 2021; Oliveira et al., 2016; Ramos et al., 2023). For instance, Makridakis et al. (2018) conducted a comprehensive analysis comparing statistical and machine learning forecasting methods and found that simpler models often outperform more complex ones, especially when the risk of overfitting is considered.

In light of this, our study contributes to this ongoing discussion by demonstrating that a simplified neural network model like TSMixer can achieve superior forecasting accuracy in certain contexts, particularly when dealing with datasets containing a large number of financial securities.

The paper employs a robust methodology inspired by Souto (2023a), accomplishing the goal of addressing the identified gap in existing literature, comparing the TSMixer model against traditional models: the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model (Bollerslev, 1986), the Heterogeneous Autoregressive (HAR) model (Corsi, 2009), and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), as well as newer models like TFT, NBEATSx, NHITS, and TimesNet. The remainder of the paper is structured as follows: Section 2 presents a literature review of recent models and methods for forecasting realized volatility and the application of the TSMixer in other domains, Section 3 delves into the architecture of the TSMixer model, Section 4 outlines the research design, Section 5 presents the results and discussion, Section 6 explores the implications of and novel insights from the results of this research, and Section 7 concludes the paper, summarizing the essential findings and implications of this study.

2. Related work

In recent years, several innovative models and methods have been proposed to address the complex task of forecasting RV in financial markets. Traditional models like HAR-type models, such as the Cluster HAR proposed by Yao et al. (2019), employ hierarchical clustering techniques to select relevant lagged volatilities, improving forecasting accuracy by distinguishing between jumps and continuous volatility components. This approach introduces a new dimension to volatility forecasting, yet the reliance on lagged volatilities and its hierarchical structure limits its ability to capture more intricate feature interactions present in financial time series data.

Moreover, Frank (2023a) demonstrated the effectiveness of TFT in predicting realized volatilities, especially during volatile periods. TFTs have shown strong predictive performance compared to other machine learning methods, such as LSTM networks and random forests, due to their ability to handle sequential dependencies and integrate different training approaches. While powerful, TFTs are complex models that require significant computational resources and are often difficult to interpret, particularly when dealing with highly volatile financial markets.

In contrast, Souto and Moradi (2023d) explored the application of BEATSx, which yielded robust predictions across different forecasting horizons. Despite the accuracy of NBEATSx in medium- and long-term forecasts, its performance in short-term volatility prediction was less consistent, particularly when applied to stock indexes from developing markets. The NHITS model proposed by Souto (2023a) also performed well in long-term forecasting tasks, but it fell short compared to NBEATSx and HAR models in short-term prediction, indicating room for improvement in capturing immediate market dynamics. Additionally, Souto (2024) evaluated the TimesNet model, a CNN-based approach, for forecasting stock realized volatility. The study compared TimesNet with traditional and contemporary models, showing competitive performance in RMSE and QLIKE but weaker results in MAE and MAPE compared to HAR, NBEATSx, and NHITS. While TimesNet demonstrates potential for long-term forecasting and extreme event prediction, its short-term forecasting and error handling need improvement.

Similarly, Yu et al. (2023) combined variational mode decomposition (VMD) with deep learning models, such as LSTM and gated recurrent units (GRU), to improve the prediction of RV. The model's architecture integrates reinforcement learning for optimal weight selection, further enhancing its forecasting capabilities. However, while this approach is innovative, it is computationally intensive and relies on complex hybrid techniques, which may not be suitable for real-time forecasting applications.

M. Liu et al. (2022)a, b, on the other hand, explored the role of trading volume in volatility forecasting by using empirical mode decomposition (EMD) with the HAR model. By decomposing trading volume into short- and long-term components, they improved out-of-sample RV forecasts, offering a new perspective on the volume–volatility relationship. Although effective, this method is computationally complex, relying on advanced decomposition techniques. Similarly, Bucci et al. (2023) addressed overfitting in volatility forecasting with neural networks by applying dimensionality reduction methods like Bayesian Model Averaging (BMA) and Lasso. These approaches reduced predictors, resulting in more accurate forecasts across stock assets. However, the reliance on extensive preprocessing can make these models less accessible.

These recent models, although highly advanced, present significant complexity, which can hinder their scalability and interpretability in real-world applications. The TSMixer model aims to fill these gaps by offering a relatively simpler architecture that leverages feature-mixing MLPs to effectively capture the temporal dependencies and feature correlations within financial time series data. Unlike traditional neural network models that require complex preprocessing or dimensionality reduction steps, the TSMixer directly processes multivariate time series data through its dual mixer architecture: the feature mixer captures the interdependencies between features, while the temporal mixer identifies patterns over time. This architecture is particularly well-suited to exploit the financial tail dependence phenomenon (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015). By efficiently capturing both temporal and feature interactions, TSMixer offers a promising approach to forecasting RV, especially in periods of or preceding financial instability, where tail dependencies play a crucial role.

Although no studies have yet explored the application of the TSMixer model in forecasting realized volatility, there are a few investigations into its implementation across other domains. These studies illustrate the versatility of TSMixer in handling multivariate time series data, offering promising results in different fields.

Ye et al. (2023) introduced the TSMixer architecture for forecasting the S&P500 Index, which is a critical economic indicator used to evaluate market performance and predict economic trends. The authors built the model on the principles of the MLP-Mixer, applying it to time series data, which traditionally employs Recurrent Neural Networks (RNNs) or attention-based networks like Transformers. TSMixer, however, differs from these models by utilizing two types of Multilayer Perceptron (MLP) layers: a feature mixer and a temporal mixer. The feature mixer captures correlations among features, while the temporal mixer extracts temporal dependencies across the input sequence. This architecture demonstrates competitive performance in S&P500 Index price prediction.

Similarly, Z. Liu et al. (2024) applied TSMixer in the context of industrial process control, where real-time model predictive control (MPC) is critical. In their work, TSMixer was employed to forecast system behavior under control constraints, with an extension that integrates a feature classification model (TSMixer with FCM). This modified architecture leverages future information and static variables, leading to superior accuracy in comparison to transformer-based models while maintaining a smaller parameter count. The model was further enhanced through the development of a two-dimensional block stochastic configuration network (2D-BSCN), which improved computational efficiency. The results showed that the TSMixer-based MPC exhibited faster processing times and reduced FLOPs, making it highly suitable for real-time industrial applications.

In the domain of healthcare, Z. Huang and He (2024) explored the use of the TSMixer in conjunction with Gated Recurrent Units (GRU) to create the GRU-TSMixer model for detecting sleep apnea and hypopnea events. By using raw respiratory signals, the model was able to accurately detect these events without the need for manual feature engineering. The combination of GRU for temporal dynamics and TSMixer for feature extraction resulted in superior performance compared to CNN, LSTM, and other state-of-the-art models. The model's success in large-scale datasets, such as the Sleep-Heart-Health-Study (SHHS) and the Multi-Ethnic Study of Atherosclerosis (MESA), highlights the potential of TSMixer in medical diagnostics, specifically for non-intrusive sleep disorder detection.

These studies collectively showcase the flexibility of the TSMixer model, suggesting that it could also be a promising candidate for financial volatility forecasting.

3. TSMixer architecture: an overview

The Time-Series Mixer (TSMixer) model, proposed by S.-A. Chen et al. (S.-A. Chen et al., 2023), introduces a novel approach in time series forecasting, particularly relevant to the complex dynamics of real-life data, such as data from financial markets. Unlike more conventional recurrent or attention-based models, TSMixer is grounded in a more straightforward yet highly effective architecture based on MLPs, mirroring the modular structure often employed in financial modelling.

TSMixer's design efficiently processes multivariate time series data, a common characteristic in financial analyses, especially for stock realized volatility forecasting. It operates on the principle of performing mixing operations across time and feature (i.e., in the stock realized volatility context, the features would be the different stocks) dimensions, reflecting the dual focus on temporal trends and cross-sectional asset characteristics in financial forecasting.

A brief description of the TSMixer architecture is given below. For further details about the TSMixer architecture, please see S.-A. Chen et al. (S.-A. Chen et al., 2023).

Central to TSMixer's architecture are the time-mixing MLPs (see Fig. 1), which model temporal patterns in time series. These layers consist of a fully-connected layer followed by an activation function and dropout. Formally, for an input matrix $X \in \mathbb{R}^{T \times F}$ (where T is the number of time steps and F the number of features), the time-mixing operation is defined as:

$$X' = \text{Dropout}(\text{Activation}(W_t \cdot X^T)), \tag{1}$$

where W_t denotes the weights of the fully-connected layer in the time-mixing MLP, and X^T is the transposed input matrix.

Complementing the time-mixing MLPs are the feature-mixing MLPs, which analyze cross-sectional, covariate information in the dataset (see Fig. 1). These MLPs employ a two-layer structure to enable complex transformations, capturing intricate relationships between different market indicators. For the output matrix X' of the time-mixing operation, the feature-mixing operation is:

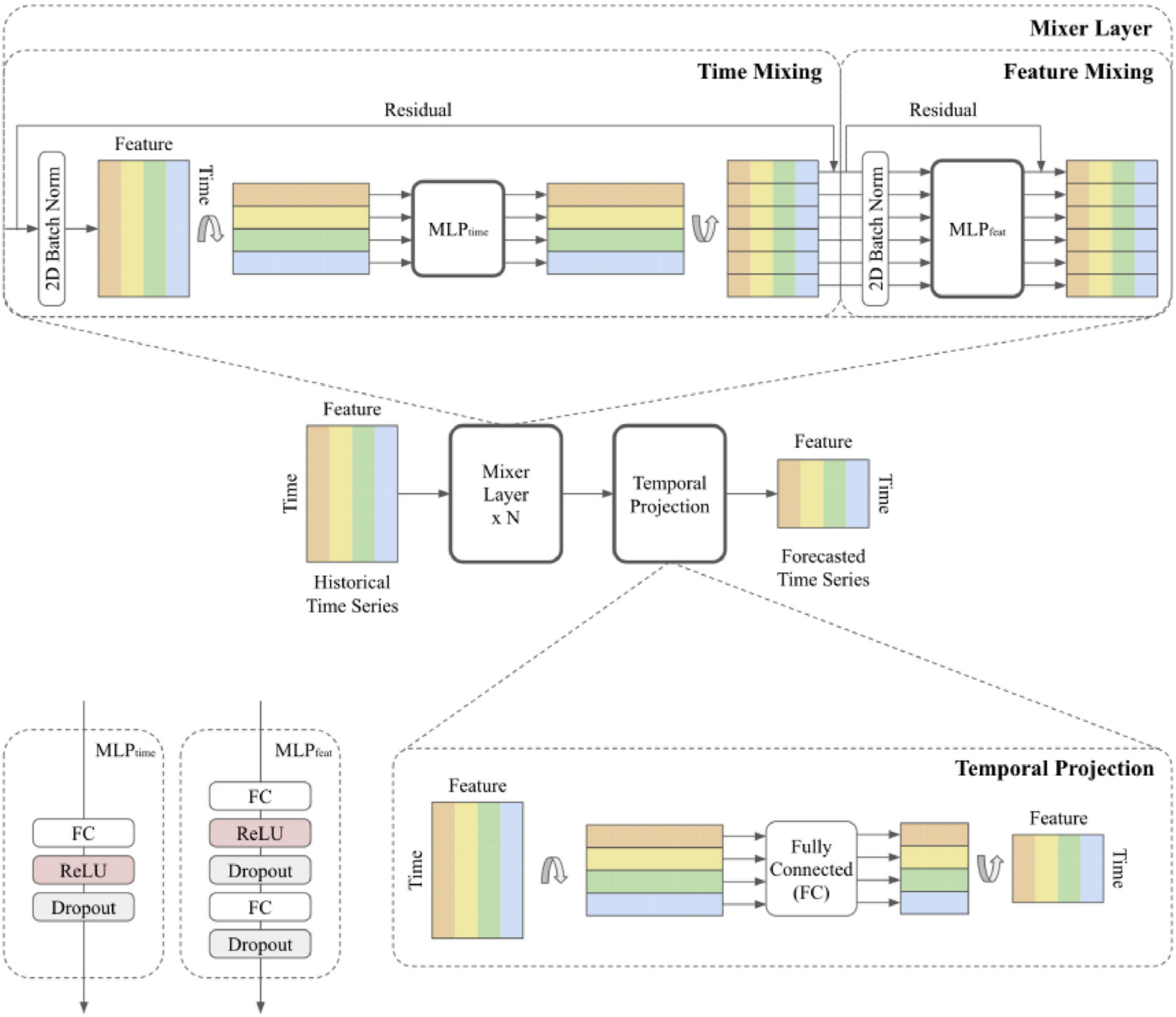

Fig. 1. TSMixer Architecture (S.-A. Chen et al., 2023). In the TSMixer model, the input matrix is organized such that its columns represent distinct features or variates, while each row corresponds to a specific time step. This structure is crucial for the fully-connected operations of the model, which are applied across the rows. At the heart of TSMixer's architecture are alternating layers of time-mixing and feature-mixing MLPs, which synthesize information effectively. A sequence of mixer layers defines the architecture, the quantity of which is indicated by N. Within this framework, the time-mixing MLPs are uniformly applied to all features, whereas the feature-mixing MLPs are consistently used across all time steps. This strategic design enables TSMixer to efficiently harness both temporal dynamics and cross–variate interactions, utilizing a compact parameter set to achieve enhanced generalization capabilities.

$$X'' = \text{Dropout}(\text{Activation}(W_f \cdot X')) \quad (2)$$

where W_f represents the weights of the fully-connected layer in the feature-mixing MLP.

Furthermore, TSMixer incorporates residual connections between each pair of time-mixing and feature-mixing layers, enhancing the model's ability to learn deeper architectures more efficiently. Moreover, as shown in Fig. 1, it employs 2D normalization on both time and feature dimensions, stabilizing the learning process by normalizing the input data. This approach is particularly effective for time-series data where both temporal and cross-sectional information is vital.

In conclusion, the TSMixer model, with its innovative approach, is well-suited for complex forecasting tasks and, thus, potentially appropriate for forecasting stock realized volatility. Its design, which focuses on analyzing temporal and cross-sectional data through time-mixing and feature-mixing MLPs, enhanced by residual connections and normalization, could make it a powerful tool in quantitative finance.

3.1. Potential limitations of the TSMixer architecture

While the TSMixer model introduces a novel framework for time series forecasting by leveraging MLPs for temporal and cross-sectional data mixing, it is important to acknowledge potential limitations inherent in its architecture.

Firstly, akin to many deep learning models, TSMixer's performance is heavily reliant on the availability of large datasets (Izbicki & dos Santos, 2020). The model's numerous parameters, resulting from its stacked time-mixing and feature-mixing MLP layers, require substantial amounts of data to adequately learn the underlying patterns without overfitting (Izbicki & dos Santos, 2020). In contexts where historical data is limited or where time series are short, this dependence may hinder the model's ability to generalize effectively, potentially making it less suitable compared to models designed to perform well with smaller datasets.

Secondly, the interpretability of TSMixer poses a significant challenge. Neural networks, including MLP-based architectures, are often criticized for being “black boxes” due to the opacity of their internal representations and decision-making processes (Izbicki & dos Santos, 2020). Stakeholders may require clear explanations of the drivers behind model predictions to ensure compliance with regulatory standards, to build trust in automated systems, or to derive actionable insights. The lack of transparency in TSMixer's outputs can thus be a considerable drawback in applications where understanding model reasoning is essential.

Beyond these common concerns, there are limitations specific to TSMixer's architectural design. The model assumes fixed-size input sequences due to its fully connected layer operations. This characteristic can limit the model's flexibility when dealing with variable-length time series or datasets with missing time points, which are common challenges in real-world financial data. Preprocessing steps to pad or truncate sequences, or to impute missing values, may be required, introducing additional complexity and potential sources of error into the modeling pipeline. Furthermore, the absence of built-in mechanisms for feature selection or the handling of multicollinearity means that redundant or irrelevant features could negatively impact the model's learning process, highlighting the importance of careful feature engineering prior to model training.

4. Research design

4.1. Sample for empirical experimentation

This study's empirical investigation relies on a meticulously curated dataset from the Standard & Poor's 100 (S&P 100) index. The dataset incorporates a select group of 80 stocks actively traded from July 1, 2007, to June 30, 2021. This period is notably significant, capturing key financial events, including the 2008 global financial crisis and the market disturbances triggered by the COVID-19 pandemic. Analyzing this interval enables a thorough examination of the performance of different volatility forecasting models across varied market environments. Furthermore, the dataset comprises daily realized volatility (RV) figures for each stock, amounting to 3409 data points.

Refer to Appendix A for a comprehensive list of the 80 chosen stocks and a statistical breakdown of their daily 5-min RV values. This appendix provides crucial statistics such as average, median, variance, and more, offering an initial overview of the volatility profiles of the included stocks. Spanning a period characterized by diverse market conditions and encompassing a broad range of stocks from the S&P 100, this dataset is an optimal foundation for evaluating the accuracy of various forecasting models, including the cutting-edge TimesNet model.

The dataset is segmented into a training set (70 % of the data) and a testing set (30 %). This division follows standard machine learning and statistical forecasting practices, ensuring sufficient model training and reliable performance evaluation. While a 70 %/30 % data split is standard (Joseph, 2022), an alternative 80 %/20 % split is also explored in the first robustness test of this study, providing insights into model robustness under varying data partitions.

Besides the main sample composed of the S&P 100 stocks, three additional datasets are used to put the empirical testing results and conclusions to the test. These datasets are used as the fourth, fifth and sixth robustness test of this study. The first dataset is composed of the stock indexes: AU200 (ÂXJO), FR40 (FCHI), JP225 (N225), UK100 (FTSE), NAS100 (NDX), SPX500 (ŜPX), and US 2000 (RUT). Its sample period is from 1 January, 2005 until 31 December, 2019, with a total of 3244 trading days. The second dataset contains the currency exchanges: AUD/JPY, AUD/USD, EUR/JPY, EUR/USD, GBP/USD, USD/CAD. Its sample period is 1 January, 2005 to 31 December, 2019, with a total of 4684 trading days. The third dataset, on the other hand, contains the commodities in USD: Corn (ZC), Gold (XAU), Natural Gas (NG), Soybeans (ZS), Sugar (SB), and West Texas Oil (WTICO). The sample period of this dataset ranges from 3 April, 2006 to 31 December, 2019, with a total of 3465 trading days. Their RV computation also adheres to the methodology outlined by L. Y. Liu et al. (L. Y. Liu et al., 2015) and their summary statistics can also be found in Appendix A.

Incidentally, The RV values of the S&P 100 dataset and the other datastes are computed using high-frequency intraday data sourced from the LOBSTER database and OANDA database respectively. The computation of RV adheres to the methodology outlined by L. Y. Liu et al. (2015), leveraging high-frequency data to capture market volatility nuances accurately. For detailed insights into this computation method, see L. Y. Liu et al. (2015). Yet, in general terms, the high-frequency data used in this study consists of 5-min price observations for each trading day. The raw data was cleaned to ensure accurate estimation of realized volatility. The following preprocessing steps were applied:

1. **Timestamp alignment**: Prices recorded at irregular intervals are aligned to the nearest 5-min mark. Observations outside the regular trading hours are excluded to avoid distortions caused by illiquidity and market closures.
2. **Missing data**: If a price observation is missing at a scheduled 5-min interval, linear interpolation is used to estimate the missing price.
3. **Outliers**: Spikes or data points that deviated by more than five standard deviations from the median price are considered outliers and removed. A rolling window approach is used to compute the median and standard deviation for outlier detection.

The daily realized volatility, on the other hand, is computed from the high-frequency 5-min returns. Let $P_{t,i}$ denote the price at the i-th 5-min interval on day t. The logarithmic returns for each interval are defined as:

$$r_{t,i} = \ln\left(\frac{P_{t,i}}{P_{t,i-1}}\right)$$

where $r_{t,i}$ represents the return from time $i-1$ to i on day t.

The 5-min realized variance for day t is calculated by summing the squared returns over all intervals $i = 1, 2, \ldots, N$, where N is the total number of 5-min intervals within the trading day:

$$RV_t = \sum_{i=1}^{N} r_{t,i}^2$$

The realized volatility, which is the square root of the realized variance, is given by:

$$\sigma_t = \sqrt{RV_t}$$

where σ_t is the daily realized volatility for day t.

The use of the 5-min interval is chosen as a balance between capturing relevant market microstructure noise and avoiding excessive noise at higher frequencies (L. Y. Liu et al., 2015).

Finally, beyond the primary focus on stock RV, the study's implications for other financial time series should be considered. Financial markets inherently exhibit universal features like volatility clustering, fat-tailed distributions, and leverage effects (Sewell, 2011; Tsay, 2005). These features are common across various financial time series, including stock prices, exchange rates, and interest rates. While focused on stock volatility, the methodologies and forecasting models used in this research are adaptable to different financial time series (Sezer et al., 2020), suggesting the potential applicability of our findings in broader financial forecasting contexts. However, the unique aspects of different financial time series warrant careful consideration, possibly requiring tailored adjustments or refinements to the applied models and methodologies when extending beyond stock realized volatility (R. Engle, 2004).

4.2. Comparing models

4.2.1. GARCH

Initiated by Bollerslev (1986), the GARCH model marks a significant advancement in the field of financial volatility analysis. Building upon the ARCH model by R. F. Engle (R. F. Engle, 1982), GARCH introduces additional complexity by incorporating past variances, thus offering a more nuanced understanding of volatility dynamics in financial markets.

The standard formulation of the GARCH(1,1) model is given by:

$$\sigma_t^2 = \omega + \gamma \epsilon_{t-1}^2 + \theta \sigma_{t-1}^2 \quad (3)$$

In this expression, σ_t^2 denotes the conditional variance at time t, ω is a constant, ϵ_{t-1} represents the previous time step's error, and σ_{t-1}^2 is the preceding conditional variance. The parameters γ and θ respectively measure recent shocks' impact and volatility's persistence over time.

The inclusion of the GARCH(1,1) model as a reference point in this research stems from its prevalent use and proven efficacy in both academic and practical settings. Its methodological soundness and track record in capturing financial market volatility dynamics establish it as a foundational benchmark for volatility forecasting (BUCCI, 2018; Corsi et al., 2008; Hansen et al., 2011, 2014; Kambouroudis et al., 2016; Sharma and Vipul, 2016). Given its established role and capabilities in volatility modelling, the GARCH model serves as an apt comparator against which to evaluate the performance of more recent, complex models.

4.2.2. HAR

The HAR model, a brainchild of Corsi (2009), marks a pivotal development in understanding volatility in financial time series. Its foundational concept is that market dynamics are shaped by influences spanning various time scales, reflecting the diverse trading activities and information assimilation among market players across different periods (Corsi, 2009).

The HAR model's mathematical framework is as follows:

$$RV_{t+1} = \alpha_0 + \alpha_d RV_t^{(d)} + \alpha_w RV_t^{(w)} + \alpha_m RV_t^{(m)} + \eta_{t+1} \tag{4}$$

In this formula, RV_{t+1} signifies the forecasted realized volatility at time $t+1$, α_0 is a constant, and η_{t+1} denotes the error term. The terms $RV_t^{(d)}$, $RV_t^{(w)}$, and $RV_t^{(m)}$ encapsulate the daily, weekly, and monthly realized volatilities, respectively. The coefficients α_d, α_w, and α_m quantify the influence of these volatilities on the prediction of future volatility levels.

Incorporating the HAR model into this study is informed by its proven track record as a reliable tool in volatility prediction, known for its simplicity and effectiveness (Audrino and Knaus, 2015; Audrino et al., 2018; Corsi et al., 2012; Y. Wang et al., 2016; Yao et al., 2019). Its adeptness at recognizing long-term dependencies and capturing the enduring nature of volatility across time frames makes it particularly relevant for financial data analysis. The model's efficacy in forecasting volatility is further bolstered by extensive empirical validation in prior research (Audrino and Knaus, 2015; Audrino et al., 2018; Corsi, 2009; Corsi et al., 2012; Y. Wang et al., 2016; Yao et al., 2019).

4.2.3. LSTM

The LSTM model, an evolution in recurrent neural networks (RNNs) introduced by Hochreiter and Schmidhuber (Hochreiter and Schmidhuber, 1997), is essential in sequential data analysis, particularly in financial time series forecasting. Its design effectively addresses the challenge of capturing long-term dependencies, a known limitation of traditional RNNs.

LSTMs excel in maintaining information over prolonged periods through a unique arrangement of gates (input, forget, and output) that manage the flow and relevance of information. This structure enables the LSTM to retain or disregard data selectively, enhancing its utility in time-dependent analyses.

Expressed mathematically, an LSTM unit involves:

$$p_t = \sigma(W_p \cdot [h_{t-1}, x_t] + b_p) \tag{5}$$

$$q_t = \sigma(W_q \cdot [h_{t-1}, x_t] + b_q) \tag{6}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{7}$$

$$C_t = p_t * C_{t-1} + q_t * \widetilde{C}_t \tag{8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t * \tanh(C_t) \tag{10}$$

where σ and tanh represent the sigmoid and hyperbolic tangent functions, and W and b denote weights and biases.

The LSTM model is chosen as a benchmark in this study for its proven effectiveness in financial time series analysis (Cao et al., 2019; Mehtab and Sen, 2022; Siami-Namini et al., 2018). Its ability to discern critical information over extended time frames makes it particularly adept at forecasting stock market volatility. The LSTM's adaptability to various data types and specific forecasting tasks, along with its proven superiority in handling complex temporal patterns (Bucci, 2020; Fischer and Krauss, 2018; Souto and Moradi, 2023b), further advocates for its inclusion and comparison with both traditional and advanced forecasting models.

4.2.4. TFT

The TFT model, introduced by Lim et al. (2021), revolutionized time series forecasting by integrating transformer attention mechanisms with elements designed explicitly for temporal data analysis. This model is ingeniously structured to adeptly navigate the complexities of multivariate time series forecasting, especially potent in discerning long-term dependencies and dynamic interrelations within financial market data.

The TFT model's unique capability lies in its interpretation of multiple input features with differing temporal significance. This capability emerges from its innovative architecture, which blends recurrent layers, attention mechanisms, and specialized temporal processing components. Such an arrangement allows the TFT model to identify key features, dynamically concentrate on relevant time periods, and disregard extraneous data. This flexibility is critical in financial markets, where the relevance of information can shift dramatically over time.

This study refrains from delving into the complete mathematical intricacies of the TFT model to maintain focus and clarity. However, a detailed explanation of its architecture and functions can be found in the work of Lim et al. (2021).

The decision to include the TFT model as a benchmark in this research is motivated by its position as a leading-edge solution in machine learning for time series forecasting. The model's proficiency in identifying and utilizing temporal dynamics makes it highly relevant for predicting volatility in financial markets (Frank, 2023a; Hu, 2021; Olorunnimbe and Viktor, 2022; H. Wu et al., 2022).

Moreover, the recent application of the TFT model in financial forecasting and its demonstrated success in various empirical studies highlight its potential as a formidable tool in volatility prediction (Frank, 2023b; Hu, 2021; Lim et al., 2021; Olorunnimbe and Viktor, 2022; H. Wu et al., 2022).

4.2.5. NBEATSx

The NBEATSx model is a groundbreaking development by Olivares et al. (2023) in neural network-driven time series forecasting. Its enhanced architecture makes it adept at handling complex, multifaceted datasets, particularly in finance (Souto and Moradi, 2023a).

Unlike traditional neural networks that rely on recurrent or convolutional structures, NBEATSx utilizes fully connected layers arranged in blocks. Each block is equipped to project future time series values, supported by backward and forward residual connections. This setup allows the model to analyze past trends and predict future changes effectively. Readers seeking a detailed mathematical breakdown of NBEATSx can refer to Olivares et al. (2023).

The inclusion of NBEATSx in this study is due to its innovative approach to time series forecasting (Olivares et al., 2023) and considerable forecast accuracy across various domains, including financial time series (Han et al., 2023; Iftikhar et al., 2023; Marcjasz et al., 2023; Mathonsi and van Zyl, 2021; X. Wang et al., 2022). Its ability to discern complex patterns and integrate external factors makes it highly suitable for financial market analysis, particularly in volatility forecasting where external influences are significant (Souto and Moradi, 2023a, 2023d).

4.2.6. NHITS

The NHITS model, introduced by Challu et al. (2023), represents an innovative step in time series forecasting, utilizing deep learning to improve prediction capabilities. NHITS is particularly distinguished by its focus on constructing hierarchical time series forecasts, enhancing accuracy and computational efficiency, especially in long-term forecasting (Challu et al., 2023). The model achieves this through multi-rate sampling and multi-scale synthesis, effectively balancing computational demands with forecasting accuracy.

Detailed mathematical insights into NHITS can be found in Challu et al. (2023), and for its specific application to stock realized volatility, refer to Souto (2023a).

NHITS has been selected as a benchmark in this research due to its exceptional performance in a variety of forecasting applications (Challu et al., 2023; Z. Chen et al., 2023a,b,c; Y. Liu et al., 2022a,b; Zheng et al., 2023), including its effectiveness in predicting stock realized volatility (Souto, 2023a).

4.2.7. TimesNet

TimesNet marks a breakthrough in handling complex temporal patterns in time series, particularly valuable in financial analysis where predicting nuanced variations is vital (Amo Baffour et al., 2019; Borup and Jakobsen, 2019; Q. Chen and Robert, 2022; W.-J. Chen et al., 2022; Souto and Moradi, 2023b; H. Wu et al., 2022).

Central to TimesNet's approach is its dual focus on intraperiod and interperiod variations in time series (H. Wu et al., 2022). The model dissects short-term oscillations within specific intervals, akin to daily stock market fluctuations, while capturing broader long-term trends that mirror the cyclical movements observed in financial markets over extended periods.

This dual analysis allows TimesNet to comprehend time series data deeply, improving its ability to predict future events affected by short-term and long-term factors. Essentially, TimesNet can simultaneously analyze daily price fluctuations and broader market trends in financial markets. The procedure provides a detailed view of how the market behaves. For mathematical specifics of TimesNet, see H. Wu et al. (H. Wu et al., 2022).

The model's proficiency in adapting to various time scales and managing intricate datasets has solidified its status as a formidable tool in forecasting endeavours (Garza and Mergenthaler-Canseco, 2023; Y. Huang et al., 2024; Wei et al., 2023; J. Wu et al., 2023; Z. Zheng et al., 2023; Zuo et al., 2023), including in the realm of stock realized volatility (Souto, 2024).

4.3. Error metrics

This research employs various error metrics to evaluate the forecasting accuracy of the models under consideration comprehensively. Standard indicators like the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) offer an overview of predictive accuracy (Naser and Alavi, 2021). The RMSE is sensitive to large deviations, serving as a rigorous accuracy gauge, while MAE provides a straightforward, linear assessment of prediction errors. These metrics are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{s \times n} \sum_{j=1}^{s} \sum_{i=1}^{n} \left(y_{i,j} - \widehat{y}_{i,j}\right)^2} \tag{11}$$

$$\mathrm{MAE} = \frac{1}{s \times n} \sum_{j=1}^{s} \sum_{i=1}^{n} y_{i,j} - \widehat{y}_{i,j} \tag{12}$$

where $y_{i,j}$ is the actual observation, $\widehat{y}_{i,j}$ the forecasted value, n the total observations, and s the number of stocks.

Furthermore, the Mean Absolute Percentage Error (MAPE) offers a scale-independent accuracy measure crucial for financial data with varying magnitudes (Naser and Alavi, 2021). It is expressed as:

$$\text{MAPE} = \frac{1}{s \times n} \sum_{j=1}^{s} \sum_{i=1}^{n} \frac{y_{i,j} - \hat{y}_{i,j}}{y_{i,j}} \tag{13}$$

Additionally, the Quasi-likelihood error (QLIKE) is included for its appropriateness in volatility forecasting, accounting for noise robustness (Souto and Moradi, 2023c; C. Zheng et al., 2023):

$$\text{QLIKE} = \frac{1}{s \times n} \sum_{j=1}^{s} \sum_{i=1}^{n} \left(\frac{y_{i,j}}{\hat{y}_{i,j}} - \log\left(\frac{y_{i,j}}{\hat{y}_{i,j}}\right) - 1 \right) \tag{14}$$

These metrics provide a nuanced understanding of the models' forecasting capabilities, allowing for a detailed assessment across different aspects of prediction accuracy and reliability.

4.4. Statistical tests

The validity of the performance assessment of TSMixer in this research is reinforced through a series of statistical tests, each tailored to evaluate distinct facets of model effectiveness and reliability. These tests are pivotal in confirming the forecasting accuracy and dependability of the models. Each model in this study undergoes statistical testing across all employed error metrics — RMSE, MAE, MAPE, and QLIKE. This exhaustive testing approach facilitates a comprehensive appraisal of the models over various aspects of forecasting accuracy.

The research employs the Dynamic Model Confidence Set (MCS) test, applying it to the average daily error of each model. The MCS test, as outlined by Hansen et al. (2011), is a robust technique for comparing multiple models, identifying a group of models with statistically similar predictive performance (Hansen et al., 2011). This test is especially relevant in financial forecasting, where discerning the most effective models is crucial (Chiriac and Voev, 2010; Gallo and Otranto, 2015; Souto and Moradi, 2023c). Souto (2023a) adapted this statistical assessment technique to assess the models in a temporal manner instead of an overall manner by having results for each considered trading month. By employing the dynamic version of MCS, one can statistically evaluate how the relative performance of the considered models changes over time and find interesting patterns that the classical MCS would hide.

Furthermore, the study utilizes the Diebold-Mariano (DM) tests, initially introduced by Diebold and Mariano (Diebold and Mariano, 1995) and refined by Harvey et al. (1997). These tests are conducted for each stock individually for a more granular comparison between the TSMixer and benchmark models. The DM test excels in determening whether the forecasting accuracy of two competing models on a per-stock basis is statistically significantly different, offering valuable insights into the relative performance of the TSMixer model against the other considered models (Chiriac and Voev, 2010; Dutta and Das, 2022; Gallo and Otranto, 2015; Souto and Moradi, 2023c; C. Zheng et al., 2023).

For the third robustness test, T-tests and F-tests are used, based on the Gaussian Distribution nature of the error measures, a fact validated by Anderson-Darling tests (Anderson and Darling, 1952). These tests are instrumental in determining the statistical significance of the error measures' mean and variance differences among various random seed selections (Souto and Moradi, 2023d).

These statistical tests collectively establish a robust framework for assessing the forecasting models in this research, ensuring a thorough and reliable evaluation of their predictive capabilities in the context of financial time series forecasting.

4.5. Robustness tests

This study conducts several robustness tests to ascertain the stability and reliability of the forecasting models under diverse conditions. These tests are pivotal in verifying that the study's conclusions genuinely reflect financial time series dynamics rather than being influenced by specific data setups or modelling assumptions.

The initial robustness test adjusts the training-to-testing data ratio. Deviating from the primary 70 %/30 % split, this test implements an 80 %/20 % split to assess model performance with different data distribution, a typical variation in forecasting research (Joseph, 2022). The aim is to evaluate model consistency with an altered data split.

A second test reduces the training dataset to half its original volume. This scenario mirrors situations like limited data availability for newly listed stocks or in markets lacking extensive historical records (Mizik, 2014; Souto et al., 2023). This reduction challenges the models to derive insights from a more restricted data set.

The third test conducts a sensitivity analysis of the neural networks' random seed selection. By implementing 20 distinct random seeds for each neural network model, this test ensures Gaussian distribution in error measures, setting the stage for T-test and F-test applications. This test highlights the impact of random seed selection on model outcomes, especially given the limited time often allocated for hyperparameter optimization in practice (Crane, 2018; Dodge et al., 2019; Hua et al., 2021; Liukis, 2022). By varying the random seed, the study gauges the models' resilience to the stochastic elements in their training, shedding light on their performance reliability (Crane, 2018; Reimers and Gurevych, 2017).

Finally, the fourth, fifth and sixth robustness tests are the employment of the stock indexes, currency exchanges and commodities datasets respectively, using the 70 %/30 % split. These robustness tests are performed to put the empirical testing results of the main sample and respective conclusions to the test.

These robustness tests substantiate the main findings, ensuring that model performances are not overly influenced by particular data characteristics or method choices. Such thorough validation is critical in financial forecasting, where models have significant decision-making implications (D'Ecclesia and Clementi, 2019; J. Li, 2022; Souto and Moradi, 2023b).

4.6. Models hyperparameters

The selection of optimal hyperparameters is a critical aspect of building effective forecasting models, particularly for complex architectures like TSMixer. To ensure the robustness and reliability of our models, we adopted a systematic approach to hyperparameter tuning, incorporating validation steps and multiple trials to refine the model's performance.

For the benchmark models used in this study, we referenced the hyperparameter analyses conducted in the prior works by Souto (2023a) and Souto (2024). These studies provide comprehensive evaluations of hyperparameter settings for similar models on financial time series data, offering a solid foundation for our initial parameter choices.

Regarding the TSMixer model, we implemented a meticulous hyperparameter optimization process using a validation-based approach. Specifically, we partitioned the original training dataset by reserving 28.5 % of it as a validation set dedicated to hyperparameter tuning. The remaining 71.5 % was used for training the model under various hyperparameter configurations. This validation set was exclusively used during the hyperparameter selection phase and was not involved in the final evaluation of the model to avoid any potential data leakage.

The hyperparameter tuning process involved conducting 40 independent trials, each corresponding to a unique combination of hyperparameters. In each trial, the models are trained on the training portion of the dataset with a specific hyperparameter configuration. We then evaluated its performance on the validation set using the considered error metrics of this paper. The criterion for selecting the optimal hyperparameters is the minimization of the error metrics on the validation set, as it directly reflects the model's accuracy.

To ensure the robustness of the chosen hyperparameters, we conducted an additional 20 trials focusing on the random seed used for initializing the model's parameters. The choice of random seed can influence the convergence and generalization of neural networks due to the stochastic nature of training processes like weight initialization and mini-batch selection. By testing multiple random seeds, we mitigated the risk that our results were contingent upon a particular random initialization, thereby enhancing the reliability of our findings.

Our hyperparameter search procedure mirrors the methodologies employed in Souto (2023a) and Souto (2024), ensuring consistency and comparability with existing literature. By adhering to established practices, we aim to avoid biases that could arise from divergent hyperparameter optimization strategies.

Comprehensive details about the optimal hyperparameters for all neural network models used in this study can be found in Appendix C. Moreover, Appendix B elucidates the hyperparameter search space for TSMixer, detailing the parameters and their respective ranges that were considered to establish the optimal model settings.

5. Results and discussion

This section is divided into the following subsections: Main Sample, Robustness Test 1, Robustness Test 2, Robustness Test 3, Robustness Test 4, Robustness Test 5, and Robustness Test 6. To facilitate the reading of the results, each subsection is composed of four chapters: Error Metrics, DM Tests, Dynamic MCS, and Conclusion (except for subsection Robustness Test 3, which is composed of three chapters: Error Metrics, Statistical Tests, and Conclusion).

5.1. Main sample

5.1.1. Error metrics

Table 1 presents the results of the error measures for the main sample.

In RMSE, the TSMixer model emerges as the most proficient, registering the lowest value of 0.371 %. The result suggests that TSMixer, with its relatively basic architecture, is optimally equipped to predict stock realized volatility, especially larger data fluctuations. Only the HAR model can follow closely, with a RMSE value of 0.377 %. Moving on to MAE, the TSMixer and NBEATSx models both achieved the lowest values of 0.241 %, suggesting that TSMixer, together with NBEATSx, performs superiorly when equal weight is assigned to all potential differences between predictions and actual outcomes. Regarding QLIKE, on the other hand, the superiority of TSMixer over other models is more present, with only TimesNet and NBEATSx having relatively close results.

Regarding MAPE, TSMixer scores poorly, only the third-best model after NBEATSx and NHITS. When paying close attention to the TSMixer architecture, this poor score can be easily explained. Thanks to the use of feature-mixing MLPs, the TSMixer can better exploit

Table 1
Error metrics results.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	0.508 %	0.337 %	4.988 %	28.652 %
HAR	0.377 %	0.248 %	3.244 %	19.879 %
LSTM	0.421 %	0.260 %	3.489 %	20.062 %
TFT	0.407 %	0.268 %	3.289 %	21.738 %
NBEATSx	0.391 %	**0.241 %**	3.109 %	**18.041 %**
NHITS	0.402 %	0.247 %	3.295 %	18.397 %
TimesNet	0.388 %	0.248 %	3.079 %	19.237 %
TSMixer	**0.371 %**	**0.241 %**	**3.056 %**	19.200 %

the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015). Consequently, we can expect a better performance of the model for periods when the average realized volatility is considerably higher, which does not influence the MAPE score as it is a relative error measure, while having a slightly worse performance for calm periods, which does influence the MAPE score more than the other error metrics.

The consistently high-ranking performance of the TSMixer model across all chosen measurements, with a superiority regarding RMSE, MAE and QLIKE, underscores its superior design for volatility prediction. These results can be viewed by some as quite striking given the fact that the TSMixer architecture is relatively less complex than the other considered neural network models, which could indicate that the innovative use of existing neural network models while keeping them simple not only leads to the need of fewer computational resources and the need of less knowledge by stakeholders to understand the underlying model but also leads to more accurate forecasts.

5.1.2. DM tests

Fig. 2 depicts the results of the DM tests for all models compared to the TSMixer model regarding RMSE.

Apart from the HAR model, all the other models yield statistically significantly less accurate forecasts than TSMixer concerning the RMSE error metric. Compared to the HAR model, the TSMixer model is statistically superior to the HAR model for double the number of stocks, whereas the HAR model is superior to TSMixer. Nevertheless, given the small number of stocks where both models are superior to each other, there is a possibility that both models have a similar performance on average, and these statistically significant differences are actually due to noise.

The results of the DM tests for all models in comparison to the TSMixer model regarding MAE can be seen in Fig. 3.

The results for MAE are similar to the results of RMSE, with the difference now that the model that presumably has, on average, a similar forecast power to TSMixer is NBEATSx and not the HAR model anymore, which now is more clearly statistically inferior to the TSMixer model.

Moving to the results of the DM tests considering QLIKE found in Fig. 4, it can be affirmed that TSMixer yields statistically significantly more precise forecasts than all models besides NBEATSx and TimesNet. These models likely have, on average, a similar forecast power to the TSMixer model.

Lastly, Fig. 5 presents the DM test results for MAPE. For this error metric, TSMixer yields statistically significantly more accurate forecasts than the GARCH, HAR, LSTM and TFT models while having a similar forecast power to TimesNet and being statistically inferior to NBEATSx and NHITS.

The results of the DM tests only confirm the results and conclusions drawn from the error metrics chapter, showcasing the potential of a model with such a simple architecture as the TSMixer.

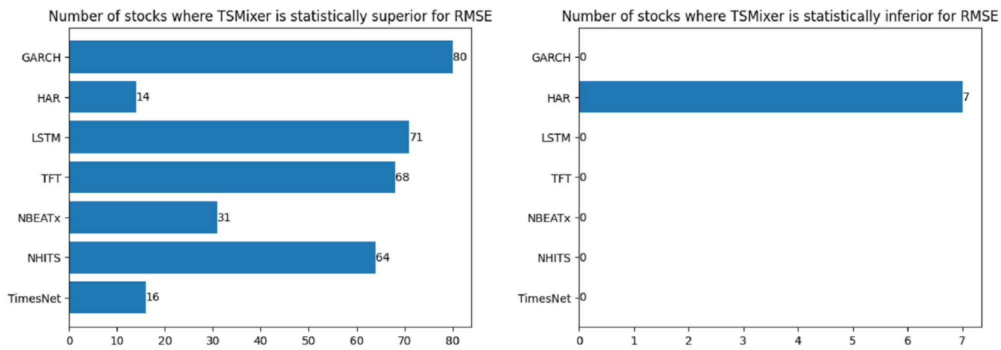

Fig. 2. DM tests results for RMSE (Main sample).

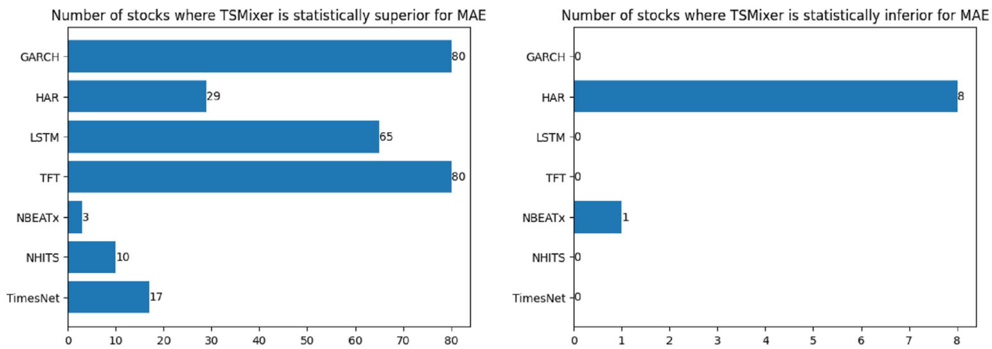

Fig. 3. DM tests results for MAE (Main sample).

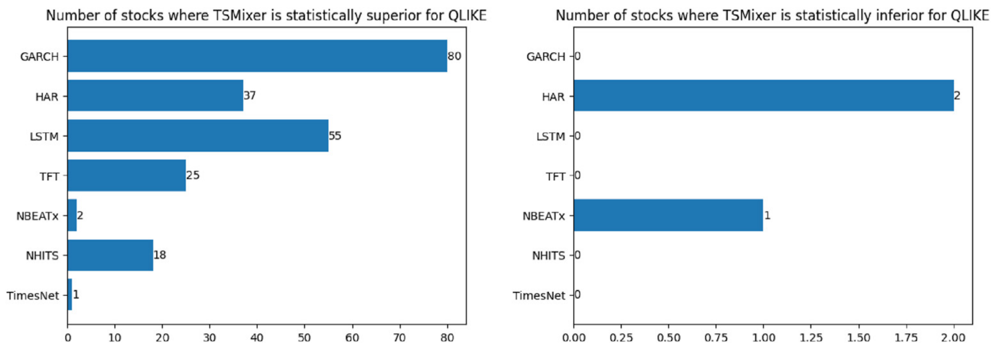

Fig. 4. DM tests results for QLIKE (Main sample).

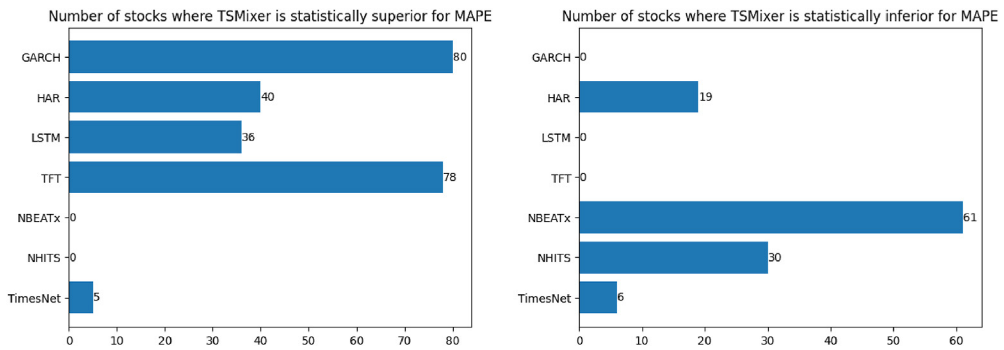

Fig. 5. DM tests results for MAPE (Main sample).

5.1.3. Dynamic MCS

Fig. 6 depicts the results of the Dynamic MCS tests. As depicted in Fig. 6, TSMixer emerges as the most superior model across QLIKE, MAE, and RMSE, as the total time spent as the top model is the largest. This superior performance is maintained during relative stability and unpredictability periods, such as the COVID-19 pandemic. Interestingly, the HAR model briefly yet consistently across all error metrics outperforms all other models during the first two months of 2020, corresponding to the pandemic's initial phase. This result could suggest that the HAR model's architecture possesses a resilience that enables it to adapt effectively during the initial stages of turbulent periods. When examining the MAPE metric, TSMixer is outperformed in the months preceding the pandemic. However, its performance is superior during the pandemic, indicating its superior resilience to unusual market behaviour even when it is not the best model in the given metric.

5.1.4. Conclusion

In conclusion, the results of the main sample indicate the superiority of TSMixer to other models when considering RMSE, MAE and QLIKE, despite its simple neural network architecture compared to the other neural network models. These results suggest that the employment of existing neural network models in a simplified form not only necessitates reduced computational resources and diminishes the complexity for stakeholders to comprehend the model's mechanics but also culminates in enhanced forecasting precision.

Alternatively, the results of the main sample indicate that the combination of time-mixing and feature-mixing MLP approaches is a novel and better approach to modelling the realized volatility of a stock portfolio or stock index for forecasting tasks. This can explain the poor MAPE results of the TSMixer since the feature-mixing MLPs incorporated into the model enable it to more effectively capture the financial tail dependence phenomenon, allowing for improved forecasting of realized volatility, particularly in the lead-up to or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015). As a result, the model is expected to perform better when the average realized volatility is significantly higher, a scenario that has minimal impact on the MAPE score due to its nature as a relative error measure. In contrast, the model's performance may decline slightly during calmer periods, which disproportionately affects the MAPE score compared to other error metrics.

5.2. Robustness test 1

5.2.1. Error metrics

The results of the error metrics for Robustness Test 1 can be found in Table 2.

When using a different data split (i.e., 80 %/20 % instead of 70 %/30 %), the results remain relatively unchanged, except for the results of MAPE, as now the TSMixer model is the best model concerning this error metric. This result confirms the hypothesis that the

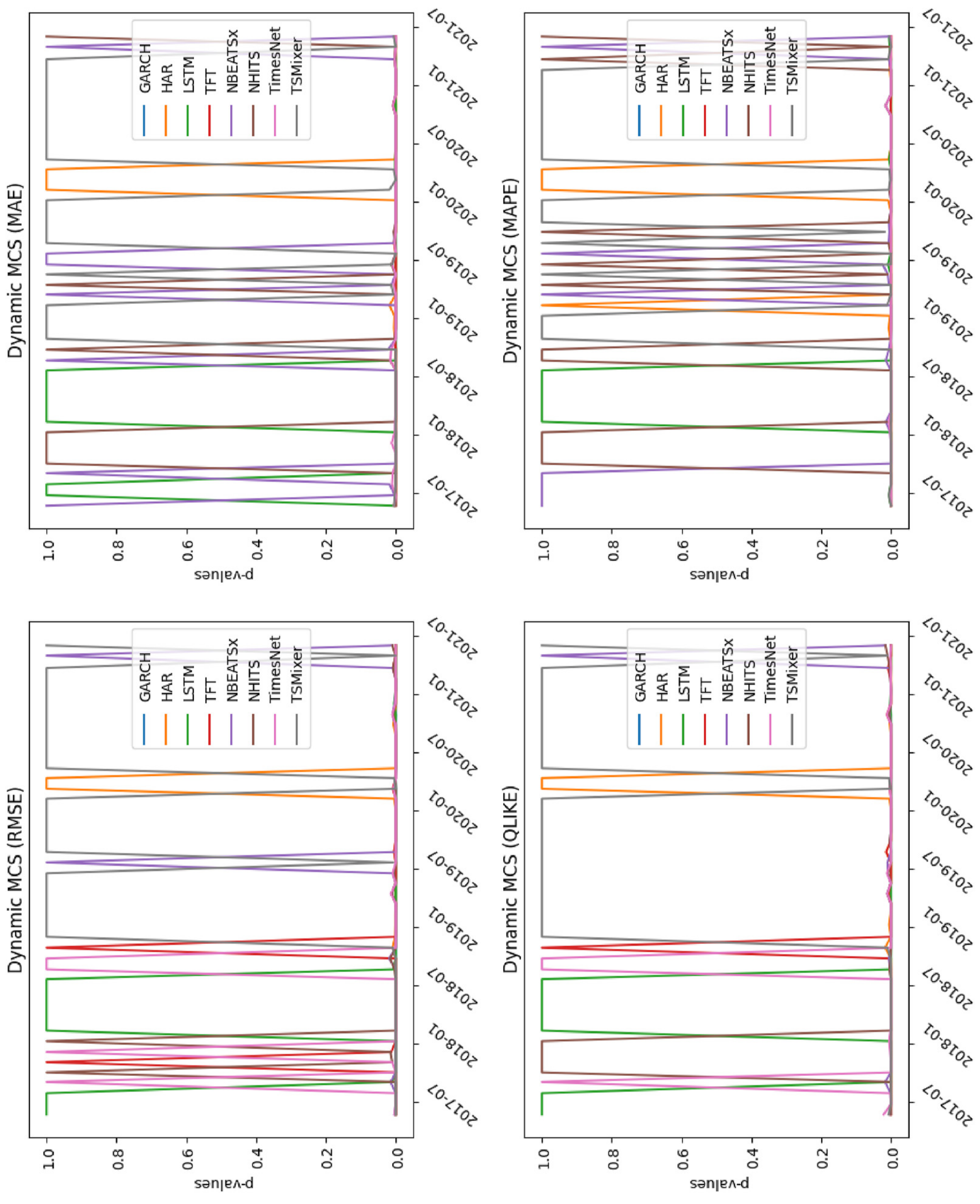

Fig. 6. MCS results (Main sample).

Table 2
Error metrics results for robustness test 1.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	0.560 %	0.363 %	4.854 %	27.184 %
HAR	0.404 %	0.268 %	3.111 %	19.505 %
LSTM	0.563 %	0.327 %	5.044 %	21.130 %
TFT	0.453 %	0.299 %	3.634 %	21.449 %
NBEATSx	0.447 %	0.287 %	3.677 %	19.953 %
NHITS	0.467 %	0.296 %	3.806 %	20.585 %
TimesNet	0.451 %	0.304 %	3.692 %	22.220 %
TSMixer	**0.399 %**	**0.264 %**	**2.922 %**	**19.214 %**

superiority of TSMixer over other models is not due to the choice of data split but instead due to its time-mixing-and-feature-mixing architecture that is more suitable to model and forecast stock realized volatility. Specifically, by changing the data split to 80 %/ 20 %, we end up giving more weight to the observations during the COVID-19 crisis, where the average volatility was considerably high. This favours the performance of the TSMixer as thanks to the use of feature-mixing MLPs, the TSMixer can better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

Additionally, this further strengthens the hypothesis that employing existing neural network models in a simplified form necessitates reduced computational resources, diminishes the complexity for stakeholders to comprehend the model's mechanics, and culminates in enhanced forecasting precision.

5.2.2. DM tests

The results of the DM tests for RMSE can be seen in Fig. 7. Similarly to the main sample results, the superiority of TSMixer over all models besides the HAR model is clear, albeit now this superiority is even more explicit. Regarding the HAR model, there is no statistically significant difference in the forecast accuracy of the HAR model and the TSMixer model, a result similar to the main sample results.

Fig. 8 depicts the results of the DM tests for MAE. In the main sample, TSMixer is significantly superior to GARCH, LSTM, TFT, NHITS, and TimesNet but lesser to HAR and NBEATx, although still superior. In Robustness Test 1, TSMixer is statistically superior for at least 77 stocks and statistically inferior for 0 stocks for all models except for HAR. The superiority of the TSMixer model over the HAR

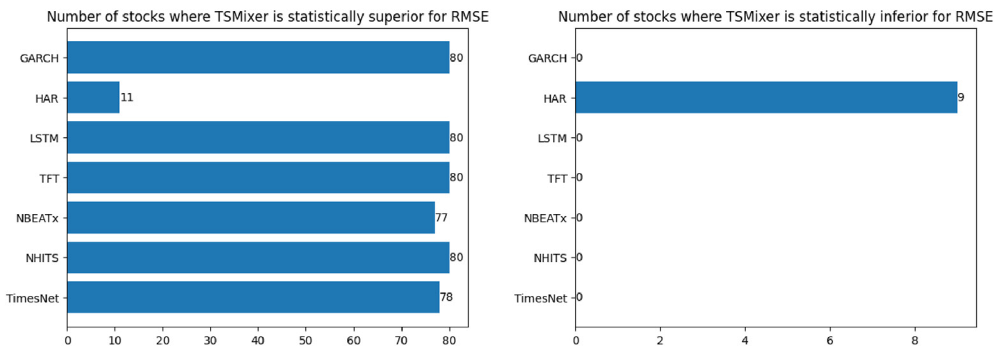

Fig. 7. DM tests results for RMSE (robustness Test 1).

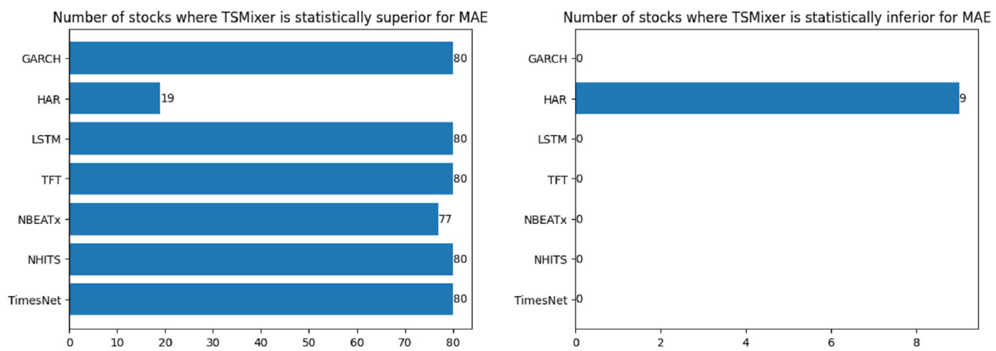

Fig. 8. DM tests results for MAE (robustness Test 1).

model decreased, now being superior for 9 stocks and inferior for 19, indicating that there is likely no statistically significant difference in the forecast power of both models regarding the MAE error metric.

The results of the DM tests for QLIKE can be found in Fig. 9. In contrast to the results of the main sample, the superiority of the TSMixer model over all other models is much more explicit, with the exception of the HAR model, over which the superiority of the TSMixer remained the same, albeit still clear. This result is likely since the use of the data split 80 %/20 % leads to allocating more relative weight to the errors during the turbulent periods caused by COVID-19, during which the TSMixer was superior to all other models, as already discussed in the results of the Dynamic MCS, tests for the main sample.

Fig. 10 shows the DM test results for the MAPE error measure. In the main sample, TSMixer proved only significant statistical superiority for GARCH, LSTM, and TFT and had slightly superior performance relative to HAR. For NBEATSx, NHITS and TimesNet, TSMixer proved inferior. However, in the test, TSMixer proved significant statistical superiority for all models except for HAR. HAR now proved superior for 29 stocks and inferior for 31 stocks, thus comparable to TSMixer. Hence, TSMixer can outperform other models regarding MAPE during financially turbulent periods.

The results of the DM tests further confirm the results and conclusions drawn from the main sample error metrics chapter with additional insight into the superiority of TSMixer over other models, besides the HAR model, during highly volatile periods, showcasing the potential that a model with such a simple architecture as the TSMixer has, especially during turbulent periods.

5.2.3. Dynamic MCS

Fig. 11 presents the Dynamic MCS results. In the revised dataset distribution, TSMixer seemingly exhibits a notable improvement across all error measures, albeit this improvement is likely since the use of the data split 80 %/20 % leads to allocating more relative weight to the errors during the turbulent periods caused by COVID-19, during which the TSMixer was superior to all other models. Similar to the main sample Dynamic MCS results, HAR remains the best model during the initial stages of the pandemic.

5.2.4. Conclusion

In summation, the outcomes of Robustness Test 1 provide substantive reinforcement to the hypothesis that the utilization of extant neural network models in a streamlined configuration not only requires fewer computational resources and reduces the intricacy for stakeholders in grasping the operational principles of the model but also leads to a marked improvement in the precision of forecasts, particularly during periods of heightened volatility. Conversely, these findings from Robustness Test 1 suggest that the integration of time-mixing and feature-mixing MLP methodologies represents a pioneering and more efficacious strategy for modeling the realized volatility of a stock portfolio or stock index in forecasting endeavors, especially in times of significant market volatility. This can be explained when having a closer look the TSMixer's architecture as its feature-mixing MLPs allow it to better capture the financial tail

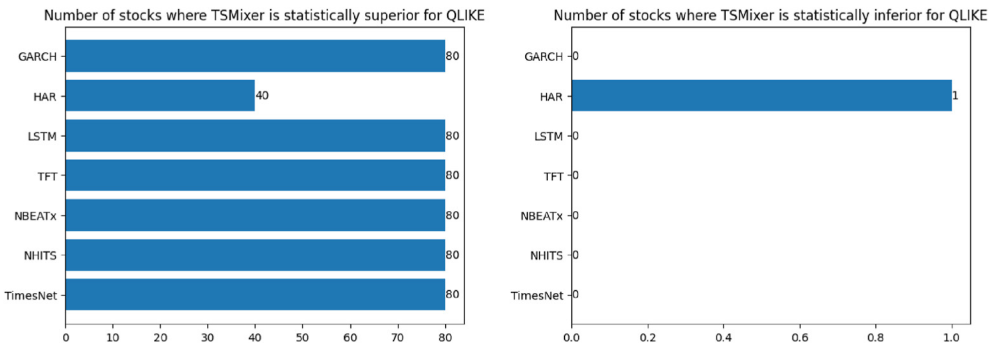

Fig. 9. DM tests results for QLIKE (robustness Test 1).

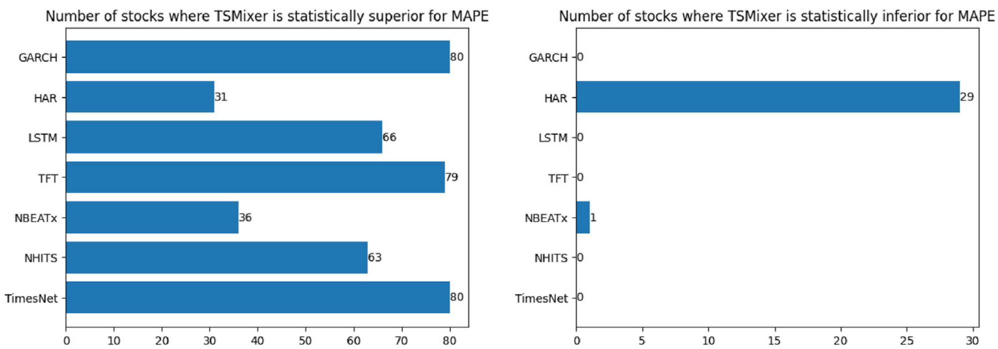

Fig. 10. DM tests results for MAPE (robustness Test 1).

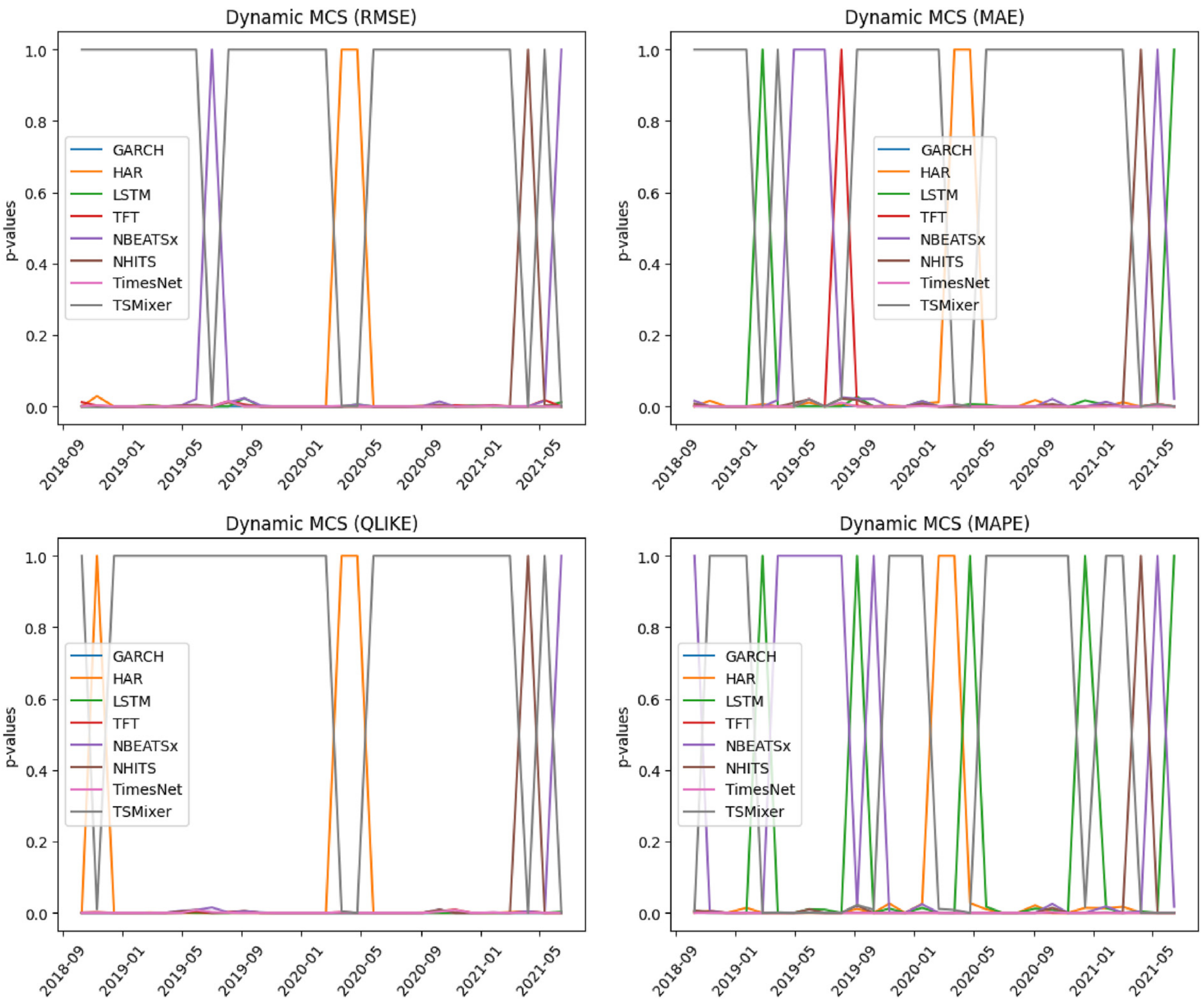

Fig. 11. MCS results (robustness Test 1).

dependence phenomenon, improving its ability to forecast realized volatility in the lead-up to or during periods of financial turbulence (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

5.3. Robustness test 2

5.3.1. Error metrics

The results of the error measures for Robustness Test 2 can be found in Table 3.

When confronted with limited historical data, TSMixer still has a relatively considerable performance. The model achieves the best RMSE and MAE values and virtually has the same QLIKE value as the best model, TFT. Considering MAPE, on the other hand, TSMixer fails to be in the top three, being behind NBEATSx, NHITS, and TFT. This can be explained by the fact that the advantage of the TSMixer model comes from its use of feature-mixing MLPs, which allows it to better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015). As a result, a better performance of the model for periods when the average realized volatility is considerably higher, which does not influence the MAPE score as it is a relative error measure, can be expected while having a slightly worse performance for calm periods, which does influence the MAPE score more than the other error metrics.

Interesting, TFT's performance even increased with limited historical data for training, which could mean that the TFT model tends to give too much importance, in comparison to other models, to parameters that are used to model data patterns in early stages of the provided historical data; patterns that over time become not applicable and relevant anymore due to the changes in stock market structure (Tsay, 2005).

These results further strengthen the hypothesis that the combination of time-mixing and feature-mixing MLP approaches is a novel and better approach to model stock realized volatility of a stock portfolio or index for forecasting tasks, even when only limited historical data for training is available.

Table 3
Error metrics results for robustness test 2.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	0.502 %	0.316 %	4.839 %	26.123 %
HAR	0.394 %	0.250 %	3.374 %	19.578 %
LSTM	0.581 %	0.295 %	5.408 %	19.337 %
TFT	0.397 %	0.246 %	**3.109 %**	18.697 %
NBEATSx	0.399 %	0.244 %	3.205 %	**18.175 %**
NHITS	0.413 %	0.250 %	3.421 %	18.499 %
TimesNet	0.394 %	0.251 %	3.183 %	19.537 %
TSMixer	**0.378 %**	**0.241 %**	3.110 %	19.237 %

5.3.2. DM tests

Fig. 12 presents the DM tests results for RMSE. In the context of RMSE, TSMixer displayed increased competitiveness relative to the LSTM, HAR, NBEATSx, and NHITS models, demonstrating statistically superior performance for a greater number of stocks when trained on limited data. However, compared to TFT and TimesNet, TSMixer's competitiveness diminished, especially regarding TFT. Moreover, it is essential to note that The HAR model was the sole model to exhibit superior performance to TSMixer in any stock, outperforming TSMixer in 3 stocks. However, TSMixer still performed better for 38 stocks, a significant increase from 14 stocks in the initial sample.

The DM test results for MAE are depicted in Fig. 13. In the context of MAE, TSMixer either maintained or enhanced its competitiveness when juxtaposed with most models, GARCH, HAR, LSTM, NHITS, and TimesNet. However, concerning TFT and NBEATSx, the reduction in the training dataset size led to a decrease in TSMixer's competitiveness, leading to the conclusion that in the context of limited historical data available, there is no statistically significant difference in the forecast power of TSMixer and these models.

Fig. 14 shows the results of the DM tests for QLIKE. Regarding QLIKE, TSMixer initially demonstrated superiority over all models in the main sample. With the limit in training data, its superiority was predominantly amplified over the models GARCH, HAR, LSTM, and NHITS. Nonetheless, the TSMixer model now does not yield statistically significantly more accurate forecasts than the TFT model anymore. At the same time, it continued to have forecast power similar to NBEATSx and TimesNet.

The DM test results for MAPE are in Fig. 15. Compared to the results of the main sample, the relative performance of TSMixer decreased significantly. In the context of limited historical data available, TSMixer is statistically superior only to the GARCH model while likely equal to the HAR, LSTM, TFT, and TimesNet and inferior to NBEATSx and NHITS.

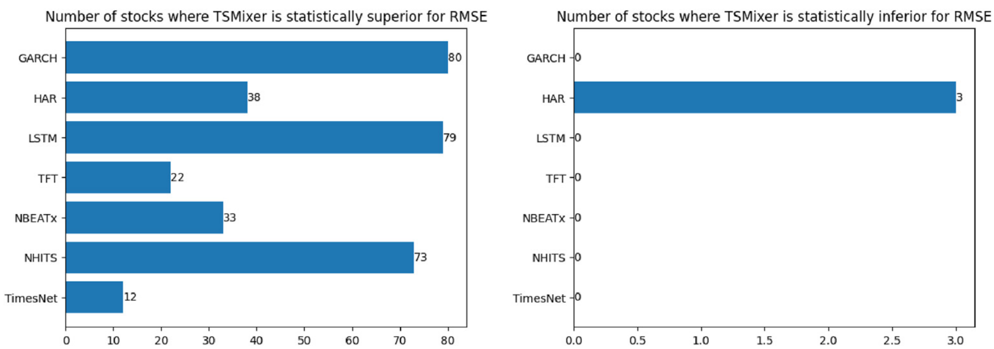

Fig. 12. DM tests results for RMSE (robustness Test 2).

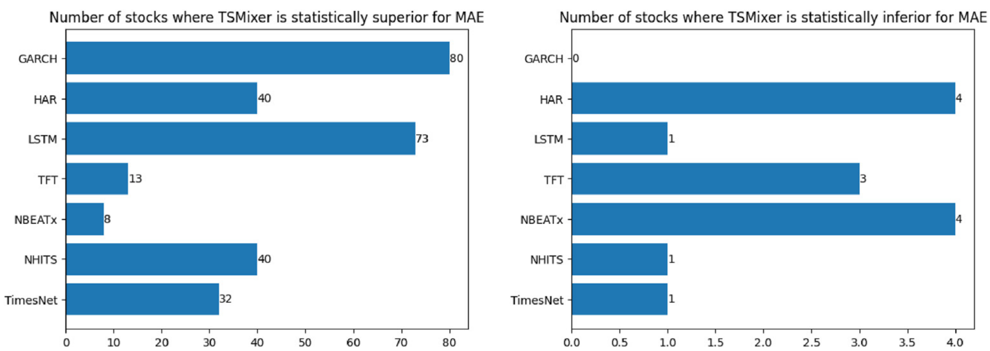

Fig. 13. DM tests results for MAE (robustness Test 2).

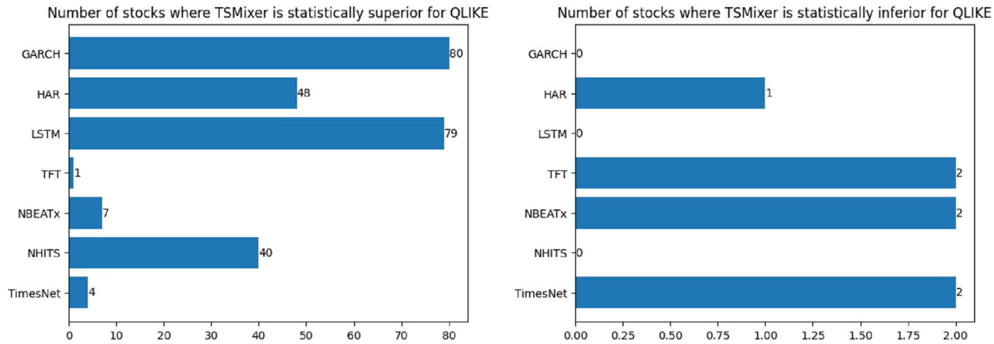

Fig. 14. DM tests results for QLIKE (robustness Test 2).

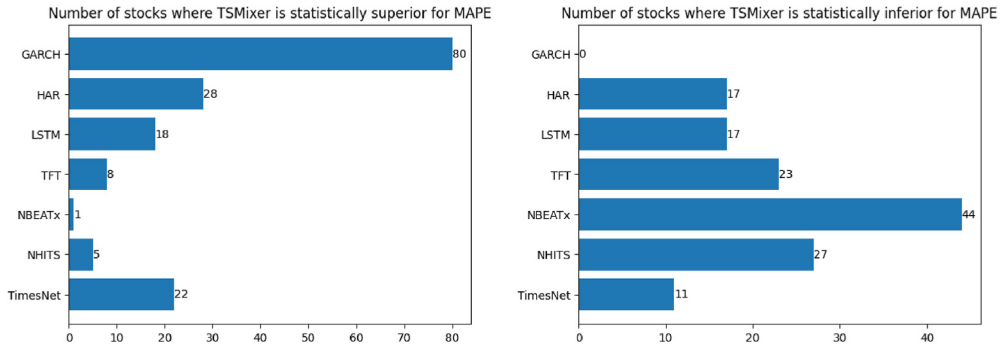

Fig. 15. DM tests results for MAPE (robustness Test 2).

The results of the DM tests indicate that when only limited historical data for training is available, the combination of time-mixing and feature-mixing MLP approaches is not necessarily a better approach to model stock realized volatility of a stock portfolio or index for forecasting tasks in comparison to the time-series stack approach used by NBEATSx.

5.3.3. Dynamic MCS

Fig. 16 depicts the resuls of the Dynamic MCS tests. Regarding RMSE and QLIKE, TSMixer's performance with a reduced training sample size is similar to its performance in the main sample, being the best model for approximately half of the time examined. Interestingly, HAR no longer illustrates the consistent, robust ability to predict volatility in the first months of the COVID-19 pandemic. TSMixer now serves as the most suited model for these months regarding the metrics mentioned above. As for MAPE and MAE, TSMixers' performance dropped noticeably, performing as the top model with less consistency and for a shorter total period, albeit continuing to be the best model during the first and highly financially turbulent year of the COVID-19 pandemic.

5.3.4. Conclusion

In conclusion, the results derived from Robustness Test 2 robustly affirm the proposition that employing existing neural network architectures in a streamlined manner not only diminishes the computational exigencies and simplifies the conceptual understanding required by stakeholders but also notably augments the accuracy of predictions, particularly amidst periods of increased market volatility. Conversely, these insights from Robustness Test 2 might also imply that incorporating time-mixing and feature-mixing within MLP frameworks constitutes an innovative and more effective approach for predicting the realized volatility of stock portfolios or stock indices, especially during marked market fluctuations. However, the findings from Robustness Test 2 also suggest that in scenarios characterized by “normal” market conditions and limited historical data for training, this hypothesis may not be as pronounced, given the comparable predictive efficacy observed between TSMixer and NBEATSx models.

5.4. Robustness test 3

5.4.1. Error metrics

Table 4 shows the means and standard deviations of the error metrics for the sensitivity analysis of the random seed choice.

Regarding RMSE, TSMixer outperforms all other models, achieving a mean of 0.369 % and a standard deviation of 0.002 %. The mean and the standard deviation are the lowest recorded, thus indicative of the model's desirableness and stability. NBEATSx is the most consistent and attractive model outside of TSMixer, with a mean of 0.390 % and equivalent standard deviation. Therefore, not only does TSMixer achieve, on average, better RMSE values than all models, but it is as robust to changes in the selection of random seed as the current most robust neural network model (Souto and Moradi, 2023d).

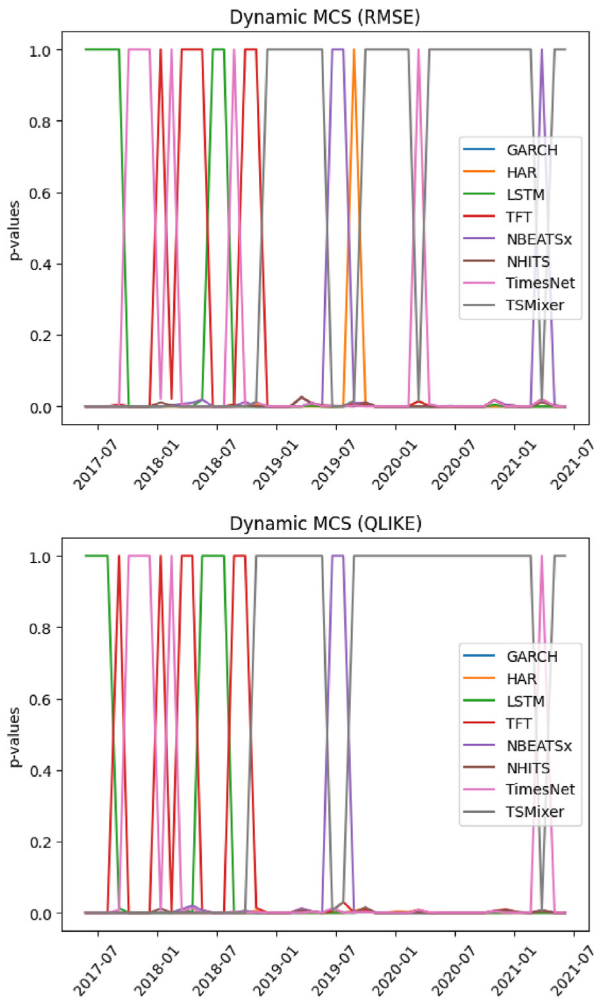

Fig. 16. MCS results (robustness Test 2).

Regarding MAE, TSMixer and NBEATSx share the same mean result of 0.241 %, whereas they have 0.002 % and 0.001 % as standard deviations, respectively. Thus, NBEATSx is superior to TSMixer by a small margin. NHITS, although having a higher mean of 0.247 %, is equivalent in standard deviation to NBEATSx, thus proving more consistent to all but NBEATSx.

Moving to QLIKE, TSMixers's mean is 3.039 % with a standard deviation of 0.008 %. This is superior performance relative to all other models. The trailing models have, at best, a mean value of 3.123 % (TFT) and a standard deviation of 0.025 % (TimesNet). These results indicate a clear robustness superiority of TSMixer over benchmark models when considering the QLIKE error measure.

As for the MAPE, NBEATSx outperforms all other models with an average of 18.058 % and a standard deviation of 0.126 %. NHITS follows closely with a mean of 18.519 % and a standard deviation of 0.283 %. Again, regarding mean, TSMixer is placed third, with an average of 19.272 % and standard deviation of 0.238 %. However, when referring to standard deviation, TSMixer is the second-best model. Therefore, even though NBEATSx is better for MAPE for both metrics, TSMixer remains competitive, ranking as the second or third most suitable model, depending on the specific metric being considered.

In summary, the results of Table 4 evidence the forecast power and robustness to changes in the selection of random seed of the TSMixer model, showing that not only is the TSMixer model's forecast power superior to the other benchmark models, but it also has a comparative robustness to the current most robust neural network model (Souto and Moradi, 2023d).

5.4.2. *Statistical tests*

Table 5 presents the results of the T-tests and F-tests based on the results of the sensitivity analysis of the random seed choice.

Concerning LSTM, TFT, and TimesNet, we can affirm that the superiority in both forecast power and robustness of TSMixer is statistically significant for all error measures, except for the MAPE error metric for TFT when considering the standard p-value threshold of 0.05. Moving to NHITS, the results of the T-tests evidence the forecast power superiority of TSMixer regarding RMSE, MAE, and QLIKE and inferiority regarding MAPE while indicating the robustness superiority of TSMixer concerning the RMSE and QLIKE error measures. Finally, considering NBEATSx, TSMixer has error measure means that are statistically significantly lower than NBEATSx's

Table 4
Sensitivity analysis results for error metrics.

Model	Metric	Mean	Std Dev
LSTM	RMSE	0.431 %	0.019 %
	MAE	0.261 %	0.009 %
	QLIKE	3.662 %	0.312 %
	MAPE	19.631 %	0.686 %
TFT	RMSE	0.397 %	0.005 %
	MAE	0.252 %	0.008 %
	QLIKE	3.123 %	0.101 %
	MAPE	19.721 %	1.024 %
NBEATSx	RMSE	0.390 %	**0.002 %**
	MAE	**0.241 %**	**0.001 %**
	QLIKE	3.127 %	0.025 %
	MAPE	**18.058 %**	**0.126 %**
NHITS	RMSE	0.402 %	0.004 %
	MAE	0.247 %	**0.001 %**
	QLIKE	3.284 %	0.057 %
	MAPE	18.519 %	0.283 %
TimesNet	RMSE	0.396 %	0.005 %
	MAE	0.253 %	0.005 %
	QLIKE	3.145 %	0.046 %
	MAPE	19.757 %	0.658 %
TSMixer	RMSE	**0.369 %**	**0.002 %**
	MAE	**0.241 %**	0.002 %
	QLIKE	**3.039 %**	**0.008 %**
	MAPE	19.272 %	0.238 %

Table 5
Statistical test results.

Model	Metric	T-statistic	F-statistic
LSTM	RMSE	14.663***	140.569***
	MAE	9.744***	33.632***
	QLIKE	8.936***	1414.845***
	MAPE	2.210**	8.306***
TFT	RMSE	25.490***	8.193***
	MAE	6.633***	22.540***
	QLIKE	3.695***	149.318***
	MAPE	1.908*	18.506***
NBEATSx	RMSE	36.521***	1.648
	MAE	0.419	2.877***
	QLIKE	14.832***	9.315***
	MAPE	−20.138***	3.551***
NHITS	RMSE	36.175***	5.545***
	MAE	14.082***	1.762
	QLIKE	19.060***	47.391***
	MAPE	−9.108***	1.413
TimesNet	RMSE	22.206***	10.645***
	MAE	9.845***	11.322***
	QLIKE	10.141***	31.247***
	MAPE	3.097***	7.645***

error measure means for RMSE and QLIKE while having the MAPE error measure mean that is statistically significantly higher than NBEATSx. Regarding the robustness of these models, on the other hand, we can affirm that NBEATSx is statistically significantly more robust than TSMixer for MAE and MAPE while being statistically significantly equally and less robust for RMSE and QLIKE, respectively.

5.4.3. Conclusion

Based on the afore-discussed results, it can be concluded that TSMixer has a superior forecast power than other benchmark models, as already seen in earlier subsections, while having robustness to changes in the selection of random seed that is better than other benchmark models, except for NBEATSx, where it has a similar, better, or worse performance depending on the choice of error metric. Therefore, given possible different conclusions when comparing TSMixer's robustness to NBEATSx's robustness depending on the choice of error measure, it can be affirmed that these models have a similar robustness level.

Now abstracting these results to a higher level, the results of Robustness Test 3 offer considerable support to the proposition that the utilization of extant neural network models in a streamlined and inventive manner not only demands fewer computational resources and reduces the complexity for stakeholders in understanding the model's operational framework, but also results in improved accuracy of forecasts. Conversely, the outcomes derived from the principal sample suggest that the amalgamation of time-mixing and feature-mixing

MLP techniques represents a pioneering and more efficacious strategy for modelling the realized volatility of a stock portfolio or stock index in forecasting endeavours, especially before or during financially turbulent times thanks to the model's capability to better model the financial tail dependence phenomenon (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

5.5. Robustness test 4

5.5.1. Error metrics

Table 6 presents the error measures results for Robustness Test 4.

It can been seem that the TSMixer model is not anymore the top-performing model for most error metrics, albeit it is still in the top two for RMSE and QLIKE. Interestingly, now the TFT model is one of the best models for almost all error measures, showing the importance of testing various datasets with different financial securities types for realized volatility model forecasting evaluation.

This could be explained by two hypothesis. The first one is that the TSMixer model works better for forecasting realized volatiltiy of individual stocks due to their unique characteristics that they do not share with stock indexes (Z. Gu and Ibragimov, 2018; Souto and Moradi, 2024). The other hypothesis is that the TSMixer model performs relatively bad for the stock indexes dataset and the following datasets as they do not have a great number of financial securities being forecasted (they have six to seven securities per dataset). Hence, the model cannot properly explore its feature-mixing MLPs, which allows it to better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

5.5.2. DM tests

Moving to the DM tests, Fig. 17 shows the DM tests results regarding RMSE.

Based on the results, it can be concluded that though the TSMixer model seemed to now be inferior to the NBEATSx model, it is actually statistically inferior for only one out of seven stock indexes. Yet, its RMSE superiority over other models is not statistically significant for most competing models, meaning that the TSMixer model is a competitive but not superior model for this dataset composed of stock indexes. Moving to MAE, its DM tests results can be found in Fig. 18.

The results demonstrate that the TSMixer model is statistically superior to the GARCH, HAR and LSTM models while being statistically inferior to TFT, NBEATSx, and NHITS. Thus, for MAE, the TSMixer would not be a preferred model for forecasting stock index realized volatility.

Fig. 19 depicts the DM tests results regarding QLIKE.

It can be seen that while the TSMixer is not statistically inferior to any model, it is statistically superior to the GARCH, HAR and LSTM models. Thus, the TSMixer model is a competitive but not the superior model for this dataset concerning QLIKE.

The DM tests results for MAPE can be seen in Fig. 20.

Table 6
Error metrics results for robustness test 4.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	1.690 %	1.120 %	32.227 %	171.097 %
HAR	0.321 %	0.178 %	4.033 %	23.276 %
LSTM	0.338 %	0.194 %	4.727 %	25.674 %
TFT	0.318 %	0.170 %	3.891 %	21.283 %
NBEATSx	**0.314 %**	**0.165 %**	3.937 %	**20.155 %**
NHITS	0.322 %	0.168 %	4.137 %	20.534 %
TimesNet	0.326 %	0.175 %	3.971 %	21.710 %
TSMixer	0.317 %	0.174 %	**3.883 %**	23.014 %

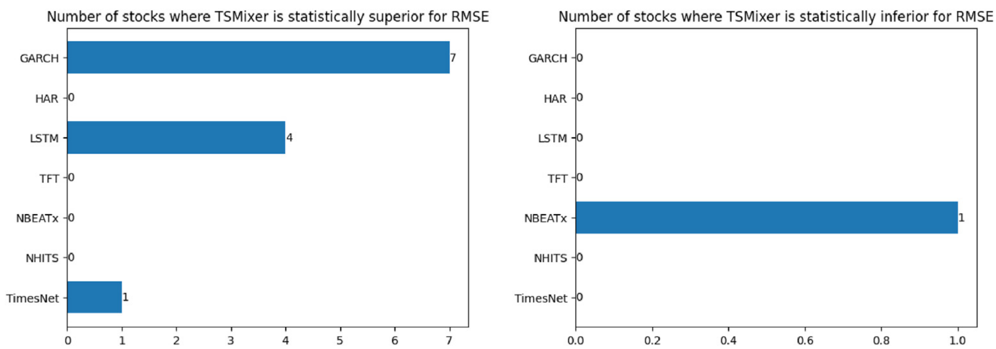

Fig. 17. DM tests results for RMSE (robustness Test 4).

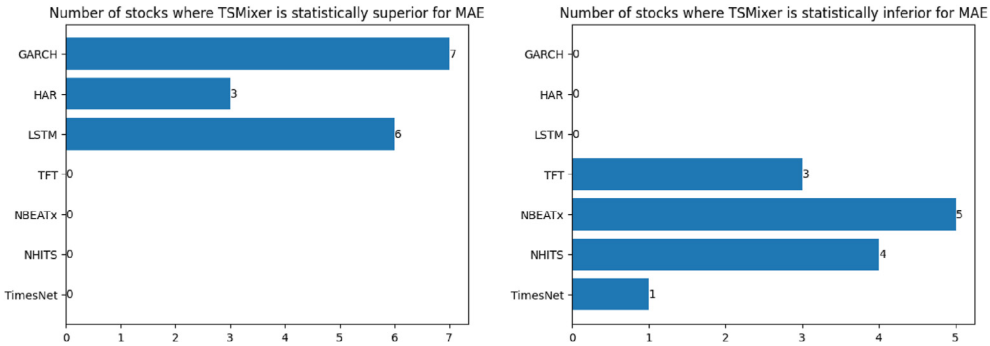

Fig. 18. DM tests results for MAE (robustness Test 4).

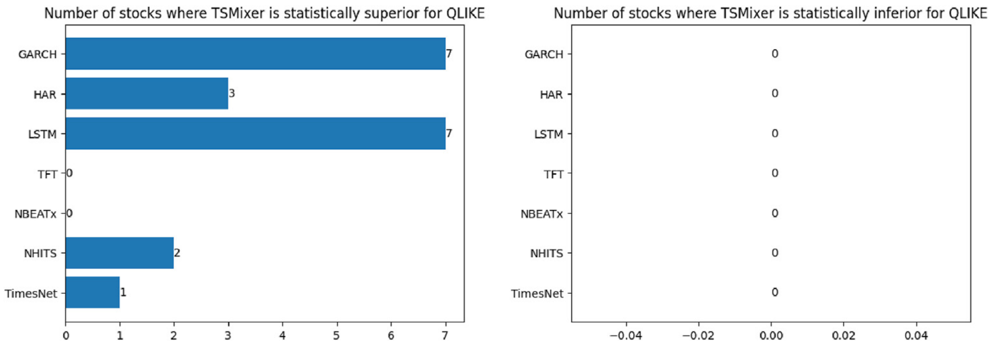

Fig. 19. DM tests results for QLIKE (robustness Test 4).

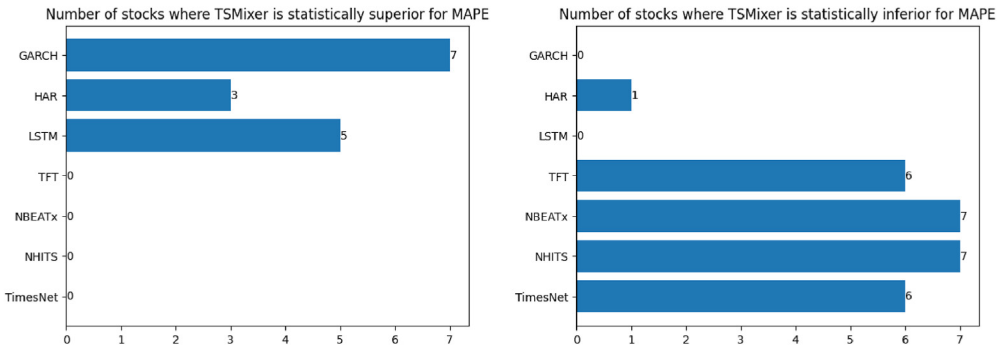

Fig. 20. DM tests results for MAPE (robustness Test 4).

The results demonstrate that the TSMixer model is statistically superior to the GARCH, HAR and LSTM models while being statistically inferior to all other models. Hence, the TSMixer would certainly not be a preferred model for forecasting stock index realized volatility for those prioritizing MAPE.

5.5.3. Dynamic MCS

The results of the Dynamic MCS can be found in Fig. 21. It can be seen that the position of the most-performing model often changes, with NBEATSx and NHITS frequently being present in this position for all error measures. Further, no interesting observations nor conclusions can be drawn from the Dynamic MCS results.

5.5.4. Conclusion

The results of Robustness Test 4 demonstrate that TSMixer, while remaining competitive, is not the top-performing model when applied to a dataset of stock indexes. Although TSMixer ranked highly for RMSE and QLIKE, it was outperformed by models like TFT, NBEATSx, and NHITS in several key metrics, particularly MAE and MAPE. The DM tests confirm that TSMixer is statistically inferior to some of these models, suggesting that while TSMixer remains viable for certain error measures, it is not the optimal choice for stock index realized volatility forecasting. These findings highlight the importance of testing models across different datasets, as performance can vary significantly depending on the characteristics of the financial instruments being forecasted.

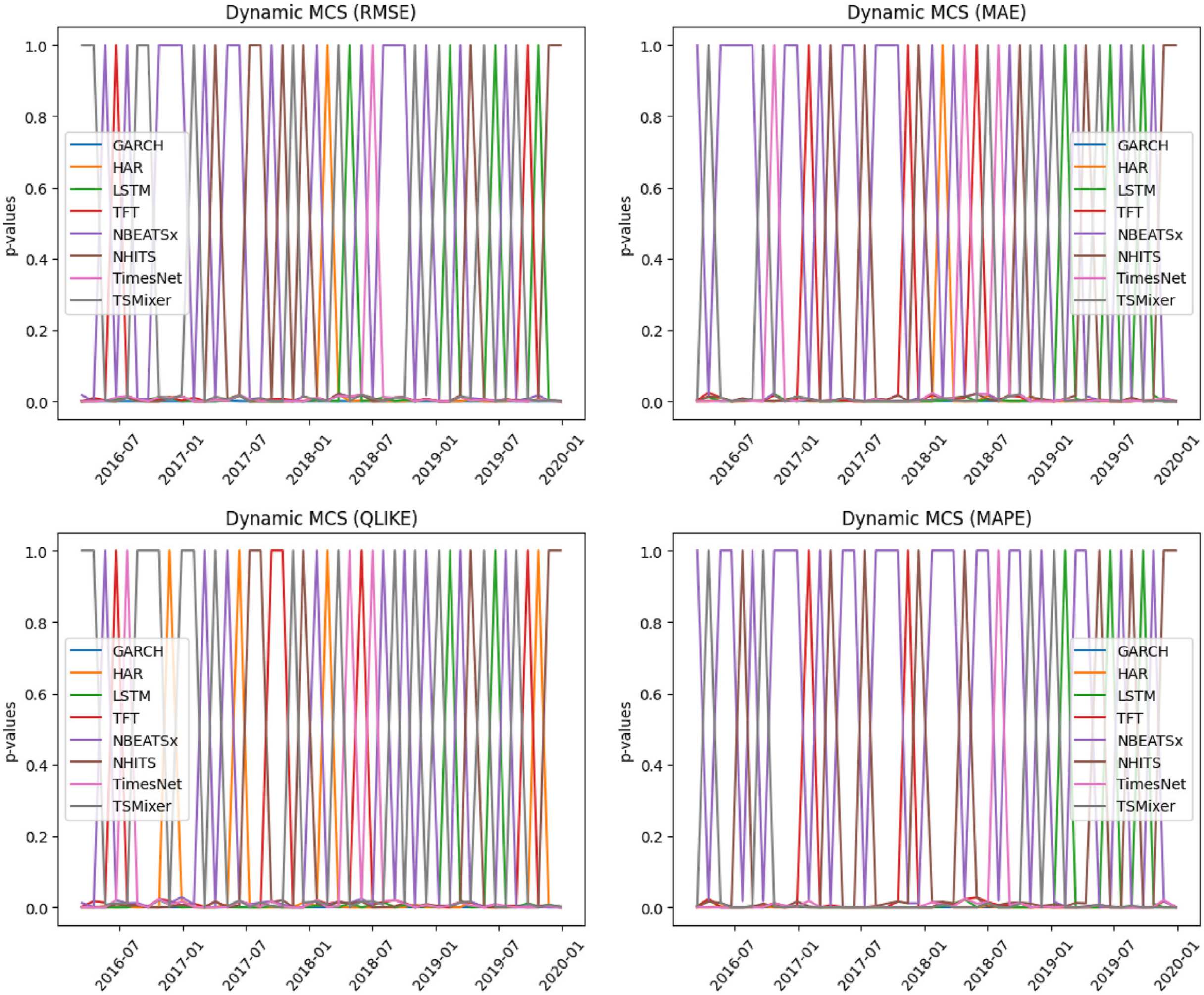

Fig. 21. MCS results (robustness Test 4).

These findings can be interpreted through two possible hypotheses. The first suggests that the TSMixer model is more effective in predicting the realized volatility of individual stocks, likely because these stocks possess distinct characteristics that are not shared with stock indexes (Z. Gu and Ibragimov, 2018; Souto and Moradi, 2024). The second hypothesis posits that the TSMixer model underperforms when applied to stock indexes and certain other datasets, as these datasets contain only a small number of financial securities (typically six to seven per dataset). As a result, the model is unable to fully leverage its feature-mixing MLPs, which are designed to take advantage of the financial tail dependence phenomenon, improving the prediction of realized volatility, particularly in or ahead of turbulent financial periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

5.6. Robustness test 5

5.6.1. Error metrics

Table 7 shows the error measures results for Robustness Test 5.

For the foreign exchange dataset, the TSMixer model is in the top three for all almost all error metrics, with the exception of MAPE, though it is not the best model for any error measure. This indicates that the novel model is competitive but likely not the preferred model for forecasting realized volatility of foreign exchange securities.

This favours the hypothesis that the TSMixer model performs relatively bad for this dataset since its does not possess numerous financial securities being forecasted, which does not allow the model to properly use its feature-mixing MLPs, which allows it to better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

Table 7
Error metrics results for robustness test 5.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	0.262 %	0.160 %	13.508 %	66.296 %
HAR	0.254 %	0.155 %	13.578 %	63.858 %
LSTM	0.259 %	0.158 %	13.315 %	62.741 %
TFT	0.250 %	0.148 %	12.567 %	57.488 %
NBEATSx	**0.203 %**	**0.103 %**	8.505 %	**24.914 %**
NHITS	0.206 %	0.108 %	**7.549 %**	28.791 %
TimesNet	0.220 %	0.122 %	41.185 %	31.476 %
TSMixer	0.214 %	0.116 %	7.683 %	31.574 %

5.6.2. DM tests

The DM tests results for RMSE can be seen in Fig. 22.

Based on the results, it can be concluded that the TSMixer model is statistically superior to all other models besides NBEATSx and NHITS, with only NBEATSx being slightly statistically significantly better than TSMixer. Thus, despite the considerable performance of TSMixer, NBEATSx would still be preferred for forecasting tasks involving foreign exchange securities.

Fig. 23 depicts the DM tests results concerning MAE.

Similarly to RMSE, the TSMixer model is statistically superior to all other models besides NBEATSx and NHITS, with now NBEATSx and NHITS being clearly statistically significantly better than TSMixer. Hence, the NBEATSx and NHITS would be preferred for forecasting tasks involving foreign exchange securities.

Moving to QLIKE, its DM tests results can be found in Fig. 24.

It can be affirmed that the TSMixer model is statistically superior to all other models besides NBEATSx and NHITS while not being inferior to these two models. Consequently, the TSMixer model would likely be preferred for forecasting tasks involving foreign exchange securities given its simplicity when compared to NBEATSx and NHITS.

Fig. 25 shows the DM tests results regarding MAPE.

For MAPE, he TSMixer model is statistically superior to all other models besides NBEATSx, NHITS, and TimesNet, with NBEATSx and NHITS being clearly statistically significantly better than TSMixer. Thus, NBEATSx and NHITS would be preferred for forecasting tasks involving foreign exchange securities.

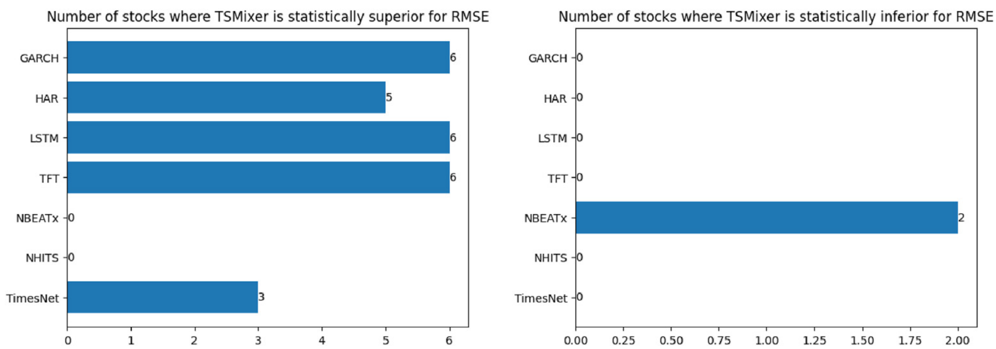

Fig. 22. DM tests results for RMSE (robustness Test 5).

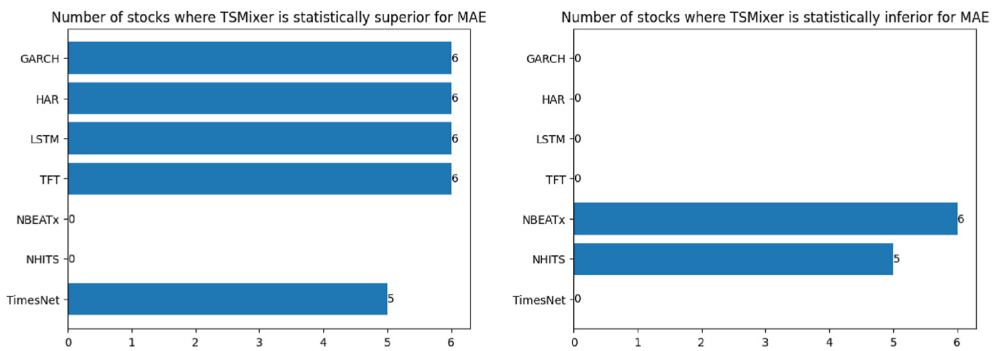

Fig. 23. DM tests results for MAE (robustness Test 5).

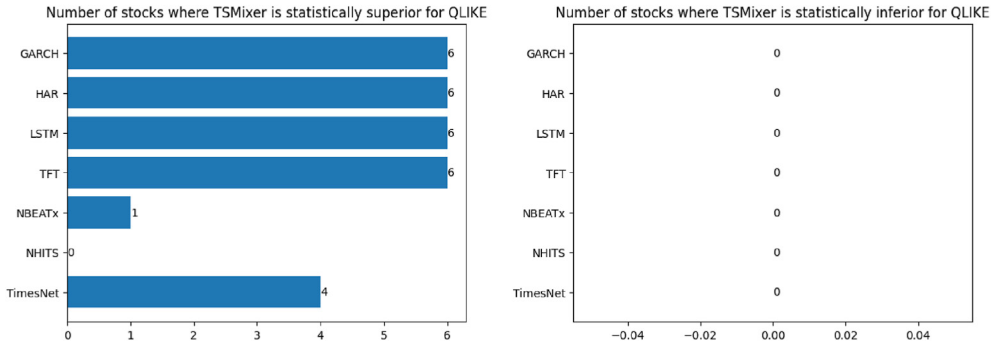

Fig. 24. DM tests results for QLIKE (robustness Test 5).

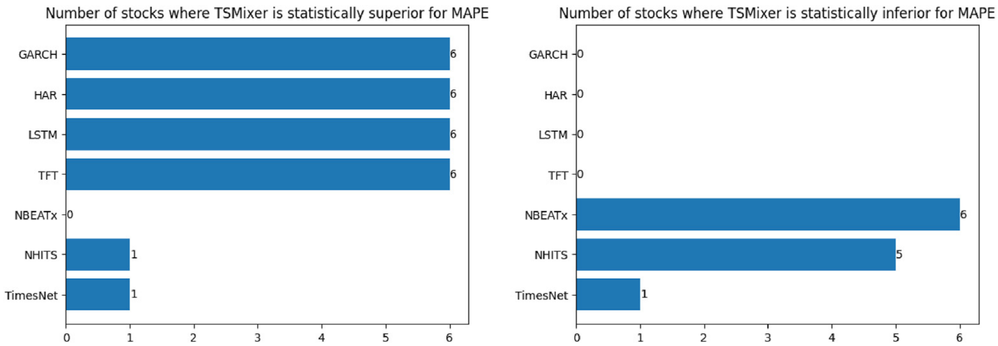

Fig. 25. DM tests results for MAPE (robustness Test 5).

5.6.3. Dynamic MCS

The results of the Dynamic MCS can be seen in Fig. 26. It can be seen that the position of the most-performing model often changes for RMSE and QLIKE, while for MAE and MAPE, the NBEATSx model is clearly the dominant model throughout time. This is another indication that the NBEATSx model would be preferred over the novel TSMixer model.

5.6.4. Conclusion

The results of Robustness Test 5, using foreign exchange securities, suggest that while the TSMixer model remains competitive, it is not the top-performing model across most error metrics. TSMixer consistently ranks in the top three for RMSE, MAE, and QLIKE but is outperformed by NBEATSx and NHITS in several instances. The DM tests indicate that, while TSMixer is statistically superior to many models, NBEATSx and NHITS consistently perform better, particularly for MAE and MAPE. However, TSMixer's simplicity compared to these models makes it a viable alternative for forecasting foreign exchange volatility, especially when prioritizing RMSE and QLIKE. Ultimately, NBEATSx and NHITS emerge as the preferred models, especially in terms of consistency and overall accuracy in this context.

These results give further confirm the hypothesis that the model is more appropriate for datasets containing a considerable number of financial securities being forecasted to allow the model to properly use its feature-mixing MLPs, which allows it to better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

5.7. Robustness test 6

The error measures results for Robustness Test 6 can be seen in Table 8.

5.7.1. Error metrics

For the commodities dataset, the TSMixer is the dominant model for RMSE and QLIKE while being in the top three for MAE and having a considerable bad position for MAPE. This shows that depending on the preference of the research and/or practitioner, the novel model could or not be preferred over the existing benchmark models.

The bad position for MAPE and not dominant position for MAE further confirm the importance of the use of the TSMixer model's feature-mixing MLPs for its advantage and the fact that the model is more appropriate for datasets containing a considerable number of financial securities being forecasted to allow the model to properly use its feature-mixing MLPs, which allows it to better exploit the financial tail dependence phenomenon to better forecast realized volatility preceding or during financially turbulent periods (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

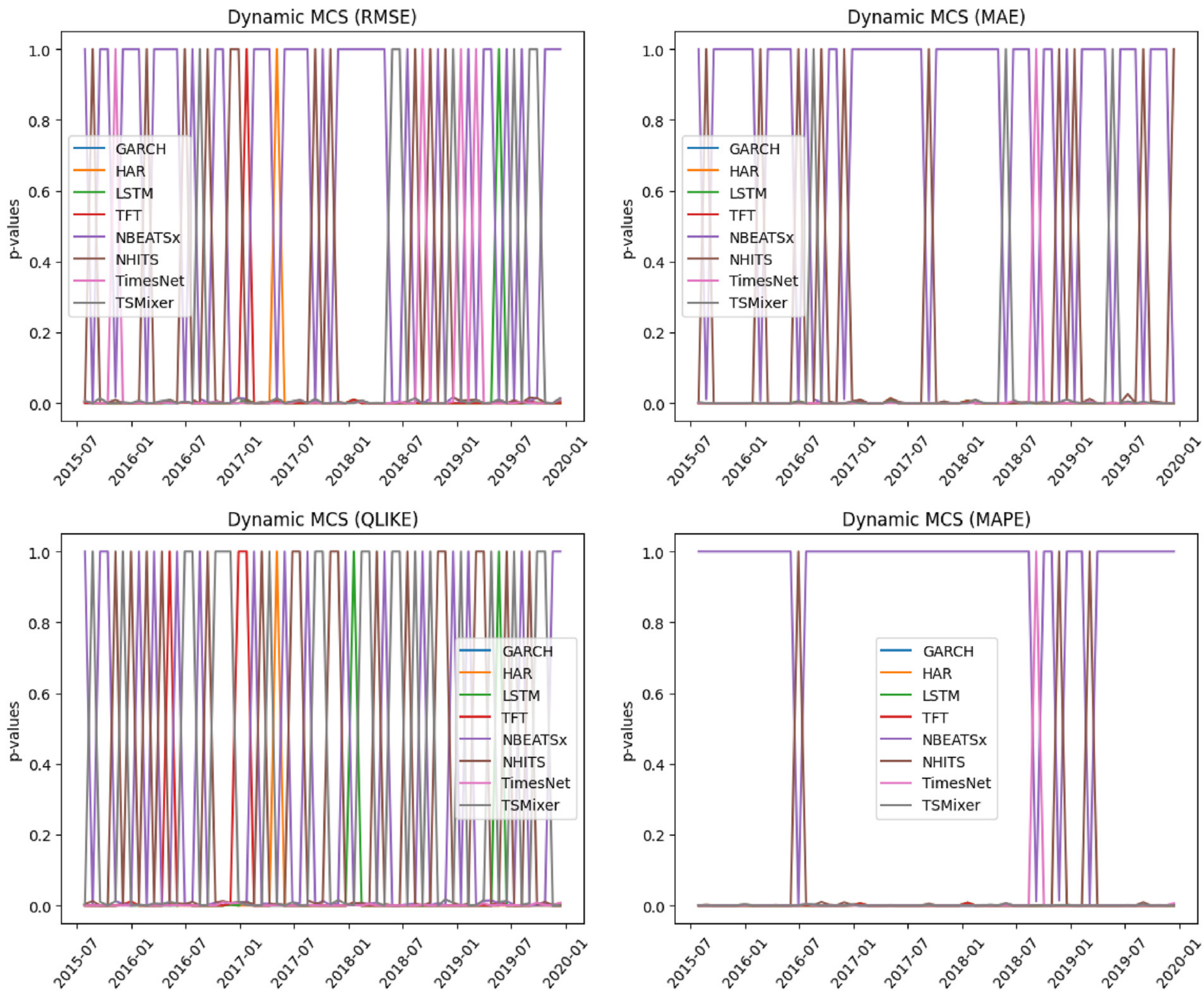

Fig. 26. MCS results (robustness Test 5).

5.7.2. DM tests

Fig. 27 depicts the DM tests results for RMSE.

It can be seen that the TSMixer model is statistically superior to all models besides NBEATSx while not being inferior to the latter. Thus, considering RMSE, the TSMixer model would likely be preferred.

Moving to MAE, Fig. 28 presents its DM tests results.

Now, the TSMixer model is statistically superior to only the GARCH and HAR model while being inferior to NBEATSx and NHITS. Hence, for MAE, the NBEATSx and NHITS models would be preferred.

Regarding QLIKE, its DM tests results can be found in Fig. 29.

While we can affirm with considerable confidence that the TSMixer model is statistically superior to the GARCH, HAR, LSTM, and NHITS model, we cannot do the same with TFT, NBEATSx and TimesNet, albeit the TSMixer model is superior for two out of seven commodities.

Finally, Fig. 30 depicts the MAPE DM tests results.

Now, the TSMixer model is statistically superior to only the GARCH and HAR model while being inferior to NBEATSx and NHITS. Hence, for MAE, the NBEATSx and NHITS models would be preferred considering this error metric.

5.7.3. Dynamic MCS

The results of the Dynamic MCS can be found in Fig. 31. Similarly to Robustness Test 5, the position of the most-performing model often changes for RMSE and QLIKE, while for MAE and MAPE, the NBEATSx model is clearly the dominant model throughout time. This is another indication that the NBEATSx model would be preferred over the novel TSMixer model.

Table 8
Error metrics results for robustness test 6.

Model	RMSE (%)	MAE (%)	QLIKE (%)	MAPE (%)
GARCH	0.750 %	0.501 %	6.648 %	38.576 %
HAR	0.439 %	0.309 %	3.865 %	24.083 %
LSTM	0.414 %	0.271 %	3.103 %	18.644 %
TFT	0.400 %	0.268 %	2.891 %	18.945 %
NBEATSx	0.395 %	**0.252 %**	2.931 %	**16.902 %**
NHITS	0.400 %	0.257 %	2.952 %	17.439 %
TimesNet	0.402 %	0.266 %	2.945 %	18.611 %
TSMixer	**0.393 %**	0.265 %	**2.874 %**	19.323 %

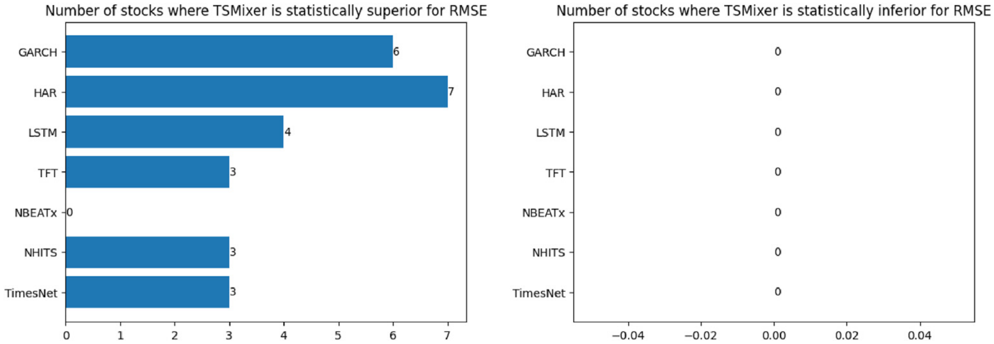

Fig. 27. DM tests results for RMSE (robustness Test 6).

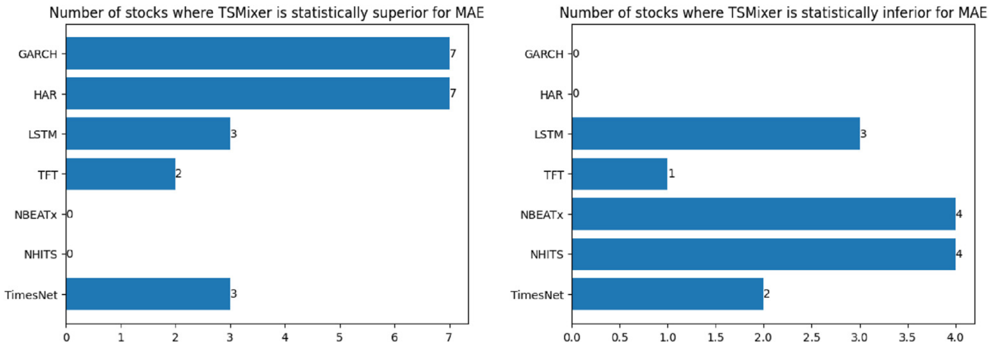

Fig. 28. DM tests results for MAE (robustness Test 6).

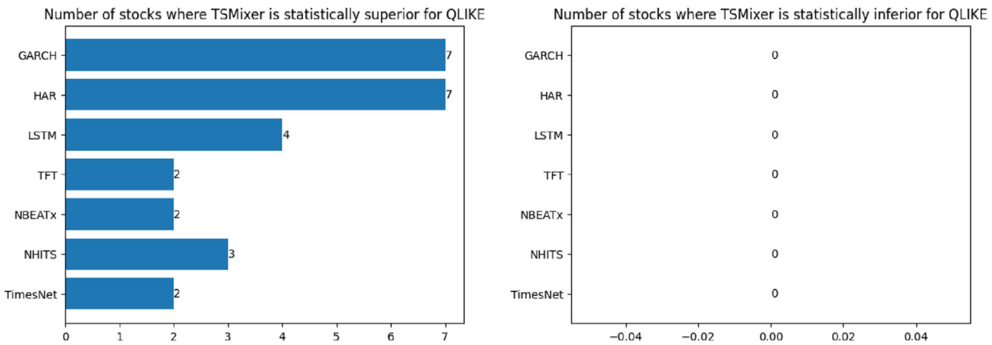

Fig. 29. DM tests results for QLIKE (robustness Test 6).

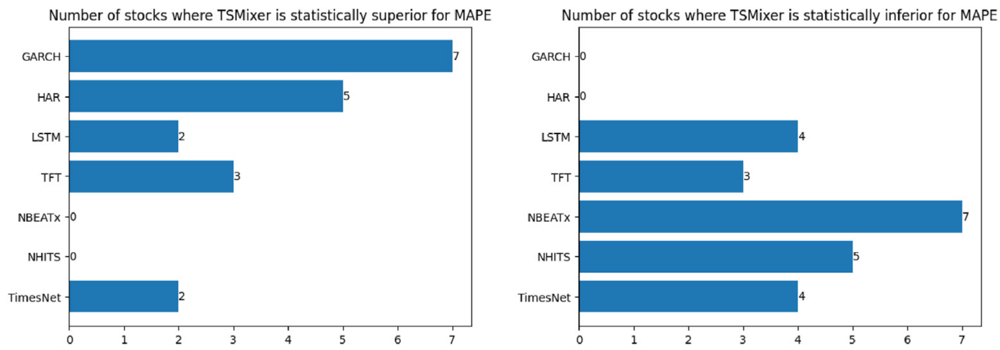

Fig. 30. DM tests results for MAPE (robustness Test 6).

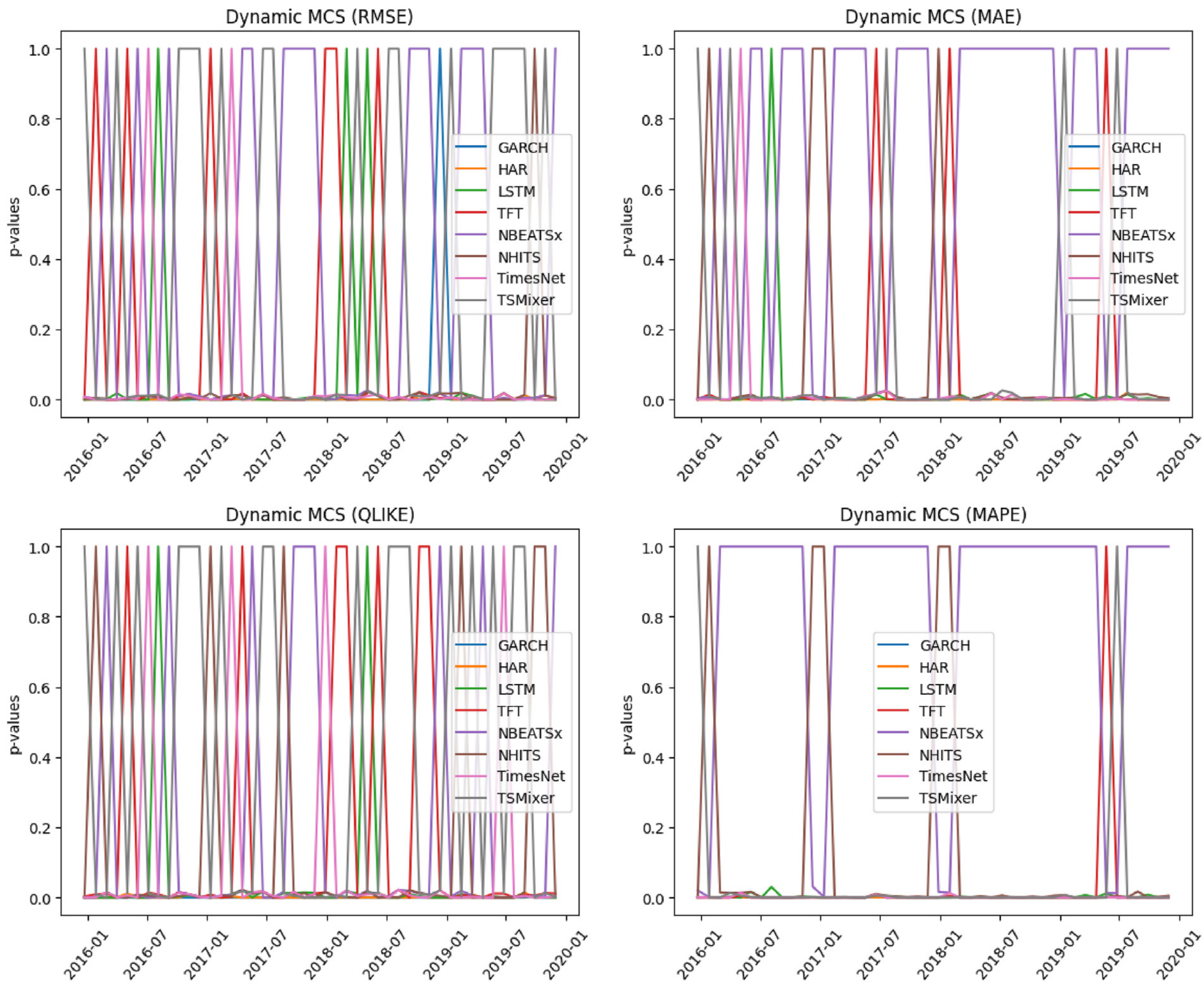

Fig. 31. MCS results (robustness Test 6).

5.7.4. Conclusion

The results of Robustness Test 6, using a commodities dataset, indicate that the TSMixer model is a strong contender, particularly for RMSE and QLIKE, where it performs as the dominant model. However, its ranking drops for MAE and MAPE, where NBEATSx and NHITS consistently outperform TSMixer. The DM tests further confirm that, while TSMixer is statistically superior for RMSE and QLIKE over many models, it is not clearly superior to NBEATSx and NHITS for MAE and MAPE. This suggests that the choice of model depends on the specific error metric prioritized. For practitioners focused on RMSE or QLIKE, TSMixer would likely be the preferred model. However, for those emphasizing MAE or MAPE, NBEATSx and NHITS would be the preferred options.

These results further confirm the importance of the use of the TSMixer model's feature-mixing MLPs for its advantage and the fact that the model is more appropriate for datasets containing a considerable number of financial securities being forecasted.

6. Novel insights and pratical considerations

The findings of this research collectively contribute significant insights into the literature of stock realized volatility forecasting and neural network time series forecasting. These insights challenge prevailing notions in neural network architecture and complexity and underscore these models' adaptability and efficacy in varying market conditions.

The overarching result from the Main Sample and the initial Robustness Tests demonstrates the superiority of the TSMixer model in forecasting accuracy when applied to individual stocks. This result is evident across various error metrics (RMSE, MAE, QLIKE) and under different data configurations. The TSMixer model, with its innovative time-mixing and feature-mixing MLP approaches, exemplifies how a simplified yet effective neural network architecture can lead to enhanced forecasting precision. This finding is important in the neural network forecasting literature, where there is often an implicit assumption that greater model complexity correlates with improved performance.

Still regarding this, interestingly more complex models, such as LSTM and TFT, underperform. This can be explained by a few limitations that these models have. Firstly, both models can at times suffer from overfitting due to their complexity (Goodfellow, 2016). Secondly, the LSTM model fails to account for cross-sectional information across different stocks (S. Gu et al., 2020), while also having issues with vanishing gradient due to its architecture (Pascanu et al., 2012). Finally, financial time series data are often noisy and exhibit abrupt changes. TFT may struggle to adapt its attention mechanisms effectively in such environments, leading to less accurate forecasts (Zerveas et al., 2021).

However, the results from Robustness Tests 4, 5, and 6 reveal that TSMixer's superiority is context-dependent. When applied to datasets such as stock indexes, foreign exchange securities, and commodities, which typically contain a smaller number of financial instruments, TSMixer remains competitive but is not consistently the top-performing model. In these cases, models like NBEATSx and NHITS often outperform TSMixer, particularly in error metrics like MAE and MAPE. The differences in performance suggest that TSMixer's advantage is more pronounced in datasets with a considerable number of securities, allowing it to fully leverage its feature-mixing MLPs to exploit financial tail dependence phenomena (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

These findings highlight the importance of testing models across different datasets, as performance can vary significantly depending on the characteristics of the financial instruments being forecasted. They also suggest that simplicity in neural network architecture does not necessarily guarantee superior performance across all contexts, challenging the conventional pursuit of complexity in model design only insofar as it may mask the necessity of model appropriateness to data characteristics.

Furthermore, the study reveals that the TSMixer model is particularly effective in forecasting the realized volatility of individual stocks, likely because these stocks possess distinct characteristics and a greater amount of data that the model can exploit (Z. Gu and Ibragimov, 2018; Souto and Moradi, 2024). In datasets with fewer securities, the model cannot fully utilize its feature-mixing capabilities, leading to relatively lower performance compared to models like NBEATSx and NHITS.

This study's findings have several implications for academic understanding and practical application of neural network models in financial forecasting. Firstly, they confirm that the effectiveness of simplified neural network architectures like TSMixer is contingent on the dataset characteristics, particularly the number of financial securities involved. Secondly, the successful application of the TSMixer model to individual stock volatility forecasting opens new avenues for employing time-mixing and feature-mixing techniques in similar contexts.

In practical terms, implementing TSMixer in real-world financial forecasting scenarios requires careful consideration of several challenges. One of the key challenges is **data availability**: robust performance of TSMixer depends on access to extensive datasets with a large number of financial securities. In markets or sectors where such comprehensive data is limited, the model's ability to capture the necessary feature interactions may be compromised. To overcome this challenge, practitioners can:

- **Augment datasets** by incorporating additional relevant financial instruments or related variables to enrich the feature space.
- Utilize **data augmentation techniques**, such as bootstrapping or synthetic data generation, to increase the effective size of the dataset while preserving underlying patterns.
- Leverage **transfer learning** from larger datasets in similar domains to enhance the model's performance on smaller datasets.

Another practical aspect is the **computational requirements** associated with training and deploying TSMixer. Although TSMixer is less computationally intensive than more complex neural network models, it still requires sufficient computational resources, especially for large-scale datasets. To address computational challenges, practitioners can.

- Utilize **cloud-based computing services** that offer scalable resources to handle intensive computations without significant capital investment.
- Implement **efficient training procedures**, such as mini-batch training, to reduce memory usage and increase training speed.
- Optimize the model's **hyperparameters** to balance complexity and performance, potentially reducing unnecessary computational overhead.

Integration with existing systems poses additional challenges. Financial institutions often have established data pipelines and forecasting systems, and integrating new models like TSMixer requires ensuring compatibility and minimal disruption. Suggestions to facilitate integration include.

- Developing the TSMixer model using **standard programming languages and frameworks** common in the industry (e.g., Python with TensorFlow or PyTorch) to ease adoption.
- Creating **modular components** for the model that can be integrated as plug-ins or extensions to existing platforms.
- Providing **comprehensive documentation and user interfaces**, allowing practitioners to customize and interact with the model without extensive retraining.

Lastly, recognizing the limitations of the TSMixer model in certain datasets underscores the importance of **model selection based on data properties**, reinforcing the need for adaptability in real-world financial market scenarios where data characteristics can vary widely. Practitioners should.

- Conduct **exploratory data analysis** to understand the dataset's characteristics before selecting a modeling approach.
- Consider **hybrid or ensemble methods** that combine TSMixer with other models like NBEATSx and NHITS to leverage their respective strengths across different datasets.
- Remain **flexible and iterative** in the modeling process, regularly evaluating model performance and adjusting techniques as necessary.

In conclusion, this research not only contributes to the theoretical advancement in neural networks and financial volatility forecasting literature but also provides practical insights for developing and applying these models in diverse financial contexts. The novel approach of using simplified yet effective neural network models like TSMixer could pave the way for more accessible, efficient, and accurate financial forecasting tools in the future, particularly when applied to datasets with a considerable number of securities. However, practitioners should be mindful of the model's limitations in contexts where the dataset contains a smaller number of financial instruments.

7. Conclusion

This study aimed to assess the effectiveness of the TSMixer neural network model in forecasting stock realized volatility, comparing its performance with both traditional models (GARCH, HAR) and newer neural network architectures (LSTM, NHITS, TimesNet, NBEATSx). The primary analysis included all stocks in the S&P 100 index with continuous trading from July 1, 2007, to June 30, 2021. The methodology involved splitting the data into a 70 % training set and a 30 % test set, and the evaluation was extended through robustness tests to ensure the reliability and generalizability of the findings. These robustness tests included alternative data splits, reduced training sample sizes, sensitivity analysis to random seed selection, and applications to different datasets comprising stock indexes, foreign exchange securities, and commodities.

The results from the main sample demonstrated the TSMixer model's superior performance in forecasting stock realized volatility for individual stocks. TSMixer outperformed the benchmark models in key error metrics such as RMSE, MAE, and QLIKE, consistently across different data configurations and stochastic conditions in the initial robustness tests. This finding underscores the effectiveness of TSMixer's innovative time-mixing and feature-mixing MLP techniques when applied to datasets with a considerable number of securities, allowing the model to fully leverage the data's complexity and exploit financial tail dependence phenomena.

However, the results from Robustness Tests 4, 5, and 6, which involved datasets with fewer securities (stock indexes, foreign exchange rates, and commodities), revealed that while the TSMixer model remained competitive, it was not always the top-performing model. In Robustness Test 4, applied to a stock indexes dataset, TSMixer was outperformed by models like TFT, NBEATSx, and NHITS in key metrics, particularly MAE and MAPE. Similarly, in Robustness Test 5, using foreign exchange securities, TSMixer consistently ranked in the top three but was often outperformed by NBEATSx and NHITS. In Robustness Test 6, with commodities data, TSMixer was the dominant model for RMSE and QLIKE but ranked lower for MAE and MAPE. These findings suggest that the TSMixer model's performance is influenced by the dataset's characteristics, particularly the number of financial instruments and the ability to exploit feature interdependencies.

These results highlight that the effectiveness of simplified neural network architectures like TSMixer is context-dependent. While TSMixer excels in forecasting realized volatility for individual stocks within a large dataset, its relative performance diminishes when applied to datasets with a smaller number of securities, where it cannot fully utilize its feature-mixing capabilities. This underscores the importance of considering dataset characteristics in model selection for financial forecasting tasks and challenges the notion that model simplicity alone guarantees superior performance across all contexts.

Furthermore, the study provides novel insights into neural network forecasting and financial volatility. It demonstrates that a simplified model like TSMixer can effectively capture the nuances of stock market volatility for individual stocks but may have limitations when applied to datasets with fewer financial instruments. This finding suggests that the model's innovative time-mixing and feature-mixing techniques are more effective when there is a considerable amount of data allowing the model to exploit the financial tail dependence phenomenon, which is prominent in larger datasets (Beine et al., 2010; Chesnay and Jondeau, 2001; Jebran et al., 2017; Souto, 2023c; White et al., 2015).

Based on the findings of this study, there are several specific areas for future research that could build upon the strengths of the TSMixer model and address some of its limitations.

Firstly, researchers should explore the application of the TSMixer model in other financial domains beyond the ones explored in this study. This could include forecasting volatility in bond markets, cryptocurrencies, or stocks of emerging markets, where the unique properties of these assets may present new challenges and opportunities for the model.

Secondly, given the observed limitations of TSMixer when dealing with datasets containing fewer securities, future work could focus on developing new hybrid models that combine the strengths of TSMixer with those of models like NBEATSx and NHITS. For instance, integrating the time-mixing and feature-mixing MLP techniques of TSMixer with the recursive and hierarchical forecasting approaches of NBEATSx and NHITS might yield a model that performs robustly across datasets of varying sizes. Such hybrid models could capitalize on TSMixer's ability to capture complex dependencies in large datasets while retaining strong performance in smaller datasets.

Furthermore, enhancing the interpretability of the TSMixer model is an important area for future research. Developing methods to interpret the model's outputs and internal workings, such as using explainable AI techniques like SHAP values or Layer-wise Relevance Propagation, could make TSMixer more transparent and accessible to practitioners. This increased interpretability would be particularly valuable in financial contexts where understanding the factors driving predictions is crucial for decision-making.

In conclusion, this research contributes to the theoretical advancement in neural networks and financial volatility forecasting literature by demonstrating that simplified neural network models like TSMixer can offer high predictive accuracy and computational efficiency in appropriate contexts. However, practitioners should carefully consider the characteristics of their datasets when selecting forecasting models, as the performance of models like TSMixer may vary depending on the number of financial instruments and the data's complexity. This study highlights the importance of matching model capabilities with dataset properties to achieve optimal forecasting performance in financial markets.

Data and code availability

Data will be shared upon request. Regarding the code used in this study, the link to the GitHub repository where the open-source code for the TSMixer model and all Jupyter Notebooks used in this research is https://github.com/hugogobato/TSMixer-for-Realized-Volatility/tree/main.

Author contribution

Hugo Gobato Souto: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Visualization, Project administration; **Storm Koert Heuvel**: Software, Formal analysis, Writing - Original Draft; **Francisco Louzada Neto**: Writing - Review & Editing, Visualization, Supervision, Project administration.

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.

Acknowledgments

The authors would like to express heartfelt gratitude to the São Paulo Research Foundation (FAPESP) for their financial support through the Master’s scholarship (Processo 2024/06274-0) of the first author of this paper and the associated project (Processo Vinculado 2013/07375-0).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

A. Summary Statistics of Datasets

Table 9
Summary Statistics of 5-min Realized Volatility Daily Values for S&P 100 Dataset

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 %
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 %

(*continued on next page*)

Table 9 (*continued*)

Tickers	Mean	Std	Min	25 %	50 %	75 %	Max
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 %
CME	1.48 %	0.94 %	0.43 %	0.92 %	1.17 %	1.65 %	8.21 %
COP	1.53 %	0.87 %	0.40 %	0.99 %	1.31 %	1.80 %	8.50 %
COST	1.07 %	0.57 %	0.31 %	0.73 %	0.92 %	1.21 %	5.32 %
CRM	1.79 %	0.89 %	0.47 %	1.20 %	1.55 %	2.15 %	7.76 %
CSCO	1.25 %	0.65 %	0.37 %	0.84 %	1.07 %	1.45 %	6.42 %
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 %
FIS	1.18 %	0.70 %	0.39 %	0.77 %	0.98 %	1.32 %	7.74 %
FISV	1.14 %	0.63 %	0.38 %	0.76 %	0.96 %	1.30 %	7.17 %
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 %
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 %
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 %
INTU	1.27 %	0.63 %	0.39 %	0.86 %	1.10 %	1.47 %	6.13 %
ISRG	1.58 %	0.82 %	0.47 %	1.05 %	1.35 %	1.84 %	6.78 %
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 %
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 %
MDT	1.10 %	0.54 %	0.36 %	0.77 %	0.96 %	1.25 %	5.96 %
MMM	1.04 %	0.58 %	0.29 %	0.68 %	0.90 %	1.22 %	5.49 %
MO	1.05 %	0.55 %	0.25 %	0.72 %	0.92 %	1.20 %	6.18 %
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 %

(*continued on next page*)

Table 9 (*continued*)

Tickers	Mean	Std	Min	25 %	50 %	75 %	Max
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 %

Table 10
Summary Statistics of 5-min Realized Volatility Daily Values for Stock Indexes Dataset

	Mean (%)	Std (%)	Min (%)	25 %	50 %	75 %	Max (%)
AU200_AUD	0.96 %	0.62 %	0.02 %	0.62 %	0.79 %	1.09 %	7.26 %
FR40_EUR	1.13 %	0.77 %	0.15 %	0.69 %	0.93 %	1.32 %	14.24 %
JP225_USD	1.24 %	0.83 %	0.14 %	0.77 %	1.04 %	1.43 %	10.89 %
NAS100_USD	1.06 %	0.66 %	0.07 %	0.68 %	0.90 %	1.23 %	8.76 %
SPX500_USD	0.90 %	0.65 %	0.06 %	0.53 %	0.72 %	1.06 %	8.17 %
UK100_GBP	0.96 %	0.70 %	0.05 %	0.59 %	0.77 %	1.09 %	14.25 %
US2000_USD	1.22 %	0.73 %	0.06 %	0.80 %	1.01 %	1.39 %	8.87 %

Table 11
Summary Statistics of 5-min Realized Volatility Daily Values for Currencies Exchange Dataset

	Mean	Std	Min	25 %	50 %	75 %	Max
AUD_JPY	0.79 %	0.57 %	0.0027 %	0.49 %	0.65 %	0.91 %	7.99 %
AUD_USD	0.64 %	0.42 %	0.0489 %	0.42 %	0.57 %	0.77 %	6.20 %
EUR_JPY	0.61 %	0.39 %	0.0263 %	0.38 %	0.53 %	0.73 %	5.76 %
EUR_USD	0.49 %	0.26 %	0.0103 %	0.32 %	0.46 %	0.61 %	2.44 %
GBP_USD	0.50 %	0.30 %	0.0143 %	0.35 %	0.47 %	0.60 %	5.33 %
USD_CAD	0.50 %	0.28 %	0.0216 %	0.32 %	0.47 %	0.63 %	2.48 %

Table 12
Summary Statistics of 5-min Realized Volatility Daily Values for Commodity Dataset

	Mean	Std	Min	25 %	50 %	75 %	Max
CORN_USD	1.61 %	0.82 %	0.33 %	1.07 %	1.44 %	1.92 %	10.72 %
NATGAS_USD	2.57 %	1.07 %	0.48 %	1.82 %	2.34 %	3.10 %	11.74 %
SOYBN_USD	1.32 %	0.62 %	0.27 %	0.93 %	1.16 %	1.53 %	7.23 %
SUGAR_USD	1.92 %	0.78 %	0.45 %	1.38 %	1.80 %	2.29 %	7.59 %
WHEAT_USD	1.89 %	0.91 %	0.35 %	1.34 %	1.70 %	2.20 %	23.85 %
WTICO_USD	1.92 %	0.95 %	0.29 %	1.33 %	1.70 %	2.22 %	7.82 %
XAU_USD	1.01 %	0.52 %	0.20 %	0.67 %	0.88 %	1.19 %	5.61 %

Appendix B

B. Hyperparameter Search Space for TSMixer

Table 13
TSMixer Hyperparameters Search Space

Hyperparameters	Options
Dropout	[0.7, 0.75, 0.8, 0.85, 0.9]
n_inputs	[3, 5, 7, 10, 15, 21]
n_blocks	[2, 3, 4, 5, 6]
ff_dim	[160, 180, 220, 260, 300, 320, 400]
learning_rate	[0.0005, 0.0001, 0.00005, 0.00001]
Epochs	[150, 250, 350, 450, 550]
loss	MSE ()
random_seed	randrange (129228148)

Appendix C

C. Optimal Hyperparameters for Each Model

Table 14
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
loss	DistributionLoss (distribution = ‘StudentT’, level = [80, 90])

Table 15
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
loss	MSE ()
num_lr_decays	1
scaler_type	“Minmax”
Randomseed	92551658

Table 16
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”
loss	DistributionLoss (distribution = ‘StudentT’, level = [80, 90])
n_harmonics	0
n_blocks	[3, 3]
n_polynomials	0
Randomseed	99670502

Table 17
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”
loss	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’
Randomseed	123422336

Table 18
TimesNet Optimal Hyperparameters

Hyperparameters	Options
n_inputs	15
top_k	5
encoder_layers	2
num_kernels	6
hidden_size	90
conv_hidden_size	90
Dropout	0.1
Epochs	150
learning_rate	0.0001
num_lr_decays	0
scaler_type	"Standard"
loss	MSE ()
Randomseed	1

Table 19
TSMixer Hyperparameters Search Space

Hyperparameters	Options
Dropout	0.7
n_inputs	21
n_blocks	2
ff_dim	220
learning_rate	0.0005
Epochs	250 (168 with early stop)
loss	MSE ()
random_seed	89102543

References

Abhishek, K., Singh, M., Ghosh, S., Anand, A., 2012. Weather forecasting model using artificial neural network. Procedia Technology 4, 311–318. https://doi.org/10.1016/j.protcy.2012.05.047.

Amo Baffour, A., Feng, J., Taylor, E.K., 2019. A hybrid artificial neural network-gjr modeling approach to forecasting currency exchange rate volatility. Neurocomputing 365, 285–301. https://doi.org/10.1016/j.neucom.2019.07.088.

Anderson, T.W., Darling, D.A., 1952. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. Ann. Math. Stat. 23 (2), 193–212. Retrieved December 20, 2023, from. http://www.jstor.org/stable/2236446.

Arnerić, J., Poklepović, T., Teai, J.W., 2018. Neural network approach in forecasting realized variance using high-frequency data. Business Systems Research Journal 9 (2), 18–34. https://doi.org/10.2478/bsrj-2018-0016.

Audrino, F., Huang, C., Okhrin, O., 2018. Flexible har model for realized volatility. Stud. Nonlinear Dynam. Econom. 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. Econom. Rev. 35 (8–10), 1485–1521. https://doi.org/10.1080/07474938.2015.1092801.

Aydin, G., Jang, H., Topal, E., 2016. Energy consumption modeling using artificial neural networks: the case of the world's highest consumers. Energy Sources B Energy Econ. Plann. 11 (3), 212–219. https://doi.org/10.1080/15567249.2015.1075086.

Azadeh, A., Babazadeh, R., Asadzadeh, S., 2013. Optimum estimation and forecasting of renewable energy consumption by artificial neural networks. Renew. Sustain. Energy Rev. 27, 605–612. https://doi.org/10.1016/j.rser.2013.07.007.

Azadeh, A., Ghaderi, S., Sohrabkhani, S., 2008. Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors. Energy Convers. Manag. 49 (8), 2272–2278. https://doi.org/10.1016/j.enconman.2008.01.035.

Baboo, S.S., Shereef, I.K., 2010. An efficient weather forecasting system using artificial neural network. Int. J. Environ. Sustain Dev. 1 (4), 321.

Bahrammirzaee, A., 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. Neural Comput. Appl. 19 (8), 1165–1195. https://doi.org/10.1007/s00521-010-0362-z.

Beine, M., Cosma, A., Vermeulen, R., 2010. The dark side of global integration: increasing tail dependence. J. Bank. Finance 34 (1), 184–192. https://doi.org/10.1016/j.jbankfin.2009.07.014.

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

Borup, D., Jakobsen, J.S., 2019. Capturing volatility persistence: a dynamically complete realized egarch-midas model. Quant. Finance 19 (11), 1839–1855. https://doi.org/10.1080/14697688.2019.1614653.

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. J. Financ. Econom. 18 (3), 502–531. https://doi.org/10.1093/jjfinec/nbaa008.

Bucci, A., He, L., Liu, Z., 2023. Combining dimensionality reduction methods with neural networks for realized volatility forecasting. Ann. Oper. Res. https://doi.org/10.1007/s10479-023-05544-7.

Cao, J., Li, Z., Li, J., 2019. Financial time series forecasting model based on ceemdan and lstm. Phys. Stat. Mech. Appl. 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, K., Zhou, Y., Dai, F., 2015. A lstm-based method for stock returns prediction: a case study of China stock market. 2015 IEEE International Conference on Big Data (Big Data), pp. 2823–2824. https://doi.org/10.1109/BigData.2015.7364089.

Chen, L., Pelger, M., Zhu, J., 2023a. Deep learning in asset pricing. Manag. Sci. https://doi.org/10.1287/mnsc.2023.4695.

Chen, Q., Robert, C.-Y., 2022. Multivariate realized volatility forecasting with graph neural network. Proceedings of the Third ACM International Conference on AI in Finance. https://doi.org/10.1145/3533271.3561663.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S.O., Pfister, T., 2023b. Tsmixer: an all-mlp architecture for time series forecasting. https://doi.org/10.48550/ARXIV.2303.06053.

Chen, W.-J., Yao, J.-J., Shao, Y.-H., 2022. Volatility forecasting using deep neural network with time-series feature embedding. Economic Research-Ekonomska Istraživanja 36 (1), 1377–1401. https://doi.org/10.1080/1331677x.2022.2089192.

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

Chesnay, F., Jondeau, E., 2001. Does correlation between stock returns really increase during turbulent periods? Econ. Notes 30 (1), 53–80. https://doi.org/10.1111/1468-0300.00047.

Chiriac, R., Voev, V., 2010. Modelling and forecasting multivariate realized volatility. J. Appl. Econom. 26 (6), 922–947. https://doi.org/10.1002/jae.1152.

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

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

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

Crane, M., 2018. Questionable answers in question answering research: Reproducibility and variability of published results. Transactions of the Association for Computational Linguistics 6, 241–252. https://doi.org/10.1162/tacl\text{_}a\text{_}00018.

Das, S.K., Kumar, A., Das, B., Burnwal, B., 2013. On soft computing techniques in various areas. Computer Science & Information Technology (CS & IT). https://doi.org/10.5121/csit.2013.3206.

Deb, A., 2011. Introduction to soft computing techniques: artificial neural networks, fuzzy logic and genetic algorithms. In: Soft Computing in Textile Engineering. Elsevier, pp. 3–24. https://doi.org/10.1533/9780857090812.1.3.

de Campos Souza, P.V., 2020. Fuzzy neural networks and neuro-fuzzy networks: a review the main techniques and applications used in the literature. Appl. Soft Comput. 92, 106275. https://doi.org/10.1016/j.asoc.2020.106275.

D'Ecclesia, R.L., Clementi, D., 2019. Volatility in the stock market: Ann versus parametric models. Ann. Oper. Res. 299 (1–2), 1101–1127. https://doi.org/10.1007/s10479-019-03374-0.

Denaï, M., Palis, F., Zeghbib, A., 2007. Modeling and control of non-linear systems using soft computing techniques. Appl. Soft Comput. 7 (3), 728–738. https://doi.org/10.1016/j.asoc.2005.12.005.

Diebold, F.X., Mariano, R.S., 1995. Comparing predictive accuracy. J. Bus. Econ. Stat. 13 (3), 253. https://doi.org/10.2307/1392185.

Dodge, J., Gururangan, S., Card, D., Schwartz, R., Smith, N.A., 2019. Show your work: improved reporting of experimental results. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. EMNLP-IJCNLP). https://doi.org/10.18653/v1/d19-1224.

Dunis, C.L., Laws, J., Sermpinis, G., 2011. Higher order and recurrent neural architectures for trading the eur/usd exchange rate. Quant. Finance 11 (4), 615–629. https://doi.org/10.1080/14697680903386348.

Dutta, A., Das, D., 2022. Forecasting realized volatility: new evidence from time-varying jumps in vix. J. Futures Mark. 42 (12), 2165–2189. https://doi.org/10.1002/fut.22372.

Engle, R., 2004. Risk and volatility: econometric models and financial practice. Am. Econ. Rev. 94 (3), 405–420. https://doi.org/10.1257/0002828041464597.

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.

Fente, D.N., Kumar Singh, D., 2018. Weather forecasting using artificial neural network. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1757–1761. https://doi.org/10.1109/ICICCT.2018.8473167.

Fischer, T., Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. 270 (2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

Frank, J., 2023a. *Forecasting Realized Volatility in Turbulent Times Using Temporal Fusion Transformers* (Working Paper) (JEL Classification: C45, C53, C58, E44). Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics, Nürnberg. https://www.economics.fau.de/publications/working-papers/03-2023.pdf.

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

Gallo, G.M., Otranto, E., 2015. Forecasting realized volatility with changing average levels. Int. J. Forecast. 31 (3), 620–634. https://doi.org/10.1016/j.ijforecast.2014.09.005.

Garza, A., Mergenthaler-Canseco, M., 2023. Timegpt-1. https://doi.org/10.48550/ARXIV.2310.03589.

Goodfellow, I., 2016. Deep learning.

Gu, S., Kelly, B., Xiu, D., 2020. Empirical asset pricing via machine learning. In: Karolyi, A. (Ed.), Rev. Financ. Stud. 33 (5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009.

Gu, Z., Ibragimov, R., 2018. The "cubic law of the stock returns" in emerging markets. J. Empir. Finance 46, 182–190. https://doi.org/10.1016/j.jempfin.2017.11.008.

Hamid, S.A., Iqbal, Z., 2004. Using neural networks for forecasting volatility of s&p 500 index futures prices. J. Bus. Res. 57 (10), 1116–1125. https://doi.org/10.1016/s0148-2963(03)00043-2.

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. Stoch. Environ. Res. Risk Assess. 37 (7), 2563–2575. https://doi.org/10.1007/s00477-023-02405-4.

Hansen, P.R., Huang, Z., Shek, H.H., 2011. Realized garch: a joint model for returns and realized measures of volatility. J. Appl. Econom. 27 (6), 877–906. https://doi.org/10.1002/jae.1234.

Hansen, P.R., Lunde, A., Voev, V., 2014. Realized beta garch: a multivariate garch model with realized measures of volatility. J. Appl. Econom. 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. Int. J. Forecast. 13 (2), 281–291. https://doi.org/10.1016/s0169-2070(96)00719-4.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 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.

Hua, H., Li, X., Dou, D., Xu, C., Luo, J., 2021. Noise stability regularization for improving bert fine-tuning. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. https://doi.org/10.18653/v1/2021.naacl-main.258.

Huang, Y., Zhou, C., Cui, K., Lu, X., 2024. A multi-agent reinforcement learning framework for optimizing financial trading strategies based on timesnet. Expert Syst. Appl. 237, 121502. https://doi.org/10.1016/j.eswa.2023.121502.

Huang, Z., He, K., 2024. Gru-tsmixers: sleep apnea and hypopnea detection based on multi scale mlp-mixers. 2024 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. https://doi.org/10.1109/IJCNN60899.2024.10650636.

Ibrahim, D., 2016. An overview of soft computing. Procedia Comput. Sci. 102, 34–38. https://doi.org/10.1016/j.procs.2016.09.366.

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.

Izbicki, R., dos Santos, T.M., 2020. Aprendizado de máquina: Uma abordagem estatística. Rafael Izbicki.

Javeed Nizami, S., Al-Garni, A.Z., 1995. Forecasting electric energy consumption using neural networks. Energy Pol. 23 (12), 1097–1104. https://doi.org/10.1016/0301-4215(95)00116-6.

Jebran, K., Chen, S., Ullah, I., Mirza, S.S., 2017. Does volatility spillover among stock markets varies from normal to turbulent periods? evidence from emerging markets of asia. The Journal of Finance and Data Science 3 (1–4), 20–30. https://doi.org/10.1016/j.jfds.2017.06.001.

Joseph, V.R., 2022. Optimal ratio for data splitting. Stat. Anal. Data Min.: 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. J. Futures Mark. 36 (12), 1127–1163. https://doi.org/10.1002/fut.21783.
Kecman, V., 2001. Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. MIT press.
Khan, N.A., Khalaf, O.I., Romero, C.A.T., Sulaiman, M., Bakar, M.A., 2021. Application of euler neural networks with soft computing paradigm to solve nonlinear problems arising in heat transfer. Entropy 23 (8), 1053. https://doi.org/10.3390/e23081053.
Kim, K.-j., 2006. Artificial neural networks with evolutionary instance selection for financial forecasting. Expert Syst. Appl. 30 (3), 519–526. https://doi.org/10.1016/j.eswa.2005.10.007.
Kim, K.-j., Ahn, H., 2011. Simultaneous optimization of artificial neural networks for financial forecasting. Appl. Intell. 36 (4), 887–898. https://doi.org/10.1007/s10489-011-0303-2.
Lara-Benítez, P., Carranza-García, M., Riquelme, J.C., 2021. An experimental review on deep learning architectures for time series forecasting. Int. J. Neural Syst. 31 (3), 2130001. https://doi.org/10.1142/s0129065721300011.
Li, J., 2022. The comparison of lstm, lgbm, and cnn in stock volatility prediction. Advances in Economics, Business and Management Research. https://doi.org/10.2991/aebmr.k.220307.147.
Li, Z., Qi, S., Li, Y., Xu, Z., 2023. Revisiting long-term time series forecasting: an investigation on linear mapping. https://doi.org/10.48550/ARXIV.2305.10721.
Lim, B., Arık, S.Ö., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int. J. Forecast. 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. J. Econom. 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., 2022a. Trading volume and realized volatility forecasting: evidence from the China stock market. J. Forecast. 42 (1), 76–100. https://doi.org/10.1002/for.2897.
Liu, Y., 2019. Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. Expert Syst. Appl. 132, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038.
Liu, Y., Wu, H., Wang, J., Long, M., 2022b. Non-stationary transformers: exploring the stationarity in time series forecasting. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems, 35. Curran Associates, Inc, pp. 9881–9893. In: https://proceedings.neurips.cc/paper/text files/paper/2022/file/4054556fcaa934b0bf76da52cf4f92cb-Paper-Conference.pdf.
Liu, Z., Xu, X., Luo, B., Yang, C., Gui, W., Dubljevic, S., 2024. Accelerated mpc: a real-time model predictive control acceleration method based on tsmixer and 2d block stochastic configuration network imitative controller. Chem. Eng. Res. Des. 208, 837–852. https://doi.org/10.1016/j.cherd.2024.07.030.
Liukis, A., 2022. Optimizing the random seed - is there any point in doing that?.
Looney, C.G., 1993. Neural networks as expert systems. Expert Syst. Appl. 6 (2), 129–136. https://doi.org/10.1016/0957-4174(93)90003-o.
Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2018. Statistical and machine learning forecasting methods: concerns and ways forward. In: Hernandez Montoya, A.R. (Ed.), PLoS One 13 (3), e0194889. https://doi.org/10.1371/journal.pone.0194889.
Malliaris, M., Salchenberger, L., 1993. A neural network model for estimating option prices. Appl. Intell. 3 (3), 193–206. https://doi.org/10.1007/bf00871937.
Maqsood, I., Khan, M., Abraham, A., 2004. An ensemble of neural networks for weather forecasting. Neural Comput. Appl. 13 (2). https://doi.org/10.1007/s00521-004-0413-4.
Marcjasz, G., Narajewski, M., Weron, R., Ziel, F., 2023. Distributional neural networks for electricity price forecasting. Energy Econ. 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.
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. Springer, Singapore, pp. 405–423. https://doi.org/10.1007/978-981-16-4807-6\text{_}39.
Miura, R., Pichl, L., Kaizoji, T., 2019. Artificial neural networks for realized volatility prediction in cryptocurrency time series. In: Lecture Notes in Computer Science. Springer International Publishing, pp. 165–172. https://doi.org/10.1007/978-3-030-22796-8\text{_}18.
Mizik, N., 2014. Assessing the total financial performance impact of brand equity with limited time-series data. J. Market. Res. 51 (6), 691–706. https://doi.org/10.1509/jmr.13.0431.
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.
O.Omolaye, P., 2017. A holistic review of soft computing techniques. Appl. Comput. Math. 6 (2), 93. https://doi.org/10.11648/j.acm.20170602.15.
Olivares, K.G., Challu, C., Marcjasz, G., Weron, R., Dubrawski, A., 2023. Neural basis expansion analysis with exogenous variables: forecasting electricity prices with nbeatsx. Int. J. Forecast. 39 (2), 884–900. https://doi.org/10.1016/j.ijforecast.2022.03.001.
Oliveira, T.P., Barbar, J.S., Soares, A.S., 2016. Computer network traffic prediction: a comparison between traditional and deep learning neural networks. International Journal of Big Data Intelligence 3 (1), 28. https://doi.org/10.1504/ijbdi.2016.073903.
Olorunnimbe, K., Viktor, H., 2022. Similarity embedded temporal transformers: enhancing stock predictions with historically similar trends. In: Foundations of Intelligent Systems. Springer International Publishing, pp. 388–398. https://doi.org/10.1007/978-3-031-16564-1\text{_}37.
Özbakır, L., Baykasoğlu, A., Kulluk, S., 2010. A soft computing-based approach for integrated training and rule extraction from artificial neural networks: Difaconn-miner. Appl. Soft Comput. 10 (1), 304–317. https://doi.org/10.1016/j.asoc.2009.08.008.
Pascanu, R., Mikolov, T., Bengio, Y., 2012. On the difficulty of training recurrent neural networks. https://doi.org/10.48550/ARXIV.1211.5063.
Pavlidis, N.G., Plagianakos, V.P., Tasoulis, D.K., Vrahatis, M.N., 2006. Financial forecasting through unsupervised clustering and neural networks. Operational Research 6 (2), 103–127. https://doi.org/10.1007/bf02941227.
Ramos, F.R., Pereira, M.T., Oliveira, M., Rubio, L., 2023. The memory concept behind deep neural network models: an application in time series forecasting in the e-commerce sector. Decision Making: Applications in Management and Engineering 6 (2), 668–690. https://doi.org/10.31181/dmame622023695.
Reimers, N., Gurevych, I., 2017. Reporting score distributions makes a difference: performance study of lstm-networks for sequence tagging. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/d17-1035.
Reisenhofer, R., Bayer, X., Hautsch, N., 2022. Harnet: a convolutional neural network for realized volatility forecasting. https://doi.org/10.48550/ARXIV.2205.07719.
Roesch, I., Günther, T., 2018. Visualization of neural network predictions for weather forecasting. Comput. Graph. Forum 38 (1), 209–220. https://doi.org/10.1111/cgf.13453.
Ruf, J., Wang, W., 2019. Neural networks for option pricing and hedging: a literature review. https://doi.org/10.48550/ARXIV.1911.05620.
Ruiz, L., Rueda, R., Cuéllar, M., Pegalajar, M., 2018. Energy consumption forecasting based on elman neural networks with evolutive optimization. Expert Syst. Appl. 92, 380–389. https://doi.org/10.1016/j.eswa.2017.09.059.
Runge, J., Zmeureanu, R., 2019. Forecasting energy use in buildings using artificial neural networks: a review. Energies 12 (17), 3254. https://doi.org/10.3390/en12173254.
Sahin, S., Tolun, M., Hassanpour, R., 2012. Hybrid expert systems: a survey of current approaches and applications. Expert Syst. Appl. 39 (4), 4609–4617. https://doi.org/10.1016/j.eswa.2011.08.130.
Sewell, M., 2011. Characterization of financial time series. Rn, 11 (1), 1.
Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Appl. Soft Comput. 90, 106181. https://doi.org/10.1016/j.asoc.2020.106181.
Sharma, P., Vipul, 2016. Forecasting stock market volatility using realized garch model: international evidence. Q. Rev. Econ. 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. Nhits for forecasting stock realized volatility. SSRN Electron. J. https://doi.org/10.2139/ssrn.4650761.
Souto, H.G., 2023b. Time series forecasting models for s&p 500 financial turbulence. J. Math. Finance 13 (1), 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., 2024. Charting new avenues in financial forecasting with timesnet: the impact of intraperiod and interperiod variations on realized volatility prediction. Expert Syst. Appl. 255, 124851. https://doi.org/10.1016/j.eswa.2024.124851.
Souto, H.G., Blackmon, J., Moradi, A., 2023. Augmented har. SSRN Electron. J. https://doi.org/10.2139/ssrn.4516177.
Souto, H.G., Moradi, A., 2023a. Realized covariance matrix nbeatsx. SSRN Electron. J. https://doi.org/10.2139/ssrn.4529219.
Souto, H.G., Moradi, A., 2023b. Forecasting realized volatility through financial turbulence and neural networks. Econ. Bus. Rev. 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 Syst. Appl. 122802. https://doi.org/10.1016/j.eswa.2023.122802.
Souto, H.G., Moradi, A., 2024. A generalization of the topological tail dependence theory: from indices to individual stocks. Decision Analytics Journal 12, 100512. https://doi.org/10.1016/j.dajour.2024.100512.
Subasi, A., 2012. Classification of emg signals using combined features and soft computing techniques. Appl. Soft Comput. 12 (8), 2188–2198. https://doi.org/10.1016/j.asoc.2012.03.035.
Tsay, R.S., 2005. Analysis of Financial Time Series. John Wiley & Sons.
Vidal, A., Kristjanpoller, W., 2020. Gold volatility prediction using a cnn-lstm approach. Expert Syst. Appl. 157, 113481. https://doi.org/10.1016/j.eswa.2020.113481.
Vortelinos, D.I., 2017. Forecasting realized volatility: har against principal components combining, neural networks and garch. Res. Int. Bus. 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. Eng. Appl. Artif. Intell. 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. J. Bank. Finance 64, 136–149. https://doi.org/10.1016/j.jbankfin.2015.12.010.
Wei, L., Zhu, T., Guo, Y., Ni, C., Zheng, Q., 2023. Cloudprednet: an ultra-short-term movement prediction model for ground-based cloud image. IEEE Access 11, 97177–97188. https://doi.org/10.1109/ACCESS.2023.3310538.
Weigand, A., 2019. Machine learning in empirical asset pricing. Financ. Mark. Portfolio Manag. 33 (1), 93–104. https://doi.org/10.1007/s11408-019-00326-3.
White, H., Kim, T.-H., Manganelli, S., 2015. Var for var: Measuring tail dependence using multivariate regression quantiles. J. Econom. 187 (1), 169–188. https://doi.org/10.1016/j.jeconom.2015.02.004.
Wing-Yi Chio, S., Li, Y., JingRan Yang, R., 2021. Realized volatility prediction. 2021 2nd European Symposium on Software Engineering. https://doi.org/10.1145/3501774.3501793.
Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022. Timesnet: temporal 2d-variation modeling for general time series analysis. https://doi.org/10.48550/ARXIV.2210.02186.
Wu, J., Zhang, Z., Tong, R., Zhou, Y., Hu, Z., Liu, K., 2023. Imaging feature-based clustering of financial time series. In: Kumar, A. (Ed.), PLoS One 18 (7), e0288836. https://doi.org/10.1371/journal.pone.0288836.
Yan, K., Li, W., Ji, Z., Qi, M., Du, Y., 2019. A hybrid lstm neural network for energy consumption forecasting of individual households. IEEE Access 7, 157633–157642. https://doi.org/10.1109/ACCESS.2019.2949065.
Yao, X., Izzeldin, M., Li, Z., 2019. A novel cluster har-type model for forecasting realized volatility. Int. J. Forecast. 35 (4), 1318–1331. https://doi.org/10.1016/j.ijforecast.2019.04.017.
Yardimci, A., 2009. Soft computing in medicine. Appl. Soft Comput. 9 (3), 1029–1043. https://doi.org/10.1016/j.asoc.2009.02.003.
Ye, J., Gu, J., Dash, A., Deek, F.P., Wang, G.G., 2023. Prediction with time-series mixer for the s&p500 index. 2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW), pp. 20–27. https://doi.org/10.1109/ICDEW58674.2023.00008.
Yu, Y., Lin, Y., Hou, X., Zhang, X., 2023. Novel optimization approach for realized volatility forecast of stock price index based on deep reinforcement learning model. Expert Syst. Appl. 233, 120880. https://doi.org/10.1016/j.eswa.2023.120880.
Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C., 2021. A transformer-based framework for multivariate time series representation learning. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3447548.3467401.
Zhang, C., Pu, X., Cucuringu, M., Dong, X., 2023a. Graph neural networks for forecasting realized volatility with nonlinear spillover effects. SSRN Electron. J. https://doi.org/10.2139/ssrn.4375165.
Zhang, G.P., Berardi, V.L., 2001. Time series forecasting with neural network ensembles: an application for exchange rate prediction. J. Oper. Res. Soc. 52 (6), 652–664. https://doi.org/10.1057/palgrave.jors.2601133.
Zhang, Z., Chen, Y., Zhang, D., Qian, Y., Wang, H., 2023b. Ctfnet: long-sequence time-series forecasting based on convolution and time–frequency analysis. IEEE Transact. Neural Networks Learn. Syst. 1–15. https://doi.org/10.1109/TNNLS.2023.3294064.
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.
Zimmermann, H.-G., Tietz, C., Grothmann, R., 2012. Forecasting with recurrent neural networks: 12 tricks. In: Neural Networks: Tricks of the Trade. Springer Berlin Heidelberg, pp. 687–707. https://doi.org/10.1007/978-3-642-35289-8\text{_}37.
Zuo, C., Wang, J., Liu, M., Deng, S., Wang, Q., 2023. An ensemble framework for short-term load forecasting based on timesnet and tcn. Energies 16 (14), 5330. https://doi.org/10.3390/en16145330.