Neural Computing and Applications (2025) 37:6527–6545
https://doi.org/10.1007/s00521-024-10787-x

ORIGINAL ARTICLE

Enhancing financial time series forecasting through topological data analysis

Luiz Carlos de Jesus Jr.[1] · Francisco Fernández-Navarro[2] · Mariano Carbonero-Ruz[1]

Received: 9 August 2024 / Accepted: 7 November 2024 / Published online: 17 January 2025

Abstract

Topological data analysis (TDA) is increasingly acknowledged within financial markets for its capacity to manage complexity and discern nuanced patterns and structures. It has been applied effectively to uncover intricate relationships and capture non-linear dependencies inherent in market data. This manuscript presents a groundbreaking study that delves into integrating features derived from TDA to improve the performance of forecasting models for univariate time series prediction. The research specifically examines whether incorporating features extracted from TDA-such as entropy, amplitude, and the number of points obtained from persistent diagrams can provide valuable supplementary information to the baseline forecasting model. Thus, the aim is to determine if these TDA-derived features can boost forecasting accuracy by offering additional insights that existing models might overlook. The N-BEATS model serves as the baseline forecasting model due to its robust generalization capabilities and flexibility in incorporating additional features into the model. The proposed methodology is compared against a univariate N-BEATS model without additional features and other strategies incorporating supplementary features such as temporal decomposition and time delay embeddings. The evaluation includes forecasting for six cryptocurrencies across four distinct time scenarios and four traditional financial instruments across two scenarios each, resulting in 32 datasets. The results obtained were promising, as the proposed method, $\mathtt{N-BEATS}_{+\mathrm{TDA}}$, achieved the best results in mean performance and mean ranking for the three metrics considered (MAPE, MAE, and RMSE). Significant differences were observed with the rest of the proposed methods using a significance level of $\alpha = 0.10$, highlighting the effectiveness of integrating TDA features to enhance forecasting models.

Keywords Topological data analysis · Time series forecasting · feature extraction

1 Introduction

Topological data analysis (TDA) utilizes concepts from algebraic topology to analyse the shape of data, making it a pivotal tool for extracting meaningful insights from complex datasets [1]. At its core, TDA employs persistent homology to unveil the underlying structure of data by identifying and tracking features such as connected components, loops, and voids across different scales [2]. Persistent homology, a cornerstone of TDA, systematically studies the multi-scale topological characteristics of datasets. It constructs simplicial complexes at different scales and tracks the evolution of features like connected components, loops, and voids [3]. This method captures the persistence of these features across scales, offering insights into the robustness and significance of topological structures within the data [4].

Persistent homology results are typically visualized using persistence diagrams or barcodes, which provide a concise yet powerful representation of the topological features and lifetimes inherent in a dataset [3]. These visual tools condense complex mathematical insights into accessible formats, enabling researchers to discern persistent

✉ Francisco Fernández-Navarro
fafernandez@uma.es

Luiz Carlos de Jesus Jr.
lluizcarlos@al.uloyola.es

Mariano Carbonero-Ruz
mcarbonero@uloyola.es

[1] Department of Quantitative Methods, Universidad Loyola Andalucía, 41704 Córdoba, Spain

[2] Department of Computer Languages and Computer Science, Universidad de Málaga, 29010 Málaga, Spain

patterns across various scales of analysis [2]. By highlighting enduring features amidst fluctuations and noise, persistence diagrams facilitate the discovery of fundamental structures that may not be discernible through conventional data analysis methods alone [5]. Moreover, these capabilities enhance the interpretability of intricate datasets and support informed decision-making across diverse scientific and practical domains, ranging from biology and neuroscience to materials science and beyond. Finally, persistence diagrams yield metrics such as entropy, amplitude, and point counts, which can be instrumental in quantitative assessments and predictive modelling [6].

Financial markets are complex systems where prices of assets evolve according to various factors, including economic indicators, market sentiment, and external events [7]. Traditional financial analysis methods rely on time series analysis, statistical models, and machine learning techniques [8–10]. However, these methods may fail to capture the intricate and multi-scale financial data structure [11]. TDA, employing persistent homology, introduces a new approach to examining financial markets [12]. It provides insights into the overall data structure, essential for comprehending market behaviour and predicting critical events like market crashes [13]. Stable and significant features in data are identified by persistent homology, thereby offering connections to underlying market dynamics. Its resilience to noise and ability to manage high-dimensional data makes it highly suitable for financial applications. Furthermore, financial time series data can be represented as a topological space by TDA, allowing for the identification of meaningful clusters, loops, and voids, which provide valuable insights for decision-making [14].

Topological data analysis (TDA) application to financial markets is relatively new but has shown promising results in various areas. One significant application of TDA is in detecting critical transitions in financial markets. For instance, Gidea (2017) [12] attempted to detect early signs of critical transitions-such as financial crashes-by tracking changes in the topology of correlation graphs as prices approached the 2007-2008 financial crisis using persistent homology. In another study focusing on the detection of critical transitions, Gidea and Katz (2018) [13] used persistent homology to identify topological patterns in a multidimensional time series formed by the four major financial indices in the USA during the technology crash of 2000 and the financial crisis of 2008. They employed persistence landscapes and their L^p norms to measure changes in one-dimensional persistence diagrams. Their results indicated that the L^p norms increased significantly days before the two crashes, while the norms remained stable during calm market periods. Continuing the exploration of critical transition detection in financial markets, researchers applied TDA to study crashes in cryptocurrency markets. Gidea et al. (2020) [15] analysed the time series of four major cryptocurrencies: Bitcoin, Ethereum, Litecoin, and Ripple. They defined the C^1 norm of persistence landscapes to identify critical transitions preceding the cryptocurrency crash in January 2018. Additionally, they employed the k-means clustering algorithm to discern patterns in price movements.

TDA has also been employed to manage risk and guide potential investment strategies. By understanding the topological features of financial data, TDA can contribute to more effective risk management strategies. Persistent homology aids in identifying periods of increased market turbulence, enabling investors to adjust their portfolios accordingly [16]. Additionally, researchers have developed TDA-based indicators to enhance investment strategies. These indicators, derived from the persistence landscapes of financial time series, offer new insights into market behaviour and have demonstrated superior performance compared to traditional indicators in specific scenarios [17].

From a methodological perspective, within the domain of machine learning, TDA has been applied across various domains, including pattern recognition, classification, and signal quality evaluation [18–22]. In the realm of time series analysis, TDA has also demonstrated its utility [23, 24]. However, its application has predominantly been limited to classification settings, for identifying certain phenomena in dynamical systems, or for purely exploratory analysis [25], rather than being utilized for forecasting purposes. To the best of our knowledge, no existing manuscripts currently explore the use of TDA specifically for forecasting tasks. This apparent gap in the literature may arise from the prevailing belief that forecasting models should, through their learning processes, inherently capture and integrate the structural information within historical observations. In contrast, the present study proposes a novel approach by directly capturing these features in the raw data through TDA. Consequently, this manuscript seeks to advance forecasting models by integrating supplementary features derived from TDA.

In pursuit of this objective, this study will construct point clouds and generate persistence diagrams to derive complex topological features, including persistent entropy, amplitude, and the number of points within the persistence diagram. These features capture critical non-linear dependencies and topological invariants, facilitating a multiscale analysis of financial time series. Persistent entropy quantifies the complexity and irregularity within the time series, indicating volatility and underlying market dynamics. Amplitude, reflecting the spread in the persistence diagram, provides insights into the stability and recurrence of observed patterns. The number of points representing significant topological features across scales highlights the persistence of structures within the data, indicating recurrent cycles or trends that may otherwise remain undetected. These attributes collectively could enhance computational forecasting models by incorporating topological summaries, offering a robust and adaptable framework suited to the inherently non-stationary nature of financial data.

Furthermore, it is essential to emphasize that these features are extracted using a sliding window approach over the time series. For each segment, a point cloud is constructed, and a persistence diagram is generated, from which the three features above are extracted and incorporated into the forecasting model to enhance performance. This innovative approach differs from traditional methods that typically generate a single point cloud for the entire time series; instead, multiple point clouds are constructed to preserve crucial temporal information and capture evolving dynamics. Consequently, local structures are detected, non-stationary data are accommodated, and robustness against outliers is achieved, resulting in a more nuanced and temporally aware analysis well-suited to the complexities of financial time series. Finally, the proposed feature extraction methodology will be integrated into the `N-BEATS` model, which has been selected due to its state-of-the-art performance in time series forecasting and its ability to handle complex patterns.

The remainder of the manuscript is structured as follows: Sect. 2 presents the proposed methodology for extracting TDA features and integrating them into the `N-BEATS` model. Section 3 outlines the experimental framework for comparing the proposed model against traditional feature extraction methods using financial datasets. Section 4 reports the forecasting results across various datasets and metrics, highlighting significant improvements in predictive accuracy. Sect. 5 emphasizes the effectiveness of TDA-derived features in enhancing time series forecasting and suggests future research directions, including applications in reinforcement learning (RL) and high-frequency trading (HFT). Finally, Sect. 6 summarizes the proposal and its principales resultados.

2 The proposed method

The forthcoming section introduces an innovative approach to time series forecasting utilizing topological data analysis (TDA). This section comprehensively describes the TDA techniques employed in this study. The primary objective is to extract features that characterize the underlying temporal patterns in a novel manner and compare them against traditional techniques to establish a benchmark. This approach aims to validate whether TDA-derived features can effectively predict a target variable.

Consequently, this section begins by defining the problem of point forecasting in univariate time series. It then justifies the necessity of incorporating feature extraction mechanisms in financial time series forecasting due to the inherent complexity of such problems. Subsequently, it details the extraction of TDA features from the time series using a sliding window approach. Finally, it outlines how these features are integrated into a forecasting model based on neural networks, specifically the `N-BEATS` model.

2.1 Problem statement

An univariate time series is defined as a sequence of data points $x(t)$, $\mathcal{T} = \{x(t) \mid t = 1, 2, \ldots, N\}$, where N represents the number of points in the time series and $x(t)$ denotes the value of the series at time t. In the case of financial time series data, $x(t)$ typically represents variables such as stock prices, trading volumes, interest rates, and exchange rates, with measurements taken at regular intervals (daily, monthly, annually).

Given this context, the feature extraction technique proposed in this manuscript aims to address the challenge of (point) forecasting in univariate time series. The primary objective is to utilize the last T observations of the time series to generate point forecasts for a specified time horizon H. Formally, this problem can be described as learning a function, parameterized as a neural network, $f : \mathbb{R}^T \rightarrow \mathbb{R}^H$, from a given collection of input–output pairs, where inputs are length-T vectors, and outputs are length-H vectors. For simplicity, a prediction horizon of one point will be considered, i.e. $H = 1$.

2.2 Justification for feature extraction techniques in financial time series

Forecasting in financial markets involves using historical data to predict future market trends, leveraging statistical methods and machine learning algorithms. The primary objective is identifying patterns and trends that inform investment decisions, risk management, and strategic planning. The necessity for robust feature extraction techniques in financial time series forecasting is underscored by the complex and dynamic nature of financial markets, which are characterized by:

- **Non-Stationarity**: Financial time series data are often non-stationary, with changing statistical properties. Techniques such as differencing are employed to achieve stationarity before model implementation [26].
- **High volatility and noise**: Unpredictable factors induce high volatility and noise in financial markets, complicating the identification of genuine patterns. Feature

extraction methods are used to reduce noise and isolate relevant patterns [27].

- **Complex interdependencies**: Financial time series are influenced by interrelated factors, necessitating their consideration for improved forecasting accuracy. Feature extraction techniques incorporate these relationships into models [28].
- **Overfitting and underfitting**: Overfitting captures noise, while underfitting results from overly simplistic models. Feature extraction mitigates these issues by generating informative features [29, 30].
- **Structural breaks**: Sudden changes in the data-generating process disrupt established patterns. Detecting structural breaks is crucial for forecast accuracy, and feature extraction techniques assist in this adaptation [31].

Given these challenges, the necessity of robust feature extraction techniques in financial time series forecasting becomes evident. Feature extraction enhances the predictive power of forecasting models by transforming raw data into informative features that better capture the underlying patterns and trends. By addressing non-stationarity, high volatility, and complex interdependencies, feature extraction techniques improve the accuracy and reliability of financial forecasts. Effective feature extraction allows models to gain deeper insights into the data, making it easier to identify trends and predict future market movements [32].

2.3 Feature extraction via topological data analysis techniques

This manuscript performs feature extraction at the segment level using a slicing window approach rather than across the entire time series. This decomposition divides the original series $\mathcal{T} = \{x(t) \mid t = 1, 2, \ldots, N\} \in \mathbb{R}^N$ into Q overlapping windows of length L, resulting in $\mathcal{T} = (\mathbf{s}_1, \ldots, \mathbf{s}_Q) \in \mathbb{R}^{Q \times L}$ where $Q = N - L + 1$. Each segment $\mathbf{s}_q = (x(q), \ldots, x(q + L - 1)) \in \mathbb{R}^L$ represents a distinct subset of the original data. Topological characteristics specific to each segment are then computed using advanced techniques from TDA, such as single Takens' embeddings, point clouds, persistence diagrams, and Vietoris-Rips complexes. The application of these techniques is detailed for each segment of the time series $\mathcal{T}$, aiming to elucidate localized structures and dynamics within financial data. This approach harnesses sophisticated analytical tools to uncover nuanced patterns and relationships that traditional methods may obscure.

Firstly, single Takens' embeddings are implemented, a method utilized to reconstruct the state space of time series and form a crucial part of time delay embedding techniques in TDA. The mathematical foundation of this approach is derived from Takens' embedding theorem, which asserts that, under certain conditions, the dynamics of a system can be fully reconstructed from a series of observations [33]. Given a time series segment $\mathbf{s}_q$, a point cloud, $\mathcal{C}_q$, in λ-dimensional space is generated by the single Takens' embedding method, where each point $\mathbf{x}_i$ in $\mathcal{C}_q$ is defined as:

$$\mathcal{C}_q = \left\{\mathbf{x}_i \mid \mathbf{x}_i = \left(s_{q,i}, s_{q,i+\tau}, \ldots, s_{q,i+(\lambda-1)\tau}\right), i = 1, 2, \ldots, J\right\},$$

where $\tau \in \mathbb{Z}_+$ denotes the time delay, $\lambda \in \mathbb{Z}_+$ represents the embedding dimension, and $J = L - (\lambda - 1)\tau$ denotes the number of points in the point cloud $\mathcal{C}_q$. Here, $s_{q,i}$ refers to the value at position i within segment $\mathbf{s}_q$. The parameters τ and λ significantly influence the dynamical system's phase space reconstruction. τ governs the temporal spacing of sequential observations in the time series, thereby ensuring the effective distribution of points in the reconstructed space. Its optimal selection minimizes redundancy between successive coordinates, enhancing their independence and capturing essential dynamical information. Conversely, λ determines the number of delayed observations utilized in constructing the phase space. A judicious choice of λ ensures a faithful representation of the original system's dynamics, thereby revealing the discernible structure of the underlying attractor.

In this proposal, the point cloud associated with each q-th segment, $\mathcal{C}_q$, serves as the initial step towards visualizing the inherent geometric structure within financial datasets. By projecting data points into multidimensional space, analysts gain a geometric perspective to observe clusters, trends, and outliers, thus providing a foundational insight into the market's geometrical configuration [34]. From the point cloud, a filtration process tracks the birth and death of topological features (such as connected points, loops, and voids) across successive stages. This process involves incrementally "thickening" the points, visualized as increasing the radius of balls centred at each point in $\mathcal{C}_q$. Mathematically, for $\mathcal{C}_q = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J\}$, the filtration can be represented as a sequence of subspaces:

$$\emptyset = \mathbf{u}_0 \subseteq \mathbf{u}_1 \subseteq \mathbf{u}_2 \subseteq \ldots \subseteq \mathbf{u}_J = \mathcal{C}_q$$

where each $\mathbf{u}_i$ includes points from $\mathcal{C}_q$ based on proximity criteria typically defined by a metric, and $\emptyset$ represents the empty set.

The Vietoris-Rips complex provides a specific approach to construct such a filtration from a point cloud. At a given scale parameter ϵ, denoted $VR_\epsilon(\mathcal{C}_q)$ for the q-th segment, it includes a simplex for every subset of points in $\mathcal{C}_q$ that are pairwise within distance ϵ of each other. This complex grows as ϵ increases:

$$VR_{\epsilon_0}(\mathcal{C}_q) \subseteq VR_{\epsilon_1}(\mathcal{C}_q) \subseteq \ldots \subseteq VR_{\epsilon_k}(\mathcal{C}_q)$$

where $0 = \epsilon_0 < \epsilon_1 < \ldots < \epsilon_k$. As the filtration progresses, new topological features may appear (*birth*) and existing features may merge or disappear (*death*) [3, 35]. For example, a loop *births* when a set of points forms a cycle that does not enclose a filled-in area, and a loop *dies* when the space inside it becomes filled-in.

Persistence diagrams, denoted as $\mathcal{D}_q$ ($q \in \{1, \ldots, Q\}$), represent these topological features by plotting (b_i, d_i) points, where b_i and d_i are the birth and death times of the i-th feature, respectively. These diagrams capture enduring structures amidst market volatility, shedding light on long-term market behaviours [36]. Hence, formally, the persistence diagram associated with the q-th segment is defined as $\mathcal{D}_q = \{(b_i, d_i) \mid i \in I_q\}$, where b_i and d_i represent the birth and death times of the i-th feature, respectively, and I_q is the index set enumerating the topological features identified in the persistence diagram associated with the q-th segment.

Once the persistence diagrams of the different segments have been computed, the next step is to calculate the characteristics of interest for each persistence diagram. In our study, these metrics will be the entropy, amplitude, and number of points, as these three measures provide deeper insights into the underlying market dynamics. Firstly, the *persistent entropy* associated with the q-th segment, $H(\mathcal{D}_q)$, is defined as:

$$H(\mathcal{D}_q) = -\sum_{i \in I_q} p_i \log p_i \tag{1}$$

where $p_i = \frac{r_i}{R}$, $r_i = d_i - b_i$ is the lifetime of the i-th feature, and $R = \sum_{i \in I_q} r_i$ is the total sum of lifetimes. It measures the uncertainty associated with the persistence diagram. High-entropy values indicate a more complex or chaotic underlying system, while lower values suggest more regular or periodic behaviour [37].

Secondly, the *amplitude* of the q-th segment, $A(\mathcal{D}_q)$, is defined as:

$$A(\mathcal{D}_q) = \max_{i \in I_q} r_i = \max_{i \in I_q} (d_i - b_i), \tag{2}$$

and refers to the size of the topological features detected in the data. This can be interpreted as a measure of the 'spread' or 'size' of the features in a particular homological dimension. Different norms or metrics (e.g. bottleneck and Wasserstein) can be used to quantify this measure.

Thirdly, the *number of points* for the q-th segment, $N(\mathcal{D}_q)$, is the number of distinct topological features (such as connected components and loops) identified in the data at various filtration values. It indicates the complexity or richness of the topological structure inherent in the data [38]. $N(\mathcal{D}_q)$ is defined as:

$$N(\mathcal{D}_q) = |I_q|. \tag{3}$$

Figure 1 illustrates the TDA process for a specific time series segment. In Subfigure a, the original time series data segment under analysis is depicted. This segment is transformed using the single Takens embedding method with a time delay $\tau = 72$ and an embedding dimension $\lambda = 2$ to create a point cloud, $\mathcal{C}$, as shown in Subfigure b. Subsequently, Subfigure c presents the persistence diagram, $\mathcal{D}$, derived from the point cloud $\mathcal{C}$. The persistence diagram indicates the presence of connected components ($H0$), loops ($H1$), and voids ($H2$), showcasing the birth and death times of these topological features across the filtration process.

As indicated at the beginning of this section, the feature extraction process is repeated for all segments of the series following a sliding window approach. To illustrate the complete process, Fig. 2 depicts the division of the series into overlapping segments of size L, generating a total of Q windows. For each window, the corresponding point cloud and persistence diagram are computed. The values of entropy, amplitude, and the number of points are calculated from these diagrams. These values are subsequently stored in a vector, which for the q-th segment, is defined as $\mathbf{v}_q = (H(\mathcal{D}_q), A(\mathcal{D}_q), N(\mathcal{D}_q)) \in \mathbb{R}^3$.

2.4 The N-BEATS forecast model

This section offers a comprehensive overview of the N-BEATS model, emphasizing its core elements and the incorporation of TDA features into its architecture. The first subsubsection outlines the fundamental components and operational mechanisms that distinguish N-BEATS as a robust framework for time series forecasting. The second subsubsection details how TDA features are integrated into the N-BEATS model to enrich its capabilities for univariate forecasting.

2.4.1 Overview of key elements of N-BEATS

The neural basis expansion analysis for time series forecasting (N-BEATS) represents a significant advancement in time series forecasting, embodying a paradigmatic shift in generating forecasts by offering a versatile and robust framework for handling univariate and multivariate time series data. Developed by a team of researchers, N-BEATS has demonstrated state-of-the-art performance across various benchmarking datasets, distinguishing itself through a combination of simplicity, effectiveness, and robustness. The model's deep learning architecture, employing fully connected layers and a backward and forward residual stacking mechanism, decomposes the time series into trend, seasonality, and residual components, thereby capturing complex patterns and dependencies in financial data.

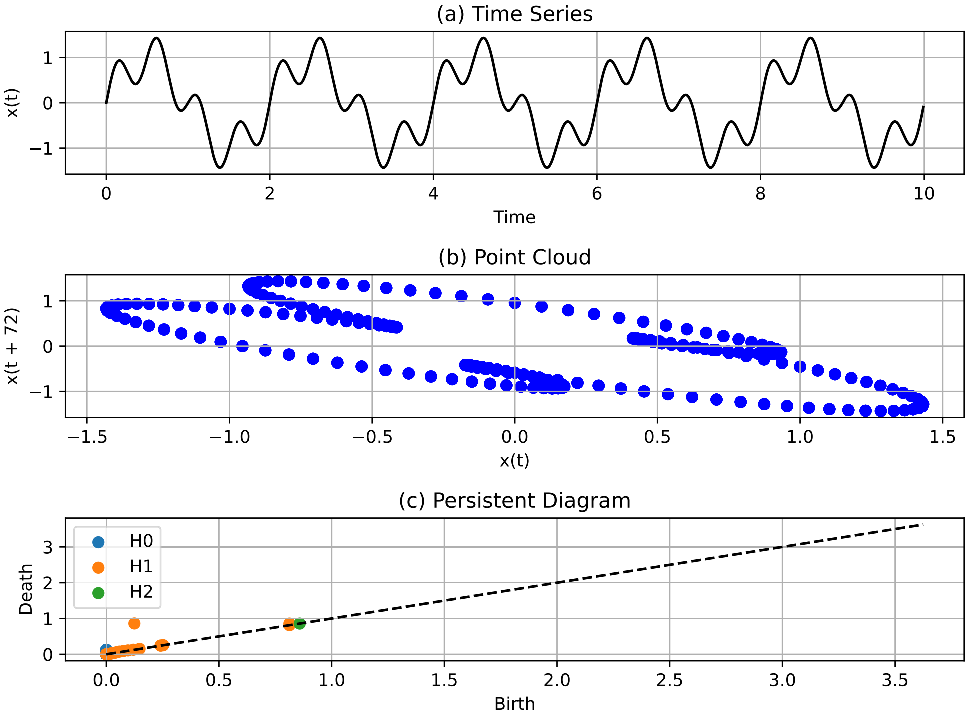


Fig. 1 Illustration of the feature extraction process for a segment of a time series: **a** Example segment of a time series on which feature extraction will be performed; **b** Resulting point cloud diagram obtained using Single Takens embedding with a time delay of $\tau = 72$ and embedding dimension $\lambda = 2$; **c** Persistence diagram generated from **b**, illustrating *connected components* ($H0$), *loops* ($H1$), and *voids* ($H2$)

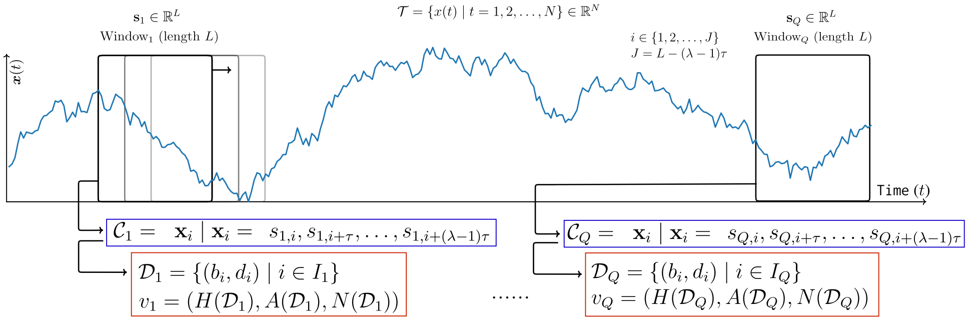

$\mathbf{s}_1 \in \mathbb{R}^L$
Window$_1$ (length L)
$\mathcal{T} = \{x(t) \mid t = 1, 2, \ldots, N\} \in \mathbb{R}^N$
$i \in \{1, 2, \ldots, J\}$
$J = L - (\lambda - 1)\tau$
$\mathbf{s}_Q \in \mathbb{R}^L$
Window$_Q$ (length L)
$x(t)$
Time (t)
$\mathcal{C}_1 = \mathbf{x}_i \mid \mathbf{x}_i = s_{1,i}, s_{1,i+\tau}, \ldots, s_{1,i+(\lambda-1)\tau}$
$\mathcal{D}_1 = \{(b_i, d_i) \mid i \in I_1\}$
$v_1 = (H(\mathcal{D}_1), A(\mathcal{D}_1), N(\mathcal{D}_1))$
……
$\mathcal{C}_Q = \mathbf{x}_i \mid \mathbf{x}_i = s_{Q,i}, s_{Q,i+\tau}, \ldots, s_{Q,i+(\lambda-1)\tau}$
$\mathcal{D}_Q = \{(b_i, d_i) \mid i \in I_Q\}$
$v_Q = (H(\mathcal{D}_Q), A(\mathcal{D}_Q), N(\mathcal{D}_Q))$

Fig. 2 Illustration of the feature extraction procedure via TDA, computed on time series observations $x(1), \ldots, x(N)$. The time series, $\mathcal{T}$, is decomposed into a collection of Q overlapping windows of length L. For each window q, its corresponding point cloud, $\mathcal{C}_q$, is computed, from which the persistence diagram, $\mathcal{D}_q$, is derived. The persistence diagram is then used to compute the values of entropy, amplitude, and number of points, which are stored in the vector $\mathbf{v}_q = (H(\mathcal{D}_q), A(\mathcal{D}_q), N(\mathcal{D}_q)) \in \mathbb{R}^3$

Its basis expansion mechanism expands the input time series into a set of basis functions and enhances adaptability to diverse forecasting tasks. The iterative refinement process within its block structure ensures progressively improved forecasts, leading to higher accuracy and reliability [39].

Notably, `N-BEATS` is flexible, capable of handling time series data with missing values, irregular intervals, and multiple frequencies, and has shown superior performance over traditional statistical models and other deep learning models across multiple benchmarks. For example, `N-BEATS` demonstrates accuracy improvements of 11% over a statistical benchmark, including the ARIMA baseline, and 3% over the 2020 M4 competition winner, which is based on a hybrid model combining a neural residual/attention dilated LSTM stack with a classical Holt-Winters statistical model. The model's interpretability, facilitated by its basis expansion mechanism, provides crucial insights into different components of the time series, aiding financial analysts in understanding the driving factors behind forecasts. Despite its complex architecture, `N-BEATS` is relatively simple to implement and use, requiring minimal pre-processing or feature engineering. It is scalable for efficiently handling large datasets, making it suitable for real-time forecasting scenarios. Its robustness to noise and outliers, critical for maintaining accuracy in unpredictable market events, further underscores its utility in various financial forecasting tasks, including stock prices, exchange rates, and economic indicators [40].

The decision to integrate TDA features as supplementary inputs in N-BEATS was primarily motivated by the model's significant goodness of fit. While the model exhibits high performance, further enhancing forecasting accuracy with additional features presents challenges, given its already strong fit. Moreover, two compelling reasons for including TDA features in N-BEATS are its modular architecture, which facilitates interpretability crucial for understanding predictions in a financial context and its scalability, which ensures robust performance when managing large datasets, thereby rendering it particularly well-suited for various financial applications.

2.4.2 Incorporation of TDA features into N-BEATS architecture

In this section, the integration of TDA features, as shown in Fig. 2, for each segment of the time series into the N-BEATS model [39], is detailed. The standard N-BEATS model is briefly described, with a note that TDA features can similarly be integrated into the basis expansion variant without modifications. The generic N-BEATS model is constructed from a stack of P double-residual blocks. For $1 \leq p \leq P$, each block consists of a non-linear map (implemented as a multi-layer perceptron, MLP). The representation of the N-BEATS model and its variant incorporating TDA features can be observed in Fig. 3, where the P double-residual blocks, their inputs, and their outputs are depicted. As can be seen, the structure of the input and output variables in the blocks for both models (the baseline N-BEATS model and the variant with TDA features, $\mathtt{N-BEATS}_{+\mathrm{TDA}}$) is the same, with the particularity that in the TDA-enhanced model, each block receives the corresponding input $\mathbf{x}$ along with the TDA characteristics as additional input, denoted as $\mathbf{v}$. The entry of these TDA characteristics is represented by a red line in Fig. 3.

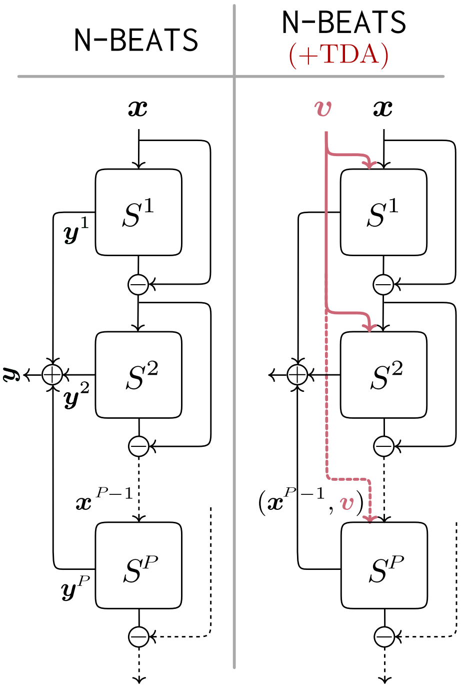


Fig. 3 Illustration of the N-BEATS architecture for univariate time series forecasting (left) and the enhanced N-BEATS architecture incorporating TDA features stored in the vector $\mathbf{v}$ (right)

In the following, the operation of the model will be formally defined starting from the input data $\mathbf{x}$. Firstly, the non-linear mapping of the p-th MLP can be defined as:

$$S^p : \mathbb{R}^T \rightarrow \mathbb{R}^h, \tag{4}$$

where T is the dimension of the inputs (length-T vectors), and h denotes the internal dimensionality. The twofold output of each block (see Fig. 3) is produced as follows:

$$\mathbf{x}^p = \mathbf{x}^{p-1} - \mathbf{U}^p(S^p(\mathbf{x}^{p-1})), \tag{5}$$

and

$$\hat{\mathbf{y}}^p = \mathbf{V}^p(S^p(\mathbf{x}^{p-1})), \tag{6}$$

with $\mathbf{U}^p \in \mathbb{R}^{T\times h}$, $\mathbf{V}^p \in \mathbb{R}^{H\times h}$, and $\mathbf{x}^0 = \mathbf{x}$, where H is the output dimension (assumed to be one for simplicity). The outputs $\mathbf{x}^p$ are used as inputs to subsequent computation blocks, while the predicted outputs $\hat{\mathbf{y}}^p$ are utilized to compute the final model prediction $\hat{\mathbf{y}} \in \mathbb{R}^H$ via component-wise summation:

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}^1 + \hat{\mathbf{y}}^2 + \cdots + \hat{\mathbf{y}}^P. \tag{7}$$

Importantly, in this deep neural network forecasting model, the outputs $\mathbf{x}^p$ can be utilized as an interface to integrate additional information. Specifically, in this manuscript, the base input signal to each block is enhanced by concatenating the TDA features stored in the vector $\mathbf{v}$ (see Fig.3), as follows:

$$\mathbf{x}^{\mathrm{p}}_{TDA} = (\mathbf{x}^p, \mathbf{v}) \quad \text{where} \quad \mathbf{v} = (H(\mathcal{D}), A(\mathcal{D}), N(\mathcal{D})), \tag{8}$$

and, as can be observed in Fig. 3, the time series signal $\mathbf{x}$ is provided to the N-BEATS model in its raw form, while the corresponding TDA features, $\mathbf{v}$, are supplied to each block as complementary signals.

2.5 Computational complexity

This section examines the computational complexity involved in incorporating TDA features into the N-BEATS model, which represents the primary contribution of this manuscript. The computation of the persistence diagram is identified as the most significant aspect of TDA feature extraction. The most computationally intensive component within this process is the calculation of the Vietoris-Rips complex, which is characterized by a runtime complexity of $\mathcal{O}(2^J)$ [41, 42]. It should also be noted that these calculations are repeated Q times for each of the Q segments of the time series. Consequently, the computational complexity of this stage is mainly determined by the number of segments and the value of J, where J is defined as $J = L - (\lambda - 1)\tau$. Thus, it is also dependent on the length of each segment L, the dimensionality of the point cloud embedding λ, and the time delay τ.

3 Computational experiments

The computational experiments in this study are designed to compare the performance of traditional feature engineering techniques with the proposed approach that incorporates TDA-derived features. This section outlines the key elements of the experimental framework. Specifically, the following subsections define the datasets, consisting of data from six cryptocurrencies evaluated across four scenarios. Pre-processing techniques, including handling missing data, outlier detection, and stationarity tests, are detailed, along with the parameters used in the baseline forecasting model, `N-BEATS`.

Given that the primary objective is to assess performance improvements due to the inclusion of additional features, the `N-BEATS` model is consistently used as the baseline forecasting model across all feature extraction techniques. Therefore, a subsection is also dedicated to defining the traditional feature extraction techniques utilized for comparison with the proposed TDA-derived features. Additionally, the performance metrics employed to evaluate the various methods are specified, ensuring a comprehensive forecasting accuracy and effectiveness assessment.

3.1 Datasets

The selected datasets include time series data for six major cryptocurrencies-BNB (Binance Coin), BTC (Bitcoin), ETH (Ethereum), XRP (Ripple), ADA (Cardano), and LTC (Litecoin)-all denominated in USDT (Tether) [43, 44], along with additional financial instruments such as the CAC40 index (FCHI), iShares Core MSCI Europe ETF (IEUR), the EUR/USD exchange rate (EURUSD = X), and crude oil futures (CL = F).

For cryptocurrencies, USDT, a stablecoin pegged to the USD at a 1:1 ratio helps mitigate fluctuations in fiat currencies, allowing a more explicit focus on cryptocurrency market behaviour. This approach enables the precise isolation of intrinsic factors and ensures consistent comparisons across assets and periods. As a result, it allows for detailed examinations of how individual assets respond to different stimuli, enhancing the understanding of the broader digital asset ecosystem [45].

In addition to cryptocurrencies, the experimental framework includes several traditional financial instruments that further broaden the analysis. The CAC40 (FCHI) is a major French stock market index representing the 40 most significant companies on the Euronext Paris, providing a benchmark for tracking Paris market performance. The iShares Core MSCI Europe ETF (IEUR) offers a diversified portfolio of European equities, serving as a proxy for European market trends and occasionally exhibiting bond-like behaviour in terms of stability. The EUR/USD exchange rate (EURUSD = X), one of the most traded currency pairs globally, reflects the relative strength of the European and US economies, making it critical for international trade analysis. Finally, crude oil futures (CL = F) track the price of West Texas Intermediate (WTI) crude oil, a vital commodity whose price movements significantly impact global economic conditions. Together, these instruments complement the cryptocurrency data by providing a holistic view of diverse asset classes, enabling cross-asset analysis and a comprehensive understanding of market dynamics.

The cryptocurrency data spans from July 1, 2019, to December 28, 2021, capturing a complete market cycle with high volatility, bull and bear markets, and stabilization phases. This period is crucial for assessing cryptocurrency performance about macroeconomic factors and regulatory changes [46]. Data are collected at 1-h intervals, balancing granularity and manageability. This frequency captures short-term market movements while reducing data overload, facilitating the analysis of intraday patterns and the impact of global events. Four distinct scenarios are created for each cryptocurrency to evaluate the model comprehensively. These scenarios allow for varied training and testing conditions, enhancing the robustness of the analysis.

In addition to cryptocurrencies, traditional financial instruments such as the CAC40 index (FCHI), crude oil futures (CL = F), the EUR/USD exchange rate (EURUSD), and the iShares Core MSCI Europe ETF (IEUR) are also analysed. Each of these instruments is evaluated using a consistent scenario structure. Specifically, two scenarios are defined for each asset: The first covers from April 1, 2023, to November 14, 2023, and the second extends from November 15, 2023, to June 29, 2024. This uniform approach ensures that model performance is assessed consistently across different asset classes and periods.

Finally, Table 1 outlines the 32 scenarios considered in the study, each assigned a unique ID for ease of reference. The table details the specific cryptocurrency or traditional financial instrument tested and the periods designated for

Table 1 Characteristics of the time series scenarios for cryptocurrencies and traditional financial instruments used in the experimental study. Each entry includes the scenario ID, asset name, training period, and testing period

ID	Asset	Start train date	End train date	Start test date	End test date
1	BNBUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
2	BNBUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
3	BNBUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
4	BNBUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
5	BTCUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
6	BTCUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
7	BTCUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
8	BTCUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
9	ETHUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
10	ETHUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
11	ETHUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
12	ETHUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
13	XRPUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
14	XRPUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
15	XRPUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
16	XRPUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
17	ADAUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
18	ADAUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
19	ADAUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
20	ADAUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
21	LTCUSDT	01/07/2019	28/12/2019	29/12/2019	13/02/2020
22	LTCUSDT	14/02/2020	12/08/2020	13/08/2020	28/09/2020
23	LTCUSDT	29/09/2020	28/03/2021	29/03/2021	14/05/2021
24	LTCUSDT	15/05/2021	11/11/2021	12/11/2021	28/12/2021
25	FCHI	01/04/2023	14/10/2023	15/10/2023	14/11/2023
26	FCHI	15/11/2023	29/05/2024	30/05/2024	29/06/2024
27	CL = F	01/04/2023	14/10/2023	15/10/2023	14/11/2023
28	CL = F	15/11/2023	29/05/2024	30/05/2024	29/06/2024
29	EURUSD	01/04/2023	14/10/2023	15/10/2023	14/11/2023
30	EURUSD	15/11/2023	29/05/2024	30/05/2024	29/06/2024
31	IEUR	01/04/2023	14/10/2023	15/10/2023	14/11/2023
32	IEUR	15/11/2023	29/05/2024	30/05/2024	29/06/2024

training and testing. These selected assets and timeframes were chosen for their ability to offer a thorough and nuanced perspective of the cryptocurrency market.

3.2 Data pre-processing

This section outlines the methodologies employed in the data pre-processing stage, including the treatment of missing values, computation of log returns, management of outliers, assessment of stationarity, normalization of data, and application of denoising techniques. These procedures have been implemented sequentially to ensure the data's integrity, accuracy, and analytical suitability [47].

Missing values were addressed using cubic spline interpolation, selected for its ability to fit a smooth curve through existing data points, thereby preserving continuity in the time series. This method is favored over linear techniques due to its effectiveness in managing the complexities inherent in financial time series data, as it approximates missing values by fitting cubic polynomials between data points, ensuring continuity of the first and second derivatives [48].

After addressing the missing values, log returns were computed to facilitate analysis. Log returns are calculated by taking the natural logarithm of the current price ratio to the previous price. This transformation is essential in financial time series analysis as it stabilizes variance and enhances the data's suitability for statistical modelling. The resulting log returns series constitutes the univariate series that will be the focus of subsequent experiments.

Identifying and adjusting outliers are essential for maintaining the robustness of the analysis, as they can skew results and lead to erroneous conclusions. In this study, outliers were detected using the Z-score method, standardizing the data to highlight values deviating from the mean. While a traditional Z-score threshold of ±3 is standard, this study adjusted the bounds to the 0.02 and 0.98 quantiles to better accommodate the heavy-tailed nature of financial returns. Observations outside these bounds were examined and adjusted to the nearest bound value, preserving data integrity while mitigating the influence of outliers.

Stationarity is a fundamental assumption in time series modelling, indicating that statistical properties such as mean, variance, and autocorrelation remain constant over time. The Augmented Dickey-Fuller (ADF) test was employed to assess stationarity, which enhances the Dickey-Fuller test by including lagged differences to address higher order autocorrelation. The null hypothesis of the ADF test states that the series has a unit root (is non-stationary), while the alternative hypothesis suggests stationarity. In this study, the ADF test was applied to the log-return series, and a sufficiently low p-value would indicate the rejection of the null hypothesis, confirming the series' stationarity [49].

After verifying stationarity, the data were normalized using the MinMaxScaler, scaling values to a range between -1 and 1, which enhances the performance of machine learning algorithms. The final pre-processing step involved denoising the data through the maximal overlap discrete wavelet transform (MODWT). This method effectively decomposes the time series into multiple resolution components, facilitating noise separation from the signal. The Daubechies 2 (db2) wavelet [50] was chosen for its compact support and smoothness, and five decomposition levels were applied. The first-level approximation, which captures the most significant underlying trends while filtering out high-frequency noise, was retained for subsequent analysis. Finally, the data pre-processing stage is illustrated in detail in the flowchart diagram presented in Fig. 4.

3.3 Feature extraction methods used for comparison purposes

Given that the proposal includes TDA-derived features in future forecasting models and, as previously indicated, the N-BEATS model is employed as the forecasting model for all tested methods, the proposed strategies for comparison can be divided into two categories. First, the benchmark model, the N-BEATS, approaches the forecasting problem from a purely univariate perspective, using the T previous data points to make the forecast. The objective of including this model is to evaluate how including new features helps improve the baseline model. The second group of methods included in the experimental study comprises alternative feature extraction models. These models, similar to the proposed strategy, $\mathtt{N-BEATS}_{+\mathrm{TDA}}$, add additional information from the extracted features to the estimation window originally used by the N-BEATS model. Therefore, the feature extraction strategies considered are:

1. The $\mathtt{N-BEATS}_{+\mathrm{LF}}$ method incorporates lagged features (LF) as inputs for predicting future values. The underlying idea of this model is that by including lagged returns, trends, and patterns in the data, such as momentum or mean reversion, can be captured [51]. This experimental study added lagged log returns with 1, 3, 7, 14, 21, and 30 days as features of this strategy.
2. The $\mathtt{N-BEATS}_{+\mathrm{DTF}}$ method integrates date and time features (DTFs) in the inputs included in the baseline N-BEATS model. These elements typically include the hour of the day, the day of the week, and whether the day is a holiday. Such features can capture time-dependent patterns, such as increased trading activity at market open/close or behavioural changes on Fridays compared to midweek [52], as financial markets are influenced by price movements, human behaviours, and operational aspects that vary over time. In this experimental study, the features added within this strategy include the year, month, day, hour, day of the week, day of the year, and number of days in a month.
3. The $\mathtt{N-BEATS}_{+\mathrm{DCF}}$ method includes difference and change features (DCF) as additional inputs in the baseline N-BEATS forecasting model. These features involve creating new variables based on the differences or percentage changes between various points in the time series. For example, this could include the log-return difference between consecutive days or the percentage change over the last 3 days. Such features help identify market trends and volatility [53]. In this study, the implementation of the $\mathtt{N-BEATS}_{+\mathrm{DCF}}$ method includes the rate of change and differences using lagged log returns of 1, 3, 7, 14, 21, and 30 days.
4. The $\mathtt{N-BEATS}_{+\mathrm{STDF}}$ method involves seasonal-trend decomposition features (STDF), which in this study are derived using the seasonal and trend decomposition using LOESS (STL) technique. STL utilizes LOESS (local regression) to decompose time series into seasonal, trend, and residual components, thereby facilitating the analysis and comprehension of intricate patterns within time series data. The trend component signifies the long-term evolution of the series, seasonality reveals recurring patterns occurring at fixed intervals, and the residual denotes what remains after the removal of trend and seasonality [54].

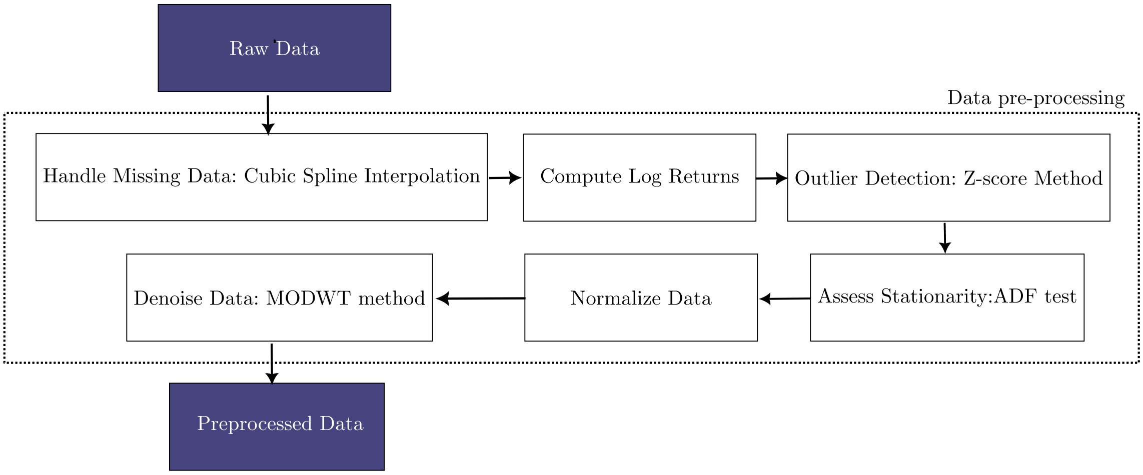

Raw Data
Data pre-processing
Handle Missing Data: Cubic Spline Interpolation
Compute Log Returns
Outlier Detection: Z-score Method
Denoise Data: MODWT method
Normalize Data
Assess Stationarity:ADF test
Preprocessed Data

Fig. 4 Flow diagram of the data pe-processing stage

5. The $\mathtt{N-BEATS}_{+\mathrm{TDEF}}$ method incorporates time delay embedding features (TDEFs), which involves constructing vectors from time series data by selecting a point and its lagged values up to a specified dimension, known as the embedding dimension. This technique is grounded in Takens' embedding theorem, which proposes that the dynamics of a system can be reconstructed using a single observable, provided that the embedding dimension and time delay are appropriately chosen. In this approach, the time delays considered in the methods are 1, 3, 7, 14, 21, and 30 days.

The proposed model, $\mathtt{N-BEATS}_{+\mathrm{TDEF}}$, is influenced significantly by two parameters, namely the time delay, τ, and the segment size used for computing TDA features, L. In the experimental study, potential values of 3 days and 7 days were considered for the time delay (which help in capturing temporal dynamics and dependencies in the data), while segment sizes of 2 weeks, 3 weeks, 1 month, and 1 month and a half were considered, offering a range of temporal granularities for feature extraction.

The hyperparameters of both the comparison methods and the proposed method, $\mathtt{N-BEATS}_{+\mathrm{TDEF}}$, were estimated using nested cross-validation with a validation window within the training set. During the model selection phase, an exhaustive grid search was performed over the predefined hyperparameter values specified earlier for each method. This ensures independent hyperparameter tuning and prevents overfitting, providing a robust estimate of model performance on unseen data.

Figure 5 portrays a detailed historical TDA feature extraction process. It outlines iterative stages for generating TDA features from pre-processed data. Initially, the data are segmented using specified parameters: time delays, τ, of 3 days and 7 days and segment sizes, L, of 2 weeks, 3 weeks, 1 month, and 1 month and a half months, ensuring comprehensive temporal coverage. Subsequently, point clouds are derived from these segments to visualize higher dimensional data structures. Persistent diagrams are then constructed to capture topological features across different scales. The entropy, amplitude, and number of points are extracted from these diagrams, distilling complex topological information into interpretable measures. This iterative process iterates through each time slice, facilitating a thorough exploration of temporal dimensions. The historical TDA features are ultimately exported for subsequent analytical or modelling tasks.

Once the historical features have been computed for both the proposed TDA strategy and the comparison methods, the next step involves merging them with the pre-processed data. In this merging process, historical features and pre-processed data are aligned based on date. This step is crucial to maintaining the temporal correspondence between different datasets.

3.4 Configuration of `N-BEATS` model parameters

The `N-BEATS` model is implemented for time series forecasting in this study, known for its robustness and effectiveness across diverse forecasting tasks. Specific hyperparameters are tailored to the dataset and objectives. The `N-BEATS` model is configured as follows: It uses an input chunk length of 7, $T = 7$, to consider one week's historical data per prediction, an output chunk length of 1 for fine-grained forecasts, $H = 1$, and a batch size of 16 to balance training speed and memory usage. The random state is fixed at 0 for reproducibility. Training spans 100 epochs to capture underlying data patterns effectively while preventing overfitting. With two layers, each 512 units wide, the model captures various levels of data

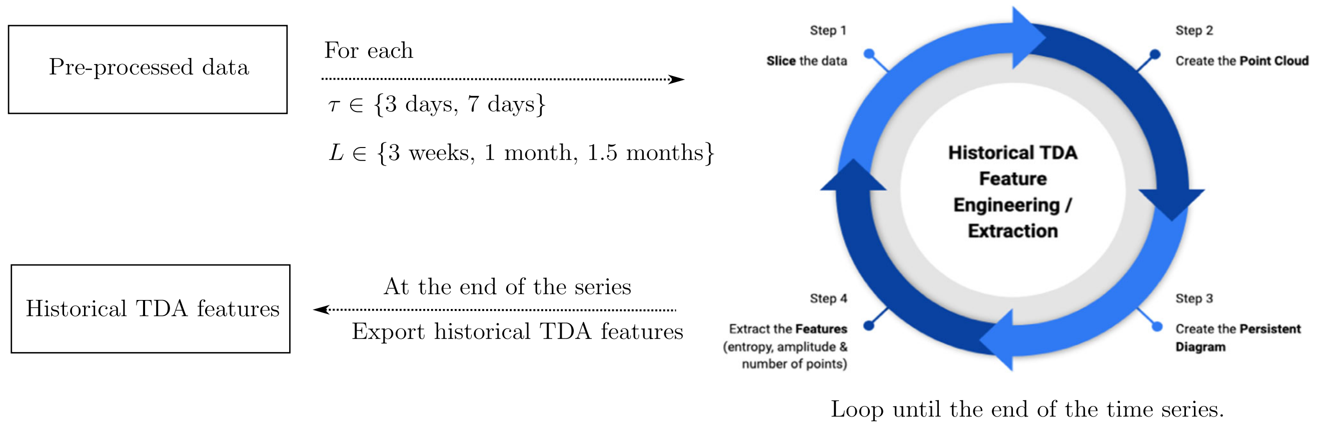

Pre-processed data
For each
$\tau \in \{3 \text{ days}, 7 \text{ days}\}$
$L \in \{3 \text{ weeks}, 1 \text{ month}, 1.5 \text{ months}\}$
Historical TDA features
At the end of the series
Export historical TDA features
Step 1
Slice the data
Step 2
Create the Point Cloud
Historical TDA Feature Engineering / Extraction
Step 3
Create the Persistent Diagram
Step 4
Extract the Features (entropy, amplitude & number of points)
Loop until the end of the time series.

Fig. 5 Workflow for historical TDA feature extraction

abstraction. The mean squared error (MSE) is the loss function, penalizing prediction errors to enhance accuracy.

These parameters collectively optimize model performance on the selected dataset, balancing complexity and computational efficiency to deliver competitive forecasts. Each parameter has been carefully selected based on preliminary experiments and domain knowledge, ensuring that the `N-BEATS` model is well-suited to handle the dataset's specific characteristics and challenges. This configuration leverages the `N-BEATS` model's strengths, making it highly effective for the time series forecasting tasks in this study.

3.5 Performance measures

To evaluate the quality of point forecasts, three widely recognized metrics are employed in this study: the mean absolute percentage error (MAPE), the mean absolute error (MAE), and the root mean squared error (RMSE). These metrics have been extensively validated in the forecasting literature for their ability to measure different dimensions of predictive accuracy [39, 55, 56]. MAPE is appreciated for its interpretability and scale independence, although issues can arise with small or zero actual values. A straightforward, unit-specific measure of error magnitude is offered by MAE, though all errors are treated equally, potentially overlooking the impact of larger deviations. A more nuanced assessment is provided by RMSE, which penalizes larger errors more severely, making it suitable for contexts where significant consequences are associated with such errors [57]. By combining these metrics, a comprehensive evaluation of forecasting accuracy is ensured, addressing various aspects of model performance and aligning with best practices in the field.

Let $\hat{\mathbf{y}} = (\hat{y}_{N+1}, \ldots, \hat{y}_{N+M}) \in \mathbb{R}^M$ denotes the forecasting in the testing set, where the testing data have a size of M and follow the training set. Let $\mathbf{y} = (x_{N+1}, \ldots, x_{N+M}) \in \mathbb{R}^M$ represents the true observations in the test set. The three performance measures are defined based on these two vectors as follows:

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100}{M}, \mathrm{MAE} = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{M},$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2}.$$

4 Computational results

In this section, the results obtained for the testing sets are analysed. Specifically, the performance of three forecasting metrics, as produced by the $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ method, is evaluated against the baseline method, `N-BEATS`, and five other feature extraction methodologies. The purpose of this analysis is to demonstrate that the incorporation of TDA features into the forecasting algorithm can enhance the performance of the models. In Table 2, the MAPE results for the testing set across 32 forecasting scenarios are presented. Based on the results from these 32 datasets, the ranking of each method for each dataset ($R = 1$ for the best-performing method and $R = 7$ for the worst) has been determined. The mean MAPE in the test set ($\overline{\mathrm{MAPE}_T}$) and the mean ranking ($\overline{R}_{\mathrm{MAPE}_T}$) is also included in Table 2.

In a similar manner, the performance of the methods used in the experimental study is shown in Tables 3 and 4 for the MAE and RMSE metrics, respectively, across the 32 forecasting scenarios considered in the test set. Additionally, the ranking for each method in each forecasting problem is calculated for each metric. The mean values of MAE and RMSE for the different methods, $\overline{\mathrm{MAE}_T}$ and $\overline{\mathrm{RMSE}_T}$, as well as the mean rankings of the different methods, $\overline{R}_{\mathrm{MAE}_T}$ and $\overline{R}_{\mathrm{RMSE}_T}$, are also computed.

Table 2 Comparative performance of the proposed model, $\mathrm{N-BEATS}_{+\mathrm{TDA}}$, the baseline method N-BEATS, and other feature extraction approaches for the MAPE metric, along with the average MAPE for the 32 scenarios considered in the test set, $\overline{\mathrm{MAPE}_T}$, and the average ranking of MAPE in the test set, $\overline{R}_{\mathrm{MAPE}_T}$

ID	MAPE						
	N-BEATS	$\mathrm{N-BEATS}_{+\mathrm{LF}}$	$\mathrm{N-BEATS}_{+\mathrm{DTF}}$	$\mathrm{N-BEATS}_{+\mathrm{DCF}}$	$\mathrm{N-BEATS}_{+\mathrm{STDF}}$	$\mathrm{N-BEATS}_{+\mathrm{TDEF}}$	$\mathrm{N-BEATS}_{+\mathrm{TDA}}$
1	139.0532	162.2351	407.7368	174.2547	148.2743	*128.4805*	**106.0596**
2	118.8600	121.5141	116.2636	*111.8443*	122.6504	196.6156	**110.2331**
3	*114.5825*	173.3484	120.9067	167.8835	149.2859	172.8479	**100.0587**
4	246.2749	*163.3496*	166.8307	188.2978	254.4878	197.2818	**125.6246**
5	663.6596	*302.4272*	459.9897	619.5670	444.0604	491.6656	**108.4535**
6	180.0241	186.9720	225.8039	200.1561	**147.1103**	184.8574	*162.9926*
7	*224.8626*	253.1376	225.8194	352.4682	272.5163	343.8078	**132.9329**
8	168.7723	*123.7964*	129.6621	178.8952	142.0266	176.9891	**93.7651**
9	140.0813	157.0952	212.9902	155.3424	*125.9206*	170.7988	**125.1705**
10	174.5808	156.4421	154.5137	161.1009	279.1564	**121.8786**	*143.1173*
11	263.3684	148.2042	*134.0920*	149.1842	176.5331	200.4140	**109.8854**
12	187.6590	187.0697	*132.8695*	182.7354	259.4022	214.1873	**120.8771**
13	*117.1035*	317.6217	638.9338	272.4830	308.5162	245.5998	**116.9503**
14	**136.1999**	*169.5358*	208.8216	270.7827	309.0858	170.5683	175.0513
15	139.2935	126.8466	203.3415	151.3640	171.5754	*112.8027*	**91.2461**
16	204.8576	197.1004	*152.5100*	186.0576	235.2554	208.5322	**132.9208**
17	*241.2468*	574.0669	2796.7173	286.3996	864.3816	725.5830	**190.5805**
18	220.0000	163.6418	222.3528	*127.8077*	234.0949	216.9085	**127.3136**
19	847.2344	1236.9432	772.0796	935.6511	*601.7099*	934.3295	**194.3418**
20	381.3435	**176.8201**	543.6985	328.6116	*214.5206*	381.3833	244.6084
21	229.5983	*179.7147*	547.4617	280.5115	335.6855	180.9279	**128.8849**
22	310.6935	286.3842	*226.7691*	414.2465	469.9199	750.6972	**112.2881**
23	*130.6911*	143.8042	135.8702	141.0580	180.4357	186.6511	**119.1654**
24	175.2214	298.4303	*142.7205*	189.2654	297.0766	314.1151	**137.5442**
25	201.1486	205.1602	257.0312	**136.0107**	391.7804	397.6362	*201.0841*
26	130.2453	*114.4795*	165.6257	123.5153	172.8619	141.9694	**106.5971**
27	1728.313	*283.2507*	1232.6260	538.3600	1151.3333	1520.2748	**160.4478**
28	1033.847	506.0304	622.0527	*443.7465*	757.9577	456.9261	**343.7265**
29	**197.5535**	243.7566	235.1660	382.4330	280.6724	259.8281	*201.9964*
30	**123.3428**	205.9400	313.4620	391.5359	205.5172	*201.4490*	218.7700
31	*115.2531*	**110.2371**	244.1184	124.0788	262.7196	122.6542	152.2304
32	338.5904	810.8122	*206.3660*	319.9599	427.9690	403.8777	**169.5470**
$\overline{\mathrm{MAPE}_T}$	300.7361	*265.1927*	386.1000	271.4252	324.8279	329.1418	**148.8895**
$\overline{R}_{\mathrm{MAPE}_T}$	3.8437	*3.8125*	4.4687	4.3750	5.1562	4.8125	**1.5312**

The best result is in boldface, and the second best result is in italics

From the analysis of the results, it can be concluded, from a purely descriptive point of view, that the $\mathrm{N-BEATS}_{+\mathrm{TDA}}$ method achieved the best results for 24 datasets in MAPE, for 18 datasets in MAE, and 19 datasets in RMSE. Additionally, the second best performance was obtained for four additional datasets in MAPE, eight datasets in MAE, and three datasets in RMSE. Furthermore, the $\mathrm{N-BEATS}_{+\mathrm{TDA}}$ method yielded the best mean performance across all three metrics ($\overline{\mathrm{MAPE}_T} = 148.8895$, $\overline{\mathrm{MAE}_T} = 0.1971$, $\overline{\mathrm{RMSE}_T} = 0.2538$), as well as the best mean rankings ($\overline{R}_{\mathrm{MAPE}_T} = 1.5312$, $\overline{R}_{\mathrm{MAE}_T} = 1.9062$, $\overline{R}_{\mathrm{RMSE}_T} = 2.0625$).

Furthermore, a graphical representation is also provided to substantiate the validity of these results. Figure 6 compares the performance of the methods included in the

Table 3 Comparative performance of the proposed model, $\mathtt{N-BEATS}_{+\mathrm{TDA}}$, the baseline method $\mathtt{N\text{-}BEATS}$, and other feature extraction approaches for the MAE metric, along with the average MAE for the 32 scenarios considered in the test set, $\overline{\mathrm{MAE}_T}$, and the average ranking of MAE in the test set, $\overline{R}_{\mathrm{MAE}_T}$

ID	MAE						
	$\mathtt{N\text{-}BEATS}$	$\mathtt{N-BEATS}_{+\mathrm{LF}}$	$\mathtt{N-BEATS}_{+\mathrm{DTF}}$	$\mathtt{N-BEATS}_{+\mathrm{DCF}}$	$\mathtt{N-BEATS}_{+\mathrm{STDF}}$	$\mathtt{N-BEATS}_{+\mathrm{TDEF}}$	$\mathtt{N-BEATS}_{+\mathrm{TDA}}$
1	0.1929	0.1920	0.3864	0.1937	0.1824	*0.1815*	**0.1757**
2	0.2821	0.2881	0.2745	**0.2596**	0.2726	0.3009	*0.2677*
3	0.3588	0.4055	**0.3205**	0.3908	0.3690	0.4182	*0.3290*
4	0.2223	0.2172	*0.1991*	0.2191	0.2233	0.2413	**0.1919**
5	0.1897	*0.1880*	0.3175	0.1954	0.1995	0.1893	**0.1796**
6	0.1860	0.2064	0.2126	0.2094	**0.1531**	*0.1741*	0.1957
7	0.2424	0.2873	**0.2311**	0.2808	0.2665	0.2581	*0.2367*
8	0.2566	0.2053	*0.1920*	0.2396	0.2175	0.2236	**0.1633**
9	*0.1957*	0.2249	0.2914	0.2526	**0.1875**	0.2144	0.2002
10	*0.2878*	0.3119	0.2991	0.3243	0.3684	**0.2669**	0.3182
11	0.2622	0.2818	*0.2516*	0.2652	0.2622	0.2521	**0.2476**
12	*0.2113*	0.2288	0.2247	0.2230	0.2506	0.2334	**0.1608**
13	*0.1469*	0.2068	0.2845	0.1910	0.1835	0.1750	**0.1403**
14	**0.1568**	0.1925	0.1959	0.2164	0.1900	0.1719	*0.1760*
15	0.3500	0.3678	*0.3429*	0.3813	0.4306	0.3814	**0.3137**
16	0.1850	0.1731	0.1785	**0.1692**	0.1896	0.1973	*0.1708*
17	*0.1917*	0.2511	0.4410	0.2206	0.2257	0.2057	**0.1759**
18	0.2442	0.2602	0.3045	0.2499	*0.2408*	0.2706	**0.2360**
19	0.2504	0.3065	0.2601	0.3216	*0.2446*	0.2685	**0.2167**
20	0.2269	0.2213	0.2744	0.2698	0.2325	*0.2192*	**0.2076**
21	0.2568	*0.2193*	0.3230	0.2463	0.2423	0.2305	**0.2063**
22	**0.1883**	0.2695	0.2870	0.2535	0.2431	0.2591	*0.2143*
23	0.2626	0.2835	*0.2530*	0.2576	**0.2455**	0.3156	0.2629
24	0.2119	0.2189	*0.2079*	0.2218	0.2036	0.2291	**0.2037**
25	0.1472	0.1484	0.1603	**0.1369**	0.2132	0.1991	*0.1436*
26	**0.1510**	0.1609	0.1738	0.1656	0.1662	0.1604	*0.1520*
27	0.2321	0.1980	0.2079	*0.1848*	0.1993	0.1967	**0.1708**
28	0.1104	*0.1001*	0.1166	0.1103	0.1243	0.1129	**0.0702**
29	*0.1486*	0.1678	0.1542	0.1713	0.1798	0.1738	**0.1353**
30	**0.1379**	0.1473	0.1500	0.1569	*0.1421*	0.1524	0.1436
31	**0.1254**	0.1332	0.2162	0.1322	0.1599	*0.1293*	0.1299
32	0.1691	0.2619	*0.1568*	0.1745	0.2029	0.2065	**0.1561**
$\overline{\mathrm{MAE}_T}$	*0.2119*	0.2289	0.2465	0.2276	0.2253	0.2252	**0.1971**
$\overline{R}_{\mathrm{MAE}_T}$	*3.2656*	4.5937	4.7187	4.6250	4.3906	4.5000	**1.9062**

The best result is in bold face and the second best result in italics

experimental study against the proposed $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ method. On the X-axis, the performance of the proposed method is displayed, while the Y-axis shows the performance of the comparison methods. Thus, in the scatter plot of Fig. 6, each point represents a comparison between $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ and another methodology on a single forecasting dataset. In this regard, points above the $y = x$ line indicate datasets for which $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ achieves better performance than the compared method since all considered metrics (MAPE, MAE, and RMSE) are to be minimized. It can be observed that the majority of points lie above this line, underscoring the proposed method's superior performance compared to the other methods.

In order to determine whether a statistical difference exists between any of these algorithms, a procedure for comparing multiple forecasting models over multiple

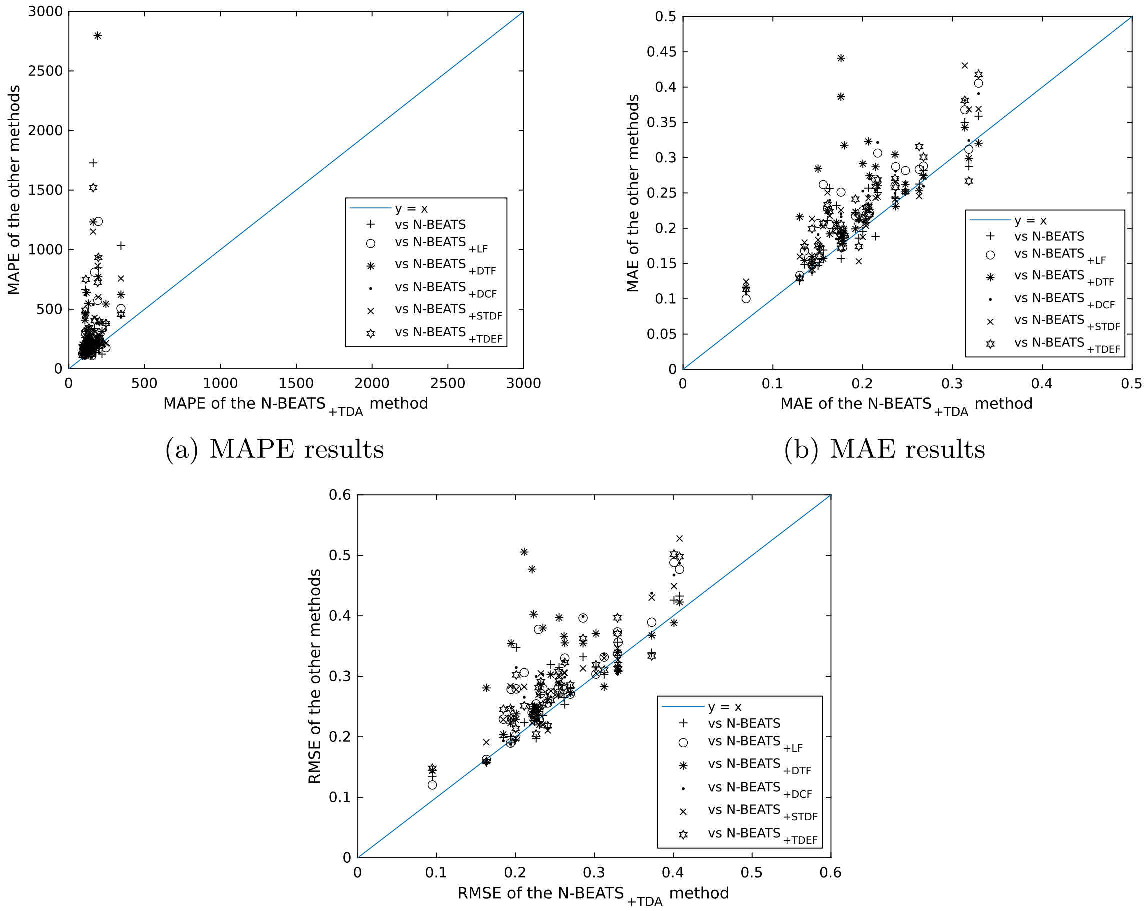
Table 4 Comparative performance of the proposed model, $\mathtt{N-BEATS}_{+\text{TDA}}$, the baseline method `N-BEATS`, and other feature extraction approaches for the RMSE metric, along with the average RMSE for the 32 scenarios considered in the test set, $\overline{\text{RMSE}_T}$, and the average ranking of RMSE in the test set, $\overline{R}_{\text{RMSE}_T}$

ID	RMSE						
	N-BEATS	$\mathtt{N-BEATS}_{+\text{LF}}$	$\mathtt{N-BEATS}_{+\text{DTF}}$	$\mathtt{N-BEATS}_{+\text{DCF}}$	$\mathtt{N-BEATS}_{+\text{STDF}}$	$\mathtt{N-BEATS}_{+\text{TDEF}}$	$\mathtt{N-BEATS}_{+\text{TDA}}$
1	0.2329	0.2402	0.4772	0.2400	*0.2232*	0.2495	**0.2210**
2	0.3565	0.3733	0.3392	*0.3385*	0.3462	0.3967	**0.3293**
3	0.4260	0.4882	**0.3884**	0.4674	0.4490	0.5021	*0.4009*
4	0.2842	0.2811	*0.2691*	0.2887	0.2897	0.3074	**0.2544**
5	0.2462	*0.2375*	0.4027	0.2529	0.2378	0.2380	**0.2230**
6	*0.2156*	0.2560	0.2611	0.2703	**0.2106**	0.2181	0.2409
7	*0.3030*	0.3318	**0.2828**	0.3361	0.3301	0.3103	0.3124
8	0.3476	0.2792	*0.2379*	0.3145	0.2780	0.3023	**0.2009**
9	*0.2355*	0.2798	0.3799	0.3032	0.2201	0.2528	**0.2347**
10	*0.3389*	0.3894	0.3679	0.4374	0.4301	**0.3336**	0.3727
11	0.3226	0.3564	**0.3092**	0.3414	0.3272	*0.3119*	0.3301
12	*0.2695*	0.2716	0.2869	0.2801	0.3048	0.2910	**0.2321**
13	**0.1899**	0.2784	0.3545	0.2474	0.2315	0.2312	*0.1944*
14	**0.1976**	0.2543	0.2483	0.2995	0.2296	*0.2045*	0.2261
15	0.4327	0.4769	*0.4228*	0.4870	0.5279	0.4977	**0.4080**
16	*0.2251*	0.2279	**0.2196**	0.2357	0.2490	0.2481	0.2302
17	*0.2238*	0.3061	0.5055	0.2653	0.2826	0.2511	**0.2110**
18	0.3154	*0.3037*	0.3708	0.3122	0.3054	0.3190	**0.3020**
19	*0.3322*	0.3964	0.3548	0.3991	0.3131	0.3626	**0.2856**
20	*0.2650*	0.2777	0.3663	0.3256	0.3060	0.2824	**0.2615**
21	0.3145	*0.2722*	0.3971	0.2909	0.3022	0.2926	**0.2552**
22	**0.2538**	0.3301	0.3552	0.2987	0.3059	0.3230	*0.2625*
23	0.3344	0.3373	*0.3160*	**0.3034**	0.3136	0.3704	0.3294
24	0.2798	0.2706	0.2729	*0.2700*	0.2727	0.2856	**0.2694**
25	0.2001	*0.1896*	0.2226	**0.1878**	0.2837	0.2466	0.1936
26	*0.2269*	0.2325	0.2498	0.2397	0.2387	0.2336	**0.2243**
27	0.3193	0.2722	0.3027	0.2659	0.2758	*0.2606*	**0.2447**
28	0.1348	*0.1204*	0.1446	0.1421	0.1439	0.1479	**0.0945**
29	0.1987	0.2291	0.2039	*0.1933*	0.2284	0.2454	**0.1845**
30	*0.1945*	0.2019	0.2282	**0.1936**	0.1973	0.2137	0.2003
31	**0.1573**	0.1625	0.2808	0.1601	0.1909	*0.1588*	0.1630
32	0.2451	0.3776	0.2419	*0.2296*	0.2695	0.2811	**0.2291**
$\overline{\text{RMSE}_T}$	*0.2693*	0.2906	0.3143	0.2880	0.2848	0.2865	**0.2538**
$\overline{R}_{\text{RMSE}_T}$	*3.2500*	4.4687	4.7500	4.3437	4.3437	4.7812	**2.0625**

The best result is in boldface, and the second best result is in italics

datasets was employed, as described by Demšar (2006) [58]. The first step involved the application of Friedman's non-parametric test [59] with a significance level of $\alpha = 0.05$ to ascertain the statistical significance of the differences in the mean ranks for each measure. The test rejected the null hypothesis (with $\alpha = 0.05$) that the differences in mean rankings of MAPE, MAE, and RMSE obtained by the different algorithms were equal. Specifically, the confidence interval for this number of datasets and algorithms was $C_0 = (0, \text{F}_{(\alpha=0.05)} = 2.14)$, and the corresponding F-values for each metric were $13.59 \notin C_0$, $9.53 \notin C_0$, and $8.35 \notin C_0$, respectively.

According to the guidelines outlined by Demšar (2006) [58], the best-performing feature extraction method, $\mathtt{N-BEATS}_{+\text{TDA}}$, was designated as the control method and compared to the remaining methods based on their



(a) MAPE results

(b) MAE results

(c) RMSE results

Fig. 6 Scatter plot comparing the performance in MAPE, MAE, and RMSE of the proposed $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ method against the baseline and other comparison methods. The X-axis represents the performance of the $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ method, while the Y-axis represents the performance of the comparison methods across these metrics

rankings. It has been observed that comparing all forecasting models to each other in a post hoc test is less sensitive than comparing all forecasting models to a specific control method. One statistical procedure for conducting this type of comparison is the Holm test. The test statistic for comparing the i-th and j-th methods using this procedure is defined as:

$$z = \frac{\overline{R}_i - \overline{R}_j}{\sqrt{\frac{k(k+1)}{6N}}}, \tag{9}$$

where k is the number of methods, N is the number of datasets, and $\overline{R}_i$ is the mean ranking of the i-th method. The z value is used to determine the corresponding probability from the normal distribution table, which is then compared with an appropriate significance level, α. Holm's test adjusts the value for α to compensate for multiple comparisons. This is accomplished through a step-up procedure that sequentially tests the hypotheses ordered by their significance. The ordered p-values are denoted as $p_1, p_2, \ldots, p_k$ such that $p_1 \leq p_2 \leq \ldots \leq p_{k-1}$. Holm's test compares each p_i with $\alpha'_{\mathrm{Holm}} = \alpha/(k-i)$, starting with the most significant p-value. If p_1 is below $\alpha/(k-1)$, the corresponding hypothesis is rejected, and p_2 is compared with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. All remaining hypotheses are retained if a null hypothesis cannot be rejected.

The results of the Holm test for $\alpha = 0.05$ are presented in Table 5, utilizing the corresponding p and α'_{Holm} values. Based on the outcomes of this test, it is inferred that higher rankings of MAPE, MAE, and RMSE are achieved by the $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ method compared to all other methods $\alpha = 0.05$. As can be observed in the statistical accuracy results of forecasting, high competitiveness is demonstrated by the baseline method, N-BEATS, and enhancing its prediction performance through additional features is deemed challenging, which appears to be accomplished by

Table 5 Adjusted p-values using MAPE, MAE, and RMSE as the test variables ($\mathtt{N-BEATS}_{+\mathrm{TDA}}$ is the control method)

Variable Test: MAPE		Variable Test: MAE		Variable Test: RMSE		Holm	
Algorithm	p-value	Algorithm	p-value	Algorithm	p-value	α'_{Holm} (0.05)	α'_{Holm} (0.10)
$\mathtt{N-BEATS}_{+\mathrm{STDF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{DTF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{TDEF}}$	$0.000_{\circ}$	0.0166	0.0083
$\mathtt{N-BEATS}_{+\mathrm{TDEF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{DCF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{DTF}}$	$0.000_{\circ}$	0.0200	0.0100
$\mathtt{N-BEATS}_{+\mathrm{DTF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{LF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{LF}}$	$1.0\mathrm{E}{-5}_{\circ}$	0.0250	0.0125
$\mathtt{N-BEATS}_{+\mathrm{DCF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{TDEF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{STDF}}$	$2.0\mathrm{E}{-5}_{\circ}$	0.0333	0.0166
N-BEATS	$2.0\mathrm{E}{-5}_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{STDF}}$	$0.000_{\circ}$	$\mathtt{N-BEATS}_{+\mathrm{DCF}}$	$2.0\mathrm{E}\text{-}5_{\circ}$	0.0500	0.0250
$\mathtt{N-BEATS}_{+\mathrm{LF}}$	$2.0\mathrm{E}{-5}_{\circ}$	N-BEATS	$0.011_{\circ}$	N-BEATS	$0.027_{\circ}$	0.1000	0.0500

$\circ$: statistically significant differences for $\alpha = 0.05$ s

the proposed method through the incorporation of features derived from TDA.

5 Discussion

The results of this study highlight the effectiveness of integrating topological data analysis (TDA)-derived features into time series forecasting models, specifically the N-BEATS model. By capturing complex, non-linear dependencies through TDA features such as persistent entropy, amplitude, and the number of points in persistence diagrams, these findings confirm the utility of topological features in enhancing predictive accuracy. The significance of these results is twofold: they validate the relevance of TDA in capturing inherent structures within financial time series and suggest several promising pathways for future research to extend TDA's applicability within the financial domain.

One potential extension for TDA-based feature engineering lies in its application to more complex modeling environments, particularly in reinforcement learning (RL) trading strategies. In recent years, RL has emerged as a pivotal tool for creating automated trading strategies within financial markets. The adaptability of RL to the dynamic and often volatile nature of financial markets underscores its potential, as it learns optimal actions through trial and error within complex environments. However, the effectiveness of RL-based strategies heavily depends on the quality and relevance of features extracted from market data. Incorporating TDA-derived features could contribute to a richer, more informative representation of market dynamics, thereby enhancing the adaptability of RL algorithms to variable market conditions. With their capacity to capture intricate structures and persistent patterns in data, TDA features could bridge the gap between the highly non-linear, high-dimensional characteristics of financial data and the decision-making framework of RL-based trading systems. This integration may improve RL model performance and contribute to innovative avenues in quantitative finance.

In addition to RL applications, TDA's role in high-frequency trading (HFT) environments warrants further investigation. High-frequency data present unique challenges due to its high dimensionality and inherent noise, which complicate traditional modelling approaches. With its inherent ability to capture persistent structures amidst chaotic data, TDA could prove valuable in extracting meaningful patterns within this context. Thus, future studies could explore the utility of TDA in high-frequency trading datasets to evaluate its effectiveness in real-time decision-making. If successful, such applications could support high-frequency trading systems by providing enhanced interpretability of market signals in rapid trading scenarios.

Another promising area for future exploration involves combining TDA-derived features with advanced deep learning architectures beyond the N-BEATS model. Although N-BEATS was selected for its robust performance in time series forecasting and its flexibility in feature integration, further exploration could assess the benefits of integrating TDA features within models like transformers or graph neural networks. These models, particularly suited for capturing complex dependencies in non-stationary time series, may enhance predictive accuracy when combined with TDA-derived features. Additionally, applying TDA features within ensemble frameworks may broaden the spectrum of time series characteristics captured, potentially leading to improved predictive outcomes.

It is essential to acknowledge certain limitations of this study. While offering a controlled environment for assessing the model's predictive power, the focus on univariate time series restricts its ability to account for interactions among multiple assets-a critical aspect in real-world financial markets. Extending future analyses to multivariate settings could reveal inter-asset relationships, potentially unveiling new patterns and interactions that enhance the predictive power of TDA-augmented models. Furthermore, this study's relatively short time horizon may limit the generalizability of findings across different market cycles and phases. Expanding the dataset to cover longer periods would allow for testing model robustness across diverse market conditions, potentially offering a more comprehensive view of financial market dynamics and further stabilizing TDA-based forecasting models.

6 Conclusions

This manuscript introduces the innovative integration of topological data analysis (TDA) into financial time series forecasting, specifically aimed at enhancing the predictive accuracy of the N-BEATS model through the utilization of TDA-derived features, such as entropy, amplitude, and the number of points from persistence diagrams and point clouds, computed via a sliding window approach. The study rigorously compares traditional feature engineering techniques with TDA-derived methods across time series data from six cryptocurrencies and four traditional financial instruments, implementing pre-processing techniques to handle missing data, outlier detection, and stationarity tests. Results obtained from performance evaluations using metrics such as MAPE, MAE, and RMSE indicate that the $\mathtt{N-BEATS}_{+\mathrm{TDA}}$ model outperformed traditional techniques, with statistical tests affirming the significant improvements in predictive accuracy attributable to the incorporation of TDA features, thereby advancing the application of TDA in financial analysis and enhancing decision-making processes.

Author contributions Luiz Carlos was responsible for the conceptualization andimplementation of the theoretical model. Francisco and Mariano supervised thework and wrote the draft of the manuscript being submitted.

Funding Funding for open access publishing: Universidad de Málaga/CBUA.

Data availability The manuscript has no associated data.

Materials availability Not applicable

Code availability The code is written in Python and will be available upon request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest with any member of thejournal.

Ethical approval and consent to participate Not applicable

References

1. Carlsson G (2009) Topology and data. Bull Am Math Soc 46(2):255–308
2. Ghrist R (2008) Barcodes: the persistent topology of data. Bull Am Math Soc 45(1):61–75
3. Edelsbrunner H, Letscher D, Zomorodian A (2002) Topological persistence and simplification. Discrete Computat Geom 28(4):511–533
4. Zomorodian AJ (2005) Topology for Computing, vol 16. Cambridge University Press, Cambridge
5. Wasserman L (2018) Topological data analysis. Annual review of statistics and its application 5(1):501–532
6. Theobald D (2001) Topology revisited: representing spatial relations. Int J Geogr Inf Sci 15:689–705
7. Mantegna RN, Stanley HE (1999) Introduction to econophysics: correlations and complexity in finance. Cambridge University Press, Cambridge
8. Tsay RS (2005) Analysis of financial time series. Wiley, Hoboken
9. Yu P, Yan X (2020) Stock price prediction based on deep neural networks. Neural Comput Appl 32(6):1609–1628
10. Chung H, Shin K-s (2020) Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. Neural Comput Appl 32(12):7897–7914
11. Khedr AM, Arif I, El-Bannany M, Alhashmi SM, Sreedharan M (2021) Cryptocurrency price prediction using traditional statistical and machine-learning techniques: a survey. Intell Syst Account Finance Manag 28(1):3–4
12. Gidea M (2017) Topological data analysis of critical transitions in financial networks. In: 3rd International Winter School and Conference on Network Science: NetSci-X 2017 3, pp. 47–59. Springer
13. Gidea M, Katz Y (2018) Topological data analysis of financial time series: landscapes of crashes. Phys A 491:820–834
14. Leykam D, Angelakis DG (2023) Topological data analysis and machine learning. Adv Phys X 8(1):2202331
15. Gidea M, Goldsmith D, Katz Y, Roldan P, Shmalo Y (2020) Topological recognition of critical transitions in time series of cryptocurrencies. Phys A 548:123843
16. Ruiz-Ortiz MA, Gómez-Larrañaga JC, Rodríguez-Viorato J (2022) A persistent-homology-based turbulence index & some applications of tda on financial markets. arXiv preprint arXiv:2203.05603
17. Baitinger E, Flegel S (2021) The better turbulence index? Forecasting adverse financial markets regimes with persistent homology. Fin Markets Portfolio Mgmt 35(3):277–308

18. Ali D, Asaad A, Jimenez M-J, Nanda V, Paluzo-Hidalgo E, Soriano-Trigueros M (2023) A survey of vectorization methods in topological data analysis. IEEE Trans Pattern Anal Mach Intell 45(12):14069–14080
19. Zhong M, Lin M, He Z (2023) Dynamic multi-scale topological representation for enhancing network intrusion detection. Comput Secur 135:103516
20. Samani EU, Banerjee AG (2024) Persistent homology meets object unity: object recognition in clutter. IEEE Trans Robot 40:886–902. https://doi.org/10.1109/TRO.2023.3343994
21. Ren Y, Liu F, Xia S, Shi S, Chen L, Wang Z (2023) Dynamic ecg signal quality evaluation based on persistent homology and googlenet method. Front Neurosci 17:1153386
22. Arfi B (2024) The promises of persistent homology, machine learning, and deep neural networks in topological data analysis of democracy survival. Qual Quant 58(2):1685–1727
23. Perea JA, Harer J (2015) Sliding windows and persistence: an application of topological methods to signal analysis. Found Comput Math 15:799–838
24. Khasawneh FA, Munch E, Perea JA (2018) Chatter classification in turning using machine learning and topological data analysis. IFAC-PapersOnLine 51(14):195–200
25. Zeng S, Graf F, Hofer C, Kwitt R (2021) Topological attention for time series forecasting. Adv Neural Inf Process Syst 34:24871–24882
26. Horváth L, Kokoszka P, Rice G (2014) Testing stationarity of functional time series. J Econom 179:66–82
27. Gatheral J, Jaisson T, Rosenbaum M (2014) Volatility is rough. Quant Finance 18:933–949
28. Gao Z-K, Small M, Kurths J (2017) Complex network analysis of time series. Europhys Lett 116(5):50001
29. Babyak M (2004) What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. Psychosom Med 66:411–421
30. Sehra S, Flores D, Montañez GD (2021) Undecidability of underfitting in learning algorithms. In: 2021 2nd International Conference on Computing and Data Science (CDS), pp. 591–594 (2021). IEEE
31. Aue A, Horváth L (2013) Structural breaks in time series. J Time Ser Anal 34(1):1–16
32. Li L, Ou Y, Wu Y, Li Q, Chen D (2018) Research on feature engineering for time series data mining. In: 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), pp. 431–435 . IEEE
33. Takens F (1981) Detecting strange attractors in turbulence, dynamical systems and turbulence. Lect Notes Math 898:366–381
34. Tran QH, Hasegawa Y (2019) Topological time-series analysis with delay-variant embedding. Phys Rev E 99(3):032209
35. Carlsson E, Carlsson JG, Sweitzer S (2022) Applying topological data analysis to local search problems. Found Data Sci 4(4):563–579
36. Chazal F, Glisse M, Labruère C, Michel B (2014) Convergence rates for persistence diagram estimation in topological data analysis. In: International Conference on Machine Learning, pp. 163–171. PMLR
37. Rucco M, Castiglione F, Merelli E, Pettini M (2016) Characterisation of the idiotypic immune network through persistent entropy. In: Battiston, S., De Pellegrini, F., Caldarelli, G., Merelli, E. (eds.) Proceedings of ECCS 2014, pp. 117–128. Springer
38. Chung MK (2012) Topological data analysis. Computational Neuroanatomy, pp 35–366
39. Oreshkin BN, Carpov D, Chapados N, Bengio Y (2020) N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 [cs.LG]
40. Putz D, Gumhalter M, Auer H (2021) A novel approach to multi-horizon wind power forecasting based on deep neural architecture. Renew Energy 178:494–505
41. Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA (2017) A roadmap for the computation of persistent homology. EPJ Data Sci 6:1–38
42. Somasundaram EV, Brown SE, Litzler A, Scott JG, Wadhwa RR (2021) Benchmarking R packages for calculation of persistent homology. R J 13(1):184
43. Mudassir M, Bennbaia S, Unal D, Hammoudeh M (2020) Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05129-6
44. Passalis N, Avramelou L, Seficha S, Tsantekidis A, Doropoulos S, Makris G, Tefas A (2022) Multisource financial sentiment analysis for detecting bitcoin price change indications using deep learning. Neural Comput Appl 34(22):19441–19452
45. Anisiuba CA, Egbo OP, Alio FC, Ifediora C, Igwemeka EC, Odidi C, Ezeaku HC (2021) Analysis of cryptocurrency dynamics in the emerging market economies: does reinforcement or substitution effect prevail? SAGE Open 11(1):21582440211002516
46. Chambino M, Dias R, Horta N (2023) Asymmetric efficiency of cryptocurrencies during the 2020 and 2022 events. Econom Anal Lett 2(2):23–33
47. Gamper J, Dignös A (2020) Processing temporal and time series data: present state and future challenges. Symposium on Advances in Databases and Information Systems, p 8–14
48. Durrleman S, Simon R (1989) Flexible regression models with cubic splines. Stat Med 8(5):551–561
49. Guo Z (2023) Research on the augmented dickey-fuller test for predicting stock prices and returns. Adv Econom Manag Polit Sci 44:101–106
50. Chernick M (2001) Wavelet methods for time series analysis. Technometrics 43:491–491
51. Borkin D, Nemeth M, Michalconok G, Mezentseva O (2019) Adding additional features to improve time series prediction. Research Papers Faculty of Materials Science and Technology Slovak University of Technology, 27:72–78
52. Lo Duca A, Marchetti A (2022) Towards the evaluation of date time features in a ship route prediction model. J Mar Sci Eng 10(8):1130
53. Adya M, Collopy F, Armstrong JS, Kennedy M (2001) Automatic identification of time series features for rule-based forecasting. Int J Forecast 17(2):143–157
54. West M (1997) Time series decomposition. Biometrika 84:489–494
55. Kim S, Kim H (2016) A new metric of absolute percentage error for intermittent demand forecasts. Int J Forecast 32(3):669–679
56. Fernández-Navarro F, Cruz MA, Gutiérrez PA, Castaño A, Hervás-Martínez C (2018) Time series forecasting by recurrent product unit neural networks. Neural Comput Appl 29:779–791
57. Willmott C (1982) Some comments on the evaluation of model performance. Bull Am Meteor Soc 63:1309–1313
58. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
59. Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Stat 11(1):86–92