Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 1, 113–127
DOI: 10.61822/amcs-2026-0009

TIME SERIES FORECAST INTERVALS USING CIRCULAR BOOTSTRAPPED TRAINING SIMULATION WITH INVARIANT DISTANCE KNN

PATCHANOK SRISURADETCHAI [a,*], PARATTAKORN KAMLANGDEE [a]

[a] Faculty of Science and Technology
Thammasat University
Khlong Luang, Pathum Thani, 12120, Thailand
e-mail: patchanok@mathstat.sci.tu.ac.th,
parattakorn.kaml@dome.tu.ac.th

This paper presents a nonparametric interval forecasting method that combines circular block bootstrap resampling with complexity-invariant K-nearest-neighbor time-series prediction. Prediction intervals are obtained directly from bootstrap-resampled training series, thereby preserving temporal dependence while accounting for forecast uncertainty. Under weak dependence and local stability assumptions, asymptotic validity of the resulting prediction intervals is established. The proposed method is evaluated using twelve time-series datasets drawn from economic, environmental, industrial, and energy applications. Empirical performance is compared with seasonal autoregressive integrated moving average models and long short-term memory neural networks using the mean absolute percentage error, empirical coverage probability, and interval score. The results show that the proposed approach yields prediction intervals of moderate width with competitive forecasting accuracy across most datasets, while empirical coverage remains close to the nominal level. Mild undercoverage is observed in short samples, attributable to limited data availability and fixed tuning parameters.

Keywords: K-nearest neighbors, circular block bootstrapping, uncertainty quantification, rolling-window forecasting.

1. Introduction

Time-series forecasting is fundamental in finance, production planning, and environmental science, where accurate predictions support informed decision-making. Although machine learning models such as long short-term memory (LSTM) networks and ensemble methods have emerged, the seasonal autoregressive integrated moving average (SARIMA) model remains a prevailing benchmark due to its simplicity, interpretability, and established role in forecasting research (Tadesse and Dinka, 2017; Bagnall *et al*., 2017; Kien *et al*., 2023; Milenkovic *et al*., 2018; Chodakowska *et al*., 2021). However, the linearity of SARIMA constrains its ability to capture nonlinear or evolving temporal structures, motivating the use of nonparametric and learning-based approaches such as K-nearest neighbors (KNN), which can adapt to complex, data-driven dynamics.

Despite its conceptual simplicity, KNN has demonstrated strong predictive capability across diverse time-series contexts. Martínez *et al.* (2019b) enhanced KNN through optimized parameter tuning and preprocessing, while Tajmouati *et al*. (2024) proposed CPTO-WNN and FPTO-WNN variants that outperform ARIMA across heterogeneous datasets. Hybrid extensions further increased flexibility: Wei *et al*. (2013) embedded KNN within a neuro-fuzzy framework to enable adaptive rule learning, while Arroyo and Maté (2009) generalized the method to histogram time series, forecasting value distributions rather than single observations. Dimensionality reduction and theoretical consistency were examined by Tang *et al*. (2018) and Yakowitz (1987), while Rady *et al*. (2020) demonstrated that KNN retains robust performance relative to support vector machines, even under varying data conditions.

Beyond nearest-neighbor methods, learning-based forecasting has been widely explored for nonlinear and complex temporal dynamics. Neuro-fuzzy systems and neural networks have been applied to financial and engineering time series characterized by nonstationarity,

*Corresponding author

chaos, or strong nonlinear interactions (Li and Chiang, 2012; Bingi and Prusty, 2021). Recent studies further show that modern deep learning architectures, including multi-task learning and temporal convolutional networks, can improve predictive accuracy in applied forecasting problems such as healthcare demand and remaining useful life estimation by exploiting shared temporal structure and extended representations (Zhou *et al*., 2023; Zhao *et al*., 2024). These approaches highlight the effectiveness of data-driven models for capturing intricate temporal dependencies, while also underscoring the need for reliable uncertainty quantification.

Deep neural networks (DNNs) have likewise advanced forecasting by capturing nonlinear dependencies and long-range interactions. Kummaraka and Srisuradetchai (2024) introduced a dual-output Monte Carlo dropout (MCDO) network that simultaneously estimates predictive means and variances, yielding reliable forecast intervals. In a focused study on sinusoidal time-series dynamics, Kummaraka and Srisuradetchai (2025) demonstrated that MCDO-based neural networks provide more accurate point forecasts and better-calibrated prediction intervals than SARIMA across a range of periodic structures, including varying amplitudes and phase shifts. Srisuradetchai and Phaphan (2024) further applied an MCDO-based DNN to agricultural export forecasting, achieving tighter and more consistent prediction intervals than traditional approaches. Complementary works—that of Güven *et al*. (2021) on local nearest-neighbor demand forecasting, of Lee (2019) on Python implementations, and Narejo *et al*. (2021) on rainfall prediction using deep belief networks—further illustrate the growing synergy between machine learning and time-series analysis. In financial applications, Thomann *et al*. (2021) combined LSTM architectures with sentiment indicators to capture behavioral signals in stock market forecasting.

Recent KNN research emphasizes scalability and robustness under diverse data conditions. The random-kernel KNN (RK-KNN) approach of Srisuradetchai and Suksrikran (2024) employs kernel smoothing and bootstrap aggregation, improving accuracy and R^2 relative to standard KNN while mitigating overfitting. Seasonality and pattern variability have been addressed through adaptive and clustering-based frameworks (Martínez *et al*., 2018; Martínez-Álvarez *et al*., 2011). In environmental contexts, Srisuradetchai and Panichkitkosolkul (2022) showed that hybrid KNN ensembles effectively model nonlinear and highly volatile $PM_{2.5}$ concentrations, underscoring the method's practical reliability.

Building on these developments, this study proposes a circular block bootstrap K-nearest neighbors time-series prediction with invariances (CBB–KNN–TSPI) framework that integrates circular block bootstrap resampling with the existing complexity-invariant KNN predictor to construct statistically grounded forecast intervals. The method embeds resampling directly into the learning process rather than applying it as a post-hoc residual adjustment, thereby enabling simultaneous modeling of sampling and model uncertainty while preserving short-range temporal dependence. This design yields sharper and statistically coherent prediction intervals without relying on restrictive distributional assumptions. Theoretical analysis establishes asymptotic coverage validity under weak dependence and local stability conditions, while empirical evaluations across multiple datasets confirm the adaptability and robustness of the CBB–KNN–TSPI framework in both stationary and weakly nonstationary environments.

2. K-nearest neighbors for time-series forecasting with invariances

This section presents the theoretical foundation of K-nearest neighbors time-series prediction with invariances (KNN–TSPI) (Parmezan and Batista, 2015; Parmezan *et al*., 2022), an extension of the classical KNN regression method that integrates complexity-aware similarity measures and transformation invariances to improve time-series forecasting. Unlike conventional KNN approaches, KNN–TSPI applies z-normalization to ensure scale-invariant distance computation, incorporates complexity-aware similarity adjustments, and enforces mechanisms to prevent trivial matches, thereby enhancing the robustness of similarity-based predictions (Lima and Souza, 2023).

2.1. Time-series representation and lagged formulation. Let $\chi = \{x_t \mid t = 1, 2, \ldots, T\}$ be a univariate time series, where $x_t \in \mathbb{R}$ denotes the observed value at time t and T represents the series length. KNN–TSPI constructs lagged feature vectors to capture temporal dependencies. The lagged representation of a time series at time t is expressed as

$$\mathbf{v}_t = [x_{t-l_1}, x_{t-l_2}, \ldots, x_{t-l_m}]^T \in \mathbb{R}^m, \tag{1}$$

where $t > \max\{l_1, l_2, \ldots, l_m\}$, $l_1, l_2, \ldots, l_m$ are the selected lag indices and m is the embedding dimension. When consecutive lags are used, the representation simplifies to

$$\mathbf{v}_t = [x_{t-1}, x_{t-2}, \ldots, x_{t-m}]^T. \tag{2}$$

Each lagged vector $\mathbf{v}_t$ serves as input to the KNN–TSPI model, while the corresponding target variable for a forecasting horizon h is x_{t+h}. This formulation ensures that each training instance captures structured temporal information to enhance future predictions (Martínez *et al.*, 2019a).

2.2. Training and test data. The training dataset is defined as

$$\mathcal{D}_{\text{train}} = \{(\mathbf{v}_t, x_{t+h}) \mid t \in I_{\text{train}}\}, \tag{3}$$

where $\mathbf{v}_t$ is the input vector, x_{t+h} is the target value h steps ahead, and $I_{\text{train}} \subseteq \{\max(l_1, l_2, \ldots, l_m), \ldots, T - h\}$ denotes the index set of training data.

The test dataset is defined as

$$\mathcal{D}_{\text{test}} = \{(\mathbf{v}_t, x_{t+h}) \mid t \in I_{\text{test}}\}, \tag{4}$$

where $I_{\text{train}} \cap I_{\text{test}} = \emptyset$, ensuring no overlap between training and testing data.

2.3. Normalization and invariance transformation in KNN–TSPI. To achieve scale and offset invariance, z-normalization is applied to each subsequence before similarity computation. The normalized version of a sequence x_t is given by

$$z'_t = \frac{x_t - \mu}{\sigma}, \tag{5}$$

where μ and σ denote the mean and standard deviation of the subsequence, respectively. This transformation guarantees that sequences with identical shapes but differing amplitudes or offsets remain comparable under the selected distance metric.

2.4. Complexity-invariant distance for similarity measurement. The complexity-invariant distance (CID) enhances the standard Euclidean distance (ED) by incorporating a complexity correction factor, ensuring that sequences with differing structural complexities are compared appropriately. The CID measure is defined as (Parmezan and Batista, 2015)

$$CID(\mathbf{v}_t, \mathbf{v}_j) = ED(\mathbf{v}_t, \mathbf{v}_j) \times CF(\mathbf{v}_t, \mathbf{v}_j), \tag{6}$$

where $ED(\mathbf{v}_t, \mathbf{v}_j)$ is the Euclidean distance,

$$ED(\mathbf{v}_t, \mathbf{v}_j) = \sqrt{\sum_{l=1}^{m} (v_{t,l} - v_{j,l})^2},$$

and $CF(\mathbf{v}_t, \mathbf{v}_j)$ is the complexity correction factor,

$$CF(\mathbf{v}_t, \mathbf{v}_j) = \frac{\max\big(CE(\mathbf{v}_t), CE(\mathbf{v}_j)\big)}{\min\big(CE(\mathbf{v}_t), CE(\mathbf{v}_j)\big)}.$$

A common complexity estimate (CE) for a vector $\mathbf{v}$ is

$$CE(\mathbf{v}) = \sqrt{\sum_{l=2}^{m} (v_l - v_{l-1})^2}.$$

This modification ensures that sequences with similar shapes but differing levels of variation (e.g., smooth versus highly fluctuating patterns) are treated equitably during similarity comparisons.

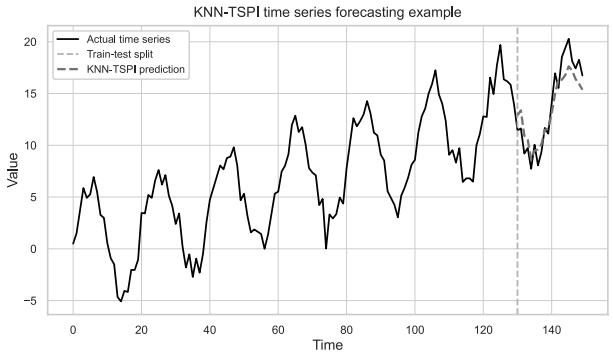

Fig. 1. KNN–TSPI time-series forecasting. The observed series is shown as a solid line. The training and test segments are indicated by different background shading, and the KNN–TSPI forecasts are shown as a dashed line.

2.5. Nearest-neighbor search with complexity-aware similarity. For a given query vector $\mathbf{v}_t$ from the test dataset, KNN–TSPI selects the K-nearest neighbors from the training set based on the CID measure:

$$\mathcal{N}_t = \arg \min_{S \subseteq \mathcal{D}_{\text{train}}, |S|=K} \sum_{\mathbf{v}_j \in S} d\big(g(\mathbf{v}_t), g(\mathbf{v}_j)\big), \tag{7}$$

where $g(\mathbf{v})$ denotes the composition of z-normalization and CID transformation applied before similarity computation.

To prevent trivial matches, KNN–TSPI enforces a minimum separation constraint such that selected neighbors are not within a window of size m around the query. This ensures that predictions emphasize structural and dynamic similarity rather than local autocorrelation.

2.6. Point prediction using KNN–TSPI. After identifying the K-nearest neighbors $\mathcal{N}_t$, the point forecast for time $t+h$ is computed as the mean of the future values associated with those neighbors:

$$\hat{x}_{t+h} = \frac{1}{K} \sum_{j \in \mathcal{N}_t} x_{t_j+h}. \tag{8}$$

This approach yields an interpretable and computationally efficient forecast, maintaining consistency with the nonparametric similarity-based paradigm.

Figure 1 illustrates an example of time-series prediction using KNN regression on a synthetic dataset generated as $x_t = 0.1t + 5\sin(\pi t/10) + \varepsilon_t$, where t ranges from 0 to 150 and ε_t represents white noise. The dataset combines trend, seasonality, and stochastic variation, making it an appropriate benchmark for evaluating forecasting models. The KNN–TSPI model learns from the training region and produces re-trended predictions, effectively capturing temporal dependencies and structural patterns in the data.

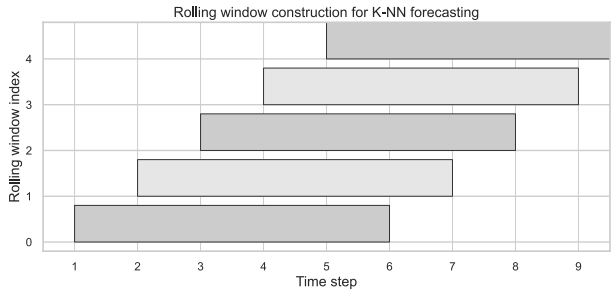

Fig. 2. Rolling-window construction for KNN-based time-series forecasting.

3. Proposed method

This section details the proposed CBB–KNN–TSPI framework, which extends the complexity-invariant KNN predictor by embedding CBB resampling directly into the training process. The resampling step generates dependent replicas of the training data that preserve short-range correlation, providing a nonparametric means to approximate the sampling distribution of forecasts. The resulting ensemble of bootstrapped predictors yields empirical forecast distributions from which prediction intervals are obtained. The theoretical properties of these intervals, including asymptotic coverage validity under weak dependence, are formally established in Appendix.

3.1. Rolling-window framework for training data generation. The forecasting procedure employs a rolling-window mechanism, in which a univariate time series $\chi = \{x_t \mid t = 1, 2, \ldots, T\}$ is systematically transformed into structured input-output pairs for supervised learning.

1. *Constructing lagged vectors for training.* To capture temporal dependencies, each observation is embedded into a lagged feature vector defined as

$$\mathbf{v}_t = [x_{t-l_1}, x_{t-l_2}, \ldots, x_{t-l_m}]^\top \in \mathbb{R}^m, \quad (9)$$

where m denotes the embedding dimension and $l_1, l_2, \ldots, l_m$ are selected lag indices. When consecutive lags are employed, this formulation reduces to (2). Each lagged vector $\mathbf{v}_t$ serves as the input to the KNN–TSPI predictor, while the corresponding target x_{t+h} represents the future observation at forecast horizon h.

The rolling-window mechanism thereby generates overlapping input-output pairs across the series, as illustrated in Fig. 2. This construction ensures that each training instance encodes recent temporal patterns while maintaining consistent feature dimensionality.

2. *Computing residuals for bootstrapping.* After fitting KNN–TSPI once on the original training set (constructed from the lagged input vectors defined above), residuals are computed as

$$e_t = x_{t+h} - \widehat{f}_T(\mathbf{v}_t), \quad (10)$$

where $\widehat{f}_T(\mathbf{v}_t)$ denotes the fitted KNN–TSPI prediction. These residuals capture model errors from the fitted predictor and serve as the foundation for circular block bootstrap resampling, in which contiguous blocks are drawn from the residual sequence to preserve short-range temporal dependence.

3.2. Generating bootstrapped training data via circular block bootstrapping. To introduce variability while preserving autocorrelation, residuals are resampled using the CBB procedure.

1. *Complexity-invariant residual resampling.* Rather than resampling residuals independently—which would disrupt temporal dependence—contiguous blocks of residuals are drawn as

$$B_i = (e_i, e_{i+1}, \ldots, e_{i+b-1}), \quad (11)$$

where b denotes the block length. Theoretically, the block length is of order $b \asymp T^{1/3}$ (Lahiri, 2003; Politis and White, 2004). In the empirical analysis, $b = 12$ is adopted for monthly data to encompass one full annual cycle within each block, ensuring stable interval calibration while satisfying the asymptotic condition $b/T \to 0$. Each bootstrapped residual sequence e_t^* is generated via the CBB procedure, allowing the sequence to wrap around the series boundary when necessary:

$$e_t^* \sim \mathrm{CBB}(\{e_t\}, b), \quad (12)$$

with each block sampled randomly and uniformly with replacement:

$$P(B_i) = \frac{1}{T}, \quad i = 1, 2, \ldots, T. \quad (13)$$

Since sampling is performed with replacement, identical blocks may be selected multiple times, introducing stochastic variation across bootstrap replicates while preserving short-range dependence.

2. *Constructing bootstrapped observations.* The resampled residuals are added to the KNN–TSPI fitted values:

$$x_{t+h}^{(r)} = \widehat{f}_T(\mathbf{v}_t) + e_t^{*(r)}, \quad (14)$$

where r denotes the bootstrap iteration. Each resulting sequence $\{x_{t+h}^{(r)}\}$ represents a new realization of the time series, incorporating both model and sampling uncertainty.

3. *Constructing bootstrapped training sets.* Each resampled series preserves the rolling-window structure, allowing the construction of corresponding lagged vectors:

$$\mathbf{v}_t^{(r)} = [x_{t-1}^{(r)}, x_{t-2}^{(r)}, \ldots, x_{t-m}^{(r)}]^T \in \mathbb{R}^m. \quad (15)$$

The bootstrapped training dataset is therefore expressed as

$$\mathcal{D}_T^{(r)} = \{(\mathbf{v}_t^{(r)}, x_{t+h}^{(r)}) \mid t \in I_{\text{train}}\}, \quad (16)$$

and the complete ensemble of B bootstrapped datasets is

$$\mathcal{D}_T^{(B)} = \{\mathcal{D}_T^{(1)}, \mathcal{D}_T^{(2)}, \ldots, \mathcal{D}_T^{(B)}\}. \quad (17)$$

These resampled datasets provide diverse yet structurally consistent representations for model re-estimation, thereby reflecting the intrinsic uncertainty of the underlying data-generating process.

3.3. Forecasting with bootstrapped training data. For each bootstrapped dataset $\mathcal{D}_T^{(r)}$, KNN–TSPI is applied independently to obtain a corresponding forecast realization.

1. *Nearest-neighbor search with complexity-aware similarity.* The K-nearest neighbors for each query vector $\mathbf{v}_t^{(r)}$ are identified using the CID-based distance:

$$d(\mathbf{v}_t^{(r)}, \mathbf{v}_j^{(r)}) = \text{CID}\big(g(\mathbf{v}_t^{(r)}), g(\mathbf{v}_j^{(r)})\big), \quad (18)$$

where $g(\mathbf{v})$ represents the combined transformation of z-normalization and complexity adjustment.

2. *Forecasting step.* The K most similar neighbors are determined as

$$\mathcal{N}_t^{(r)} = \arg \min_{S \subseteq \mathcal{D}_T^{(r)}, |S|=K} \sum_{\mathbf{v}_j^{(r)} \in S} d\big(g(\mathbf{v}_t^{(r)}), g(\mathbf{v}_j^{(r)})\big), \quad (19)$$

and the bootstrap-specific forecast is then computed as

$$\hat{x}_{t+h}^{(r)} = \frac{1}{K} \sum_{j \in \mathcal{N}_t^{(r)}} x_{t_j+h}^{(r)}, \quad (20)$$

where t_j denotes the time index associated with the j-th nearest neighbor.

3.4. Constructing forecast intervals from bootstrapped predictions. After generating multiple bootstrapped forecasts, prediction intervals are constructed from their empirical distribution, enabling nonparametric uncertainty quantification.

1. *Aggregating forecast distributions.* The collection of bootstrap forecasts at time $t+h$ is defined as

$$\mathcal{X}_{t+h}^{(B)} = \{\hat{x}_{t+h}^{(1)}, \hat{x}_{t+h}^{(2)}, \ldots, \hat{x}_{t+h}^{(B)}\}, \quad (21)$$

and the aggregated point estimate is the mean across all bootstrap realizations:

$$\hat{x}_{t+h}^{(B)} = \frac{1}{B} \sum_{r=1}^{B} \hat{x}_{t+h}^{(r)}. \quad (22)$$

2. *Estimating prediction intervals.* The bootstrap forecast ensemble $\{\hat{x}_{t+h}^{(r)}\}_{r=1}^B$ provides an empirical approximation to the conditional predictive distribution $F_{t+h}(z \mid \mathbf{v}_t) = \Pr(x_{t+h} \le z \mid \mathbf{v}_t)$. Accordingly, the empirical quantile operator $Q_\alpha(\hat{x}_{t+h}^{(r)})$ serves as a nonparametric estimator of the theoretical quantile $F_{t+h}^{-1}(\alpha \mid \mathbf{v}_t)$. For a desired prediction level $(1-\alpha)$ (e.g., 95%), the lower and upper bounds of the prediction interval are obtained from the empirical quantiles of the bootstrap distribution (Errouissi *et al.*, 2015):

$$\begin{aligned} L_{t+h} &= Q_{\alpha/2}(\hat{x}_{t+h}^{(r)}), \\ U_{t+h} &= Q_{1-\alpha/2}(\hat{x}_{t+h}^{(r)}), \end{aligned} \quad (23)$$

where Q_α denotes the empirical quantile operator,

$$Q_\alpha = \inf \left\{ x \in \mathbb{R} : B^{-1} \sum_{r=1}^{B} \mathbf{1}\big(\hat{x}_{t+h}^{(r)} \le x\big) \ge \alpha \right\}. \quad (24)$$

Hence, the prediction interval at time $t+h$ is given by $[L_{t+h}, U_{t+h}]$, where L_{t+h} and U_{t+h} correspond to the 2.5th and 97.5th empirical percentiles, respectively, for a nominal 95% prediction level.

The large-sample validity of these intervals follows from the asymptotic properties of the bootstrap predictive distribution under weak dependence and local stability.

4. Evaluation criteria

To comprehensively evaluate both the deterministic accuracy and the probabilistic reliability of the proposed forecasting methods, three complementary measures are employed: the mean absolute percentage error, coverage probability, and interval score.

1. *Mean absolute percentage error (MAPE).* The MAPE quantifies the average relative deviation between the predicted and observed values, providing a normalized measure of point forecast accuracy:

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{x_t - \hat{x}_t}{x_t} \right|. \quad (25)$$

Lower values indicate more accurate forecasts.

2. *Coverage probability (CP).* The CP represents the empirical proportion of true observations that lie within their respective prediction intervals (Chatfield, 1993; Kummaraka and Srisuradetchai, 2023). It reflects the calibration of interval forecasts relative to their nominal confidence levels:

$$\mathrm{CP} = \frac{100}{n}\sum_{t=1}^{n}\mathbf{1}(x_t \in [L_t, U_t]), \tag{26}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function, equal to one if the realized value x_t lies within the interval $[L_t, U_t]$, and zero otherwise.

3. *Interval score (IS).* To jointly assess both interval *sharpness* (narrowness) and *calibration* (coverage accuracy), the proper scoring rule known as the interval score is adopted (Gneiting and Raftery, 2007; Jordan *et al*., 2019). For a nominal coverage level of $(1-\alpha)$, the score is defined as

$$\begin{aligned}\mathrm{IS}_\alpha = (U_t - L_t) &+ \frac{2}{\alpha}(L_t - x_t)\,\mathbf{1}(x_t < L_t)\\ &+ \frac{2}{\alpha}(x_t - U_t)\,\mathbf{1}(x_t > U_t),\end{aligned} \tag{27}$$

where smaller values indicate prediction intervals that are both sharp and well-calibrated.

5. Datasets

To evaluate the proposed method, 12 heterogeneous time-series datasets from economic, environmental, industrial, and energy domains were analyzed. They vary in frequency, length, and temporal complexity, covering energy, transportation, industrial production, and climate processes. This diversity enables a comprehensive assessment of the method's forecasting and interval performance under real-world conditions. A brief description of each dataset follows.

1. *UAH mid-troposphere temperature (UAHTEMP).* Provided by the University of Alabama in Huntsville (UAH), this dataset records global temperature anomalies in the mid-troposphere (approximately 8 km altitude) using satellite-based microwave measurements. It spans December 1978–present and reports monthly deviations from the 1991–2020 baseline (NOAA CDR Program, 2017).

2. *Producer price index for synthetic rubber manufacturing (PPISRM).* From the US Bureau of Labor Statistics (BLS), this monthly index (PCU325212325212P) tracks price changes in synthetic rubber manufacturing, with a base of 100 in June 1981 (US Bureau of Labor Statistics, 2025b).

3. *Renewable energy consumption (REWENGY).* Published by the US Energy Information Administration (EIA), this monthly series measures US renewable energy consumption (in quadrillion BTUs), representing 9% of total energy use in 2023 (US Energy Information Administration, 2025).

4. *Total vehicle sales (VEHSALE).* Compiled by the US Bureau of Economic Analysis (BEA) and available via the FRED database, this monthly series reflects US automotive demand (US Bureau of Economic Analysis, 2025).

5. *US unemployment rate (UNEMPRATE).* A BLS dataset hosted on FRED, reporting the monthly US unemployment rate (not seasonally adjusted) as an indicator of labor-market conditions (US Bureau of Labor Statistics, 2025c).

6. *Domestic air conditioner sales (DOMAIR).* Monthly sales of domestic air conditioners in Thailand, 2000–2023, measured in units sold. It captures strong seasonal demand linked to climate and economic factors (Office of Industrial Economics, 2025).

7. *Durian export volume (DUREXPV).* From the Thai Ministry of Commerce, this monthly series records fresh durian exports. In early 2024, exports to China exceeded 225,000 tons, illustrating agricultural seasonality and trade variation (Office of Agricultural Economics, 2025).

8. *Slaughter chicken price index (SCPI).* A BLS producer price index tracking monthly poultry prices, reflecting food-sector inflation and supply-chain volatility (US Bureau of Labor Statistics, 2025a).

9. *Australian wine sales (AUSWINE).* A standard benchmark dataset of monthly wine sales in Australia, January 1980–August 1994 (176 observations), exhibiting strong seasonality and trend (Hyndman and Yang, 2018; Hyndman *et al*., 2025).

10. *Beer production (BEERPROD).* Part of the CBE dataset (chocolate, beer, and electricity), this monthly record of Australian beer output, 1958–1990 (396 observations), is widely used to study cyclical industrial behavior (Cowpertwait and Metcalf, 2009).

11. *Airline passenger traffic (AIRPAS).* The classic "AirPassengers" dataset of monthly international airline counts (in thousands), January 1949–December 1960 (144 observations),

with multiplicative trend and seasonality (Box *et al.*, 2015).

12. *Texas coal power consumption (COALTX).* Quarterly coal consumption (in short tons) for electric power generation in Texas, 2001–2020, illustrating regional fossil-fuel dependence and energy transition trends (US Energy Information Administration, 2023).

6. Results

This section presents the comparative performance of the CBB–KNN–TSPI framework against the SARIMA benchmark and LSTM models across 12 heterogeneous datasets covering climatic, industrial, energy, and economic domains. The forecasting horizon was fixed at one year, i.e., 12 monthly or four quarterly observations, reflecting typical short-term decision horizons in applied forecasting (Arroyo-Marioli *et al.*, 2023).

6.1. Visual comparison. Figures (3)–(5) show, for each dataset, the simulated CBB training series (left) and the corresponding 95% prediction intervals on the test set (right). The circular block resampling preserves local dependence and short-term seasonality, producing training realizations that reflect realistic temporal variability.

Across most datasets, SARIMA yields the widest intervals, CBB–KNN–TSPI produces intervals of moderate width, and LSTM gives the narrowest bands. This width comparison, however, is only descriptive. The actual predictive reliability is better reflected by the interval score discussed below. The observed differences are consistent with model construction: SARIMA relies on analytical variance formulas that are typically conservative, while the Monte Carlo dropout (MCDO) LSTM may understate uncertainty due to limited weight sampling. By contrast, the proposed CBB–KNN–TSPI balances both tendencies through block-resampled residuals that preserve serial dependence while calibrating the forecast distribution empirically. Although not always the narrowest, its intervals align closely with the observed series and maintain coverage near the nominal 95% level.

6.2. Quantitative evaluation. Table 1 reports the MAPE, CP, and IS, assessing forecast accuracy, reliability, and sharpness. A lower MAPE and IS indicate higher accuracy and narrower intervals, while a CP close to the nominal level reflects adequate coverage.

For clarity, results are summarized across three temporal structures: (1) cyclical and irregular, (2) highly volatile and irregular, and (3) seasonal or trend-dominated series, illustrating how CBB–KNN–TSPI adapts to different dependence patterns with balanced precision and coverage.

1. *Cyclical and irregular datasets.* For global temperature anomalies (UAHTEMP), the synthetic rubber index (PPISRM), and beer production (BEERPROD), CBB–KNN–TSPI achieves competitive point accuracy and consistently sharper, better-calibrated intervals than LSTM, with coverage close to nominal. Specifically, UAHTEMP attains the lowest IS among all methods (0.53) with the CP = 91.67%, the PPISRM improves the IS to 61.90 (vs. 73.63 for SARIMA and 158.49 for LSTM) with the CP = 75%, and BEERPROD yields an IS = 56.40, i.e., much lower than LSTM's 131.53 despite its slightly higher point accuracy (MAPE = 5.53), with the CP = 91.67%. These results suggest that circular block resampling preserves cyclical dependence while preventing over-dispersed intervals, leading to sharper yet reliable uncertainty estimates.

2. *Highly volatile and irregular datasets.* For the durian export volume (DUREXPV), slaughter-chicken price index (SCPI), and Texas coal power consumption (COALTX), all exhibiting spikes or heavy-tailed behavior, CBB–KNN–TSPI produces consistently sharper, well-calibrated intervals. In the SCPI, the MAPE drops to 3.08% (vs. 10.79% for SARIMA), and the IS decreases from 189.41 (SARIMA) to 123.18 (CP = 100%). For DUREXPV, the IS improves from 142,611.34 (SARIMA) to 138,769.00 (CP = 100%), and is dramatically lower than LSTM's 1,300,036.29. For COALTX, the IS reduces from 1.67×10^7 (SARIMA) to 1.29×10^7 (CP = 75% across methods). While occasional extremes fall outside the 95% band, overall coverage remains adequate (75–100%).

3. *Seasonal and trend-dominated series.* For renewable energy consumption (REWENGY), airline passenger traffic (AIRPAS), Australian wine sales (AUSWINE), total vehicle sales (VEHSALE), the US unemployment rate (UNEMPRATE), and domestic air conditioner sales (DOMAIR), CBB–KNN–TSPI maintains a balanced accuracy–calibration trade-off under pronounced seasonality and long-term trends. REWENGY attains the lowest MAPE (1.99%) and the best IS (87.06) with the CP = 100%. AUSWINE and UNEMPRATE show notable interval gains (IS = 6678.82 and 1.56, respectively) with the CP between 91.67% and 100%. VEHSALE achieves compact intervals (IS = 366.49) with the CP = 100%, despite SARIMA's slightly lower MAPE (2.95%). For AIRPAS and DOMAIR, performance

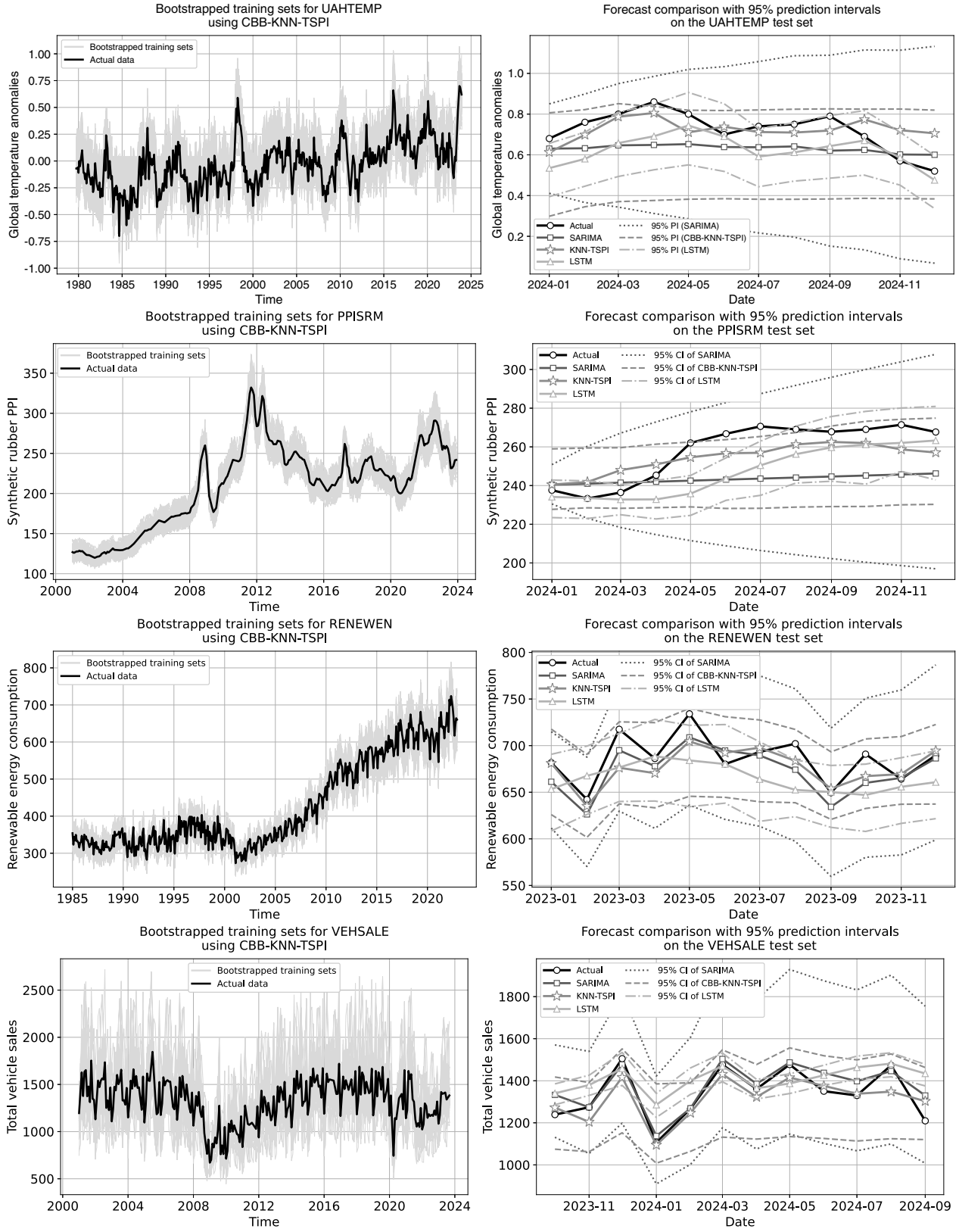

Fig. 3. Simulated bootstrap training sets (left) and forecast comparisons (right) using KNN–TSPI with the circular bootstrap (Part 1). For grayscale reproduction, 95% prediction intervals are distinguished by line style: dotted (SARIMA), solid (CBB–KNN–TSPI), and dash-dot (LSTM).

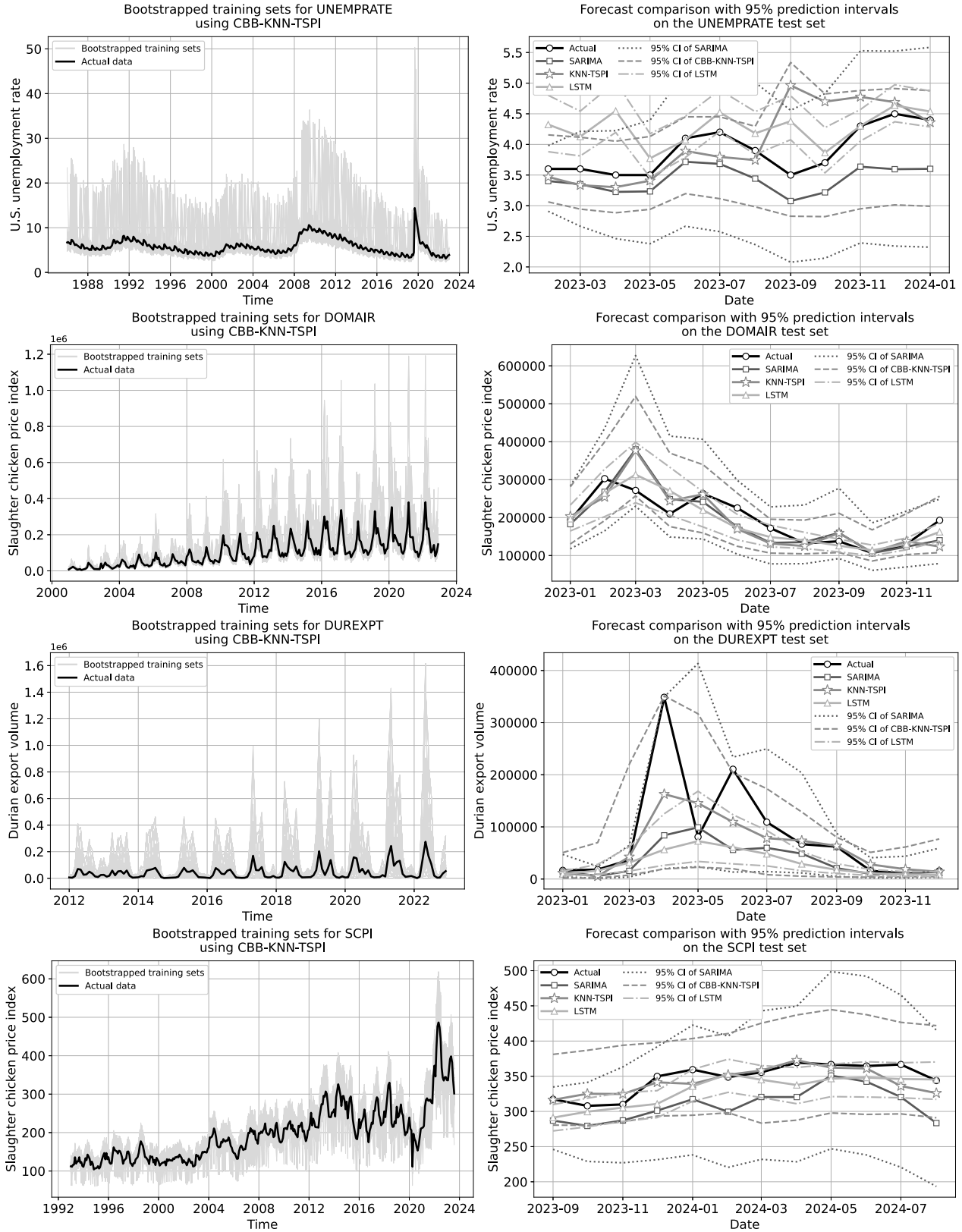

Fig. 4. Simulated bootstrap training sets (left) and forecast comparisons (right) using KNN–TSPI with the circular bootstrap (Part 2). For grayscale reproduction, 95% prediction intervals are distinguished by line style: dotted (SARIMA), solid (CBB–KNN–TSPI), and dash-dot (LSTM).

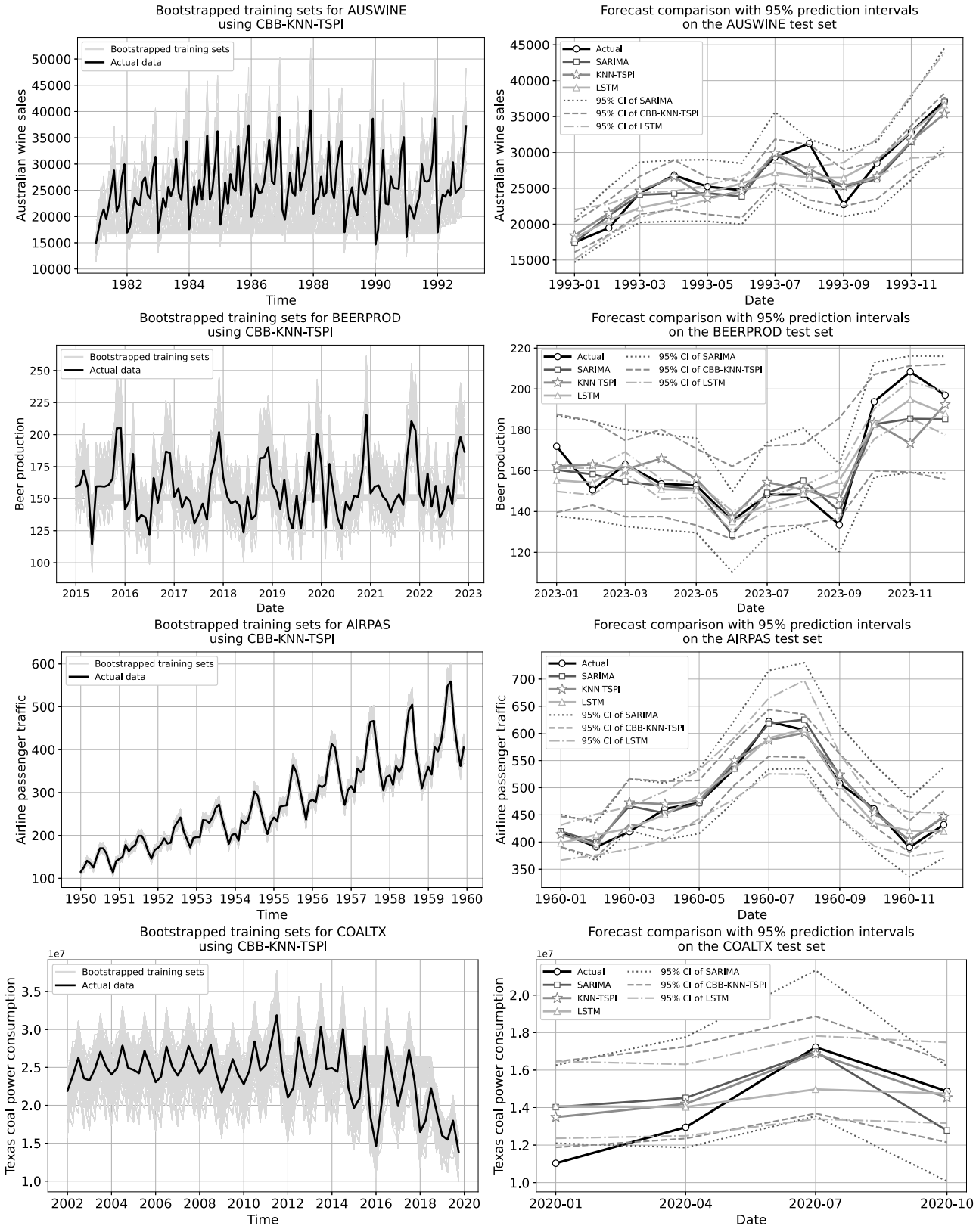

Fig. 5. Simulated bootstrap training sets (left) and forecast comparisons (right) using KNN–TSPI with the circular bootstrap (Part 3). For grayscale reproduction, 95% prediction intervals are distinguished by line style: dotted (SARIMA), solid (CBB–KNN–TSPI), and dash-dot (LSTM)

Table 1. Forecast performance across 12 datasets using KNN–TSPI for point forecasts and CBB–KNN–TSPI for interval forecasts. Bold values denote the best result within each metric. For the MAPE and IS, lower values indicate better performance, while a higher CP indicates better coverage. CBB–KNN denotes CBB–KNN–TSPI (proposed method).

Dataset	MAPE (%)			CP (%)			IS		
	KNN–TSPI	SARIMA	LSTM	CBB–KNN	SARIMA	LSTM	CBB–KNN	SARIMA	LSTM
UAHTEMP	**11.26**	14.62	13.51	91.67	**100.00**	66.67	**0.53**	0.79	0.61
PPISRM	**3.32**	6.55	4.18	75.00	**100.00**	66.67	**61.90**	73.63	158.49
REWENGY	**1.99**	2.33	3.68	**100.00**	**100.00**	66.67	**87.06**	152.20	226.11
VEHSALE	3.78	**2.95**	4.79	**100.00**	**100.00**	50.00	**366.49**	658.35	916.81
UNEMPRATE	**10.25**	11.68	10.31	**100.00**	**100.00**	58.33	**1.56**	2.42	6.63
DOMAIR	15.82	14.87	**12.47**	**100.00**	**100.00**	83.33	**144,758.31**	207,575.40	157,129.58
DUREXPV	42.01	**41.56**	42.26	**100.00**	**100.00**	41.67	**138,769.00**	142,611.34	1,300,036.29
SCPI	**3.08**	10.79	4.92	**100.00**	**100.00**	75.00	**123.18**	189.41	161.50
AUSWINE	**5.41**	5.57	6.73	91.67	**100.00**	58.33	**6678.82**	9292.22	33138.66
BEERPROD	5.53	4.73	**4.38**	91.67	**100.00**	58.33	56.40	**48.70**	131.53
AIRPAS	3.15	**2.60**	3.28	91.67	91.67	**100.00**	116.12	140.79	**97.84**
COALTX	**9.04**	13.67	12.46	**75.00**	**75.00**	**75.00**	$\mathbf{1.29\times10^7}$	1.67×10^7	1.75×10^7

remains competitive (IS = 116.12 and 144,758.31, respectively) with stable coverage (91.67–100%), indicating robust behavior without over-dispersion.

Overall, CBB–KNN–TSPI achieves a consistent trade-off between predictive accuracy and interval reliability across all temporal regimes. The observed empirical coverage aligns with the theoretical guarantees in Appendix, confirming that circular block resampling effectively preserves dependence while yielding statistically coherent, data-driven intervals.

7. Conclusion and discussion

Time-series forecasting remains essential in finance, production planning, and environmental monitoring, where dependable projections inform critical decisions. Classical models such as SARIMA provide transparency and interpretability but are constrained by their linear structure, which limits their capacity to capture nonlinear or evolving temporal dynamics. Conversely, deep learning models such as LSTM offer flexible function approximation and perform well on smooth, trend-dominated data but often require extensive tuning and large samples to ensure stable uncertainty quantification. These challenges motivate the development of nonparametric alternatives such as the KNN family, which balances model interpretability with the ability to represent complex, data-driven temporal patterns.

This study introduced the CBB–KNN–TSPI framework, integrating circular block bootstrap resampling with a complexity-invariant KNN predictor to construct data-driven prediction intervals. Resampling at the data level preserves temporal dependence while introducing controlled stochastic variation, allowing empirical estimation of both model and sampling uncertainty. Theoretical analysis in Appendix establishes asymptotic coverage under mild α-mixing and local-stability conditions, providing a rigorous foundation for the procedure.

Experiments on 12 heterogeneous datasets show that CBB–KNN–TSPI produces sharp, well-calibrated intervals and competitive point accuracy. It yields lower interval scores and mean absolute percentage errors for nonlinear and irregular series (e.g., PPISRM, UAHTEMP) and achieves better calibration than SARIMA and LSTM for volatile data such as the SCPI and DUREXPV. Minor coverage deviations in short samples (e.g., COALTX) arise from limited evaluation windows rather than systematic bias.

Overall, CBB–KNN–TSPI offers a statistically consistent and interpretable approach to probabilistic forecasting. By combining circular resampling with complexity-aware similarity, it achieves a stable trade-off between predictive precision and interval reliability without relying on restrictive distributional assumptions. The framework is especially effective for weakly nonstationary or moderately nonlinear processes. Future work will extend the theory to stronger dependence regimes and develop multivariate or online variants for large-scale applications.

Acknowledgment

The authors thank the Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, for providing computational resources and administrative support. Special appreciation goes to the colleagues and reviewers whose valuable suggestions significantly improved this manuscript.

References

Arroyo, J. and Maté, C. (2009). Forecasting histogram time series with K-nearest neighbours methods, *International Journal of Forecasting* **25**(1): 192–207.

Arroyo-Marioli, F., Khadan, J., Ohnsorge, F. and Yamazaki, T. (2023). Forecasting industrial commodity prices: Literature review and a model suite, *Policy Research Working Paper 10611*, World Bank, Washington DC.

Bagnall, A., Lines, J., Bostrom, A., Large, J. and Keogh, E. (2017). The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances, *Data Mining and Knowledge Discovery* **31**(3): 606–660.

Bingi, K. and Prusty, B.R. (2021). Forecasting models for chaotic fractional-order oscillators using neural networks, *International Journal of Applied Mathematics and Computer Science* **31**(3): 387–398, DOI: 10.34768/amcs-2021-0026.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*, 5th Edn, John Wiley & Sons, Hoboken.

Bühlmann, P. (1999). Bootstraps for time series, *Research Report 87*, ETH Zürich, Zürich.

Chatfield, C. (1993). Calculating interval forecasts, *Journal of Business & Economic Statistics* **11**(2): 121–135.

Chodakowska, E., Nazarko, J. and Nazarko, Ł. (2021). ARIMA models in electrical load forecasting and their robustness to noise, *Energies* **14**(23): 7952.

Cowpertwait, P.S.P. and Metcalf, A.V. (2009). *Introductory Time Series with R*, Springer, New York.

Errouissi, R., Cardenas-Barrera, J., Meng, J., Castillo-Guerra, E., Gong, X. and Chang, L. (2015). Bootstrap prediction interval estimation for wind speed forecasting, *2015 IEEE Energy Conversion Congress and Exposition (ECCE), Montreal, Canada*, pp. 1919–1924.

Gneiting, T. and Raftery, A.E. (2007). Strictly proper scoring rules, prediction, and estimation, *Journal of the American Statistical Association* **102**(477): 359–378.

Güven, İ., Uygun, Ö. and Şimşir, F. (2021). Machine learning algorithms with intermittent demand forecasting: An application in retail apparel with plenty of predictors, *Textile and Apparel* **31**(2): 99–110.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E. and Yasmeen, F. (2025). *forecast: Forecasting Functions for Time Series and Linear Models*, R package, Version 8.24.0, `https://cran.r-project.org/package=forecast`.

Hyndman, R.J. and Yang, Y. (2018). *tsdl: Time Series Data Library,* R package, Version 0.1.0, `https://github.com/FinYang/tsdl`.

Jordan, A., Krüger, F. and Lerch, S. (2019). Evaluating probabilistic forecasts with `scoringRules`, *Journal of Statistical Software* **90**(12): 1–37.

Kien, D., Huong, P. and Minh, N. (2023). Application of SARIMA model in load forecasting in Hanoi city, *International Journal of Energy Economics and Policy* **13**(3): 164–170.

Kummaraka, U. and Srisuradetchai, P. (2023). Interval estimation of the dependence parameter in bivariate clayton copulas, *Emerging Science Journal* **7**(5): 1478–1490.

Kummaraka, U. and Srisuradetchai, P. (2024). Time-series interval forecasting with dual-output Monte Carlo dropout: A case study on durian exports, *Forecasting* **6**(3): 616–636.

Kummaraka, U. and Srisuradetchai, P. (2025). Monte Carlo dropout neural networks for forecasting sinusoidal time series: Performance evaluation and uncertainty quantification, *Applied Sciences* **15**(8): 4363.

Lahiri, S.N. (2003). *Resampling Methods for Dependent Data*, Springer-Verlag, New York.

Lee, W. (2019). *Python Machine Learning*, 1st Edn, Wiley, Indianapolis.

Li, C. and Chiang, T.-W. (2012). Intelligent financial time series forecasting: A complex neuro-fuzzy approach with multi-swarm intelligence, *International Journal of Applied Mathematics and Computer Science* **22**(4): 787–800, DOI: 10.2478/v10006-012-0058-x.

Lima, F. and Souza, V.M.A. (2023). A large comparison of normalization methods on time series, *Big Data Research* **34**: 100407, DOI: 10.1016/j.bdr.2023.100407.

Martínez-Álvarez, F., Troncoso, A., Riquelme, J. and Aguilar-Ruiz, J. (2011). Energy time series forecasting based on pattern sequence similarity, *IEEE Transactions on Knowledge and Data Engineering* **23**(8): 1230–1243.

Martínez, F., Frías, M.P., Charte, F. and Rivera, A.J. (2019a). Time series forecasting with KNN in R: The tsfknn package, *The R Journal* **11**(2): 229–242.

Martínez, F., Frías, M.P., Pérez, M. and Rivas, A.R. (2019b). A methodology for applying k-nearest neighbor to time series forecasting, *Artificial Intelligence Review* **52**: 2019–2037, DOI: 10.1007/s10462-017-9593-z.

Martínez, F., Frías, M.P., Pérez-Godoy, M. and Rivas, A.R. (2018). Dealing with seasonality by narrowing the training set in time series forecasting with kNN, *Expert Systems with Applications* **103**: 38–48, DOI: 10.1016/j.eswa.2018.03.005.

Milenkovic, M., Svadlenka, L., Melichar, V., Bojovic, N. and Avramovic, Z. (2018). SARIMA modelling approach for railway passenger flow forecasting, *Transport* **33**(5): 1113–1120.

Narejo, S., Jawaid, M., Talpur, S., Baloch, R. and Pasero, E. (2021). Multi-step rainfall forecasting using deep learning approach, *PeerJ Computer Science* **7**: e514.

NOAA CDR Program (2017). NOAA climate data record (CDR) of MSU and AMSU-A mean layer temperatures, UAH version 6.0, NOAA National Centers for Environmental Information, Asheville, DOI: 10.7289/V5MC8X31.

Office of Agricultural Economics (2025). Trade report: Thailand's international trade statistics, Ministry of Agriculture and Cooperatives of Thailand, Bangkok, https://www.oae.go.th/home/article/386.

Office of Industrial Economics (2025). Industrial statistics report: Production and distribution of cooling machines, Ministry of Industry of Thailand, Bangkok, https://i.index.oie.go.th/industrialStatistics1.aspx.

Parmezan, A. and Batista, G. (2015). A study of the use of complexity measures in the similarity search process adopted by kNN algorithm for time series prediction, *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, USA*, pp. 45–51.

Parmezan, A., de Souza, V.A. and Batista, G. (2022). Time series prediction via similarity search: Exploring invariances, distance measures and ensemble functions, *IEEE Access* **10**: 78022–78043, DOI: 10.1109/ACCESS.2022.3192849.

Politis, D.N. and White, H. (2004). Automatic block-length selection for the dependent bootstrap, *Econometric Reviews* **23**(1): 53–70.

Rady, E.H.A., Fawzy, H. and Fattah, A.M.A. (2020). Comparison between support vector machines and K-nearest neighbor for time series forecasting, *Journal of Mathematical and Computational Science* **10**(6): 2342–2359.

Srisuradetchai, P. and Panichkitkosolkul, W. (2022). Using ensemble machine learning methods to forecast particulate matter ($PM_{2.5}$) in Bangkok, Thailand, *Multi-Disciplinary Trends in Artificial Intelligence (MIWAI 2022) Proceedings, Chiang Mai, Thailand*, pp. 204–215.

Srisuradetchai, P. and Phaphan, W. (2024). Using Monte-Carlo dropout in deep neural networks for interval forecasting of durian export, *WSEAS Transactions on Systems and Control* **19**: 10–21, DOI: 10.37394/23203.2024.19.2.

Srisuradetchai, P. and Suksrikran, K. (2024). Random kernel K-nearest neighbors regression, *Frontiers in Big Data* **7**: 1402384, DOI: 10.3389/fdata.2024.1402384.

Stone, C.J. (1977). Consistent nonparametric regression, *The Annals of Statistics* **5**(4): 595–620.

Tadesse, K. and Dinka, M. (2017). Application of SARIMA model to forecasting monthly flows in waterval river, South Africa, *Journal of Water and Land Development* **35**: 229–236, DOI: 10.1515/jwld-2017-0088.

Tajmouati, S., Wahbi, B.E.L., Bedoui, A., Abarda, A. and Dakkon, M. (2024). Applying k-nearest neighbors to time series forecasting: Two new approaches, *Journal of Forecasting* **43**(5): 1559–1574, DOI: 10.1002/for.3093.

Tang, L., Pan, H. and Yao, Y. (2018). K-nearest neighbor regression with principal component analysis for financial time series prediction, *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, China*, pp. 127–131.

Thomann, M., Farchmin, J., Weisser, C., Kruse, R., Säfken, B. and Silbersdorff, A. (2021). Stock price predictions with LSTM neural networks and twitter sentiment, *Statistics, Optimization & Information Computing* **9**(2): 268–287, DOI: 10.19139/soic-2310-5070-1202.

US Bureau of Economic Analysis (2025). Total vehicle sales [TOTALSA] (millions of units, seasonally adjusted annual rate), FRED, Federal Reserve Bank of St. Louis, St. Louis, https://fred.stlouisfed.org/series/TOTALSA.

US Bureau of Labor Statistics (2025a). Producer price index by commodity: Farm products: Slaughter chickens [WPU0141] (index 1982 = 100, not seasonally adjusted), FRED, Federal Reserve Bank of St. Louis, St. Louis, https://fred.stlouisfed.org/series/WPU0141.

US Bureau of Labor Statistics (2025b). Producer price index by industry: Synthetic rubber manufacturing: Primary products [PCU325212325212P] (index Jun 1981 = 100, not seasonally adjusted), FRED, Federal Reserve Bank of St. Louis, St. Louis, https://fred.stlouisfed.org/series/PCU325212325212P.

US Bureau of Labor Statistics (2025c). Unemployment rate [UNRATENSA] (percent, not seasonally adjusted), FRED, Federal Reserve Bank of St. Louis, St. Louis, https://fred.stlouisfed.org/series/UNRATENSA.

US Energy Information Administration (2023). Table CT8: Electric power sector consumption estimates, 1960–2022, US Department of Energy, Washington DC, https://www.eia.gov/electricity/data/browser/.

US Energy Information Administration (2025). Monthly energy review, US Department of Energy, Washington DC, https://www.eia.gov/totalenergy/data/monthly/.

Wei, C., Chen, T. and Lee, S. (2013). k-NN based neuro-fuzzy system for time series prediction, *Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Honolulu, USA*, pp. 569–574, DOI: 10.1109/SNPD.2013.68.

Yakowitz, S. (1987). Nearest-neighbour methods for time series analysis, *Journal of Time Series Analysis* **8**(2): 235–247.

Zhao, J., Liu, D. and Meng, L. (2024). Remaining useful life prediction of a lithium-ion battery based on a temporal convolutional network with data extension, *International Journal of Applied Mathematics and Computer Science* **34**(1): 105–117, DOI: 10.61822/amcs-2024-0008.

Zhou, M., Huang, X., Liu, H. and Zheng, D. (2023). Hospitalization patient forecasting based on multi-task deep learning, *International Journal of Applied Mathematics and Computer Science* **33**(1): 151–162, DOI: 10.34768/amcs-2023-0012.

Patchanok Srisuradetchai holds a BSc in statistics from Thammasat University (2006) and an MSc in applied statistics from NIDA (2008). He also holds an MSc and a PhD in statistics from Montana State University, Bozeman, USA (2012, 2015). He is an associate professor in the Department of Mathematics and Statistics, Thammasat University. His interests include computational statistics, simulation, machine learning, and statistical computing in R.

Parattakorn Kamlangdee holds a BSc in statistics (high distinction) from Thammasat University (2024) and is pursuing an MSc in applied statistics at the same institution. He works in life-insurance operations and services, focusing on dashboards and automated monitoring. His interests include machine learning, time-series forecasting, and uncertainty quantification.

Appendix

Asymptotic validity of CBB–KNN–TSPI forecast intervals

Notation. Let $\{x_t\}_{t=1}^{T}$ be a strictly stationary time series and define the lagged feature vector

$$\mathbf{v}_t = [x_{t-1}, x_{t-2}, \ldots, x_{t-m}]^\top.$$

For a fixed forecast horizon $h \geq 1$, let the training set be

$$\mathcal{D}_T = \{(\mathbf{v}_t, x_{t+h}) : t \in I_{\text{train}}\},$$

and let $\widehat{f}_T$ denote the fitted KNN–TSPI predictor trained on $\mathcal{D}_T$. Define the h-step-ahead residuals by

$$e_t = x_{t+h} - \widehat{f}_T(\mathbf{v}_t), \qquad t \in I_{\text{train}}.$$

To preserve serial dependence, the circular block bootstrap (CBB) with block length $b = b(T) \to \infty$ and $b/T \to 0$ is applied to the residual sequence $\{e_t\}$, producing bootstrap residual replicates,

$$e_t^{*(r)} \sim \text{CBB}(\{e_t\}, b),$$

for $r = 1, \ldots, B$. The corresponding bootstrap pseudo-responses are constructed as

$$x_{t+h}^{(r)} = \widehat{f}_T(\mathbf{v}_t) + e_t^{*(r)}.$$

These pseudo-responses define a bootstrap pseudo-series, from which the lagged feature vectors are reconstructed as

$$\mathbf{v}_t^{(r)} = [x_{t-1}^{(r)}, x_{t-2}^{(r)}, \ldots, x_{t-m}^{(r)}]^\top.$$

Each bootstrap training set is therefore given by

$$\mathcal{D}_T^{(r)} = \{(\mathbf{v}_t^{(r)}, x_{t+h}^{(r)}) : t \in I_{\text{train}}\},$$

and the refitted bootstrap predictor $\widehat{f}_T^{(r)}$ obtained from $\mathcal{D}_T^{(r)}$ yields a bootstrap forecast $\widehat{x}_{t+h}^{(r)}$ at horizon h.

Asymptotic regime. Let $T \to \infty$, $K = K(T) \to \infty$ with $K/|I_{\text{train}}| \to 0$, and $b = b(T) \to \infty$ with $b/T \to 0$. The number of bootstrap replications satisfies $B = B(T) \to \infty$. All expectations and probabilities below are taken with respect to the true data-generating process of $\{x_t\}$.

Assumption A1. *(Data dependence)* The process $\{x_t\}$ is strictly stationary and α-mixing with

$$\sum_{\ell=1}^{\infty} \alpha(\ell)^{\delta/(2+\delta)} < \infty$$

for some $\delta > 0$, and possesses finite $(2+\delta)$ moments. Consequently, the lagged process $\{\mathbf{v}_t\}$ is also α-mixing.

Assumption A2. *(Local stability of the learner)* Let $\widehat{f}_T : \mathcal{P} \to \mathcal{F}$ denote the learning map from empirical distributions on $\mathbb{R}^m \times \mathbb{R}$ to measurable predictors. Assume that $\widehat{f}_T$ is Hadamard-directionally differentiable and locally Lipschitz under the bounded-Lipschitz metric $d_{\text{BL}}(\cdot, \cdot)$, in the sense that

$$\|\widehat{f}_T(P_1) - \widehat{f}_T(P_2)\|_\infty \leq L_T \, d_{\text{BL}}(P_1, P_2), \qquad L_T = O_p(1).$$

This condition holds for K-NN regression under $K \to \infty$ and $K/n \to 0$ (see Stone, 1977), and is assumed here for the CID-based KNN–TSPI learner.

Assumption A3. *(Validity of the CBB for residuals)* The residual sequence $\{e_t\}$ is weakly dependent with

$$\sum_{\ell=-\infty}^{\infty} |\text{Cov}(e_t, e_{t+\ell})| < \infty.$$

Under $b(T) \to \infty$ and $b(T)/T \to 0$, the circular block bootstrap consistently reproduces the joint law of the paired process $\{(\mathbf{v}_t, e_t)\}$ in the bounded-Lipschitz metric d_{BL} (Lahiri, 2003).

Assumption A4. *(Stochastic equicontinuity)* The predictor class $\{\widehat{f}_T(\mathbf{v}) : \mathbf{v} \in \mathbb{R}^m\}$ is stochastically equicontinuous and uniformly bounded in probability, so that the empirical process indexed by $\mathbf{v}$ satisfies a functional central limit theorem (Bühlmann, 1999).

Assumption A5. *(Quantile regularity)* The conditional predictive distribution function $F_{t+h}(z \mid \mathbf{v}_t)$ is continuous at its $\alpha/2$ and $1 - \alpha/2$ quantiles.

Theorem A1. (Asymptotic coverage of the CBB–KNN–TSPI interval) *Assume* A1–A5 *hold. Let the equal-tailed prediction interval* $[L_{t+h}, U_{t+h}]$ *be constructed from the empirical quantiles of the bootstrap forecasts* $\{\widehat{x}_{t+h}^{(r)}\}_{r=1}^{B}$ *as*

$$L_{t+h} = Q_{\alpha/2}\big(\widehat{x}_{t+h}^{(r)}\big), \qquad U_{t+h} = Q_{1-\alpha/2}\big(\widehat{x}_{t+h}^{(r)}\big),$$

where $Q_\alpha(\cdot)$ denotes the empirical quantile operator. Then

$$\Pr\{x_{t+h} \in [L_{t+h}, U_{t+h}]\} \longrightarrow 1-\alpha \qquad \text{as } T \to \infty.$$

Proof. (Sketch of proof)
(i) By Assumption A3, the circular block bootstrap consistently reproduces the joint law of $\{(\mathbf{v}_t, e_t)\}$ in the bounded-Lipschitz metric d_{BL}.

(ii) By the local Lipschitz stability in Assumption A2, the perturbation $\mathcal{D}_T \mapsto \mathcal{D}_T^{(r)}$ induces an $\mathcal{O}_p\big(d_{\mathrm{BL}}(\mathcal{D}_T, \mathcal{D}_T^{(r)})\big)$ deviation of the fitted learner $\widehat{f}_T$. Combined with stochastic equicontinuity (Assumption A4), this implies that the bootstrap predictive distribution function $\widehat{F}_{t+h}^{(r)}(z)$ converges in probability, uniformly on compact sets, to the true conditional distribution $F_{t+h}(z \mid \mathbf{v}_t)$.

(iii) By the continuous mapping theorem and the regularity condition in Assumption A5, the empirical quantiles at levels $\alpha/2$ and $1-\alpha/2$ converge in probability to their population counterparts. Consequently,

$$\Pr\{x_{t+h} \in [L_{t+h}, U_{t+h}]\} \to 1-\alpha,$$

which establishes the desired asymptotic coverage. ■

Remark. Assumptions A1–A5 parallel standard conditions in nonparametric time-series bootstrap theory (Bühlmann, 1999; Lahiri, 2003), but are adapted here to the learning-based predictor $\widehat{f}_T$ underlying the KNN–TSPI framework. The key requirement is the local stability of the learner, which ensures that bootstrap perturbations of the training distribution induce asymptotically equivalent perturbations of the fitted predictor. This yields first-order validity of the empirical quantiles used to construct the prediction intervals.

Received: 12 June 2025
Revised: 4 December 2025
Accepted: 15 December 2025