E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia Science & Technology Asia
Vol. 30 No.1 January - March 2025 Page: [79-94]
Original research article

Circular Bootstrap on Residuals for Interval Forecasting in K-NN Regression: A Case Study on Durian Exports

Parattakorn Kamlangdee, Patchanok Srisuradetchai*

Faculty of Science and Technology, Thammasat University, Department of Mathematics and Statistics, Pathum Thani 12120, Thailand

Received 2 August 2024; Received in revised form 19 February 2025
Accepted 24 February 2025; Available online 24 March 2025

ABSTRACT

Traditional k-nearest neighbor (K-NN) methods for time series forecasting often fail to capture prediction uncertainty. This study addresses this limitation by integrating circular bootstrap on residuals with K-NN regression to forecast Thailand's monthly durian export volumes, which exhibit strong non-linearity and seasonality. The proposed methodology includes data scaling and explicit seasonality handling. Circular bootstrap generates multiple residual samples, constructing forecast intervals that quantify uncertainty while preserving temporal dependencies. Comparative analysis demonstrates that the proposed method outperforms the seasonal autoregressive integrated moving average (SARIMA) and exponential smoothing state space (ETS) models by producing forecast intervals that are, on average, 20% narrower. However, coverage is slightly lower, with actual values falling within the intervals in 11 out of 12 months, compared to full coverage by SARIMA and ETS. The results highlight the potential of combining statistical resampling with machine learning to enhance K-NN forecasting, offering a practical solution for improving time series forecast reliability, as demonstrated in the case study of Thailand's durian exports.

Keywords: Circular bootstrap; ETS; Interval forecasting; Probabilistic forecasting; SARIMA

1. Introduction

Time-series forecasting is essential across diverse domains, such as finance, production planning, and environmental monitoring. Traditional methods such as autoregressive integrated moving average (ARIMA) and Holt-Winters models have been widely used but often struggle with nonlinear and complex data patterns. Recently, machine learning techniques, particularly k-nearest neighbors (K-NN) algorithms, have gained prominence for their flexibility and effectiveness in diverse forecasting tasks.

***Corresponding author:** patchanok@mathstat.sci.tu.ac.th **doi:** 10.14456/scitechasia.2025.6

The K-NN algorithm, despite its simplicity, has proven to be a robust method for time-series forecasting. Tajmouati et al. [1] introduced two methodologies to enhance K-NN forecasting performance: classical parameter tuning in weighted nearest neighbors (CPTO-WNN) and fast parameter tuning in weighted nearest neighbors (FPTO-WNN). These methods optimize the selection of neighbors and parameters through a cross-validation-inspired approach, demonstrating superior performance compared to classical methods like ARIMA and Holt-Winters in various real-world datasets. Martínez et al. [2] developed a comprehensive methodology for applying K-NN regression to time-series forecasting, focusing on automatic parameter selection and preprocessing techniques to improve efficiency and accuracy.

Expanding on these methods, Zhang et al. [3] proposed a novel methodology combining ensemble empirical mode decomposition (EEMD) with a multidimensional K-NN model (MKNN) to forecast financial time series. The EEMD technique addresses the mode mixing problem inherent in EMD, enhancing forecast accuracy. This two-stage approach successfully forecasts multiple stock indices, outperforming traditional EMD-KNN and ARIMA models. Tang *et al*. [4] further explored financial time series prediction by integrating principal component analysis (PCA) with K-NN, which reduces data dimensionality and redundancy, thereby improving prediction performance.

In the realm of visual recognition, Liu and Liu [5] introduced a locally linear K-NN (LLK) method, which integrates sparse representation and optimization techniques to enhance robustness and accuracy across various visual datasets. Ni and Nguyen [6] proposed an adaptive K-NN algorithm for image interpolation, leveraging global optimization via Markov random fields to improve image quality based on sufficient training data.

Addressing seasonality in time-series forecasting, Martínez et al. [7] proposed narrowing the training set to specific seasons. This strategy involves training specialized K-NN learners for each season, reducing the likelihood of misleading forecasts and enhancing prediction accuracy. Similarly, Martínez-Álvarez et al. [8] used pattern sequence similarity for energy time-series forecasting, where clustering techniques group samples to improve forecast accuracy.

In the context of environmental monitoring, Srisuradetchai and Panichkitkosolkul [9] explored ensemble machine learning methods, including K-NN, to forecast particulate matter (PM2.5) concentrations in Bangkok. Their study demonstrated that hybrid models combining multiple algorithms significantly enhance forecasting accuracy, underscoring the value of ensemble approaches. Troncoso et al. [10] applied weighted nearest neighbors to forecast electricity market prices, highlighting the relevance of parameter tuning for improved accuracy.

Further expanding the application of K-NN, Srisuradetchai [11] introduced a novel interval forecasting approach integrating K-NN with bootstrapping to capture forecast uncertainty. The bootstrap samples were taken from observed forecasts with multiple k values. Fernández-Rodríguez et al. [12] applied nearest-neighbor predictors to foreign exchange markets, showing potential economic benefits. The tsfknn package for R, described by Martínez et al. [13], allows users to specify K-NN models and generate multi-step forecasts, showcasing practical applications in various time-series datasets. Srisuradetchai and Phaphan [14] applied Monte Carlo dropout (MCDO) in deep neural networks for interval forecasting of agricultural exports, demonstrating superior performance compared to traditional

models. Kück and Freitag [15] evaluated the forecasting performance of local nearest neighbor models for demand forecasting in production planning, achieving high forecast accuracy with low computation times. Lee [16] discussed the implementation of K-NN in Python, emphasizing its application in classification and the determination of optimal parameters. Narejo et al. [17] investigated multi-step rainfall forecasting using deep learning approaches, including temporal deep belief networks, highlighting the advancements in leveraging deep learning for time-series forecasting.

Forecast intervals are essential in time-series forecasting because they provide a range within which future values are expected to fall, thereby quantifying the uncertainty inherent in predictions. Unlike point forecasts, which offer a single value, forecast intervals give a probabilistic estimate, enhancing decision-making processes by accounting for potential variability [18]. This approach aligns with modern probabilistic machine learning, which focuses on uncertainty quantification and risk assessment. By integrating techniques like Monte Carlo dropout and bootstrapping with K-NN [19], we can produce more robust and informative forecasts, crucial for applications requiring high reliability. Here, we use the circular bootstrap with time-series K-NN regression to produce the forecast interval.

To further validate the efficacy of our proposed interval forecasting method, we will compare its performance with interval forecasts generated by traditional SARIMA and Holt-Winters models. This comparison aims to demonstrate the superior capability of our approach in handling nonlinear and complex data patterns, thereby providing more reliable and comprehensive forecasts.

2. Circular Bootstrap

The circular bootstrap on residuals is a robust resampling technique specifically designed for time-series data, preserving the inherent serial correlation essential for accurate forecasting. Traditional bootstrap methods may disrupt temporal dependencies, but the circular bootstrap on residuals treats the residual series as a circular structure, maintaining these dependencies effectively [20, 21]. The following steps outline the circular bootstrap process:

1) Fit a Model and Calculate Residuals: Given a time series $T = \{y_1, y_2, \ldots, y_n\}$ of length n, fit a predictive model and calculate the residuals as $e_i = y_i - \hat{y}_i$, where $\hat{y}_i$ represents the predicted value.
2) Concatenate Residual Series: Extend the residual series R by concatenating it to itself, forming $E^* = \{e_1, e_2, \ldots, e_n, e_1, e_2, \ldots, e_n\}$. This step ensures continuity and preserves the serial correlation inherent in the residuals.
3) Random Block Selection: Define a block length l. Randomly select starting indices i_j from $\{1,2,..., n\}$ This randomness helps capture variability while preserving serial dependencies [22].
4) Form Blocks: Construct blocks B_j of length l starting from i_j in E^*, such that $B_j = \{e_{i_j}, e_{i_j+1}, \ldots, e_{i_j+l-1}\}$. This step ensures each block retains the temporal structure of the series [23, 24].
5) Reconstruct Series: Concatenate these blocks to form a bootstrap sample of residuals $E_b = \{B_1, B_2, ..., B_m\}$, where R_b is of length n. This reconstructed series mimics the structure of the original residuals, preserving the inherent dependencies.
6) Generate Bootstrap Samples: Repeat the process to generate multiple bootstrap samples of residuals $E_b^{(1)}, E_b^{(2)}, \ldots, E_b^{(M)}$. Add these samples to the predicted values $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ to generate

new bootstrap samples of the forecasted series: $V_b^{(j)} = \{\hat{y}_1 + e_1^{(j)}, \hat{y}_2 + e_2^{(j)}, \ldots, \hat{y}_n + e_n^{(j)}\}$, $j = 1, 2, ..., M$.

7) Repeat the Process: Continue generating multiple bootstrap series $V_b^{(1)}, V_b^{(2)}, ..., V_b^{(M)}$ for further analysis and modeling, ensuring robustness in statistical inference.

Mathematically, for a block length l and the number of bootstrap samples M, the bootstrap series $V_b^{(j)}$ for $j = 1, 2, \ldots, M$ is constructed as:

$$V_b^{(j)} = \{\hat{y}_i + e_{i_j}, \hat{y}_{i+1} + e_{i_j+1}, \ldots, \hat{y}_{i+n-1} + e_{i_j+n-1}\},$$

where i_j are the randomly selected starting indices. This approach ensures that each bootstrap sample T_b retains the temporal dependencies of the original series, which is crucial for accurate forecasting and inference [25, 26].

The circular bootstrap on residuals, with its ability to preserve serial correlations and handle non-stationary data, provides a powerful tool for generating robust forecast intervals in time-series analysis. By combining this technique with K-NN regression, we aim to enhance the accuracy and reliability of our time-series forecasts.

3. SARIMA and ETS Models

3.1 SARIMA model

The seasonal autoregressive integrated moving average (SARIMA) model is a widely used statistical technique for time-series forecasting, particularly for datasets exhibiting seasonal patterns. It extends the autoregressive integrated moving average (ARIMA) model by incorporating seasonal autoregressive (SAR) and seasonal moving average (SMA) components, along with seasonal differencing to address seasonality.

A SARIMA model is represented as SARIMA (p, d, q) (P, D, Q)$_s$, where p, d, and q are the non-seasonal orders, P, D, and Q are the seasonal orders, and s denotes the seasonal period. The SARIMA model is mathematically expressed as shown in Eq. (3.1).

$$\phi_P(B)\Phi_P(B^s)\nabla^d\nabla_s^D y_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t, \quad (3.1)$$

where B is the backshift operator, defined as $B^k y_t = y_{t-k}$. ∇^d is the differencing operator, defined as $\nabla^d y_t = (1-B)^d y_t$. ∇_s^D is the seasonal differencing operator, defined as $\nabla_s^D y_t = (1-B^s)^D y_t$. $\phi_p(B)$ is the autoregressive polynomial:

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p.$$

$\Phi_P(B^s)$ is the seasonal autoregressive polynomial. $\theta_q(B)$ is the moving average polynomial:

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q.$$

$\Theta_Q(B^s)$ is the moving average polynomial:

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_1 B^{Qs}.$$

ϵ_t are white noise error terms with zero mean and constant variance.

The SARIMA model can also be written as an infinite moving average (MA) process, as shown in Eq. (3.2).

$$y_t = \sum_{i=0}^{\infty} \psi_i \epsilon_t, \quad (3.2)$$

where the coefficients ψ_i are derived from the infinite series expansion of the rational function, as presented in Eq. (3.3).

$$\frac{\theta_q(B)\Theta_Q(B^s)}{\phi_p(B)\Phi_P(B^s)}. \quad (3.3)$$

The standard error of the forecast $\mathrm{SE}(\hat{y}_{t+h})$ for an SARIMA model at h-steps ahead is

calculated based on the accumulated forecast error variance. This is given in Eq. (3.4).

$$\mathrm{Var}(\hat{y}_{t+h}) = \sigma^2\left(1+\sum_{i=1}^{h-1}\psi_i^2\right), \qquad (3.4)$$

where σ^2 is the variance of the SARIMA residuals. ψ_i are the coefficients from the MA representation.

The standard error of the forecast is shown in Eq. (3.5).

$$\mathrm{SE}(\hat{y}_{t+h}) = \sqrt{\mathrm{Var}(\hat{y}_{t+h})} = \sigma\sqrt{1+\sum_{i=1}^{h-1}\psi_i^2}\ . \qquad (3.5)$$

Finally, the $(1-\alpha)\times 100\%$ forecast interval for h-steps ahead is presented in Eq. (3.6).

$$\hat{y}_{t+h} \pm z_{\alpha/2}\cdot\sigma\sqrt{1+\sum_{i=1}^{h-1}\psi_i^2}, \qquad (3.6)$$

where $z_{\alpha/2}$ is the critical value from the standard normal distribution corresponding to the chosen confidence level.

3.2 Exponential Smoothing State Space Model

The exponential smoothing state space model (ETS), specifically the additive Holt-Winters' method, is a widely used approach for forecasting time-series data with trend and seasonal components. This method incorporates error, trend, and seasonality additively. The model is formulated as shown in Eqs. (3.7)–(3.10):

$$y_t = l_{t-1} + b_{t-1} + s_{t-s} + \varepsilon_t \qquad (3.7)$$

$$l_t = \alpha(y_t - s_{t-s}) + (1-\alpha)(l_{t-1} + b_{t-1}) \qquad (3.8)$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \qquad (3.9)$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-s}, \qquad (3.10)$$

where y_t is the observed value at time t. l_t is the level component at time t. b_t is the trend component at time t. s_t is the seasonal component at time t. α,β,γ are smoothing parameters. s is the length of the seasonal cycle. ε_t is the error term at time t.

The forecast for h-steps ahead in the ETS(A,A,N) model is given in Eq. (3.11).

$$\hat{y}_{t+h} = l_t + hb_t + s_{t+h-s(k+1)}\ . \qquad (3.11)$$

The forecast interval for the h-step-ahead forecast in the ETS method is presented in Eq. (3.12).

$$\hat{y}_{t+h} \pm z_{\alpha/2} S.E.\left(\hat{y}_{t+h}\right), \qquad (3.12)$$

where

$$S.E.\left(\hat{y}_{t+h}\right) = \sigma\sqrt{h\left(1+\alpha^2+\alpha\beta h+\frac{1}{6}\beta^2 h(2h-1)\right)}\ .$$

Here, σ is the standard deviation of the residuals and α is a smoothing parameter [27].

4. Time-series K-NN regression

Time-series K-NN regression enhances traditional K-NN by effectively capturing temporal dependencies in univariate time-series forecasting. This approach leverages the relationships between different points in time, leading to improved forecasting accuracy. The steps are as follows [28-30]:

1) Lagged Feature Vectors: Construct feature vectors using lagged values:

$$Z_i = \{y_{i-p}, y_{i-(p-1)}, \ldots, y_{i-1}\},$$

where p is the number of lags.

2) Target Values: Define the target value y_i, corresponding to the future values to be predicted.

3) Distance Calculation: compute the Euclidean distance between the current instance and all historical feature vectors:

$$d\left(Z_i, Z_j\right) = \sqrt{\sum_{k=1}^{p}\left(y_{i-k} - y_{j-k}\right)^2}, \qquad (4.1)$$

where $i = 1,2,\ldots,n-p$.

4) K-Nearest Neighbors: Identify the k nearest neighbors of Z_i :

$$\Omega_i = \left\{ Z_{i_1}, Z_{i_2}, \ldots, Z_{i_k} \right\}.$$

5) Forecasting: Predict future values by averaging the target values of the k nearest neighbors. The corresponding equation is shown in Eq. (4.2).

$$\hat{y}_i = \frac{1}{K} \sum_{k=1}^{K} y_{i_k}, \qquad (4.2)$$

where y_{i_k} are the target values corresponding to the k nearest neighbors.

K-NN with circular bootstrap enhances predictive performance by quantifying uncertainty, a capability that the standalone K-NN model lacks. Without bootstrapping, predictions would lack interval estimates, reducing their practical utility. By integrating statistical resampling, this approach provides more robust and reliable forecasts [31, 32].

5. Dataset

Forecasting durian export volumes holds significant importance for Thailand's agricultural economy. As one of the world's largest durian exporters, accurate predictions help exporters align production with demand, reduce excess inventory, and support governmental agencies in developing effective policies for trade and resource allocation. Additionally, timely forecasts assist logistics and distribution companies in optimizing transportation schedules to meet seasonal peaks.

The dataset used as a case study consists of monthly durian export volumes in tonnes from Thailand, spanning from January 2011 to December 2023. The data was extracted from the website https://impexpth.oae.go.th/export. This dataset provides an extensive view of durian exports over multiple years, enabling a comprehensive analysis of trends, seasonality, and variations in export volumes.

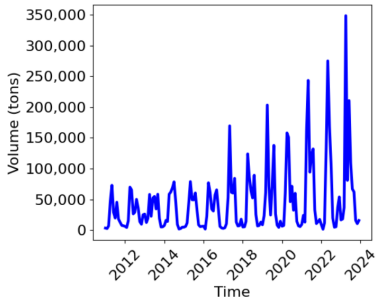

Fig. 1. Monthly durian export volumes from 2011 to 2023, showing seasonal patterns and an increasing long-term trend.

Fig. 1 illustrates the monthly durian export volumes from 2011 to 2023. This plot highlights several key features. There is a clear seasonal pattern with peaks and troughs occurring at regular intervals. The export volumes show significant variation year-to-year, with some years experiencing higher peaks than others. Additionally, the general trend appears to be increasing over the period, suggesting growth in durian export volumes.

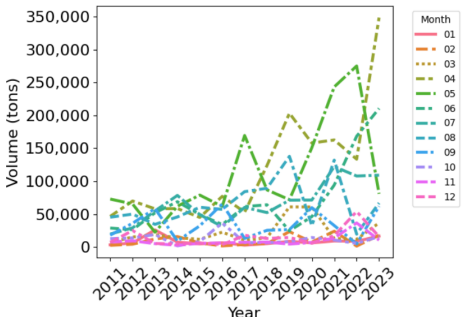

Fig. 2. Monthly durian export volumes (training dataset) from 2011 to 2023, separated by month to illustrate seasonal patterns.

Fig. 2 represents the durian export volumes separated by month over the years. This plot provides insights into seasonal variations. Each line represents a specific month, allowing us to observe monthly trends across different years. Months like April, May, and June tend to have the higher export volumes, indicating peak harvesting and export periods. In contrast, months such as January, November, and December generally show lower export volumes.

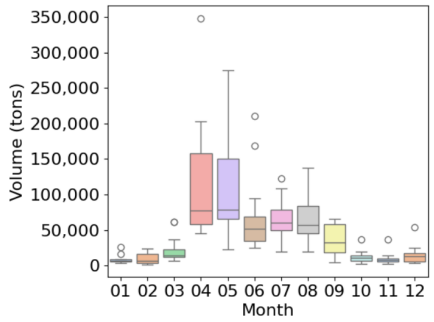

Fig. 3. Boxplot of durian export volumes (training dataset) by month, highlighting seasonal variability and distribution.

Fig. 3 shows the distribution of durian export volumes separated by month. This plot helps in understanding the variability and spread within each month. The boxplot shows the median, quartiles, and outliers for each month's export volumes. April and May exhibit the highest median export volumes, with a wider spread, indicating variability in exports. Lower median export volumes and narrower spreads are observed in months like January, November, and December.

6. Performance Criteria

This section outlines the performance criteria used to evaluate the accuracy and reliability of both point and interval forecasts generated by the proposed method.

6.1 Point forecast performance

1) Root Mean Squared Error (RMSE): Root RMSE is a standard metric for assessing the accuracy of point forecasts. It is defined as follows [33, 34]:

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

RMSE provides a measure of the average magnitude of forecast errors, penalizing larger errors more heavily due to the squaring operation.

2) Mean Absolute Percentage Error (MAPE): MAPE is another commonly used metric for evaluating forecast accuracy, expressed as a percentage:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%\,.$$

MAPE provides an intuitive measure of forecast accuracy by indicating the average absolute percentage error of the forecasts [35, 36].

6.2 Interval forecast performance

1) Coverage Percentage: The coverage percentage evaluates the proportion of actual values that fall within the forecast intervals. It is defined as:

$$\text{Coverage} = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \in [L_i, U_i]\right) \times 100\%.$$

A higher coverage percentage indicates that the forecast intervals effectively capture the variability in the data [37, 38].

2) Interval Width: The width of the interval forecast provides a measure of the uncertainty associated with the predictions. It is calculated as

$$\text{Width} = \frac{1}{n}\sum_{i=1}^{n}\left(L_i - U_i\right).$$

A narrower interval width, given a high coverage percentage, indicates more precise forecasts [39, 40].

7. Methodology

This section outlines the methodology used to propose the interval forecast for durian export volumes in Thailand using K-NN regression with circular bootstrap on residuals. The process is described in the following steps.

7.1. Data preparation

To stabilize the variance, a log transformation is applied to the durian export volumes. The time-series model is additive, as shown in Eq. (7.1):

$$\log(y) = T + S + C + I\,, \qquad (7.1)$$

where $T, S, C,$ and I represent the trend, seasonal, cyclical, and irregular components, respectively. The trend

component in our case study is estimated using Eq. (7.2):

$$\hat{T}_{t+h,\log(y)} = a + bt\,, \qquad (7.2)$$

where a and b are the estimated coefficients. The detrended data is then obtained by subtracting $\hat{T}_{t+h,\log(y)}$ from $\log(y_i)$ as defined in Eq. (7.3):

$$R_i = \log(y_i) - \hat{T}_{t+h,\log(y)}, \qquad (7.3)$$

where R is the "remains" component, corresponding to $S + C + I$, also referred to as the detrended transformed data. This transformation is essential for applying K-NN time-series regression [11]. The detrended transformed data is illustrated in Fig. 4.

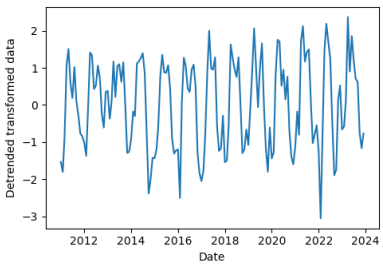

Fig. 4. Detrended transformed data obtained after removing the trend component.

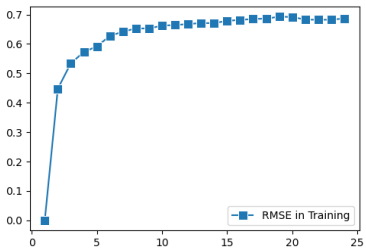

Fig. 5. RMSE in the training dataset as a function of k.

7.2 K-nearest neighbors regression

A K-NN regression model is fitted to the detrended transformed target values. This involves searching for an optimal k. The RMSEs of the training dataset for each k are plotted in Fig. 5.

The RMSE curve, while not perfectly flat, shows negligible improvement beyond $k = 3$. Reducing the number of neighbors also minimizes computational costs and model complexity, which is particularly beneficial when dealing with high-frequency data.

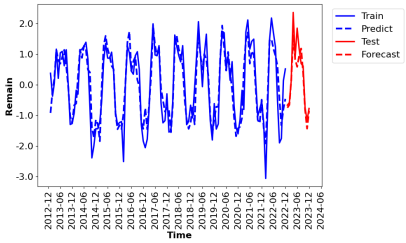

Fig. 6. Actual, predicted, and forecasted remains $(S + C + I)$ in log scale for the training and test datasets.

In our case study, 24 lags were used as independent variables. This choice is particularly useful for capturing long-term seasonal trends and patterns. The number of lags can be determined through trial and error, domain knowledge, or cross-validation [41–43].

7.3 Generating Bootstrap Samples on Residuals

The circular bootstrap on residuals, as described in Section 2, involves generating multiple bootstrap samples by resampling the residuals:

$$e_i = R_i - \hat{R}_i\,.$$

The residuals exhibit an approximately normal distribution, as shown in Fig. 7. The autocorrelation values at all lags beyond lag 1 lie within the blue confidence interval bands. This indicates that the autocorrelations are not statistically significant, suggesting that the K-NN model has effectively captured the underlying data structure, and the residuals do not exhibit any systematic patterns.

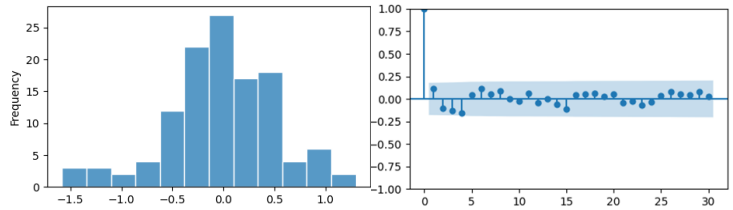

Fig. 7. Histogram of residuals in log scale (left) and autocorrelation function plot (right).

Then, the bootstrap samples of the remains with added residuals are defined as:

$$V_b^{(j)} = \{\hat{R}_i + e_{i_j}, \hat{R}_{i+1} + e_{i_j+1}, \ldots, \hat{R}_{i+n-1} + e_{i_j+n-1}\},$$

where $j = 1, 2, \ldots, 10000$. In this case study, the block length is varied from 1 to 12, and the resulting bootstrap samples of forecasted remains in the test dataset are illustrated in Fig. 8.

It is observed that the length of the bootstrap block (BL) does not significantly affect the forecast distribution for any future months, except for the first month. For the first month, the forecast distributions tend to have a heavy tail. In contrast, for BL values ranging from 3 to 12, the forecast distributions exhibit lighter tails and are more concentrated around the point forecasts.

It is worth noting that when the BL is set to 1, the procedure resembles classical bootstrap, which disrupts the serial correlation in time-series data. This results in wider, less reliable forecast intervals. In contrast, for BL values of 3 or higher, the circular bootstrap preserves these dependencies, producing more concentrated distributions around the point forecasts.

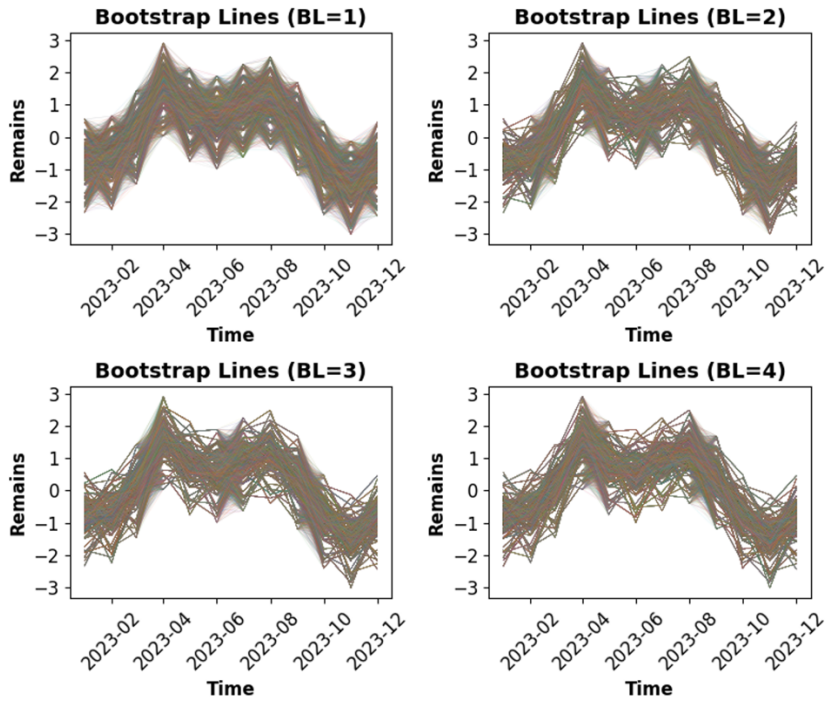

Fig. 8. Simulated remains in the log scale with varying bootstrap block lengths.

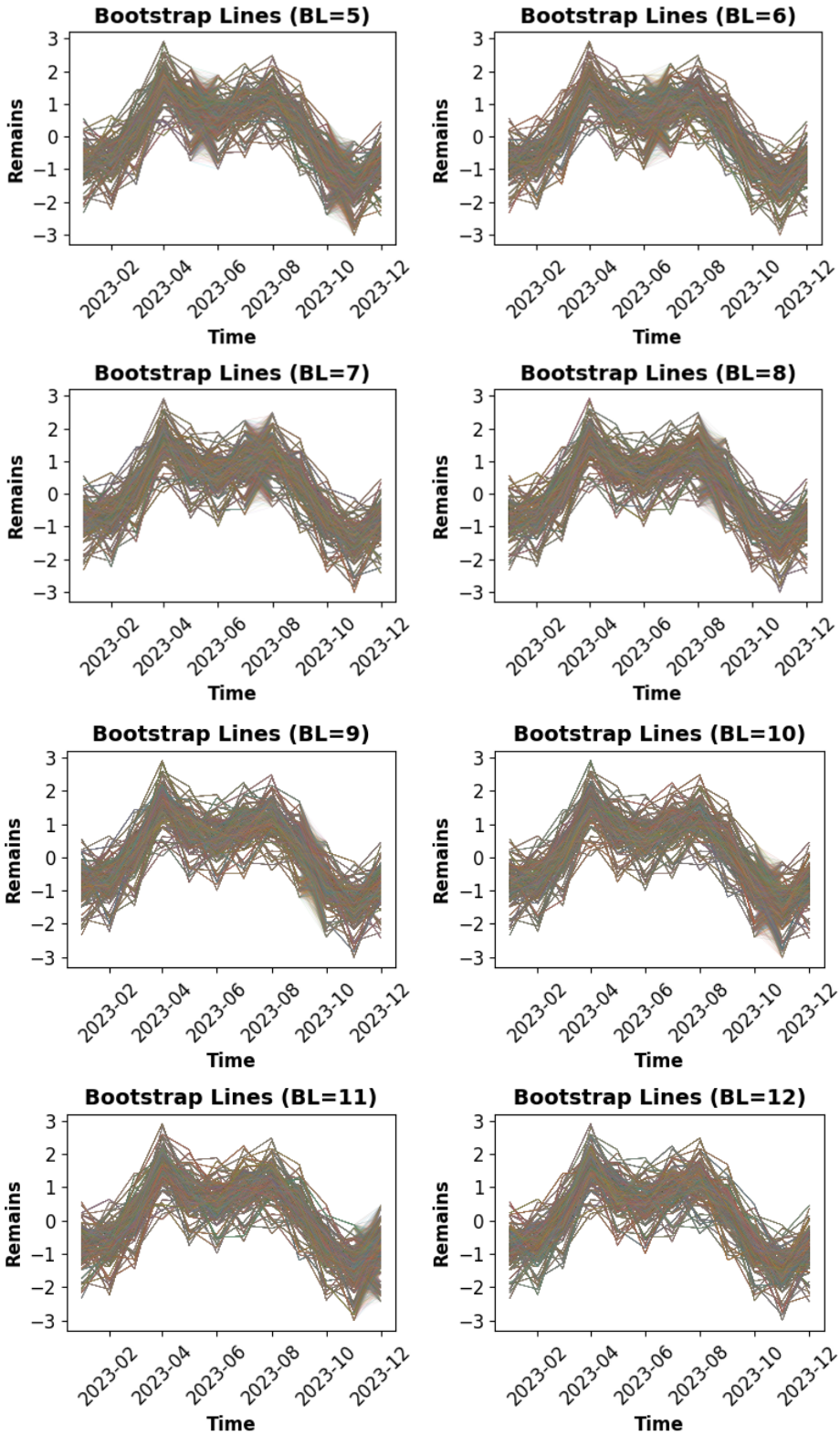

Fig. 8. Simulated remains in the log scale with varying bootstrap block lengths (continued).

7.4 Extract the forecast interval

When using bootstrap resampling with BL = 12, the lower (LB) and upper (UB) bounds of the forecast intervals are determined from the quantiles of the bootstrap samples. Specifically, for a given confidence level $(1-\alpha)100\%$, the $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap samples define the lower and upper bounds, respectively [44–47].

The forecast intervals of R_{t+h} at each future time step $t+h$, where h represents the forecast horizon, are computed as follows:

$$LB_{R_{t+h}} = \text{Quantile}_{\alpha/2}\left(\left\{V_b^{(1)}, V_b^{(2)}, \ldots, V_b^{(10,000)}\right\}\right),$$

$$UB_{R_{t+h}} = \text{Quantile}_{1-\alpha/2}\left(\left\{V_b^{(1)}, V_b^{(2)}, \ldots, V_b^{(10,000)}\right\}\right),$$

where $V_b^{(1)}, V_b^{(2)}, \ldots, V_b^{(10000)}$ are the bootstrap samples for the *h*-th forecast period.

7.5 Adding the trend component to the forecast interval

Before transforming back to the original scale, the trend component must be added to the forecasted values and their corresponding interval bounds while still in the log scale. This ensures that the forecasts and intervals account for the underlying trend present in the data.

- Forecasted Values:

$$\log(y_{t+h}) = \hat{R}_{t+h} + \hat{T}_{t+h}.$$

- Forecasted Intervals:

$$LB_{t+h,\log(y)} = LB_{R_{t+h}} + \hat{T}_{t+h}$$

$$UB_{t+h,\log(y)} = UB_{R_{t+h}} + \hat{T}_{t+h}.$$

7.6 Transformation to the original scale

After adding the trend component, the final step is to transform the forecasted values and forecasted intervals back to the original scale. This involves exponentiating the forecasted values and their corresponding interval bounds:

- Forecasted Values:

$$\hat{y}_{t+h} = \exp[\log(y_{t+h})].$$

- Forecasted Intervals:

$$LB_{t+h,y} = \exp\left[LB_{t+h,\log(y)}\right]$$

$$UB_{t+h,y} = \exp\left[UB_{t+h,\log(y)}\right].$$

8. Results

This section presents the point forecasts for durian export volumes obtained using the proposed K-NN regression with circular bootstrap on residuals. Both point and forecasted intervals are compared with those from SARIMA and ETS.

8.1 Point forecasts

Although point forecasts are not the primary focus of this study, they play a crucial role in constructing forecasted intervals when using circular bootstrap on residuals. The point forecasts generated by the K-NN, ETS, and SARIMA models are presented in Table 1.

Table 1. Comparison of point forecasts in the test dataset.

Month	Actual	K-NN	ETS	SARIMA
Jan 23	16,598.6	15,236.2	9,953.4	11,796.1
Feb 23	18,118.3	16,845.2	9,135.6	5,240.1
Mar 23	36,684.6	36,889.7	25,341.6	14,703.9
Apr 23	348,527.7	165,036.7	128,624.5	100,374.4
May 23	80,854.3	75,602.4	136,178.2	126,790.6
Jun 23	210,559.7	59,503.5	76,081.3	64,907.5
Jul 23	109,118.5	85,654.6	81,836.9	70,138.1
Aug 23	66,855.3	109,204.3	84,170.2	55,607.3
Sep 23	62,045.4	49,440.6	37,143.0	21,684.8
Oct 23	15,921.2	14,848.7	14,269.3	9,074.0
Nov 23	10,551.5	8,085.8	11,121.7	9,677.0
Dec 23	15,711.9	14,567.4	15,621.7	13,613.2

The point forecasts for April 2023 show a significant deviation across all models. This discrepancy can be attributed to the high variability in April 2023, where the actual durian export volume reached 348,527.7 tons—substantially higher than the maximum observed value in the training dataset (approximately 250,000 tons), as shown in Figs. 2 and 3. This extreme value highlights the limitations of all models in capturing unseen extreme variations.

A detailed performance comparison, including RMSE and MAPE across the training and test datasets, is presented in Table 2. The K-NN model outperformed ETS and SARIMA, achieving the lowest RMSE and MAPE values. Specifically, K-NN achieved RMSEs of 65,300 (training) and 70,136.5 (test), compared to 77,148.8 (training) and 86,654.5 (test) for ETS, and 86,073.5 (training) and 90,210.2 (test) for SARIMA.

Table 2. A comparison of RMSE and MAPE across training and test datasets.

Model	RMSE (Training)	RMSE (Test)	MAPE (Training)	MAPE (Test)
K-NN	65,300.0	70,136.5	22.5%	24.1%
ETS	77,148.8	80,654.3	34.6%	35.3%
SARIMA	86,073.5	90,210.2	44.2%	44.9%

Table 3. Comparison of forecasted intervals in the test dataset.

Month	Actual	Lower	Upper	Width
Jan 23	16,598.6	4,188	40,576	36,388
Feb 23	18,118.3	4,630	44,861	40,231
Mar 23	36,684.6	10,140	98,243	88,103
Apr 23	348,527.7	42,920	439,518	396,598
May 23	80,854.3	19,661	191,423	171,762
Jun 23	210,559.7	15,475	150,661	135,187
Jul 23	109,118.5	23,544	216,875	193,331
Aug 23	66,855.3	28,400	276,503	248,103
Sep 23	62,045.4	12,858	125,182	112,325
Oct 23	15,921.2	4,076	39,544	35,469
Nov 23	10,551.5	2,223	20,473	18,251
Dec 23	15,711.9	4,004	38,795	34,791

8.2 Forecasted intervals

The proposed interval forecasting method, which applies circular bootstrapping on residuals, produces substantially narrower forecasted intervals compared to ETS and SARIMA, particularly in May and June 2023, as shown in Fig. 9.

On average, the mean widths of the forecasted intervals for the K-NN, ETS, and SARIMA models are approximately 125,615.14, 161,188.08, and 160,176.04, respectively. The interval coverage percentages are 91.67% (11 out of 12 months), 100%, and 100%, respectively.

Table 3 summarizes the lower and upper bounds along with the corresponding interval widths. Although ETS and SARIMA models provide full coverage, their intervals are, on average, 22% wider than those of the proposed method. This suggests that while ETS and SARIMA offer greater coverage, they do so at the cost of less precise intervals.

Notably, the proposed method's interval failed to cover the actual value in only one instance (June 2023), demonstrating its effectiveness in balancing interval width and coverage.

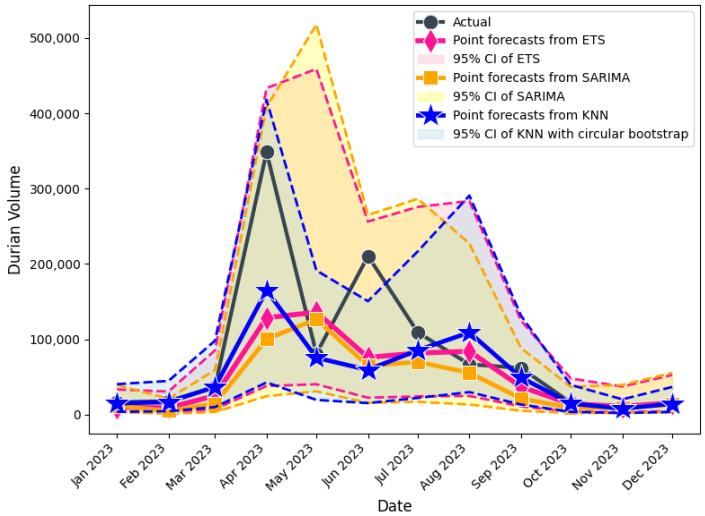

Fig. 9. Forecasted intervals for durian export volumes in 2023 using K-NN with circular bootstrap, ETS, and SARIMA models.

9. Conclusions and Discussions

This study introduced a novel interval forecasting approach that integrates circular bootstrap on residuals with K-NN regression to predict durian export volumes from Thailand. The proposed method effectively captures non-linearities and seasonal complexities, offering significant improvements over traditional forecasting models such as SARIMA and ETS.

The comparative analysis demonstrated that K-NN with circular bootstrap outperforms SARIMA and ETS in forecast accuracy, achieving lower RMSE and MAPE values, which indicate superior point forecast performance. Moreover, the forecasted intervals generated by the proposed method were substantially narrower, particularly in May and June 2023, while maintaining a high coverage percentage.

While the proposed method produces narrower intervals than ETS and SARIMA, it comes with slightly lower coverage. However, in practical applications, more precise intervals provide better guidance for decision-making, especially when a slight reduction in coverage is acceptable in exchange for higher specificity in predictions.

The superior performance of K-NN with circular bootstrap can be attributed to its non-parametric nature, which allows it to adapt dynamically to non-linear patterns, irregular seasonal peaks, and emerging trends. In contrast, SARIMA and ETS rely on predefined functional forms, which limit their adaptability to complex variations in durian export data.

However, this study also identifies limitations. In April 2023, all models, including the proposed method, deviated significantly from actual export volumes due to an unprecedented surge in durian exports, which was not represented in the training data. This highlights the challenges of forecasting extreme values and outliers, emphasizing the need for further refinement in capturing rare but impactful market shifts.

Acknowledgment

The authors gratefully acknowledge the financial support provided by Faculty of Science and Technology, Thammasat University, Contract No. SciGR 5/2567.

References

[1] Tajmouati S, Wahbi BEL, Bedoui A, Abarda A, Dakkon M. Applying k-nearest neighbors to time series forecasting: Two new approaches. J Forecasting. 2024 Feb;1:1–16.

[2] Martínez F, Frías MP, Pérez MD, Rivera AJ. A methodology for applying k-nearest neighbor to time series forecasting. Artif Intell Rev. 2019;52(3):2019–37.

[3] Zhang N, Lin A, Shang P. Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting. Physica A: Stat Mech Appl. 2017;477:161–73.

[4] Tang L, Pan H, Yao Y. K-Nearest Neighbor Regression with Principal Component Analysis for Financial Time Series Prediction. ICCAI '18: Proceedings of the 2018 International Conference on Computing and Artificial Intelligence. 2018;127–31.

[5] Liu Q, Liu C. A Novel Locally Linear KNN Method With Applications to Visual Recognition. IEEE Trans Neural Netw Learn Syst. 2017 Sep;28(9):2010–21.

[6] Ni KS, Nguyen TQ. An adaptable k-nearest neighbors algorithm for MMSE image interpolation. IEEE Trans Image Process. 2009 Sep;18(9):1976–87.

[7] Martínez F, Frías MP, Pérez-Godoy MD, Rivera AJ. Dealing with seasonality by narrowing the training set in time series forecasting with kNN. Expert Syst Appl. 2018 Mar;103:38–48.

[8] Martínez-Álvarez F, Troncoso A, Riquelme JC, Aguilar-Ruiz JS. Energy time series forecasting based on pattern sequence similarity. IEEE Trans Knowl Data Eng. 2011 Sep;23(8):1230–43.

[9] Srisuradetchai P, Panichkitkosolkul W. Using ensemble machine learning methods to forecast particulate matter ($PM_{2.5}$) in Bangkok, Thailand. In: Multi-disciplinary Trends in Artificial Intelligence: 15th International Conference, MIWAI 2022, Virtual Event, November 17–19, 2022, Proceedings. 2022 Nov. p. 204–15.

[10] Troncoso A, Riquelme Santos JM, Gomez-Exposito A, Martinez-Ramos JL. Electricity market price forecasting based on weighted nearest neighbors techniques. IEEE Trans Power Syst. 2007 Sep;22(3):1294–301.

[11] Srisuradetchai P. A novel interval forecast for k-nearest neighbor time series: A case study of durian export in Thailand. IEEE Access. 2024;12:2032–44.

[12] Fernández-Rodríguez F, Sosvilla-Rivero S, Andrada-Félix J. Nearest-neighbour predictions in foreign exchange markets. In: Chen SH, Wang PP, editors. Computational Intelligence in Economics and Finance. Berlin, Heidelberg: Springer; 2004. p. 193–207.

[13] Martínez F, Frías MP, Charte F, Rivera AJ. Time series forecasting with KNN in R: The tsfknn package. The R Journal. 2019;11(2):229–42.

[14] Srisuradetchai P, Phaphan W. Using Monte-Carlo dropout in deep neural networks for interval forecasting of durian export. WSEAS Trans Syst Control. 2024;19:10–21.

[15] Kück M, Freitag M. Forecasting of customer demands for production planning by local k-nearest neighbor models. Int J Prod Econ. 2021;231:107837.

[16] Lee WM. Python machine learning. 1st ed. Wiley; 2019 Apr. ISBN: 978-1119545637.

[17] Narejo S, Jawaid MM, Talpur S, Baloch R, Pasero EGA. Multi-step rainfall forecasting using deep learning approach. PeerJ Comput Sci. 2021;7:e514.

[18] Panichkitkosolkul W, Srisuradetchai P. Bootstrap confidence intervals for the parameter of zero-truncated Poisson-Ishita distribution. Thailand Statistician. 2022 Oct;20(4):918–27.

[19] Kummaraka U, Srisuradetchai P. Time-series interval forecasting with dual-output Monte Carlo dropout: A case study on durian exports. Forecasting. 2024 Sep;6(3):616–36.

[20] Efron B, Tibshirani RJ. An introduction to the bootstrap. 1st ed. Boca Raton, FL, USA: Chapman & Hall/CRC; 1993.

[21] Politis DN, Romano JP. The stationary bootstrap. J Am Stat Assoc. 1994 Dec;89(428):1303–13.

[22] Lahiri SN. Resampling methods for dependent data. New York, NY, USA: Springer-Verlag; 2003. p. 1–374.

[23] Davison AC, Hinkley DV. Bootstrap methods and their application. Cambridge, U.K.: Cambridge University Press; 1997.

[24] Härdle W, Horowitz J, Kreiss JP. Bootstrap methods for time series. Int Stat Rev. 2013 Aug;71(2):435–59.

[25] Shao X, Politis DN. The dependent wild bootstrap. J Am Stat Assoc. 2010 Mar;105(489):218–35.

[26] Bühlmann P, Künsch HR. Block length selection in the bootstrap for time series. Comput Stat Data Anal. 1999 Sep;31(3):295–310.

[27] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and Practice. 2nd

ed. Melbourne, Australia: OTexts; 2018. Available from: https://otexts.com/fpp2 [Accessed 2024 Jun 24].

[28] Micchelli CA, Pontil M. On learning vector-valued functions. Neural Comput. 2005 Jan;17(1):177–204.

[29] Taieb SB, Bontempi G, Sorjamaa A, Lendasse A. Long-term prediction of time series by combining direct and MIMO strategies. In: 2009 International Joint Conference on Neural Networks; 2009 Jun 14-19; Atlanta, GA, USA. IEEE; 2009. p. 3054–61.

[30] Box GEP, Jenkins GM, Reinsel GC. Time series analysis: Forecasting and control. 1st ed. Hoboken, NJ, USA: Wiley; 2008. p. 1–784.

[31] Gel YR, Lyubchich V, Ramirez Ramirez LL. Bootstrap quantification of estimation uncertainties in network degree distributions. Sci Rep. 2017;7:5807.

[32] de Luna X, Johansson P, Sjostedt-de Luna S. Bootstrap inference for k-nearest neighbour matching estimators. IZA Discussion Paper No. 5361. 2010. Available from: https://ssrn.com/abstract-=1723999.

[33] Srisuradetchai P, Suksrikran K. Random kernel k-nearest neighbors regression. Front Big Data. 2024;7:1402384.

[34] Huadsri S, Phaphan W. The development of forecasting models for life insurance data by employing time-series analysis and machine learning techniques. WSEAS Trans Math. 2024;23.

[35] Simmachan T, Manopa W, Neamhom P, Poothong A, Phaphan W. Detecting fraudulent claims in automobile insurance policies by data mining techniques. Thailand Statistician. 2023 Jul;21(3): 552–68.

[36] De Myttenaere A, Golden B, Le Grand B, Rossi F. Mean absolute percentage error for regression models. Neurocomputing. 2016;192:38–48.

[37] Kummaraka U, Srisuradetchai P. Interval estimation of the dependence parameter in bivariate Clayton copulas. Emerg Sci J. 2023 Oct;7(5):1478–90.

[38] Srisuradetchai P, Dangsupa K. On interval estimation of the geometric parameter in a zero-inflated geometric distribution. Thailand Statistician. 2023;21(1):93–109.

[39] Løhre E, Juanchich M, Sirota M, Teigen KH, Shepherd TG. Climate scientists' wide prediction intervals may be more likely but are perceived to be less certain. Weather Clim Soc. 2019 Jul;11(3):565–75.

[40] Zhang P, Liu S, Lu D, Zhang G, Sankaran R. A prediction interval method for uncertainty quantification of regression models. In: Ninth International Conference on Learning Representations (ICLR); 2021 May; Virtual Conference. OSTI ID: 1785172.

[41] Huang YC, Padarian J, Minasny B, McBratney AB. Using Monte Carlo conformal prediction to evaluate the uncertainty of deep learning soil spectral models. EGUsphere [Preprint]. 2025.

[42] Chen KY, Wang CH. A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Syst Appl.* 2007;32(1):254–64.

[43] Kalyan R, Nayak SYH, Pavitra R, Tripathy K, Prathyusha K. Forecasting foreign currency exchange price using long short-term memory with k-nearest neighbor method. Int J Eng Adv Technol. 2019. Available from: https://api.semanticscholar.org/CorpusID:242676291.

[44] Ialongo C. Confidence interval for quantiles and percentiles. Biochem Med (Zagreb). 2019;29:010101.

[45] Flowers-Cano RS, Ortiz-Gómez R, León-Jiménez JE, López Rivera R, Perera Cruz LA. Comparison of bootstrap confidence intervals using Monte Carlo simulations. Water. 2018;10(2):166.

[46] Tarr G. Small sample performance of quantile regression confidence intervals. J Stat Comput Simul. 2011;82(1):81–94.

[47] Diciccio TJ, Romano JP. A review of bootstrap confidence intervals. J R Stat Soc Ser B (Methodol). 1989 Jul;51(3):470.