Applying regression conformal prediction with nearest neighbors to time series data

Samya Tajmouati\textsuperscript{a}, Bouazza EL Wahbi\textsuperscript{a}, and Mohamed Dakkon\textsuperscript{b}

\textsuperscript{a}Department of Mathematics, Ibn Tofail University, Faculty of Sciences, Kénitra, Morocco; \textsuperscript{b}Department of Economics and Management, Abdelmalek Essaâdi University, FSJES, Tétouan, Morocco

\section*{ABSTRACT}

In this paper, we apply conformal prediction to time series data. Conformal prediction is a method that produces predictive regions given a confidence level. The regions outputs are always valid under the exchangeability assumption. However, this assumption does not hold for the time series data because there is a link among past, current, and future observations. Consequently, the challenge of applying conformal predictors to the problem of time series data lies in the fact that observations of a time series are dependent and therefore do not meet the exchangeability assumption. This paper aims to present a way of constructing reliable prediction intervals by using conformal predictors in the context of time series. We use the nearest neighbors method based on the fast parameters tuning technique in the weighted nearest neighbors (FPTO–WNN) approach as the underlying algorithm. Data analysis demonstrates the effectiveness of the proposed approach.

\section*{1. Introduction}

Many traditional machine-learning algorithms focus on providing point forecasts. However, in many cases, it is necessary to output the regions in which the unknown values should fall. As a result, prediction regions should be examined (Vovk, Gammerman, and Shafer 2005; Balasubramanian, Ho, and Vovk 2014; Kowalczewski 2019). To provide prediction regions, some machine learning methods do exist where Conformal Prediction is considered one of the most popular methods that provides valid regional predictions (Vovk, Gammerman, and Shafer 2005; Papadopoulos, Vovk, and Gammerman 2011; Balasubramanian, Ho, and Vovk 2014). Besides, unlike other machine learning methods such as the Bayesian approach, the application of Conformal Prediction requires only the exchangeability assumption. We say that \((x_1, x_2, \ldots)\) is an infinitely exchangeable sequence of random variables if, for any \(n\), the joint probability \(p(x_1, x_2, \ldots, x_n)\) is invariant to permutation of the indices. That is, for any permutation \(\pi\),

\[ p(x_1, x_2, \ldots, x_n) = p(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}) \]

Given a significance level \(\alpha\) and a machine learning algorithm for making a prediction \(\hat{y}\), conformal prediction produces a \(1 - \alpha\) prediction region – a set \(\Gamma^\alpha\) that contains \(y\) with probability at least \(1 - \alpha\) (Shafer and Vovk 2008). For regression problems, where \(y\) is numerical, \(\Gamma^\alpha\) is an interval. For classification problems, where \(y\) is a label, \(\Gamma^\alpha\) is a set of labels. In both versions, the
prediction regions are built according to a nonconformity measure—a function based on some machine learning algorithm that evaluates the dissimilarity between an example and a set of other examples. It should be noted that Conformal Prediction provides valid prediction regions under the exchangeability assumption, regardless the particular algorithm or nonconformity measure definition it uses. Nevertheless, the efficiency and therefore the tightness of prediction regions depends on the nonconformity measure used by Conformal Prediction. In this context, Papadopoulos, Vovk, and Gammerman (2011) defined six novel nonconformity measures based on the k-nearest neighbors regression algorithm and developed their corresponding prediction intervals. They demonstrated that the predictive regions produced by the new measures are tighter than those of the typical measure. Johansson et al. (2014) used random forests as the learning algorithm for regression Conformal Prediction, which is an effective algorithm to use compared with k-nearest neighbors and neural networks. In particular, the study showed that using random forests together with a nonconformity measure based on out-of-bag errors normalized using a nearest-neighbor-based difficulty estimate produced the tight regions. To lower the computational complexity observed in the nearest neighbor procedure (Boström et al. 2017) suggested a modified nonconformity measure, where the difficulty estimate employed for normalization is based on the variance of the predictions made by the trees in a forest. The modified approach is comparably good and significantly faster. Moreover, conformal prediction has been used effectively in various research areas. In healthcare domain, Zhan et al. (2020) introduced a novel application of conformal prediction in lung cancer prediction. Samples of breath air from lung cancer patients were gathered with an electronic nose. Next, the conformal prediction based on k-nearest neighbors was used to assess the reliability of cancer diagnosis. In drug discovery, Alvarsson et al. (2021) used conformal prediction to quantify the reliability of predictions from QSAR (quantitative structure–activity relationships) models and provided an application example for modeling ABC transporters. In time series setting, tough the exchangeability assumption is hardly satisfied, successful tries in implementing conformal prediction have recently shown up. Kowalczewski (2019) applied normalized conformal prediction to time series data using various point forecast algorithms and a nonconformity measure normalized using the error of nearest neighbors. They evaluated the efficiency of the proposed method and compared it with non-conformal methods, conformal methods without normalization, and conformal methods with different normalization techniques. Overall, the proposed method is efficient and similar to the other methods in term of performance. Kath and Ziel (2021) applied conformal prediction to short-term electricity price forecasting in three different power markets. They compared its performance with quantile regression averaging and empirical error distribution approaches, where they used naive expert learner, lasso regression, k-nearest neighbors, and SVM regression as underlying point forecast models. Their findings suggest that conformal prediction output reliable prediction intervals. Xu and Xie (2021) developed a method called EnbPI that constructs multiple prediction intervals at once for dynamic time series. EnbPI method doesn’t require exchangeability assumption and is similar to Conformal Prediction. The produced prediction intervals are approximately valid for some ensemble estimators. Wisniewski, Lindsay, and Lindsay (2020) applied Conformal Prediction to a financial time series that comprises the net positions of a given Market Maker in order to build a reliable range of position bounds. They compared Conformal Prediction to benchmark models like quantile regression and found that the application of conformal prediction leads to efficient results. Dashevskiy and Luo (2011) considered the problem of applying conformal predictors to time series prediction in general and the network traffic demand prediction problem in particular. They showed that in the case when the time series data do not meet the requirement of exchangeability, conformal predictors provide reliable prediction intervals, which indicates empirical validity. Moreover, they tested different point forecasts algorithms in order to determine the points that are empirical
efficient. Balasubramanian, Ho, and Vovk (2014) considered an application of conformal prediction to the network traffic demand time series and showed the empirical validity of the conformal prediction when the time series are not exchangeable.

Following the work of Dashevskiy and Luo (2011) and Balasubramanian, Ho, and Vovk (2014), we extend the application of conformal predictors to time series data when the exchangeability assumption is not met. Our approach differs from the existing one by the fact that it builds prediction regions in the multidimensional case and employs as the underlying algorithm the weighted nearest neighbors (WNN) based on the fast parameters tuning method (FPTO) as described in Tajmouati et al. (2021). Also, we extend our approach such that it provides efficient point forecasts, which lead to efficient prediction intervals (Tajmouati et al. 2021). Our method shows promising results.

The paper is organized as follows. Section 2 introduces the conformal prediction method. Sections 3 and 4 describe the general idea behind the FPTO–WNN method and the application of the conformal prediction to time series with the FPTO–WNN approach, respectively. A simulation study and an example with real data are provided in Sec. 5. Section 6 contains a summary.

2. Conformal prediction

Conformal prediction approach uses past experience to determine precise levels of confidence in new predictions. It is a method introduced first by Gammerman, Vovk, and Vapnik (1998) and studied by Vovk, Gammerman, and Shafer (2005) and Shafer and Vovk (2008). Its main objective is to produce valid prediction intervals under the exchangeability assumption of the data. The original version, which was implemented in the transductive manner, is defined as follows. Given an exchangeable sequence \( z_1, z_2, \ldots, z_l \) where \( z_i = (x_i, y_i) \) is the \( i^{th} \) pair such that \( x_i \in \mathbb{R}^d \) and \( y_i \in \mathbb{R} \) are the object and the label, respectively. Given a new unlabeled example \( x_{l+1} \), the task of the conformal prediction is to output an interval that contains label \( y_{l+1} \) of \( x_{l+1} \):

To do this, one has to define some non-conformity measure: \( A_{l+1} : Z^l \times Z \rightarrow \mathbb{R} \), which attributes a numerical score \( a_i = A_{l+1}(\{z_1, z_2, \ldots, z_{l-1}, z_{l+1}, \ldots, z_{l+1}\}, z_i) \) to each example \( z_i \) measuring the degree of disagreement between its label \( y_i \) and the predicted label \( \hat{y}_i = D_{\{z_1, z_2, \ldots, z_{l-1}, z_{l+1}, \ldots, z_{l+1}\}}(x_i) \), where \( D_{\{z_1, z_2, \ldots, z_{l-1}, z_{l+1}, \ldots, z_{l+1}\}} \) is the prediction rule created by the underlying algorithm used to predict \( y_i \) based on all the examples except \( z_i \). The degree of disagreement measures the deviation between the predicted value and the observed value. That is, if the predicted value is close to the observed one, the disagreement is considered weak. In the conformal prediction, the underlying algorithm means the algorithm like Neural Network, SVM, KNN, and regression, to name a few that provides point forecasts, which are later used to generate a prediction interval. Usually, one uses the function \( |y - \hat{y}| \) as a non-conformity score. The non-conformity score \( z_{l+1} \) does not reflect on its own any information and need to be compared with all other non-conformity measures. This comparison can be performed through the \( p \) value function calculated for the new example \( z_{l+1} \) as:

\[
p(\hat{y}) = \frac{\#\{i = 1, \ldots, l + 1 : a_i \geq a_{l+1}\}}{l + 1}
\]  

According to Eq. (1), the higher the value of \( a_{l+1} \), the less probable it is that \( y_{l+1} \) takes \( \hat{y} \). Finally, given a significance level \( \delta \), a regression conformal predictor outputs the following prediction region:

\[
\Gamma = \{ \hat{y} : p(\hat{y}) > \delta \}
\]
To overcome the computational inefficiency in the transductive conformal prediction, an inductive conformal prediction is proposed. The general steps in the inductive approach are as follows:

- Split the training dataset $z_1, z_2, \ldots, z_l$ into two smaller datasets: the calibration dataset with $q < l$ examples and the proper training dataset with $m:=l-q$ examples.
- Use the proper training data $z_1, z_2, \ldots, z_m$ to generate the prediction rule $D_{\{z_1, z_2, \ldots, z_m\}}$ created by the underlying algorithm.
- Attribute a non-conformity score to each one of the examples in the calibration set. Note that the non-conformity score $x_{m+i}$ of each example $z_{m+i}$ in the calibration dataset $z_{m+1}, z_{m+2}, \ldots, z_{m+q}$ is calculated as the degree of disagreement between $\hat{y}_{m+i} = D_{\{z_1, z_2, \ldots, z_m\}}(x_{m+i})$ and the real value $y_{m+i}$.
- Define $p$ value of $\hat{y}$ of $x_{l+g}$ as:
  
  $$p(\hat{y}) = \frac{\#\{i = m+1, \ldots, m+q, l+g : x_i \geq x_{l+g}\}}{q+1},$$

  where $x_{l+g}$ is the degree of disagreement between $\hat{y}_{l+g} = D_{\{z_1, z_2, \ldots, z_m\}}(x_{l+g})$ and $\hat{y}$.
- Sort the non-conformity scores of the calibration examples $x_{m+1}, \ldots, x_{m+q}$ in descending order: $x_{(m+1)}, \ldots, x_{(m+q)}$.
- For a significance level $\delta$ and $x_i = |y_i - \hat{y}|$, output the prediction region as:
  
  $$\Gamma = \{\hat{y} : p(\hat{y}) > \delta\} := |\hat{y}_{l+g} - x_{(m+i)}: \hat{y}_{l+g} + x_{(m+i)}|,$$

  where $s = \lceil \delta(q+1) \rceil$.

### 3. FPTO–WNN

FPTO–WNN is an automatic method that selects the optimal values of the nearest neighbors and performs feature selection in the weighted nearest neighbors (WNN) approach for time series data. The method works as follows. Let $a = (a_1, a_2, \ldots, a_T)$ and $E = \{E_1, \ldots, E_I\}$ be a time series and a set of $I$ training datasets of $a$, respectively. For $i \in \{1, \ldots, I\}$, $E_i = \{a_1, a_2, \ldots, a_{T-n}\}$ is the $i^{th}$ training dataset used to forecast the $i^{th}$ test dataset: $a_{T-n+1}, \ldots, a_{T-n+n}$. The optimal values of $p$ and $k$ are denoted by $p^*, k^*$, respectively, and are the values that minimize

$$\text{MAPE}^*(p,k) = \arg \min_{p,k} \text{MAPE}^*(p,k),$$

where

$$\text{MAPE}^*(p,k) = \frac{1}{I} \sum_{i=1}^{I} \frac{100}{n} \sum_{j=1}^{n} \frac{|\hat{a}_{T-i,n+j} - a_{T-i,n+j}|}{a_{T-i,n+j}}.$$
where \( w_j = \frac{1}{k} \sum_{i=j}^{k} \frac{1}{2} \) is a weighting factor related to the degree of closeness of \( s_{t_j} \) to \( s_{T_j} \). \( \mathbb{I}(x = y) \) represents the indicator function of event \( x = y \) that takes the value 1 if the assertion \( x = y \) is true and 0 otherwise. The interested reader may refer to the work of Tajmouati et al. (2021) paper for further theoretical analysis. Overfitting can be identified by splitting the time series data into training and testing datasets. The training dataset is split into a new train dataset and a validation dataset. Then, the model is iteratively trained and validated on the new train and validation sets according to the FPTO–WNN approach. Finally, the performance of the obtained model on the test dataset is assessed.

4. Conformal prediction for time series with FPTO–WNN

Section 3 presents an efficient way to provide point predictions for time series data. However, these point predictions are not associate with confidence information. In this section, we introduce a method that builds the prediction region for the point predictions for time series data. The conformal prediction is combined with the FPTO–WNN approach and works as follows. First, the time series \( a = (a_1, a_2, ..., a_T) \) is transform to pairs:

\[
z_t = (\text{object, label}) = (x_t, y_t) := \left( (a_{t-n.p+1}, a_{t-n.p+2}, ..., a_t), (a_{t+1}, ..., a_{t+n}) \right),
\]

for \( t = \{T - n, T - 2.n, ..., T - n.c\} \), where \( p \) is an integer and \( c \) is the last integer that holds the following inequality \( T - n.c \geq n.p \). Then, the prediction region of \( y_T = (a_{T+1}, ..., a_{T+n}) \) is calculated by defining the non-conformity measure based on the WNN approach as defined in Sec. 2. The implemented non-conformity scores are defined as follows:

\[
x_{t,j} = |a_{t+j} - \hat{a}_{t+j}| \quad \text{for} \; j = 1, 2, ..., n,
\]

where \( \hat{a}_{t+j} \) is the \( j^{th} \) component of \( \hat{y}_t = D_{\{z_{t-n}, ..., z_{T-n} \}}(x_t) \), such that \( D_{\{z_{t-n}, ..., z_{T-n} \}} \) is the prediction rule created by the WNN’s approach using all the previous examples of \( z_t \). For implementation, we consider the following scheme:

- Calculate the non-conformity scores: \( x_{T-n,j}, x_{T-2.n,j}, ..., x_{T-h.n,j} \), for \( j = 1, ..., n \), where \( h \) is an integer inferior or equal to \( c \).
- Define \( p \) value of \( y_{T,j} \) for \( x_T \) as:

\[
p(\tilde{y}_{T,j}) = \frac{\#(t = T - n, T - 2.n, ..., T - h.n, T : x_{t,j} \geq x_{T,j})}{h + 1}.
\]

- For each \( j = 1, ..., n \), sort the non-conformity scores: \( x_{T-n,j}, x_{T-2.n,j}, ..., x_{T-h.n,j} \) in descending order obtaining the sequences: \( x_{(T-n,j)}, x_{(T-2.n,j)}, ..., x_{(T-h.n,j)} \)
- For a significance level \( \delta \), output the prediction interval:

\[
\Gamma_{T,j} = \{ \tilde{y}_{T,j} : p(\tilde{y}_{T,j}) > \delta \} = [\hat{a}_{T+j} - x_{(T-s.n,j)}, \hat{a}_{T+j} + x_{(T-s.n,j)}],
\]

where \( s = \lfloor \delta(h + 1) \rfloor \).
- For a significance level \( \delta \), output the prediction region:

\[
\Gamma_T = \prod_{j=1}^{n} \Gamma_{T,j} = \prod_{j=1}^{n} [\hat{a}_{T+j} - x_{(T-s.n,j)}, \hat{a}_{T+j} + x_{(T-s.n,j)}],
\]
4.1. **Checking the validity and efficiency of the proposed method**

To check the validity of our proposed method, we split the time series into training and test datasets. Then, we implement the FPTO–WNN approach to the training dataset to tune the WNN model’s parameters: $p^*$ and $k^*$. We apply the proposed method to build the prediction intervals of the points that constitute the test dataset. We check the empirical reliability of the resulting predictive intervals by reporting the percentage of examples in the test dataset such that the true label is inside the corresponding prediction interval. Finally, we check the efficiency of the method by measuring the tightness of these regions. For this purpose, we use either the mean or the median function. That is, we follow the steps as defined in Algorithm 1 to check the efficiency and the reliability of the proposed method. Note that, since $|\delta(I_1 + i + 1)| \geq 1$ for $i = \{0, ..., I_2 - 1\}$, $I_1$ must be superior than $\frac{1}{\delta} - 1$. In order to build the prediction region of $y_T$, we set $h = I_1 + I_2$, $p = p^*$ and $k = k^*$.

**Algorithm 1**: CheckCP($a, p^*, n, k^*, I_1, I_2, \delta$)

**Require**: Time series ($a$), Window’s length ($p^*$), Number of predictions to be made ($n$), Number of neighbors ($k^*$), Number of validation sets used in FPTO–WNN ($I_1$), Number of test examples ($I_2$), significance level $\delta$

$M \leftarrow$ empty matrix with $n$ columns and $I_2$ rows

$$
\begin{pmatrix}
\mathcal{A}_{n-I_2-nI_2,1} & \cdots & \mathcal{A}_{n-I_2-nI_2,n} \\
\mathcal{A}_{n-I_2-I_2+1,1} & \cdots & \mathcal{A}_{n-I_2-I_2+1,n} \\
\vdots & \cdots & \vdots \\
\mathcal{A}_{n-I_2-I_2,1} & \cdots & \mathcal{A}_{n-I_2-I_2,n} \\
\end{pmatrix}
$$

$P \leftarrow$

$$
\begin{pmatrix}
\mathcal{A}_{n-I_2-nI_2,1} & \cdots & \mathcal{A}_{n-I_2-nI_2,n} \\
\mathcal{A}_{n-I_2-I_2+1,1} & \cdots & \mathcal{A}_{n-I_2-I_2+1,n} \\
\vdots & \cdots & \vdots \\
\mathcal{A}_{n-I_2-I_2,1} & \cdots & \mathcal{A}_{n-I_2-I_2,n} \\
\end{pmatrix}
$$

for $i = 0$ to $I_2 - 1$ step 1 do

$P \leftarrow$ sort each column of $P$ in descending order

$s \leftarrow [\delta(I_1 + i + 1)]$

$M[i + 1,] \leftarrow P[s,]$

Add the raw $(\mathcal{A}_{n-I_2+i,n,1}, ..., \mathcal{A}_{n-I_2+i,n,n})$ to $P$

end for

$\text{FIND} \leftarrow \begin{pmatrix} 
\mathcal{A}_{n-I_2,1} & \cdots & \mathcal{A}_{n-I_2,n} \\
\vdots & \cdots & \vdots \\
\mathcal{A}_{n-I_2,1} & \cdots & \mathcal{A}_{n-I_2,n} 
\end{pmatrix} \leq M$

$\text{FIND}_{i,j}$ takes 1 when $\mathcal{A}_{nI_2+i-1,n,j} \leq M_{i,j}$ and 0 otherwise.

**Output**: Percentage of examples in the test set for which the true label is inside the corresponding prediction region: $\sum_{nI_2} \sum_{nI_2} \text{FIND}_{i,j}$; Percentage of examples in the test set for which the $j^{th}$ component of the true label is inside the corresponding prediction interval: $\sum_{nI_2} \text{FIND}_{i,j}$; Mean width: $2.\text{mean}(M_{i,j} : j = 1, ..., n)$; Median width: $2.\text{median}(M_{i,j} : j = 1, ..., n)$

5. **Examples**

To evaluate the method derived in Sec. 4, we investigate the efficiency of the method utilizing two datasets. A simulated time series dataset and a real dataset. We set the size of the test set be
equal to 20% of the available data \((n.I_2 = 20\%, T)\) and the number of the test sets used in FPTO–WNN to be equal to 20% of training data if \(\frac{20\%}{n}.(T - n.I_2) \geq \frac{1}{\delta} - 1\) and \(\frac{1}{\delta} - 1\) otherwise. As mentioned above, \(I_1\) must be greater than \(\frac{1}{\delta} - 1\). Thus, the test set’s size must take at least \(\frac{1}{\delta} - 1\). \(I_1\) takes its possible minimal values \(\frac{1}{\delta} - 1\) and is defined as follows:

\[
I_1 = \begin{cases} \frac{20\%}{n}.(T - n.I_2), & \text{if } \frac{20\%}{n}.(T - n.I_2) \geq \frac{1}{\delta} - 1 \\ \frac{1}{\delta} - 1, & \text{otherwise} \end{cases}
\] (4)

### 5.1. Simulations studies

In this section we conduct simulations to investigate the performance of our derived approach. We simulate time series data to estimate the theoretical optimal intervals and then compare them with our method’s prediction intervals. We simulate monthly data from two models ETS\((A, N, A)\) and ETS\((A,Ad, A)\) where the residual errors follow the standard normal distribution. We set the number of predictions and the confidence level to 3 and 0.95, respectively. For most ETS models, a prediction interval can be written as: \(\hat{a}_{T+h|T} \pm c\sigma_h\) where \(c\) depends on the coverage probability and \(\sigma_h^2\) is the forecast variance (Hyndman and Athanasopoulos 2018). Thus, the theoretical widths are calculated as: \(2c\sigma_h\). When the forecast errors follow the normal distribution and using a 95% confidence level, the constant \(c\) is assumed to be equal to 1.96. The forecast variance expressions are:

\[
\sigma_h^2 = \sigma^2 \left[ 1 + x^2(h - 1) + \gamma k(2x + \gamma) \right], \quad \text{for ETS}(A,N,A) \\
\sigma_h^2 = \sigma^2 \left[ 1 + x^2(h - 1) + \gamma k(2x + \gamma) + \frac{\beta\phi h}{(1 - \phi)^2} \left\{ 2x(1 - \phi) + \beta\phi \right\} - \frac{\beta\phi(1 - \phi h)}{(1 - \phi)^2(1 - \phi^2)} \left\{ 2x(1 - \phi^2) + \beta\phi(1 + 2\phi - \phi^h) \right\} + \frac{2\beta\gamma\phi}{(1 - \phi)(1 - \phi^m)} \left\{ k(1 - \phi^m) - \phi^m(1 - \phi^mk) \right\} \right], \quad \text{for ETS}(A,Ad,A)
\]

where \(\sigma^2\) is the residual variance, \(m\) is the seasonal period, \(k\) is the integer part of \((h - 1)/m\), and \(\phi\) is a damping parameter. \(x\) and \(\gamma\) are the smoothing parameters. For the ETS \((A, N, A)\) model, we generate two series. The first series is of length \(T = 300\) with the smoothing parameters \(x = 0.5\) and \(\gamma = 0.2\). The second series is of length \(T = 400\) with the smoothing parameters \(x = 0.8\) and \(\gamma = 0.4\). Similarly, we generate two series from ETS\((A,Ad, A)\). The first series is of length \(300\) and the model parameters \(x = 0.7, \beta = 0.3, \gamma = 0.2\) and \(\phi = 0.82\). The second series is of length \(400\) and the model parameters \(x = 0.8, \beta = 0.2, \gamma = 0.1\) and \(\phi = 0.9\).

From Table 1, one can see that most of the empirical widths for the last three predictions are close to the theoretical widths. Moreover, from Table 2, one can conclude that the percentage inside predictive regions is high and exceeds the 95% confidence level in almost all the scenarios. Overall, the results guarantee the empirical validity and efficiency of our approach.
5.2. Cow’s milk production in the United Kingdom (UK)

We use the milk production data in the UK (Eurostat 2021) to evaluate the proposed method derived in Sec. 4. The time series data include 634 points ranging from January 1968 to October 2020, and are delineated by month. We implement the proposed algorithm CheckCP in R. To determine the efficiency of conformal prediction based on the WNN approach, we compare it with two underlying algorithms: Autoregressive Integrated Moving Average (ARIMA) and ETS. We fit the models on the training set using the functions `auto.arima` and `ets` from the R package `forecast` (Hyndman et al. 2018). Then, we use a similar algorithm CheckCP to measure the validity and the efficiency of conformal predictions using these models.

Figure 1 depicts the monthly cow’s milk production in a hundred thousand tonnes in the UK and shows the trend across the time. Table 3 presents the MAPE values of the four models on the test dataset for each horizon \( n = 1, 2, 3, 4 \). All the models provide low MAPE value, which indicates a good forecast accuracy. Overall, our approach outperforms other models across the horizon.

Table 4 presents the reliability of the obtained prediction regions for each underlying algorithm. \( \sum \sum_{n,j}^{\text{FINO}_{ij}} \) finds the percentage of the true label inside the predictive regions. It reports the percentage of examples for which the true label is inside the region output by each method. It is clear that for the confidence levels 90%, 92% and 95%, the underlying algorithms are
Table 3. MAPE values for FPTO–WNN, SARIMA, and ETS methods applied to the cow’s milk production data in the UK.

| Horizon n | Method                                  | MAPE  |
|-----------|-----------------------------------------|-------|
| 1         | FPTO–WNN ($p^* = 11, k^* = 7$)          | 1.2919|
|           | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 1.2269|
|           | ETS($M, N, A$)                          | 1.3456|
| 2         | FPTO–WNN($p^* = 6, k^* = 6$)           | 1.5768|
|           | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 1.6621|
|           | ETS($M, N, A$)                          | 1.6406|
| 3         | FPTO–WNN($p^* = 4, k^* = 7$)           | 1.8893|
|           | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 1.8608|
|           | ETS($M, N, A$)                          | 1.8727|
| 4         | FPTO–WNN($p^* = 2, k^* = 10$)          | 2.0759|
|           | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 2.0703|
|           | ETS($M, N, A$)                          | 1.9541|

Table 4. Reliability of the obtained prediction regions for FPTO–WNN, SARIMA, and ETS models using Cow’s milk production dataset in the UK.

| Horizon n | $l_1$ | $l_2$ | Method                                  | 90%   | 92%   | 95%   | $\sum \sum P_{(\text{find}_i)}$  |
|-----------|-------|-------|-----------------------------------------|-------|-------|-------|----------------------------------|
| 1         | 102   | 127   | FPTO–WNN ($p^* = 11, k^* = 7$)          | 91.34 | 93.70 | 96.85 | 91.34 |
|           |       |       | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 89.76 | 92.13 | 96.06 | 91.34 |
|           |       |       | ETS($M, N, A$)                          | 96.85 | 99.21 | 100   | 91.34 |
| 2         | 51    | 64    | FPTO–WNN ($p^* = 6, k^* = 6$)           | 90.62 | 92.19 | 95.31 | 93.54 |
|           |       |       | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 89.06 | 91.41 | 96.87 | 93.54 |
|           |       |       | ETS($M, N, A$)                          | 96.09 | 97.66 | 99.22 | 93.54 |
| 3         | 34    | 43    | FPTO–WNN ($p^* = 4, k^* = 7$)           | 92.25 | 93.80 | 97.67 | 94.53 |
|           |       |       | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 87.60 | 91.47 | 95.35 | 94.53 |
|           |       |       | ETS($M, N, A$)                          | 95.35 | 98.45 | 99.22 | 94.53 |
| 4         | 26    | 32    | FPTO–WNN ($p^* = 2, k^* = 10$)          | 91.40 | 94.53 | 96.90 | 96.72 |
|           |       |       | SARIMA(2, 1, 1)(0.1.1)$_{12}$           | 86.72 | 91.41 | 93.75 | 96.72 |
|           |       |       | ETS($M, N, A$)                          | 94.53 | 96.87 | 98.44 | 96.72 |

Table 5. Reliability and the tightness of FPTO–WNN, SARIMA, and ETS methods when $n = 1$ using the Cow’s milk production data in the UK dataset.

| Method            | Mean width | Median width | $\sum \sum P_{(\text{find}_i)}$  |
|-------------------|------------|--------------|----------------------------------|
|                   | 90%   | 92%   | 95%   | 90%   | 92%   | 95%   |                                  |
| FPTO–WNN          | 0.633 | 0.692 | 0.823 | 0.634 | 0.692 | 0.836 | 91.34 |
| SARIMA            | 0.625 | 0.681 | 0.840 | 0.626 | 0.686 | 0.840 | 91.34 |
| ETS               | 0.889 | 0.982 | 1.141 | 0.886 | 0.957 | 1.130 | 91.34 |

Table 6. Reliability and the tightness of FPTO–WNN, SARIMA, and ETS methods when $n = 2$ using the Cow’s milk production data in the UK dataset.

| $p^{th}$ | Method            | Mean width | Median width | $\sum \sum P_{(\text{find}_i)}$  |
|----------|-------------------|------------|--------------|----------------------------------|
| 1        | FPTO–WNN          | 0.692      | 0.756        | 0.881                           | 0.726 | 0.769 | 0.855 | 87.50 |
|          | SARIMA            | 0.730      | 0.775        | 0.979                           | 0.744 | 0.787 | 1.016 | 87.50 |
|          | ETS               | 0.990      | 1.096        | 1.248                           | 0.948 | 1.103 | 1.261 | 98.44 |
| 2        | FPTO–WNN          | 0.822      | 0.990        | 1.244                           | 0.827 | 1.066 | 1.253 | 93.75 |
|          | SARIMA            | 1.064      | 1.164        | 1.401                           | 1.092 | 1.162 | 1.375 | 96.87 |
|          | ETS               | 1.125      | 1.155        | 1.268                           | 1.119 | 1.143 | 1.207 | 93.75 |

empirically valid since the percentages reported are close to and exceed, in some cases, the confidence levels. Tables 5–8 report the reliability and the tightness of the three methods using 1, 2, 3, and 4 components, respectively. It is clear that the models with high percentage inside predictive
regions, such as the ETS, have low efficiency. Also, one can see that, in general, the FPTO–WNN method provides a low mean width and a low median width. That is, the FPTO–WNN method is the most efficient method.

### 6. Conclusion

This paper addressed the problem of forecasting prediction intervals. We used the conformal prediction and applied the weighted nearest neighbors based on the fast parameters tuning technique in the weighted nearest neighbors as its underlying algorithm. In particular, we showed how to extend the conformal prediction in time series data and how to apply it with nearest neighbors in the multidimensional case. We introduced a new algorithm that checks the empirical validity and efficiency of the proposed method. With the violation of the exchangeability assumption, the results on the simulated time series data and the Cow’s milk production dataset in the UK showed that the conformal prediction can still end up to be empirically valid and efficient. Moreover, implementing the WNN method as the underlying algorithm leads to a tight regions with efficient results.

### ORCID

Samya Tajmouati  [http://orcid.org/0000-0001-8221-1706](http://orcid.org/0000-0001-8221-1706)
Bouazza EL Wahbi  [http://orcid.org/0000-0001-6206-2512](http://orcid.org/0000-0001-6206-2512)

---

### Table 7. Reliability and the tightness of FPTO–WNN, SARIMA, and ETS methods when $n = 3$ using the Cow’s milk production data in the UK dataset.

| $j^{th}$ Method | Mean width | Median width | $\sum \frac{\alpha_{MN_j}}{\beta_{MN_j}}$ |
|-----------------|------------|--------------|-------------------------------------|
|                | 90% 92% 95% | 90% 92% 95% | 90% 92% 95% |
| 1 FPTO–WNN     | 0.676 0.737 0.825 | 0.679 0.682 0.811 | 0.70 0.70 0.9535 |
| SARIMA         | 0.625 0.709 0.837 | 0.636 0.744 0.822 | 0.88 0.93 0.9535 |
| ETS            | 0.840 0.960 1.127 | 0.848 0.920 1.130 | 0.93 100 100 |
| 2 FPTO–WNN     | 0.862 1.039 1.291 | 0.844 1.127 1.289 | 0.93 0.95 100 |
| SARIMA         | 1.090 1.171 1.215 | 1.125 1.162 1.210 | 0.93 0.93 0.9767 |
| ETS            | 1.194 1.303 1.523 | 1.126 1.337 1.541 | 0.97 100 100 |
| 3 FPTO–WNN     | 1.184 1.262 1.441 | 1.198 1.259 1.361 | 0.93 0.95 0.9767 |
| SARIMA         | 1.165 1.303 1.526 | 1.196 1.394 1.565 | 0.81 0.88 0.9302 |
| ETS            | 1.377 1.428 1.605 | 1.407 1.407 1.584 | 0.95 0.95 0.9767 |

### Table 8. Reliability and the tightness of FPTO–WNN, SARIMA, and ETS methods when $n = 4$ using the Cow’s Milk Production data in the UK dataset.

| $j^{th}$ Method | Mean width | Median width | $\sum \frac{\alpha_{MN_j}}{\beta_{MN_j}}$ |
|-----------------|------------|--------------|-------------------------------------|
|                | 90% 92% 95% | 90% 92% 95% | 90% 92% 95% |
| 1 FPTO–WNN     | 0.777 0.842 0.935 | 0.765 0.889 0.924 | 0.75 0.75 0.9687 |
| SARIMA         | 0.848 0.997 1.136 | 0.787 1.016 1.1056 | 0.90 100 100 |
| ETS            | 0.912 1.059 1.387 | 0.948 0.957 1.261 | 0.96 100 100 |
| 2 FPTO–WNN     | 1.062 1.135 1.488 | 1.027 1.132 1.353 | 0.96 0.96 0.9687 |
| SARIMA         | 1.240 1.435 1.596 | 1.180 1.418 1.506 | 0.93 0.93 0.9767 |
| ETS            | 1.131 1.156 1.326 | 1.119 1.143 1.187 | 0.97 100 100 |
| 3 FPTO–WNN     | 1.182 1.291 1.555 | 1.143 1.195 1.591 | 0.93 0.96 0.9687 |
| SARIMA         | 1.096 1.169 1.320 | 1.123 1.198 1.413 | 0.84 0.84 0.8437 |
| ETS            | 1.422 1.513 1.592 | 1.363 1.465 1.595 | 0.96 0.96 0.9687 |
| 4 FPTO–WNN     | 1.551 1.687 1.883 | 1.675 1.709 1.922 | 0.87 0.90 0.9687 |
| SARIMA         | 1.281 1.383 1.543 | 1.369 1.417 1.631 | 0.78 0.87 0.9062 |
| ETS            | 1.547 1.643 1.844 | 1.519 1.601 1.853 | 0.90 0.90 0.9687 |
Data availability statement

For our analysis, Cow’s Milk Production in the UK are provided by Eurostat, which are available at: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=apro_mk_colm&lang=en

References

Alvarsson, J., S. A. McShane, U. Norinder, and O. Spjuth. 2021. Predicting with confidence: Using conformal prediction in drug discovery. *Journal of Pharmaceutical Sciences* 110 (1):42–9. doi:10.1016/j.xphs.2020.09.055.

Balasubramanian, V. N., S.-S. Ho, and V. Vovk. 2014. *Conformal prediction for reliable machine learning*. Boston: Morgan Kaufmann.

Boström, H., H. Linusson, T. Löfström, and U. Johansson. 2017. Accelerating difficulty estimation for conformal regression forests. *Annals of Mathematics and Artificial Intelligence* 81 (1–2):125–44. doi:10.1007/s10472-017-9539-9.

Dashevskiy, M., and Z. Luo. 2011. Time series prediction with performance guarantee. *IET Communications* 5 (8):1044–51. doi:10.1049/iet-com.2010.0121.

Eurostat. 2021. Cow’s milk collection and products obtained – Monthly data.

Gammerman, A., V. Vovk, and V. Vapnik. 1998. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* UAI’98, 148–55. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hyndman, R., and G. Athanasopoulos. 2018. *Forecasting: Principles and practice*, 2nd ed. Melbourne, Australia: OTexts.

Hyndman, R., G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O’Hara-Wild, F. Petropoulos, S. Razbash, and E. Wang, and F. Yasmeen. 2018. `forecast`: Forecasting functions for time series and linear models.

Johansson, U., H. Boström, T. Löfström, and H. Linusson. 2014. Regression conformal prediction with random forests. *Machine Learning* 97 (1–2):155–76. doi:10.1007/s10994-014-5453-0.

Kath, C., and F. Ziel. 2021. Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting* 37 (2):777–99. doi:10.1016/j.ijforecast.2020.09.006.

Kowalczewski, J. 2019. Normalized conformal prediction for time series data.

Papadopoulos, H., V. Vovk, and A. Gammerman. 2011. Regression conformal prediction with nearest neighbours. *Journal of Artificial Intelligence Research* 40:815–40. doi:10.1613/jair.3198.

Shafer, G., and V. Vovk. 2008. A tutorial on conformal prediction. *Journal of Machine Learning Research* 9:371–421.

Tajmouati, S., B. El Wahbi, A. Bedoui, A. Abarda, and M. Dakkoun. 2021. Applying k-nearest neighbors to time series forecasting: Two new approaches, *arXiv e-prints* arXiv:2103.14200.

Vovk, V., A. Gammerman, and G. Shafer. 2005. *Algorithmic learning in a random world*. Berlin, Heidelberg: Springer.

Wisniewski, W., D. Lindsay, and S. Lindsay. 2020. Application of conformal prediction interval estimations to market makers’ net positions. In Conformal and Probabilistic Prediction and Applications, PMLR, pp. 285–301.

Xu, C., and Y. Xie. 2021. Conformal prediction interval for dynamic time-series. International Conference on Machine Learning, PMLR, pp. 11559–69.

Zhan, X., Z. Wang, M. Yang, Z. Luo, Y. Wang, and G. Li. 2020. An electronic nose-based assistive diagnostic prototype for lung cancer detection with conformal prediction. *Measurement* 158:107588. doi:10.1016/j.measurement.2020.107588.