Proceedings of Machine Learning Research 230:1–21, 2024 Conformal and Probabilistic Prediction with Applications

ConForME: Multi-horizon conformal time series forecasting

Aloysio Galvão Lopes GALVAOLOPES@LIX.POLYTECHNIQUE.FR
Eric Goubault GOUBAULT@LIX.POLYTECHNIQUE.FR
Sylvie Putot PUTOT@LIX.POLYTECHNIQUE.FR
LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France

Laurent Pautet LAURENT.PAUTET@TELECOM-PARIS.FR
LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France

Editor: Simone Vantini, Matteo Fontana, Aldo Solari, Henrik Boström and Lars Carlsson

Abstract

Split conformal prediction is a statistical method known for its finite-sample coverage guarantees, simplicity, and low computational cost. As such, it is suitable for predicting uncertainty regions in time series forecasting. However, in the context of multi-horizon forecasting, the current literature lacks conformal methods that produce efficient intervals and have low computational cost.

Building on the foundation of split conformal prediction and one of its most prominent extensions to multi-horizon time series forecasting (CF-RNN), we introduce ConForME, a method that leverages the time dependence within time series to construct efficient multi-horizon prediction intervals with probabilistic joint coverage guarantees. We prove its validity and support our claims with experiments on both synthetic and real-world data. Across all instances, our method outperforms CF-RNN in terms of mean, min, and max interval sizes over the entire prediction horizon, achieving improvements of up to 52%. The experiments also suggest that these improvements can be further increased by extending the prediction horizon and through hyperparameter optimization.

Keywords: split conformal prediction, multi-horizon time series forecasting, uncertainty quantification.

1. Introduction

Multi-horizon time series forecasting is essential in various fields, including predicting COVID-19 infection rates, modeling stock market trends, and motion prediction, a core component of the autonomous driving stack.

Within safety-critical applications, it is important to provide not only point predictions but also uncertainty estimates in the form of interval predictions. That is, for a time series of length T and a prediction horizon H, given a sequence of past observations $y_1 \dots y_{T-H}$, output $\hat{\mathbf{y}}_{\mathbf{T-H+1}} \dots \hat{\mathbf{y}}_{\mathbf{T}}$ prediction intervals. However, providing intervals that are both *efficient* (i.e., have small widths) and *valid* (i.e., contain the true future values of the series with a given target error rate α, $0 < \alpha < 1$) is a challenging task. It becomes even harder when we consider that state-of-the-art predictors, such as (Salzmann et al., 2020), are often large neural networks.

A widely known choice to quantify uncertainty for a predictor neural network is through the means of Bayesian neural networks (Fortunato et al., 2017). Nevertheless, this computation often proves to be intractable for large machine learning models and lacks solid guarantees. With that in mind, split conformal prediction (Vovk et al., 2005) has gained

significant attention due to its ability to provide finite-sample probabilistic guarantees. Additionally, it has a low computational cost and can work with essentially any type of predictor under only mild conditions[1]. It is in this context that CF-RNN (Stankeviciute et al., 2021) was introduced. This method extends split conformal prediction to provide *valid* prediction intervals for multi-horizon time series, effectively solving the problem of lack of guarantees and high computational cost in previous works (Wen et al., 2017; Alaa and Van Der Schaar, 2020; Gal and Ghahramani, 2016). However, it suffers from a fundamental limitation: due to its reliance on approximations, it often produces intervals that are not *efficient*, underestimating too conservatively the target error rate.

To address this limitation, we propose ConForME, a method that considers the prediction intervals not individually, but grouped into blocks whose validity is considered as a whole. Thanks to this, our method uses considerably fewer approximations, thus generating more efficient prediction intervals in the context of multi-horizon time series forecasting. ConForME maintains the same validity guarantees as CF-RNN and has the same sequential computational cost during calibration and prediction. To the best of our knowledge, no other work proposes an *efficient* method in these settings without the need for an additional calibration dataset and an additional optimization step. Our main contributions are twofold:

- We propose a new method to compute *valid* prediction intervals for any measurable predictor. Our method has low computational cost, with execution time dominated by the predictor's cost.
- We demonstrate the validity of our method both theoretically and through experiments on synthetic data and three real-world datasets. The method is detailed for 1-D time series, but we also extend it to 2-D time series to work with a dataset containing trajectory data.

This paper is structured as follows: in Section 2, we formally state the problem, also introducing the core notions of validity and efficiency; then, in Section 3 we present the related work; following that, Section 4 details ConForME, as well as CF-RNN and split conformal prediction; finally we present and discuss our experiments in Section 5.

2. Problem formulation

Let us consider sequences of length T and a prediction horizon $H < T$. We assume that we are given observations $y_1, \dots, y_{T-H}$ and a point predictor f that produces predictions $\hat{y}_{T-H+1}, \dots, \hat{y}_T$ of the ground truth future values $y_{T-H+1}, \dots, y_T$. Our problem is to compute prediction intervals $\hat{\mathbf{y}}_{\mathbf{T-H+1}}, \dots, \hat{\mathbf{y}}_{\mathbf{T}}$ around these predictions that are both small (efficient) and contain the ground truth future values with at least a given coverage probability $1 - \alpha$. In particular, the main weakness of the current literature is the lack of efficiency.

Formally, let $(Y_i)_{i=1}^T$ be a sequence of T random variables with unknown distribution. Given the realization of its first $T - H$ values $(y_i)_{i=1}^{T-H}$ and a coverage probability $1 - \alpha$, compute *valid* (Definition 1) and *efficient* (Definition 2) prediction intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^T$.

1. Split conformal prediction requires a measurable predictor f and a calibration dataset $\mathcal{D}_{cal}$ which is *exchangeable* with the observed data, as detailed in Section 4.1.

Definition 1 *Let $(y_i)_{i=T-H+1}^{T}$ be samples from the random variables $(Y_i)_{i=T-H+1}^{T}$. The prediction intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T}$ are valid if, for an error rate $0 < \alpha < 1$:*

$$\mathbb{P}\left(\bigcup_{i=T-H+1}^{T} (Y_i \notin \hat{\mathbf{y}}_\mathbf{i})\right) \leq \alpha \;\Leftrightarrow\; \mathbb{P}\left(\bigcap_{i=T-H+1}^{T} (Y_i \in \hat{\mathbf{y}}_\mathbf{i})\right) > 1 - \alpha$$

Definition 2 *Let the upper and lower bounds of an interval $\hat{\mathbf{y}}_\mathbf{i}$ be $\overline{\hat{y}_i}$ and $\underline{\hat{y}_i}$ respectively. We define our efficiency metric for the intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T}$ as follows:*

$$mean\_size\left((\hat{\mathbf{y}}_\mathbf{i})_{i=1}^{n}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\overline{\hat{y}_i} - \underline{\hat{y}_i}\right)$$

Remark 3 *We also evaluate the maximum and minimum interval sizes in our experiments, and the results also support the same general claims as the mean sizes.*

We assume that we have at our disposal a measurable predictor f, described as follows:

$$f(y_1, \ldots, y_{T-H}) = (\hat{y})_{i=T-H+1}^{T} \approx (y)_{i=T-H+1}^{T} \tag{1}$$

The predictor in Equation (1) approximates the ground truth future values $(y_i)_{i=T-H+1}^{T}$ given the past observations $(y_i)_{i=1}^{T-H}$. We also assume that we are given a set of sequences $\mathcal{D}_{cal}$, each having length T, such that any observed ground truth trajectory (y_i) is *exchangeable* with $\mathcal{D}_{cal}$. That is, the set $\mathcal{D}_{cal} \cup \{(y_i)\}$ obeys Definition 4:

Definition 4 *Given a set $\mathcal{D}$ with n elements $\{s_i\}_{i=1}^{n}$. Let $s_1, \ldots, s_n$ be samples from the random variables $S_1, \ldots, S_n$. We say that $\mathcal{D}$ is exchangeable if the tuples $(S_i)_{i=1}^{n}$ and $(S_{\sigma(i)})_{i=1}^{n}$ have the same distribution for any permutation σ i.e.:*

$$(S_i)_{i=1}^{n} \overset{d}{=} (S_{\sigma(i)})_{i=1}^{n}$$

3. Literature review

Although we employ Recurrent Neural Networks (RNNs) (Schmidt, 2019) as the foundation for our predictions, the primary focus of this study is on the interval bounds around the predictions. Consequently, this literature review does not delve into the details of the state-of-the-art predictors.

3.1. Conformal prediction

Conformal prediction (CP) was first introduced in (Vovk et al., 2005), but it has gained more attention in the last decade, mainly because it is a very general uncertainty quantification technique with theoretical guarantees. These guarantees hold even in the finite-sample case, and work for complex functions such as neural networks (Papadopoulos and Haralambous, 2011). Its main requirement is a calibration dataset that is *exchangeable* with the observed data. The main drawback of this technique, however, is the lack of conditional validity, which is a topic well explored in (Vovk, 2012).

More recently, CP has been extended to handle distribution shifts of the observed data under certain conditions (Tibshirani et al., 2019). It has also been extended by (Romano et al., 2019) to produce interval sizes that adapt to each input using quantile regression. Other extensions include adaptations to classification problems, see e.g. (Romano et al., 2020; Angelopoulos et al., 2021). However, most of these extensions do not consider data with temporal dependence.

3.2. Conformal prediction for time series

Our primary focus is on the application of CP to time series forecasting. In this context, we find it useful to distinguish between two scenarios: data generated from a single time series and data derived from multiple time series. In the first scenario, maintaining independence between data points is a significant challenge. In the latter scenario, however, it is plausible to assume that we are dealing with independent and identically distributed (i.i.d.[2]) time series.

When considering data from a single time series, the common approach is to continuously update a sequence of points with new data. The predictor then uses the most recent data points to make predictions and estimate uncertainty by adjusting its prediction interval. Notable developments in this area include the ACI technique (Gibbs and Candès, 2021), which is further refined in (Feldman et al., 2023). These conformal techniques update the prediction interval for the next discrete time step, ensuring asymptotic coverage. Under similar settings, (Chernozhukov et al., 2021) is able to approximate conditional validity and, in (Chernozhukov et al., 2018), the same author considers a prediction horizon larger than one and develops a randomization scheme to achieve approximate validity. Other work, such as EnbPI (Xu and Xie, 2021) and SPCI (Xu and Xie, 2023), also extend this setting. The recent study in (Auer et al., 2023) uses modern Hopfield networks to better handle temporal dependence in this context. However, these methods have a major limitation because they can only provide asymptotic guarantees.

The second scenario involves a dataset comprising multiple i.i.d. time series, where the goal is to perform multi-step forecasts. This approach was initially explored in CF-RNN (Stankeviciute et al., 2021). A significant limitation of their method is the dependence on the Bonferroni correction to provide guarantees over the entire prediction horizon. This becomes particularly problematic in time series with substantial temporal dependence. To address this, (Sun and Yu, 2023) attempts to model dependence using Copulas, although this introduces the requirement for an additional calibration dataset. (Lindemann et al., 2023) employs the same technique as CF-RNN, but integrates the forecasts with motion planning to create a guaranteed planning framework. Building upon (Lindemann et al., 2023), (Cleaveland et al., 2024) optimizes a single nonconformity score for the entire prediction horizon, an analogous idea to what (Diquigiovanni et al., 2021) uses in the context of functional data analysis, albeit at the expense of an extra calibration dataset. Similarly, (Lin et al., 2022) considers a conformal score that assigns different weights to time steps, but they only provide guarantees over average coverage, not joint coverage.

When considering multi-step joint coverage (formalized in Definition 1), to the best of our knowledge, the current literature lacks *efficient* methods (formalized in Definition 2).

2. i.i.d. implies *exchangeability*, for this reason, CP can be applied as well.

The current state-of-the-art tackles this problem by introducing an extra calibration step, which requires some optimization and more data, or by using inefficient approximations. We propose a novel approach that is *efficient*, does not require extra calibration data, and adds minimal computational overhead.

4. Approach

Our method, ConForME, which stands for **Con**formal **For**ecast for **M**ulti-horizon pr**E**diction can be seen as a generalization of CF-RNN. We solve the problem of lack of efficiency by dividing the prediction horizon into blocks, where we can compute more precise intervals. We also present a general formulation where the error rate can vary over the horizon for each block.

To lay the groundwork for our discussion, we first delve into the details of split conformal prediction, which forms the foundation of our guarantees. Subsequently, we introduce CF-RNN, the method against which we will benchmark our approach. Lastly, we elaborate on ConForME, providing insight into its validity proof and hyperparameter selection.

4.1. Split conformal prediction

Conformal prediction (CP) (Vovk et al., 2005) is a statistical framework that enables the construction of valid probabilistic prediction regions (following a similar definition of validity as Definition 1) for any given measurable point predictor. In general, it only requires *exchangeability* between a calibration dataset and the observed data, and *measurability* of the predictor.

We detail, here, the variant of CP called split conformal prediction (SCP) which was introduced as inductive conformal prediction alongside CP. Its main advantage over CP is the low computational cost, which is necessary when dealing with neural network predictors. The main idea of SCP is to split an available dataset $\mathcal{D}$ of observations and labels $(x_i, y_i) \in \mathcal{D}$ into two disjoint parts, the training data, and calibration data i.e. $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{cal}$. $\mathcal{D}_{train}$ is used to train the predictor g and $\mathcal{D}_{cal}$ is used to calibrate the conformal predictor $C_g^\alpha(x)$. The conformal predictor computes, for any observation x, a prediction region which contains the true label y with coverage probability of at least $1-\alpha$ (α is also called the error rate), as shown in Equation (2).

$$\mathbb{P}\left(y \in C_g^\alpha(x)\right) \geq 1-\alpha \tag{2}$$

More broadly, SCP guarantees Equation (2) for any measurable predictor g as long as $\mathcal{D}_{cal} \cup \{(x, y)\}$ is exchangeable (Definition 4). In our case, g corresponds to a single output f_i of a multi-horizon predictor neural network f; the observations x correspond to the past observations $(y_i)_{i=1}^{T-H}$ and y corresponds to one of the ground truth future values y_i with $i > T-H$.

To compute the prediction regions, as described in (Papadopoulos and Haralambous, 2011), it suffices to compute the residuals in Equation (3) and select the $r^* = \lceil (1-\alpha)(1+|\mathcal{D}_{cal}|) \rceil - th$ largest. Afterwards, r^* is used as the radius of the prediction region shown in Equation (4).

$$\mathcal{R} = \{||g(x_i) - y_i|| \mid (x_i, y_i) \in \mathcal{D}_{cal}\} \tag{3}$$

$$C_g^\alpha(x) = \{y \mid ||g(x) - y|| \leq r^*\} \tag{4}$$

The calibration dataset used to construct $C_g^\alpha(x)$ is left implicit in most cases. To make it explicit, we use the notation $C^\alpha_{g,\mathcal{D}_{cal}}(x)$, this notation is necessary when we condition x to belong to some subspace, that is: given $x \in X$, where X is some measurable space, restrict x to $X' \subset X$. In this case, the following holds:

$$\mathbb{P}\left(y \in C^\alpha_{g,\{(x,y)\in\mathcal{D}_{cal} | x\in X'\}}(x) \mid x \in X'\right) \geq 1 - \alpha \tag{5}$$

As an example, let us consider the case where we are given ground truth sequences of length $T = 3$, and a prediction horizon $H = 2$, therefore each sequence (y_i) has the form (y_1, y_2, y_3) and the predictor f is such that $f(y_1) = (\hat{y}_2, \hat{y}_3)$. In our example, the problem is to compute intervals $\hat{\mathbf{y}}_\mathbf{2}$ and $\hat{\mathbf{y}}_\mathbf{3}$ given a set of ground truth sequences $\mathcal{D}_{cal}$ and respecting the following condition:

$$\mathbb{P}(y_2 \in \hat{\mathbf{y}}_\mathbf{2} \cap y_3 \in \hat{\mathbf{y}}_\mathbf{3}) \geq 1 - \alpha$$

Our solution is to compute $\hat{\mathbf{y}}_\mathbf{2} = C^\gamma_{f_1,\mathcal{D}_{cal}}(y_1)$ and $\hat{\mathbf{y}}_\mathbf{3} = C^\gamma_{f_2,\mathcal{D}'_{cal}}(y_1)$ where $\mathcal{D}'_{cal} = \{(y_1, y_2, y_3) \in \mathcal{D}_{cal} \mid y_2 \in \hat{\mathbf{y}}_\mathbf{2}\}$. Split conformal prediction and Equation (5) give us, respectively:

$$\mathbb{P}(y_2 \in \hat{\mathbf{y}}_\mathbf{2}) \geq 1 - \gamma \text{ and } \mathbb{P}(y_3 \in \hat{\mathbf{y}}_\mathbf{3} \mid y_2 \in \hat{\mathbf{y}}_\mathbf{2}) \geq 1 - \gamma$$

By simply setting $(1-\gamma)^2 = 1-\alpha$ and using Bayes' theorem we get the desired probability $\mathbb{P}(y_2 \in \hat{\mathbf{y}}_\mathbf{2} \cap y_3 \in \hat{\mathbf{y}}_\mathbf{3}) \geq 1-\alpha$. The described formulation is precisely the base for the conditional part of our method. It is used in ConForME to compute intervals that respect conditional probabilities in the form $\mathbb{P}(y_i \in \hat{\mathbf{y}}_\mathbf{i} \mid y_{i-1} \in \hat{\mathbf{y}}_\mathbf{i-1}, \dots)$, which allows for an efficient coverage.

4.2. Conformal time series forecasting (CF-RNN)

Introduced in (Stankeviciute et al., 2021), CF-RNN extends split conformal prediction to time series for multi-horizon forecasting.

CF-RNN produces valid prediction intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^T$ given an error rate α. The method computes the prediction intervals as follows:

$$\hat{\mathbf{y}}_i = C_{f_i}^{\alpha/H}(y_1, \dots, y_{T-H}) \tag{6}$$

where f is an RNN in the same form as Equation (1) and $\hat{\mathbf{y}}_i$ is constructed using split conformal prediction on each $\hat{y}_i$ separately with an error rate of α/H. These prediction intervals ensure error rates of α/H for each prediction. It follows from Boole's inequality that the joint error rate is less than α.

The term α/H is called the Bonferroni correction. Note that this method relies on Boole's inequality which is very conservative if the events are not disjoint, which can lead to inefficient coverage.

4.3. ConForME: Proposed method

The general idea of the method is to divide the prediction horizon H into blocks B_j and use Bayes' theorem within each block to compute the coverage probability. The main motivation for this block partitioning is that, with the formulation in Equation (5), validity can be guaranteed without the need for Bonferroni's correction. This way, we reduce the amount of approximations which are made in the process of computing the intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T}$.

More formally, the output of our method are intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T}$ for each input sequence $(y_i)_{i=1}^{T-H}$. These H intervals are partitioned into k blocks B_j such that:

- Each block B_j has a size $b_j = |B_j|$.
- If all blocks are concatenated in increasing order, we obtain the prediction intervals $(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T}$, i.e.:

$$(\hat{\mathbf{y}}_\mathbf{i})_{i=T-H+1}^{T} = \Big(\underbrace{\hat{\mathbf{y}}_{\mathbf{T-H+1}} \cdots \hat{\mathbf{y}}_{\mathbf{T-H+b_1}}}_{B_1} \underbrace{\cdots\cdots}_{B_j} \underbrace{\hat{\mathbf{y}}_{\mathbf{T-b_k+1}} \cdots \hat{\mathbf{y}}_{\mathbf{T}}}_{B_k}\Big)$$

To lighten the index notation, $(l)^j$ is used to translate block indexing to whole sequence indexing, with $l = 1, \ldots, b_j$ and $j = 1, \ldots, k$. It is defined as follows:

$$(l)^j = \left(\sum_{m=1}^{j-1} b_m\right) + l + (T - H)$$

This way, for example, if we write $\hat{\mathbf{y}}_{\mathbf{(1)^2}}$ we refer to the first element of the second block.

The core of our method is to ensure validity within each block with an error rate α_j (Equation (7)) and then combine the blocks to ensure that the overall error rate is at most α. The process of combining the blocks relies on Bonferroni's correction, which means that we have the constraints described in below in Equation (8):

$$\mathbb{P}\left(\bigcup_{l=1}^{b_j} \left(y_{(l)^j} \notin \hat{\mathbf{y}}_{\mathbf{(l)^j}}\right)\right) \leq \alpha_j \qquad (7) \qquad\qquad \alpha_j > 0, \ \sum_{j=1}^{k} \alpha_j \leq \alpha \qquad (8)$$

Then, for each block B_j, the conformal intervals $\hat{\mathbf{y}}_{(l)^j}$ with $l = 1, \ldots, b_j$ are built to ensure an error rate of α_j for the block. The idea is to build the intervals $\hat{\mathbf{y}}_{(l)^j}$ such that $y_{(m)^j} \in \hat{\mathbf{y}}_{(m)^j}$ for all $0 < m < l$. So, for example, $\hat{\mathbf{y}}_{(3)^j}$ is built considering only the sequences that fell inside both $\hat{\mathbf{y}}_{(2)^j}$ and $\hat{\mathbf{y}}_{(1)^j}$. We can translate that to the following expression:

$$\hat{\mathbf{y}}_{(l)^j} = C^{\alpha_j^l}_{f_{(l)^j}, \mathcal{D}_{cal}^{(l)^j}}(y_1, \ldots, y_{T-H}) \qquad (9)$$

$$\text{where } \mathcal{D}_{cal}^{(l)^j} = \{(y_i) \in \mathcal{D}_{cal} \mid y_{(m)^j} \in \hat{\mathbf{y}}_{\mathbf{(m)^j}} \forall m \in ((1)^j, \ldots, (l)^j - 1)\}$$

This means that $\hat{\mathbf{y}}_{(l)^j}$ is the interval given by a conformal predictor built with an error rate of α_j^l and calibrated only on sequences that fell within the previous intervals of the same block.

This conditional formulation, alongside the identity $\mathbb{P}(\bigcup E_i) = \mathbb{P}(1 - \bigcap \overline{E_i})$, allows the computation of the error rate α_j of the block B_j as:

$$1 - \prod_{l=1}^{b_j}(1 - \alpha_j^l) = \alpha_j \tag{10}$$

Note that no additional approximation is made with respect to the individual conformal predictors. The inequality $(1-a)(1-b) \geq 1 - a - b$ for $a, b \geq 0$ allows us to write:

$$\sum_{l=1}^{b_j} \alpha_j^l \geq \alpha_j \tag{11}$$

We note that the approximation error in the approximation used above tends to zero for small α_j^l. Finally, the constraints in Equation (8) can be rewritten as function of the individual error rates α_j^l :

$$\sum_{j=1}^{k} \sum_{l=1}^{b_j} \alpha_j^l \leq \alpha \tag{12}$$

These building blocks are put together in detail in Algorithm 1 below:

Algorithm 1: ConForME intervals computation

`/* Calibration sequences, predictor function, sequences length, prediction horizon, block sizes, and individual error rates */`

Input: $\mathcal{D}_{cal}$, f, T, H, $(b_j)_{j=1}^k$, α_j^l for $l \in \{1, \ldots, b_j\}$ and $j \in \{1, \ldots, k\}$

Output: Calibrated conformal predictors $C^{\alpha_j^l}_{f_{(l)^j}, \mathcal{D}_{cal}^{(l)^j}}$

$\mathcal{C} \leftarrow \{\}$
for $j \leftarrow 1$ **to** k **do**
  $\mathcal{D}'_{cal} \leftarrow \mathcal{D}_{cal}$
  for $l \leftarrow 1$ **to** b_j **do**
    if $l > 1$ **then**
      $\mathcal{D}'_{cal} \leftarrow \{(y_i) \in \mathcal{D}'_{cal} \mid y_{(l-1)^j} \in C^{\alpha_j^{l-1}}_{f_{(l-1)^j}, \mathcal{D}_{cal}^{(l-1)^j}}(y_1, \ldots, y_{T-H})\}$
    end
    $C^{\alpha_j^l}_{f_{(l)^j}, \mathcal{D}_{cal}^{(l)^j}} \leftarrow C^{\alpha_j^l}_{f_{(l)^j}, \mathcal{D}'_{cal}}$ ▷`As described in Section 4.1`
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{C^{\alpha_j^l}_{f_{(l)^j}, \mathcal{D}_{cal}^{(l)^j}}\}$
  end
end
return $\mathcal{C}$

The algorithm takes as inputs a set of calibration sequences (dataset) $\mathcal{D}_{cal}$, a predictor function f, block sizes $(b_j)_{j=1}^k$, and all α_j^l already chosen, respecting Equation (12). For every block (Line 2), for every interval in a block (Line 4), it filters the calibration dataset $\mathcal{D}_{cal}$ in Line 6 to produce a filtered calibration dataset. The filtering is made such that the sequences that fell within the intervals given by the previous conformal predictors of the same block are the only ones that remain. In Line 8, it uses only the filtered calibration dataset to compute the conformal predictor. This process implements what is described in Equation (9). Finally, in Line 9 the computed conformal predictor is added to the set of conformal predictors and the whole set is returned in Line 12.

4.4. Proof of validity

In this section, we provide the supporting arguments to show that, for any choice of positive α_j^l that satisfies Equation (12), the ConForME intervals $\hat{\mathbf{y}}_{(l)^j}$ built as in Equation (9) are valid as defined in Definition 1.

First, if we have block validity as in Equation (7), by the definition of the block and Boole's inequality, we have:

$$\mathbb{P}\left(\underbrace{\bigcup_{i=T-H+1}^{T}(y_i \notin \hat{\mathbf{y}}_\mathbf{i})}_{validity}\right) = \mathbb{P}\left(\underbrace{\bigcup_{j=1}^{k}\bigcup_{l=1}^{b_j}\left(y_{(l)^j} \notin \hat{\mathbf{y}}_{(\mathbf{l})^\mathbf{j}}\right)}_{block\ validity}\right) \leq \sum_{j=1}^{k}\alpha_j$$

This way, with the constraint in Equation (8), we only need to show block validity. To show that, let $E_{(l)^j}$ be the random event $y_{(l)^j} \notin \hat{\mathbf{y}}_{(\mathbf{l})^\mathbf{j}}$ and let $\overline{E_{(l)^j}}$ be its complement. The conformal predictor which is used to build $\hat{\mathbf{y}}_{(l)^j}$ provides the following guarantee:

$$\mathbb{P}\left(\overline{E_{(l)^j}} \mid \overline{E_{(l-1)^j}}, \ldots, \overline{E_{(1)^j}}\right) \geq 1 - \alpha_j^l$$

By applying Bayes' theorem, we arrive at:

$$\mathbb{P}\left(\bigcap_{l=1}^{b_j}\overline{E_{(l)^j}}\right) = \prod_{l=1}^{b_j}\mathbb{P}\left(\overline{E_{(l)^j}} \mid \overline{E_{(l-1)^j}}, \ldots, \overline{E_{(1)^j}}\right) \geq \prod_{l=1}^{b_j}(1-\alpha_j^l)$$

$$\Rightarrow \mathbb{P}\left(\bigcup_{l=1}^{b_j}\left(y_{(l)^j} \notin \hat{\mathbf{y}}_{(\mathbf{l})^\mathbf{j}}\right)\right) \leq 1 - \prod_{l=1}^{b_j}(1-\alpha_j^l) = \alpha_j \leq \sum_{l=1}^{b_j}\alpha_j^l$$

Then, it follows directly that Equation (12) implies validity as in Definition 1.

4.5. Hyperparameter selection

Algorithm 1 produces conformal predictors that are capable of computing valid prediction intervals for any selection of α_j^l satisfying Equation (12).

Searching for the optimal α_j^l, however, is costly. Therefore, for our experiments, we use the natural choice of evenly distributed error rates as detailed in Section 4.5.1. We also discuss two other possible hyperparameter choices that have been implemented.

4.5.1. Evenly distributed error rates

This is the natural approach used in our experiments. It consists on choosing α_j^l as:

$$\alpha_j^l = \frac{\alpha}{H} \tag{13}$$

This choice has proven to achieve good results, has low computational cost, and also allows us to better measure the impact of the number of blocks k as it compares directly with CF-RNN (for evenly distributed α_j^l, CF-RNN is equivalent to ConForME when $k = H$). Regarding block sizes b_j, they are also distributed evenly over the horizon.

4.5.2. Pairwise evenly distributed error rates

Another possible choice is to group the prediction horizon in pairs and balance the error rates between the two elements of the pair with a parameter β. This is done by setting $k = \lceil H/2 \rceil$, $b_j = 2$ for all $j < k$ and, for $j = k$, setting b_k to either 1 or 2 depending on H. Then, by selecting α_j^l as follows:

$$\alpha_j^l = \begin{cases} \frac{\alpha}{\lceil H/2 \rceil} & \text{if } j = k \text{ and } b_k = 1 \\ \frac{1-\beta}{1-\beta\frac{\alpha}{\lceil H/2 \rceil}}\frac{\alpha}{\lceil H/2 \rceil} & \text{if } l \text{ is } 2 \\ \beta\frac{\alpha}{\lceil H/2 \rceil} & \text{else} \end{cases} \tag{14}$$

Here, β is a new hyperparameter. The interest of this approach is that we found in our test cases that if β is optimized by using binary search, the results are close to the true optimal β. This way, we get a direct improvement over Section 4.5.1 with only a small added computational cost.

4.5.3. Globally optimized error rates

For a given number of blocks k with evenly distributed sizes b_j, this choice consists on using stochastic gradient descent's (SGD) implementation from Pytorch (Paszke et al., 2019) to optimize the mean size of the intervals generated by the conformal predictors that are given as output of Algorithm 1. The loss function is defined below:

$$loss = mean\_size\left(conforme\left(\left(\alpha_j^l\right)\right), k, \left(b_j\right)\right) + \lambda \cdot tan\left(\sum_{j=1}^{k}\sum_{l=1}^{b_j}\alpha_j^l - \alpha\right) \tag{15}$$

$conforme$ is Algorithm 1 and its other inputs are omitted for simplicity. (b_j), k are fixed and λ is a hyperparameter tuned manually, so that the second term is small. The idea is that we minimize via SGD the mean size of the conformal intervals, while ensuring that the target coverage is close to $1-\alpha$ (coverage term weighted by the parameter λ). The tangent function is used to heavily penalize high deviations from the target error rate.

5. Experiments and discussion

We validate the empirical performance of our approach by comparing it to CF-RNN on four datasets, each of which is intended to highlight a different aspect of our method.

The comparison is made against CF-RNN because it is the only approach in the literature that shares the same settings. Other methods either lack the joint coverage guarantee in Definition 1 or require an additional optimization step as well as more data.

We start in Section 5.1 by describing how we set up the experiments, then we give an overview of our results in Section 5.2. We end up in Section 5.3 by looking more closely at each single example we used.

5.1. Setting up the experiments

The four datasets presented in this section are: synthetic 1-D data, 1-D data composed of EEG (electroencephalogram) scans, 2-D trajectory data of autonomous vehicles from the Argoverse (Chang et al., 2019) dataset, and finally real-world 1-D data from a small dataset following COVID-19 cases. These are fully described in Sections 5.3.1, 5.3.2, 5.3.3 and 5.3.4 respectively.

All experiments were performed on a laptop equipped with an Intel Core i9-12900H CPU, 32 GB of RAM, and a single NVIDIA RTX A2000 GPU. To deal with 2-D data, we compute the distances between the predicted positions and the true positions, which allows us to compute prediction intervals with respect to these distances as usual. For the Argoverse results, therefore, instead of interval widths, we report the area corresponding to circles centered in the prediction with radius equal to the widths of the prediction intervals. A similar approach is used in (Lindemann et al., 2023), we do not discuss this case further as it is not the main focus of the article. The reader can refer to our code available at `https://github.com/aloysiogl/conforme` for more details.

We compare the mean size of the predicted intervals from ConForME to CF-RNN. We also experimentally verify the validity property on the proposed datasets by testing the empirical joint coverage. That is, we divide each dataset into three disjoint parts: training, calibration, and test ($\mathcal{D}_{train}$, $\mathcal{D}_{cal}$, and $\mathcal{D}_{test}$). We repeat the experiment 5 times with different random seeds and present the mean and standard deviation of the results in the tables. General information about each dataset is summarized in Table 1:

Dataset	$\lvert\mathcal{D}_{train}\rvert$	$\lvert\mathcal{D}_{cal}\rvert$	$\lvert\mathcal{D}_{test}\rvert$	Input size $T-H$	Horizon H	$1-\alpha$ (%)
Synthetic	1000	1000	500	15	10	90
EEG_{10}	15360	3840	19200	40	10	90
EEG_{40}	15360	3840	19200	40	40	90
Argoverse	208272	5210	5211	30	20	90
COVID-19	200	100	80	100	50	30

Table 1: Summary of datasets, where $1-\alpha$ is the target coverage. Note that the EEG dataset has two variants corresponding to different prediction horizons.

In the first three experiments, namely the synthetic 1-D data, the EEG data, and the COVID-19 data, we use the same data preparation and underlying predictor as in CF-RNN: a vanilla RNN. For the Argoverse data, we follow (Sun and Yu, 2023) using LaneGCN (Liang et al., 2020) as the underlying predictor.

5.2. Main findings

The main results we obtained are summarized in Tables 2 and 3. The first table shows the sizes of the prediction zones and the coverage, while the second reports the runtime of our experiments. In these tables, we only consider the natural hyperparameter choice of uniformly distributed error rates (see Section 4.5), and ConForME$_l$ denotes a choice of number of blocks $k = l$. We measure the empirical joint coverage to evaluate the validity condition using the following metric:

$$\text{Empirical coverage} = \frac{1}{|\mathcal{D}_{test}|} \sum_{(y_i)_{i=1}^{T} \in \mathcal{D}_{test}} \left(\mathbf{1}_{y_i \in \hat{\mathbf{y}}_\mathbf{i}} : i \in \{T - H + 1, \ldots, T\} \right) \tag{16}$$

The first dataset emphasizes the coverage guarantee: as the dataset is synthetic, we can guarantee exchangeability. The second dataset shows the performance gain on real data with more temporal dependence. The third demonstrates the performance of the method on 2-D data and reinforces the claims made with the second dataset. Finally, the fourth dataset shows the behavior of the method when the data is scarce.

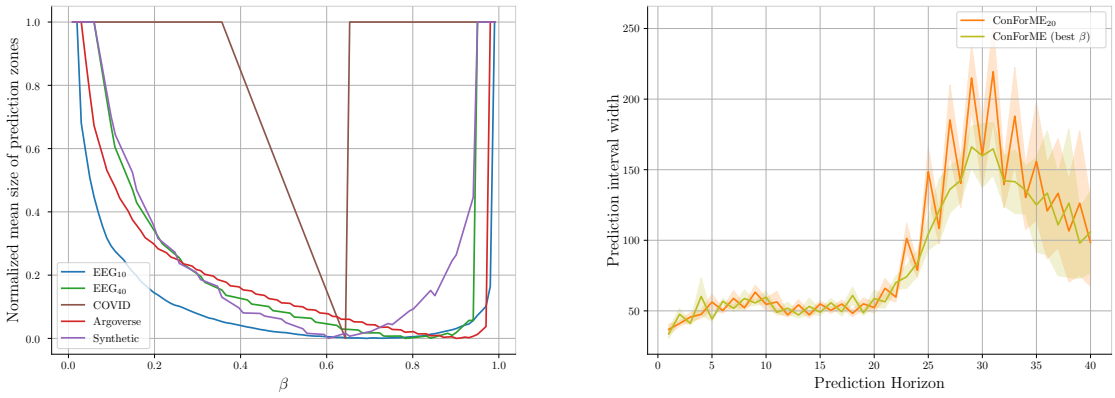
Besides the natural hyperparameter selection, we discussed two other options in Section 4.5: pairwise evenly distributed error rates and global optimization with SGD. The second option did not show significant improvements in our experiments, it also requires significantly more computation time during calibration. The first alternative, however, showed promising results.

We can see from Figure 1(a) that, for all experiments, expressing the intervals sizes in function of β (defined in Section 4.5.2) gives a function that is ϵ-close to a convex function for every β. This implies that the optimal β can be found in these examples by binary search with high precision, making this optimization very inexpensive. We believe that this is a general claim given some mild assumptions about the predictor, but we have not been able to prove it for the moment. Moreover, with this optimization, the interval widths oscillate significantly less with the horizon as it can be seen in Figure 1(b). In the case of the EEG$_{40}$ dataset, the interval sizes are 19% smaller than the baseline, as opposed to 13% for the corresponding natural hyperparameter choice (ConForME$_{20}$).

For calibration, the time complexity of our method is the same as CF-RNN in a sequential execution. Our calibration pass is equivalent to one predictor model evaluation on the calibration data plus a per block calibration pass. CF-RNN can be parallelized for each horizon, while we can only do it in blocks. It is, however, important to notice that the time complexity is the same during test time, because we only need to store interval sizes after the calibration for both methods, regardless of the number of blocks.

Analyzing the results of table Table 2 we clearly see a trend towards better results with fewer blocks. This is due to the fact that the more blocks we have, the more we rely on Bonferroni's approximation. We also see that the coverage tends to get more exact when we have fewer blocks. In this sense, it is recommended to use as fewer blocks as possible. However, when exchangeability cannot be guaranteed or when there is not enough data, we can get insufficient empirical coverages. In such cases, it might be better to use a more conservative approach, choosing a number of blocks greater than one.

Considering now the results of Table 3, we can see that the test time is the same for both ConForME and CF-RNN, as expected. We also see no significant difference between the run



(a) (b)

Figure 1: (a) shows the normalized mean interval widths/areas for each dataset when β ranges from 0 to 1 in steps of 0.01. When the intervals are infinite, their normalized width is displayed as 1. (b) shows the mean interval widths per horizon for the optimal β compared with ConForME$_{20}$ on the EEG$_{40}$ dataset. The standard deviations are represented with the semi-transparent zones.

times of ConForME with one block and ConForME with a number of blocks equal to the horizon during calibration. There is a difference during calibration between ConForME and CF-RNN, due to some constant time overhead. We remind the reader that ConForME with number of blocks equal to the horizon is equivalent to CF-RNN, the observed difference comes from the fact that ConForME is implemented for any number of blocks, therefore it must spend more time initializing code structures. Finally, we decided to omit the results for some intermediate number of blocks in Table 3 for the ConForME method, as they do not provide any additional insight.

5.3. The examples, in detail

5.3.1. Synthetic data

Following CF-RNN (Stankeviciute et al., 2021), for the synthetic dataset, we generate points following a Gaussian with a memory parameter a. This memory creates a dependence in time between the values y_t. An extra normally distributed noise is added, which overall corresponds to Equation (17):

$$y_t = \sum_{k=1}^{t} a^{t-k} x_k + \epsilon_t, \forall t \in \{1, \dots, T\} \tag{17}$$

where $x_k \sim \mathcal{N}(\mu_x, \sigma_x^2)$, $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ and $a = 0.9$. We consider only the case where $\sigma_t^2 = 0.1$, $\mu_x = 1$ and $\sigma_x^2 = 4$. CF-RNN considers time-varying noise as well, but we consider that these experiments bring no additional insight to the comparison. It is, however, possible to replicate them as well with our code. Sequence lengths and generated dataset sizes are shown in Table 1.

Dataset	Method	Zone width/area			Coverage (%)
		Mean (% reduction)	Min	Max	
Synthetic	CF-RNN	21.4 ± 0.8 (0.0)	**11.8 ± 0.5**	28.6 ± 1.5	94.8 ± 1.4
	ConFormE_{10}	21.4 ± 0.8 (0.0)	**11.8 ± 0.5**	28.6 ± 1.5	94.8 ± 1.4
	ConFormE_{5}	20.9 ± 0.6 (2.5)	**11.8 ± 0.5**	27.4 ± 1.0	94.2 ± 1.3
	ConFormE_{3}	20.7 ± 0.6 (3.3)	**11.8 ± 0.5**	27.4 ± 1.0	94.0 ± 1.2
	ConFormE_{2}	20.1 ± 0.5 (5.9)	**11.8 ± 0.5**	26.0 ± 1.2	92.7 ± 0.9
	ConFormE_{1}	**19.3 ± 0.5 (9.7)**	**11.8 ± 0.5**	**23.7 ± 0.9**	**91.0 ± 0.8**
EEG_{10}	CFRNN	58.3 ± 1.6 (0.0)	**30.7 ± 1.3**	75.1 ± 1.9	96.4 ± 0.2
	ConFormE_{10}	58.3 ± 1.6 (0.0)	**30.7 ± 1.3**	75.1 ± 1.9	96.4 ± 0.2
	ConFormE_{5}	51.3 ± 0.7 (12)	**30.7 ± 1.3**	75.1 ± 2.8	94.4 ± 0.3
	ConFormE_{3}	49.1 ± 1.1 (15)	**30.7 ± 1.3**	75.1 ± 2.8	93.4 ± 0.2
	ConFormE_{2}	43.8 ± 0.7 (25)	**30.7 ± 1.3**	74.7 ± 2.6	**91.5 ± 0.5**
	ConFormE_{1}	**38.1 ± 0.8 (35)**	**30.7 ± 1.3**	**42.4 ± 1.2**	89.1 ± 0.6
EEG_{40}	CF-RNN	106 ± 14.8 (0.0)	**34.1 ± 3.8**	269 ± 34	96.4 ± 0.6
	ConFormE_{40}	106 ± 14.8 (0.0)	**34.1 ± 3.8**	269 ± 34	96.4 ± 0.6
	ConFormE_{20}	92.7 ± 10.9 (13)	**34.1 ± 3.0**	264 ± 31	95.1 ± 0.4
	ConFormE_{8}	74.7 ± 6.6 (30)	**34.1 ± 3.0**	264 ± 35	93.2 ± 0.5
	ConFormE_{4}	63.4 ± 3.4 (40)	**34.1 ± 3.0**	264 ± 35	91.5 ± 0.6
	ConFormE_{2}	53.8 ± 1.9 (49)	**34.1 ± 3.0**	79.0 ± 6.5	**89.8 ± 0.6**
	ConFormE_{1}	**50.9 ± 1.5 (52)**	**34.1 ± 3.0**	**77.8 ± 7.8**	88.7 ± 0.6
Argoverse	CFRNN	4423 ± 122 (0.0)	122 ± 2.3	13875 ± 436	98.5 ± 0.2
	ConFormE_{30}	4423 ± 122 (0.0)	122 ± 2.3	13875 ± 436	98.5 ± 0.2
	ConFormE_{15}	4159 ± 111 (6.0)	**106 ± 1.7**	12546 ± 290	97.8 ± 0.3
	ConFormE_{10}	3954 ± 90.9 (11)	**106 ± 1.7**	11560 ± 247	97.4 ± 0.3
	ConFormE_{3}	3224 ± 63.2 (27)	**106 ± 1.7**	8402.8 ± 238	94.8 ± 0.5
	ConFormE_{1}	**2364 ± 29.9 (47)**	**106 ± 1.7**	**6101.1 ± 65.5**	**90.4 ± 0.5**
COVID	CFRNN	631 ± 253 (0.0)	90.2 ± 75	2341 ± 477	88.5 ± 5.8
	ConFormE_{50}	631 ± 253 (0.0)	90.2 ± 75	2341 ± 477	88.5 ± 5.8
	ConFormE_{25}	570 ± 167 (9.8)	85.9 ± 42	2341 ± 480	85.7 ± 6.3
	ConFormE_{10}	492 ± 100 (22)	90.2 ± 34	2341 ± 477	82.3 ± 5.4
	ConFormE_{5}	427 ± 76.4 (32)	69.9 ± 32	2341 ± 552	76.7 ± 5.8
	ConFormE_{2}	**336 ± 48.5 (47)**	**58.8 ± 5.8**	**1793 ± 336**	**67.2 ± 9.5**
	ConFormE_{1}	$\infty \pm \infty (-\infty)$	108 ± 24	$\infty \pm \infty$	71.0 ± 5.0

Table 2: Summary of coverage and prediction zone sizes for all datasets with empirical means and standard deviations reported over 5 experiments with different random seeds. In the case of width/area, the minimum (best) is highlighted in bold. Empirical joint coverage is reported and the closest valid result to the target coverage rate $1 - \alpha$ is highlighted in bold.

Dataset	Method	Calibration time (ms) Mean	Min	Max	Test time (ms) Mean	Min	Max
Synthetic	CF-RNN	**1.41 ± 0.53**	**1.13**	**2.36**	6.93 ± 6.3	3.98	18.1
	ConFormE_{10}	6.62 ± 0.50	6.04	7.35	4.09 ± 0.16	**3.96**	4.29
	ConFormE_{1}	5.10 ± 0.19	4.94	5.43	**4.05 ± 0.03**	4.01	**4.08**
EEG_{10}	CF-RNN	**3.73 ± 0.2**	**3.54**	**4.10**	**213 ± 5.6**	208	222
	ConFormE_{10}	15.5 ± 8.6	9.96	30.6	**213 ± 3.8**	**206**	**215**
	ConFormE_{1}	9.58 ± 1.0	8.82	11.4	217 ± 6.2	208	224
EEG_{40}	CFRNN	**9.09 ± 1.0**	**8.00**	**10.3**	223 ± 9.7	214	238
	ConFormE_{40}	45.8 ± 22	33.6	85.4	**214 ± 2.5**	211	**218**
	ConFormE_{4}	32.8 ± 2.4	30.4	36.2	**214 ± 6.1**	**207**	222
	ConFormE_{1}	34.7 ± 4.0	30.8	41.0	231 ± 19	213	255
Argoverse	CFRNN	**12.0 ± 1.3**	**10.7**	**14.3**	3.38 ± 0.2	3.09	3.65
	ConFormE_{30}	35.7 ± 12	29.1	57.4	**2.93 ± 0.9**	**2.16**	4.45
	ConFormE_{3}	44.7 ± 13	30.5	56.3	3.01 ± 0.3	2.54	3.38
	ConFormE_{1}	43.0 ± 20	27.3	78.4	2.94 ± 0.4	2.25	**3.31**
COVID	CFRNN	**0.61 ± 0.017**	**0.60**	**0.64**	1.90 ± 0.06	1.81	1.98
	ConFormE_{50}	22.3 ± 4.2	20.2	29.89	**1.28 ± 0.02**	**1.26**	**1.30**
	ConFormE_{4}	15.6 ± 2.1	14.5	19.37	1.29 ± 0.02	1.27	1.32
	ConFormE_{1}	7.06 ± 0.83	6.50	8.52	1.32 ± 0.04	1.28	1.37

Table 3: Summary of observed runtimes for CFRNN and ConForME for all datasets and the most relevant number of blocks for ConForME. We report calibration and test runtimes. Best results are highlighted in bold.

We expect this dataset to obey the validity condition as we can assure exchangeability of the synthetically generated data. The results in Table 2 support this claim as the measured coverage is always above the target coverage of 90%. As expected, when the number of blocks decreases (larger blocks), the Bonferroni approximation is used less often and the coverage gets closer to the target coverage, while the interval sizes reduce monotonically with the number of blocks k. For $k = H$, as expected, ConForME is equivalent to CF-RNN.

5.3.2. EEG data

The EEG data is taken from (Begleiter, 1999) and follow the same preparation as in *CF-RNN*. This dataset was generated by recording EEG signals from healthy patients subject to different visual stimuli.

It is composed of two halves with 19200 sequences of size 256. The underlying RNN model is trained on 15360 sequences and calibrated on the remaining sequences of the first half. The second half is used for testing. To achieve the sizes that are shown in Table 1, we downsample the sequences.

We present two variants, one where the sequences are downsampled to a prediction horizon of 10 and another where the prediction horizon is 40. Additionally, we present the interval sizes per horizon, per method, in Figure 2(a). We can see that the bigger prediction horizon favors ConForME, as the gains in interval size are more significant for the latter intervals of a block. Effectively, with the default choice of hyperparameters, the first interval of each block has, by definition, the same size of CF-RNN's interval, as can be seen in the Figure 2(a). We also highlight that our method leverages the conditional dependence of the predictions, for this reason, the EEG data which should have a higher time dependence, has shown significant improvements. Finally, for the horizon of 40 and only one block, we have a coverage which is slightly lower on average than the target coverage of 90%; this could be explained by the lack of exchangeability between the two halves of the dataset. Since no approximations are used when the number of blocks is one, we could not expect exact coverage if the exchangeability assumption is not fully met. Of course, this is a fundamental limitation of any method based on conformal prediction, and to mitigate it one should verify this assumption or possibly choose a more conservative coverage rate.

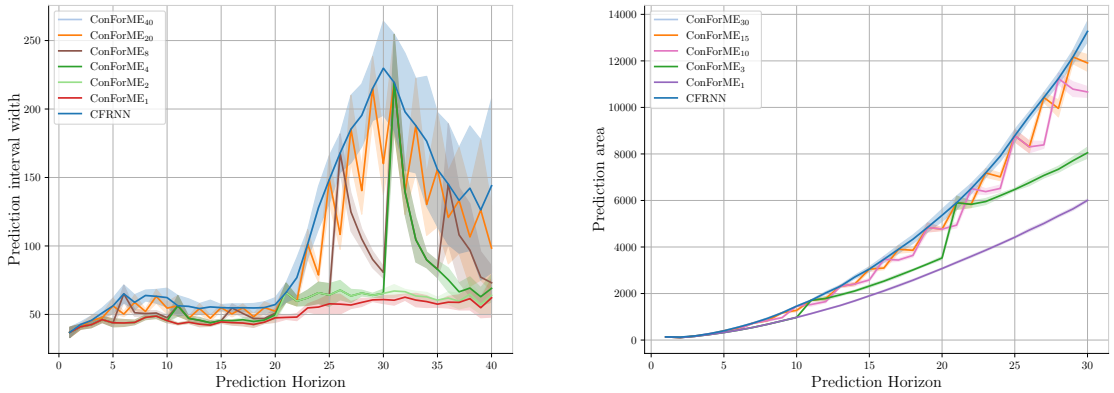


Figure 2: Interval widths/areas for both EEG$_{40}$ (a) and Argoverse (b). The standard deviations are represented with semi-transparent zones. Same colors represent equal block sizes. In (a) ConForME$_{40}$ and CF-RNN are equivalent, and in (b) ConForME$_{30}$ and CF-RNN are equivalent.

5.3.3. Argoverse

We take data from the version 1 of the Argoverse dataset (Chang et al., 2019), which is composed of 327790 sequences. Each sequence has size 50, which corresponds to 5 seconds of data sampled at 10Hz. 30 past points are used to predict 20 future points. The pre-trained model LaneGCN (Liang et al., 2020) is used, which utilizes 208272 sequences for training. The validation data, consisting of 40127 sequences, is split for calibration and testing. Sequences where the prediction gave invalid points are filtered out, totaling 5210 sequences for calibration and 5211 sequences for testing.

The overall performance improvement is comparable to that seen in the EEG data with horizon 40. The main difference can be seen in Figure 2, where we note that for the EEG dataset there is a peak just after the middle of the prediction horizon, while for Argoverse the largest area is present towards the end. This can be explained by the fact that the prediction of trajectories is more certain for the near future, since the movement of the vehicles is dictated by their dynamics, whereas brainwaves are not subject to these dynamics.

5.3.4. COVID-19 data

Our COVID-19 dataset comes from (UK Health Security Agency, 2021), which tracks the number of infected people in 380 regions over the course of 150 days. The dataset is split into 200 training sequences, 100 calibration sequences, and 80 test sequences. The prediction horizon is 50 days, based on the previous 100 days.

In this case, as noted in (Lin et al., 2022), we found that there was not enough data to calibrate the conformal predictors. In general, if $\alpha < 1/(n+1)$, where n is the number of points used to calibrate a conformal interval, the size of the conformal intervals should be infinite to ensure validity. For this reason, ConForME and CF-RNN are not applicable with a coverage rate of 90% for the COVID dataset. Even with only 30% coverage, as seen in Table 2, some intervals for ConForME with a single block still have infinite width. This shows an important limitation of both our method and CF-RNN: small calibration dataset sizes (in this case 100 samples) lead intervals of infinite size. In practice, the data needed to achieve finite size intervals during calibration is often a small fraction of the training data. It is also not realistic to train a complex predictor with only 200 sequences, as it is done here.

6. Conclusions

We presented in this paper ConForME, a low computational cost method for computing valid and efficient prediction intervals for multi-horizon time series forecasting. We demonstrated its validity and experimentally compared its efficiency with CF-RNN, showing that its natural hyperparameter choice outperforms CF-RNN by a significant margin in all cases, achieving up to 52% improvement in one dataset and, in all real-world datasets, achieving at least an improvement of 35%.

As future work, we will explore different hyperparameter choices that could potentially lead to greater efficiency gains, starting from the ones mentioned herein. In particular, it is an open question to find low computational cost alternatives to our hyperparameter choice of globally optimized error rates. Finally, we plan to integrate ConForME with a planner for safe autonomous vehicle navigation, and evaluate its real-time performance.

Acknowledgments

This work has been partially funded by Agence de l'Innovation de Défense – AID - via Centre Interdisciplinaire d'Études pour la Défense et la Sécurité – CIEDS - (project 2021 - FARO).

References

Ahmed Alaa and Mihaela Van Der Schaar. Frequentist uncertainty in recurrent neural networks via blockwise influence functions. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 175–190. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/alaa20b.html.

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael I. Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=eNdiU_DbM9.

Andreas Auer, Martin Gauch, Daniel Klotz, and Sepp Hochreiter. Conformal prediction for time series with modern hopfield networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/aef75887979ae1287b5deb54a1e3cbda-Abstract-Conference.html.

Henri Begleiter. EEG Database. UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C5TS3D.

Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8748–8757. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00895. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Chang_Argoverse_3D_Tracking_and_Forecasting_With_Rich_Maps_CVPR_2019_paper.html.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 732–749. PMLR, 2018. URL http://proceedings.mlr.press/v75/chernozhukov18a.html.

Victor Chernozhukov, Kaspar Wüthrich, and Yinchu Zhu. Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118, 2021.

Matthew Cleaveland, Insup Lee, George J. Pappas, and Lars Lindemann. Conformal prediction regions for time series using linear complementarity programming. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative*

Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, pages 20984–20992. AAAI Press, 2024. doi: 10.1609/AAAI.V38I19.30089. URL https://doi.org/10.1609/aaai.v38i19.30089.

Jacopo Diquigiovanni, Matteo Fontana, and Simone Vantini. The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data. *arXiv preprint arXiv:2102.06746*, 2021.

Shai Feldman, Liran Ringel, Stephen Bates, and Yaniv Romano. Achieving risk control in online learning settings. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=5Y04GWvoJu.

Meire Fortunato, Charles Blundell, and Oriol Vinyals. Bayesian recurrent neural networks. *CoRR*, abs/1704.02798, 2017. URL http://arxiv.org/abs/1704.02798.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Isaac Gibbs and Emmanuel J. Candès. Adaptive conformal inference under distribution shift. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1660–1672, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, pages 541–556. Springer, 2020. doi: 10.1007/978-3-030-58536-5_32. URL https://doi.org/10.1007/978-3-030-58536-5_32.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Conformal prediction intervals with temporal dependence. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=8QoxXTDcsH.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics Autom. Lett.*, 8(8): 5116–5123, 2023. doi: 10.1109/LRA.2023.3292071. URL https://doi.org/10.1109/LRA.2023.3292071.

Harris Papadopoulos and Haris Haralambous. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842–851, 2011. ISSN 0893-6080. doi: 10.1016/j.neunet.2011.05.008. URL https://www.sciencedirect.com/science/article/pii/S089360801100150X.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf.

Yaniv Romano, Matteo Sesia, and Emmanuel Candes. Classification with valid and adaptive coverage. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/244edd7e85dc81602b7615cd705545f5-Paper.pdf.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVIII*, volume 12363 of *Lecture Notes in Computer Science*, pages 683–700. Springer, 2020. doi: 10.1007/978-3-030-58523-5_40. URL https://doi.org/10.1007/978-3-030-58523-5_40.

Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. *CoRR*, abs/1912.05911, 2019. URL http://arxiv.org/abs/1912.05911.

Kamile Stankeviciute, Ahmed Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6216–6228. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/312f1ba2a72318edaaa995a67835fad5-Paper.pdf.

Sophia Huiwen Sun and Rose Yu. Copula conformal prediction for multi-step time series prediction. In *The Twelfth International Conference on Learning Representations*, 2023.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.

UK Health Security Agency. Number of COVID-19 cases in the Low Tier Local Authorities area between 2020-04-22 and 2021-05-24, 2021. URL https://ukhsa-dashboard.data.gov.uk/topics/covid-19?areaType=Lower+Tier+Local+Authority. Accessed: 2024-04-11.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In Steven C. H. Hoi and Wray Buntine, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pages 475–490, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR. URL https://proceedings.mlr.press/v25/vovk12.html.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv: Machine Learning*, 2017.

Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. PMLR, 2021. URL http://proceedings.mlr.press/v139/xu21h.html.

Chen Xu and Yao Xie. Sequential predictive conformal inference for time series. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38707–38727. PMLR, 2023. URL https://proceedings.mlr.press/v202/xu23r.html.