Conformal Prediction Interval for Dynamic Time-Series

Chen Xu [1] **Yao Xie** [1]

Abstract

We develop a method to construct distribution-free prediction intervals for dynamic time-series, called `EnbPI` that wraps around any bootstrap ensemble estimator to construct sequential prediction intervals. `EnbPI` is closely related to the conformal prediction (CP) framework but does not require data exchangeability. Theoretically, these intervals attain finite-sample, *approximately valid* marginal coverage for broad classes of regression functions and time-series with strongly mixing stochastic errors. Computationally, `EnbPI` avoids overfitting and requires neither data-splitting nor training multiple ensemble estimators; it efficiently aggregates bootstrap estimators that have been trained. In general, `EnbPI` is easy to implement, scalable to producing arbitrarily many prediction intervals sequentially, and well-suited to a wide range of regression functions. We perform extensive real-data analyses to demonstrate its effectiveness.

1. Introduction

In many modern applications, including energy and supply chains (Cochran et al., 2015), we need sequential prediction with uncertainty quantification for dynamic time-series observations that are spatially and temporally correlated. Time-series are dynamic as they can be non-stationary with highly complex spatio-temporal dependency. The uncertainty quantification is often in the form of prediction intervals, whose constriction is a fundamental problem in statistics and machine learning. For example, to incorporate renewable energy into existing power systems, it is crucial to accurately predict energy levels from wind farms and solar roof panels using data collected from solar sensors or wind turbines and construct prediction intervals.

However, this task is highly challenging, especially for dynamic time series. For instance, as outlined in the NERL report (Cochran et al., 2015), solar and wind power generation data are often non-stationary and highly stochastic, with significant variations and spatial-temporal correlations across different regions. To tackle such problems, we often use regression functions for prediction, such as random forest (Breiman, 2001) and various deep neural network structures (Lathuilière et al., 2019), which can be arbitrarily complex. Multiple regression functions are often combined into an ensemble estimator to increase accuracy and decrease variance (Breiman, 1996). However, after making predictions, existing theories and methods usually do not efficiently construct prediction intervals, especially for such complex regression models and time-series, unless restrictive assumptions on the underlying time-series distribution are in place.

Contribution. This paper directly addresses the challenges above by building distribution-free prediction intervals for dynamic time-series data with marginal coverage guarantee. In particular, we efficiently build prediction intervals for point estimates from ensemble estimators that have been trained without refitting any more ensemble models. We summarize the main contributions as follows:

- We present a robust and computationally efficient predictive inference method around ensemble estimators, called `EnbPI`, which constructs multiple/a batch of prediction intervals at once and do so sequentially. It requires no data splitting and works well for small-sample problems.
- Theoretically, we show that `EnbPI` prediction intervals enjoy approximately valid marginal coverage under mild assumptions on time-series' stochastic errors and regression estimators. In particular, the method is suitable for non-stationary time-series and may attain conditional validity. Approximately valid coverage means we can upper bound the non-coverage at each sample size T by a real sequence $\tau_T \to 0$, where τ_T depends on the underlying assumptions on data and estimators.
- Empirically, we extensively study the performance of `EnbPI` on the renewable energy estimation application, using solar and wind data. We show that `EnbPI` maintains coverage when competing methods fail to do

[1]Industrial and Systems Engineering, Georgia Institute of Technology. Correspondence to: Chen Xu <cxu310@gatech.edu>, Yao Xie <yao.xie@isye.gatech.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

so. It can handle network data and data with missing entries as well. We also demonstrate its broad applicability on time-series from other application domains, on which `EnbPI` intervals are often shorter than those by competing methods.

- Furthermore, `EnbPI` can be directly used for unsupervised sequential anomaly detection. Modifications of the procedure yield promising results on supervised anomaly detection as well.

Literature Review. A broad family of conformal prediction (CP) methods is becoming popular for constructing distribution-free prediction intervals. Formally introduced in (Shafer and Vovk, 2008), this method assigns "conformity scores"[1] to training data and test data. Inverting the hypothesis test using these scores generates prediction intervals for test data. Under exchangeability in data, this procedure generates exactly valid marginal coverage of the test point. Many works such as (Papadopoulos et al., 2007; Romano et al., 2019; Barber et al., 2019b; Kivaranovic et al., 2020; Izbicki et al., 2020) operate under this logic. For comprehensive surveys and tutorials, we refer readers to (Shafer and Vovk, 2008; Zeni et al., 2020). Although no assumption is imposed on functions that assign conformity scores and coverage is marginally exact, the exchangeability assumption is hardly reasonable for time-series.

Recently, adapting CP methods beyond exchangeable data has also been an important area. The work by (Tibshirani et al., 2019) uses weighted conformal prediction intervals when the shifted distribution on test data is proportional to the pre-shift training distribution. Another recent work by (Cauchois et al., 2020) provides a coverage guarantee when the shifted distribution lies in an f-divergence ball around the training distribution. However, both works still assume i.i.d. or exchangeable training data, making them not directly suitable for time-series. On the other hand, the works by (Chernozhukov et al., 2018; 2020) study conformal inference for time-series data and their assumptions and proof techniques motivate our theoretical analyses. Nevertheless, their methods do not avoid data-splitting and are computationally intensive for ensemble methods. Moreover, we refine their proof techniques to improve the convergence rates (see Lemma 1 and 2 proofs) and extend results under different assumptions (Corollary 1—3).

The work closest to ours in construction is the Jackknife+-after-bootstrap (J+aB) (Kim et al., 2020), which also efficiently applies conformal prediction to ensemble methods. However, that work assumes data exchangeability and during prediction, does not leverage new observations as they are sequentially revealed. In contrast, we replace the assumption on data exchangeability with mild assumptions on the error process and the estimation quality of regressors, under which the method still has performance guarantee.

[1] In this paper, conformity scores are calculated as residuals from fitting a regression algorithm $\mathcal{A}$ on the training data.

Table 1 summarizes the coverage guarantee of some CP methods under various assumptions. We remark that the table presents the best attainable coverage guarantees. However, doing so may not be ideal in practice since intervals may be too wide under these guarantees.

Table 1. Theoretical and empirical coverage guarantee of various CP methods.

Distribution Assumption	In Theory	Empirically
Exchangeable (Papadopoulos et al., 2007)	$1-\alpha$	$1-\alpha$
Covariate Shift (Tibshirani et al., 2019)	$1-\alpha$	$1-\alpha$
Strongly Mixing Errors (Chernozhukov et al., 2018)	$\approx 1-\alpha$	$1-\alpha$

Although this paper's main focus is to combine CP and ensemble methods for time-series efficiently, non-CP time-series prediction interval methods are abundant. Traditional time-series methods, such as ARIMA(p,d,q), exponential smoothing, state-space models (e.g., Kalman Filter), have been widely successful (Brockwell et al., 1991). On the other hand, (Rosenfeld et al., 2018) uses a discriminative learning framework to optimize the expected error rate under a budget constraint on interval sizes, with PAC-style, finite-sample guarantees.

2. Problem Setup

Given an unknown model $f : \mathbb{R}^d \to \mathbb{R}$, where d is the dimension of the feature vector, we observe x_t and y_t generated according to the following model

$$Y_t = f(X_t) + \epsilon_t, t = 1, 2, \dots \tag{1}$$

where ϵ_t are identically distributed according to a common cumulative distribution function (CDF) F; note that we do not need to require ϵ_t to be independent. Features X_t can be either exogenous time-series sequences and/or the history of Y_t. In the following, we assume that the first T sample points $\{(x_t, y_t)\}_{t=1}^{T}$ are training data or initial state of the random process that are given to us. Above, upper case X_t, Y_t denote random variables and lower case x_t, y_t denote data.

Our goal is to construct a sequence of prediction intervals. Initially, using the past T sample points, we construct $s \geq 1$ prediction intervals $\{C_{T,T+i}\}_{i=1}^{s}$ for $\{Y_{T+i}\}_{i=1}^{s}$, where the *batch size* s is a pre-specified parameter corresponding to how many steps we would like to look ahead. Once new sample points $\{(x_{T+i}, y_{T+i})\}_{i=1}^{s}$ become available, we use

the most recent T points to produce prediction interval for $Y_j, j = T + s + 1$ onward. Note that in the special case $s = 1$, we build intervals one after another sequentially and receive immediate feedback.

The prediction intervals are constructed as follows. Since given a significance level α, $C_{T,t}$ often depends on α, we henceforth denote it as $C^{\alpha}_{T,t}$. Furthermore, denote $\hat{f}_{-i}$ as the i-th "leave-one-out" (LOO) estimator of model f, whose training data excludes the i-th datum (x_i, y_i) and may include all the rest $T - 1$ points. Let the prediction interval at time t be

$$C^{\alpha}_{T,t} := \hat{f}_{-t}(x_t) \pm (1-\alpha) \text{ quantile of } \{\hat{\epsilon}_i\}_{i=t-1}^{t-T}, \quad (2)$$

where the prediction residual

$$\hat{\epsilon}_i := |y_i - \hat{f}_{-i}(x_i)|.$$

Thus, the interval is centered at the point prediction $\hat{f}_{-t}(x_t)$ and its width is the quantile over the past T residuals. When $s > 1$, y_{t-1} may not have be revealed when `EnbPI` constructs $C^{\alpha}_{T,t}$, so we take the quantile over past T latest available residuals.

Theoretically, we require each prediction interval $C^{\alpha}_{T,t}, t > T$ to satisfy the following *marginal* coverage guarantee:

$$P(Y_t \in C^{\alpha}_{T,t}) \geq 1 - \alpha. \quad (3)$$

It is challenging to ensure (3) under complex data dependency and without distributional assumptions. In particular, conventional conformal prediction methods that require exchangeability do not work. However, under certain assumptions on the error process $\{\epsilon_t\}_{t\geq 1}$ and LOO estimators of f, we can ensure (3) holds approximately, meaning that the probability of under-coverage can be bounded at any finite sample size T and approaches zero as sample size reaches infinity. From now on, we call a prediction interval *valid* if it achieves (3).

We lastly distinguish between *marginal versus conditional* coverage guarantee. Assume X_t belongs to a subspace $\mathcal{X} \subset \mathbb{R}^d$, whereby conditional coverage guarantee means that

$$P(Y_t \in C^{\alpha}_{T,t} | X_t \in \mathcal{X}) \geq 1 - \alpha. \quad (4)$$

As a practical example, suppose a doctor reports a prediction interval for one patient's future blood pressure. An interval satisfying (3) averages over all patients in *different* age groups, but may not satisfy (4) for the current patient in a *specific* age group. In fact, satisfying (4), even for exchangeable data, is more difficult than satisfying (3) and can be impossible without further assumptions on data (Barber et al., 2019a). Nevertheless, we will show in our experiments that `EnbPI` has the ability to satisfy (4) in many cases.

3. EnbPI Algorithm

We now present Algorithm 1, named `EnbPI`, which has several noticeable benefits: it efficiently constructs $\hat{f}_{-i}$ in (2) as ensemble estimators, requires no data-splitting, avoids model overfitting, and does not refit models during test time. In the algorithm, $\hat{f}^b$ is the b-th bootstrap estimator and variables with superscript ϕ come from either using the aggregation function ϕ on multiple inputs or other variables with superscript ϕ.

Algorithm 1 Sequential Distribution-free Ensemble Batch Prediction Intervals (`EnbPI`)

Require: Training data $\{(x_i, y_i)\}_{i=1}^{T}$, regression algorithm $\mathcal{A}$, decision threshold α, aggregation function ϕ, number of bootstrap models B, the batch size s, and test data $\{(x_t, y_t)\}_{t=T+1}^{T+T_1}$, with y_t revealed only after the batch of s prediction intervals with t in the batch are constructed.

Ensure: Ensemble prediction intervals $\{C^{\phi,\alpha}_{T,t}(x_t)\}_{t=T+1}^{T+T_1}$

for $b = 1, \ldots, B$ **do**
  Sample with replacement an index set $S_b = (i_1, \ldots, i_T)$ from indices $(1, \ldots, T)$.
  Compute $\hat{f}^b = \mathcal{A}(\{(x_i, y_i) \mid i \in S_b\})$.
end for
Initialize $\boldsymbol{\epsilon} = \{\}$
for $i = 1, \ldots, T$ **do**
  $\hat{f}^{\phi}_{-i}(x_i) = \phi(\{\hat{f}^b(x_i) \mid i \notin S_b\})$
  Compute $\hat{\epsilon}^{\phi}_i = |y_i - \hat{f}^{\phi}_{-i}(x_i)|$
  $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} \cup \{\hat{\epsilon}^{\phi}_i\}$
end for
for $t = T + 1, \ldots, T + T_1$ **do**
  Let $\hat{f}^{\phi}_{-t}(x_t) = (1-\alpha)$ quantile of $\{\hat{f}^{\phi}_{-i}(x_t)\}_{i=1}^{T}$
  Let $w^{\phi}_t = (1-\alpha)$ quantile of $\boldsymbol{\epsilon}$.
  Return $C^{\phi,\alpha}_{T,t}(x_t) = [\hat{f}^{\phi}_{-t}(x_t) \pm w^{\phi}_t]$
  if $t - T = 0 \bmod s$ **then**
    for $j = t - s, \ldots, t - 1$ **do**
      Compute $\hat{\epsilon}^{\phi}_j = |y_j - \hat{f}^{\phi}_{-j}(x_t)|$
      $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon} - \{\hat{\epsilon}^{\phi}_1\}) \cup \{\hat{\epsilon}^{\phi}_j\}$ and reset index of $\boldsymbol{\epsilon}$.
    end for
  end if
end for

We briefly comment on inputs to the algorithm:

(1) In general, $\mathcal{A}$ can consist of a family of regression algorithms (e.g., parametric and non-parametric models), each of which maps data to predictors.
(2) Smaller thresholds α mean higher coverage and yield wider intervals.
(3) Different aggregation functions ϕ bring different benefits, such as reducing mean squared error (MSE) under mean, avoiding sensitivity to outliers under median, or achieving both under trimmed mean.

(4) A larger number of bootstrap models B yields shorter and more stable intervals. Empirically, letting $B = 20 - 30$ is sufficient, especially for computationally intensive methods such as neural networks.
(5) The larger the batch size s is, the more number of prediction intervals EnbPI has to construct, and the less frequently feedback comes. When $s = \infty$, no feedback is available, so that EnbPI only uses training residuals to calibrate interval widths. All prediction intervals thus have equal width, which may not be reasonable or beneficial. We recommend choosing s as small as possible but its value depends on the data collection process.

3.1. Properties of EnbPI

Computational efficiency. The main cost lies in training T LOO ensemble predictors. EnbPI trains $\mathcal{A}$ for B times to get B bootstrap models (line 1-4) before aggregation (line 6-10), so the cost is $\mathcal{O}(B)$ in terms of the number of calls to $\mathcal{A}$. In comparison, the naive way requires training $\mathcal{A}$ for B many times on *each* leave-i-out sample $\{(x_j, y_j)\}_{j=1, j\neq i}^{T}$, so the cost drastically increases to $\mathcal{O}(BT)$. Doing so is infeasible in reality, especially for computationally intensive models such as a deep neural network. In short, EnbPI merely requires the computational power of building one ensemble model but constructs T LOO ensemble models.

No overfitting or data splitting. Unlike traditional CP methods such as ICP (Papadopoulos et al., 2007) that use data-splitting to avoid overfitting, EnbPI avoids this issue through thoughtful aggregations in line 6-10. In particular, to construct the i-th LOO ensemble predictor, EnbPI aggregates all B bootstrap models that are *not* trained on the training datum (x_i, y_i). Note that the Chernoff bound ensures that each ensemble predictor is aggregated on a balanced number of bootstrap models. On the other hand, EnbPI avoids data-splitting as bootstrap estimators are trained on random subsamples from the *full data* (line 1-4). The J+aB procedure in (Kim et al., 2020) inspires these constructions.

Leverage new data without model refitting. During the prediction time $t > T$ (line 11-21), EnbPI constructs sequential prediction intervals without refitting $\mathcal{A}$ on test data. Instead, it leverages feedback by updating past residuals using a sliding window of size T. Doing so is important in reality, as f in model (1) may change during prediction, leading to larger residuals afterwards that need to be consider. EnbPI thus allows for dynamic and accurate calibration of prediction interval widths even under potential data shift.

3.2. EnbPI on Challenging Tasks

Handling missing data. Suppose missing data are present in training data. We can properly increase the size of each bootstrap sample from the rest available data to include the same number of unique data points as if no missing data exist. Doing so is often enough since we assume a common data model f. Meanwhile, suppose EnbPI encounters a missing index t' during prediction. It can still construct the prediction interval at time t', since the feature observation $x_{t'}$ is available; the sliding window then skips over the residual $\epsilon_{t'}^{\phi}$. If the whole time-series is univariate, under which $x_{t'}$ is the history of $y_{t'}$, we need to impute $y_{t'}$ so that *future features* $x_t, t > t'$ have no missing entries.

Network prediction. Suppose a network has K nodes, so that observations at node $k \in [K]$ are $\{(y_t^k, x_t^k)\}_{t\geq 1}$. To handle spatial-temporal correlations and incorporate information from neighboring nodes, we can define the new feature $\tilde{x}_t^k$ at node k and time t as the collection of features from neighbors of k at time t and earlier. The primary benefits of doing so are two-fold: firstly, one incorporates spatial-temporal information for constructing the prediction interval of Y_t^k. Secondly, the coverage guarantee equally applies to each node as long as one applies EnbPI once for each node.

Unsupervised Sequential Anomaly detection. Suppose there is an anomaly y_{t^*} at time t^*, due to either a change in model f at t^* or an unusually large stochastic error ϵ_{t^*}. As a result, y_{t^*} will likely lie far outside the interval (equivalently, $\epsilon_{t^*}^{\phi}$ is well above the $(1-\alpha)$ quantile of past T residuals), so it is detected as an anomaly. All the benefits of EnbPI carry over as it detect anomalies in this way. A modified version of EnbPI works for supervised anomaly detection as well (see Section 5.3 and 8.5).

4. Theoretical Analysis

Without loss of generality and for notation simplicity, we only consider the validity of EnbPI on the first test point with index $T + 1$. We comment on why validity holds for all prediction intervals from $T + 2$ onward at the end of this section. From now on, we drop superscript ϕ on outputs in EnbPI for simplicity. In particular, our proof removes the assumptions on data exchangeability by replacing with general and verifiable assumptions on the error process $\{\epsilon_t\}_{t\geq 1}$ and estimation quality of ensemble predictors.

We first define the empirical p-value at $T + 1$:

$$\hat{p}_{T+1} := T^{-1} \sum_{i=1}^{T} \mathbf{1}\{\hat{\epsilon}_i > \hat{\epsilon}_{T+1}\}.$$

The following equivalence holds under basic algebraic manipulation:

$$Y_{T+1} \in C_{T,T+1}^{\alpha} \text{ if and only if } \hat{p}_{T+1} > \alpha.$$

Therefore, our method covers Y_{T+1} with probability at least $1 - \alpha$, hence being valid, if the distribution of $\hat{p}_{T+1}$ is approximately uniform. More precisely, we aim to ensure that $|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha|$ is small.

Furthermore, let

$$\tilde{p}_{T+1} := T^{-1} \sum_{i=1}^{T} \mathbf{1}\{\epsilon_i > \epsilon_{T+1}\},$$
$$\tilde{F}(x) := T^{-1} \sum_{i=1}^{T} \mathbf{1}\{\epsilon_i \leq x\},$$

where $\tilde{p}_{T+1}$ is the counterpart of $\hat{p}_{T+1}$ but uses *actual* residuals and $\tilde{F}(\epsilon_{T+1}) = 1 - \tilde{p}_{T+1}$. Equivalently define $\hat{F}(x)$ for $\hat{p}_{T+1}$.

4.1. Main Result

Recall F is the unknown true CDF for $\{\epsilon_t\}_{t\geq 1}$. Under the following assumptions, we can bound the worst deviation between $\tilde{F}(x)$ and $F(x)$ in Lemma 1, as well as between $\hat{F}(x)$ and $\tilde{F}(x)$ in Lemma 2, which are essential to proving our main theoretical results in Theorem 1. All proofs of Lemmas and Theorem 1 can be found in Section 7 of the Appendix.

Assumption 1 (Stationary and strongly mixing error process)**.** *Assume $\{\epsilon_t\}_{t\geq 1}$ are stationary and strongly mixing, with sum of mixing coefficients bounded by M. Their common CDF F also satisfies a Lipschitz condition with constant $L > 0$.*

Lemma 1. *Suppose Assumption 1 hold. Define $C_1 := (M/2)^{1/3}$. Then, for any training size T, there is an event A_T in the probability space of $\{\epsilon_t\}_{t=1}^{T}$, such that conditional on the event A_T,*

$$\sup_x |\tilde{F}(x) - F(x)| \leq C_1 (\log T/T)^{1/3}.$$

Moreover

$$P(A_T^C) \leq C_1 (\log T/T)^{1/3}.$$

Assumption 2 (Estimation quality)**.** *There exists a real sequence $\{\delta_T\}_{T\geq 1}$ that converges to zero such that*

$$\sum_{t=1}^{T} (\hat{\epsilon}_t - \epsilon_t)^2 / T \leq \delta_T^2.$$

Lemma 2. *Assume Assumption 1 and 2 hold. Define $C_2 := L + 1$ whereby*

$$\sup_x |\hat{F}(x) - \tilde{F}(x)| \leq C_2 \delta_T^{2/3} + 2 \sup_x |\tilde{F}(x) - F(x)|.$$

Our main theoretical result is the following Theorem 1, which follows as a consequence of Lemma 1 and 2.

Theorem 1 (Approximately uniform p-value)**.** *Let $C_1 = (M/2)^{1/3}, C_2 = L + 1$. For any training size T and $\alpha \in (0, 1)$, the empirical p-value $\hat{p}_{T+1}$ obeys:*

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \leq 12 C_1 \left(\log T/T\right)^{1/3} + 2 C_2 \delta_T^{2/3}.$$

We make several comments for Theorem 1:

(1) To build prediction intervals that have at least $1 - \alpha$ coverage, one needs to incorporate the upper bounds above into the prediction interval construction. `EnbPI` does not do so, as we aim to design a general wrapper that can be applied to most regression models $\mathcal{A}$, whose coverage guarantee also varies by models.

(2) The rate of convergence of order $\mathcal{O}((\log T/T)^{1/3} + \delta_T^{2/3})$ is a worst-case analysis. Empirical results show that even at small training data size T, $|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \approx 0$, which likely happens because LOO ensemble predictors approximate f in (1) well.

(3) When data are exchangeable, we can easily modify `EnbPI` to match the J+aB algorithm (Kim et al., 2020), which guarantees $1 - 2\alpha$ coverage regardless of estimation quality (Assumption 2). Specifically, we aggregate a random number of bootstrap estimators B ($B \sim \text{Binom}(\tilde{B}, (1 - \frac{1}{T+1})^T)$, $\tilde{B}$ fixed) and do not slide the past residuals ($s = \infty$).

Remark 1 (Wider Applicability of Theorem 1)**.** *In general, Theorem 1 also applies to other conformal prediction methods, such as the split/inductive conformal (Papadopoulos et al., 2007). However, there are two major disadvantages when using split conformal (and its variants) that requires "calibration data":*

1) The value T on the RHS of Theorem 1 becomes the size of the calibration data, which is typically much smaller than T. In contrast, all the T training data in `EnbPI` act as calibration data because we train LOO ensemble estimators.

2) In general, ensemble predictors in `EnbPI` are better approximators to the unknown f than split conformal predictors. Hence, Assumption 2 is often more easily satisfied even when the calibration data for split conformal is as large as the training data in `EnbPI`.

Therefore, `EnbPI` is more favorable, especially when the size of training data is much smaller than that of test data. In that case, subsetting a part of the training data as calibration data is simply impractical.

4.2. Discussions on Assumptions

In the remainder of this section, we discuss the implications, extensions, and examples of Assumption 1 and 2. In particular, we show how to replace the "stationary and strongly mixing" condition and give specific examples of δ_T.

Assumption 1. In general, this is a very mild assumption on the original process $\{(X_t, Y_t)\}_{t\geq 1}$, even when the error process $\{\epsilon_t\}_{t\geq 1}$ is iid. This is because the series can exhibit arbitrary dependence and be highly non-stationary, but still have strongly-mixing (or even i.i.d.) errors. Common time-series with i.i.d errors include non-stationary ran-

dom walk and ARIMA(p, d, q) models. Meanwhile, we refer to (Doukhan, 2012) for a comprehensive list of mixing processes, including Gaussian random fields, Gibbs fields, continuous-time processes, etc.

Moreover, the "stationary and strongly mixing" condition in Assumption 1 can be relaxed, tightened, or replaced by other conditions. Doing so yields rates of convergence different from $(\log T/T)^{1/3}$ (see RHS of Theorem 1). We provide three examples, whose precise statements and proofs are in Corollaries 1, 2, and 3 of the appendix, respectively:

(1) Suppose $\{\epsilon_t\}_{t\geq 1}$ is independent. The rate is improvable to $(\log(16T)/T)^{1/2}$.
(2) Suppose $\epsilon_t = \sum_{j=1}^{\infty} \delta_j z_{t-j}$, which captures a broad class of stationary linear processes. Under mild assumptions on δ_j and z_{t-j}, the rate is improvable to $(\log T/\sqrt{T})$, so that it is faster than assuming strongly mixing errors but slower than independent errors.
(3) Suppose $\{\epsilon_t\}_{t\geq 1}$ are generated by random symmetric matrices $\psi_{jk}/\sqrt{n}, 1 \leq j, k \leq t$. If the density of ϵ_t satisfies a logarithmic Sobolev inequality, the rate is $(\log(cT)/T)^{1/3}$ for some constant c, almost the same as Theorem 1.

Assumption 2. Firstly, one needs to avoid overfitting, since this assumption requires the closeness between predicted and actual *residuals* (i.e., ϵ), not just *responses* (i.e., Y). In other words, using in-sample residuals or interpolating data is unfavorable. Moreover, so long as estimators using $\mathcal{A}$ satisfy Assumption 2, the theoretical guarantee holds; we favor ensemble predictors in `EnbPI` as they tend to reduce function approximation errors.

Under model (1), Assumption 2 is in fact equivalent to requiring asymptotically exact function approximation, in the sense $\sum_{t=1}^{T}(\hat{f}_t(X_t) - f(X_t))^2/T \leq \delta_T, \delta_T \to 0$. The famous *No Free Lunch Theorem* (Wolpert and Macready, 1997) implies that assumptions on the underlying unknown function f are necessary, so this condition does not hold for all $\mathcal{A}$ and f.[2] Still, we can find δ_T for two classes of f and the corresponding $\mathcal{A}$:

(1). if f is sufficiently smooth, $\delta_T = o_P(T^{-1/4})$ for general neural networks sieve estimators (see Chen and White, 1999, Corollary 3.2).
(2). If f is a sparse high-dimensional linear model, $\delta_T = o_P(T^{-1/2})$ for the Lasso estimator and Dantzig selector. (see Bickel et al., 2009, Equation 7.7).

In general, one needs to analyze the rate of convergence of estimators $\hat{f}$ from a given function class (specified using $\mathcal{A}$) to the unknown true f. This task is different from analyzing the MSE of ensemble estimators (Breiman, 1996) by our previous observation, so it requires case-by-case analyses. The task can be even harder for ensemble estimators used in `EnbPI`, so finding its answers will be a part of the future research.

Lastly, the previous arguments show why the same approximate coverage guarantee holds at every point after $T + 1$. The whole error sequence is subject to Assumption 1. Moreover, when each LOO ensemble predictor approximates the unknown *fixed* f well, all residuals beyond $T + 1$ satisfy Assumption 2. In other words, there are no inherent differences between coverage at $T + 1$ or at future time indices, so long as Assumptions 1 and 2 hold.

Remark 2 (When Assumption 2 Fails)**.** *In reality, change points can alter the underlying f, whereby residual differences $|\hat{\epsilon}_t - \epsilon_t|, t \geq T$ become large. Such failures likely also cause predicted interval centers $\hat{y}_t = \hat{f}(x_t)$ to be far from true observations y_t. In this case, coverage can be poor when s is too large (i.e., the algorithm looks too far ahead), because pre-changed residuals are used to calibrate widths of post-change prediction intervals. We demonstrate such behaviors in later experiments (see Figure 3).*

5. Experimental Results

We primarily apply `EnbPI` on solar and wind energy data. In Section 5.1, we show that `EnbPI` is approximately valid as it sequentially produces intervals *one after another* (i.e., $s = 1$) and that its validity is robust under different input parameters, whereas competing methods fail to maintain validity. In Section 5.2, we then use `EnbPI` to produce multiple prediction intervals and examine its conditional coverage validity. In Section 5.3, we studies a supervised credit card fraud detection example by using a modified version of `EnbPI`. In the appendix (Section 8.4), we further demonstrate that `EnbPI` is valid on time-series from other application domains, where we often see `EnbPI` intervals are shorter than those by competing methods.

5.1. Interval Validity of EnbPI

We use 2018 hourly solar radiation data from Atlanta and 9 cities in California, as well as 2019 hourly wind energy data from the Hackeberry wind farm in Austin. In total, we have 11 time-series from 11 sensors (one from each sensor) and each time-series contains 8760 recordings (24*365), with ambient features such as temperature, humidity, wind speed, etc. In particular, California solar data constitute a network, where each node is a sensor. See Section 8.1 for detailed data descriptions and visualizations. Note that we omit using simulated data with a known data-generating model f, because we aim to examine how well prediction intervals by `EnbPI` cover actual observations, not how well ensemble predictors predict the true model. From now on, we call X_t *multivariate* if it contains ambient features and

[2] Our approach under exchangeable data imposes no condition on $\mathcal{A}$, as explained in comment (3) for Theorem 1.

univariate if it is the history of Y_t.

Comparison methods. We primarily compare EnbPI with two other CP methods and with ARIMA(10,1,10). In Section 8.2 of the appendix, we also compare EnbPI with J+aB (Kim et al., 2020), which motivates the construction of EnbPI. The first CP method is the widely adopted split conformal/inductive conformal prediction (ICP) by (Papadopoulos et al., 2007). In particular, (Chernozhukov et al., 2018) guarantees approximate validity of ICP. The other CP method is the weighted ICP (WeightedICP) proposed by (Tibshirani et al., 2019), which is proven to work when the test distribution shifts in proportion to the training distribution; it generalizes to more complex settings than ICP. We use logistic regression to estimate the weights for WeightedICP. We do a 50:50 split into proper training set and calibration set for ICP and WeightedICP. We acknowledge that CP methods for time-series are currently lacking, so that WeightedICP is chosen as a natural competitor among those that work beyond purely exchangeable data. Lastly, we use ARIMA implemented in Python's `statsmodel` package under default parameter specification.

Regression Algorithm $\mathcal{A}$. We choose four regression algorithms: ridge, random forest (RF), neural networks (NN), and recurrent neural networks (RNN) with LSTM layers. The first two are implemented in the Python `sklearn` library, and the last two are built using the `keras` library. See Section 8.1 for their parameter specifications.

Other specifications. Since the three CP methods train on random subsets of training data, we repeat all experiments below for 10 trials, where each trial splits training data into bootstrap samples independently. On the other hand, ARIMA is deterministic given training data, so we only train it once. Throughout this subsection, we fix $s = 1$, so every observation comes in sequence without delay. We let $\alpha = 0.1$ and use the first 20% of total hourly data for training unless otherwise specified. Doing so mimics a setting of small-sample problems with long-term predictive inference goals. Lastly, we use EnbPI under $B = 30$ and ϕ as taking the sample mean. Thus, each of the ensemble predictors in EnbPI is a leave-i-out bagging predictor.

Results. All results in Section 5.1 and 5.2 come from using the Atlanta solar data. Similar results using California solar data and Hackberry wind data are in the appendix (Section 8.3). Figure 1 compares average coverage and width vs. $1 - \alpha$ under EnbPI with different regression models and ARIMA. It is clear that EnbPI maintains coverage under any regression model we have chosen, whereas coverage failure by ARIMA is more severe as $1 - \alpha$ increases. Ensuring $1 - \alpha$ coverage under small α values is important in reality, making ARIMA not applicable for such dynamic time-series. Although ARIMA intervals are much shorter in terms of widths than those by EnbPI, the severe coverage failure by ARIMA makes such benefits not meaningful.

Figure 2 shows grouped boxplots of coverage and width for CP methods using ridge, NN, and RNN. Since ARIMA is non-randomized, its results are not shown here. All coverage boxes by EnbPI tightly center around the target coverage and have very small variance. Moreover, EnbPI is very suitable for small-sample problems since its coverage barely varies across different training data sizes. On the other hand, ICP and WeightedICP show significant under coverage, regardless of whether X_t is multivariate or univariate. Thus, they are neither valid nor applicable to time-series data. We believe such behaviors align with our observations in Remark 1. In terms of width, although intervals by ICP and WeightedICP are much shorter than EnbPI, the severe coverage failure by the former two methods makes such benefits not meaningful. On the other hand, EnbPI intervals under univariate X_t are shorter than those under multivariate ones, likely because response series' historical observations are more predictive of the current value than ambient information.

Remark 3 (Practical Usefulness)**.** *While theoretical guarantee of EnbPI requires $\mathcal{A}$ to satisfy Assumption 2, empirical results are valid even under potentially misspecified models, and coverage is almost always exactly valid. We think this property is particularly appealing since one may only need simple, computationally friendly, and interpretable models in EnbPI without losing coverage.*

5.2. Multi-step Ahead Predictive Inference

We let $s > 1$ in EnbPI, so it constructs multiple intervals for these hourly energy observations. We have two particular goals: firstly, we aim to attain valid *conditional coverage* at each hour, as multiple intervals correspond to different hours in a day; secondly, we show how well EnbPI can handle time-series with missing data, which is a common problem if sensors malfunction. We choose to only use EnbPI for these tasks since other CP methods and ARIMA failed even to maintain marginal validity.

Parameter Specification. All parameters into EnbPI except choices of s are kept the same unless otherwise specified. We pick $s = 14$ for Atlanta solar data, because recordings before sunrise (i.e., 6AM) and after sunset (i.e., 8PM) are zero; EnbPI thus constructs 14 prediction intervals one day ahead. For missing data experiments, we randomly drop 25% of both training and test data. Meanwhile, to use univariate X_t as features, we impute the missing entries by sampling from a normal distribution with parameters being empirical mean and standard error of past s observations. We assume the ambient features for multivariate X_t are readily available and performs no imputation.

Results. Figure 3 shows *conditional* coverage of EnbPI under RF at certain hours of the day with the presence

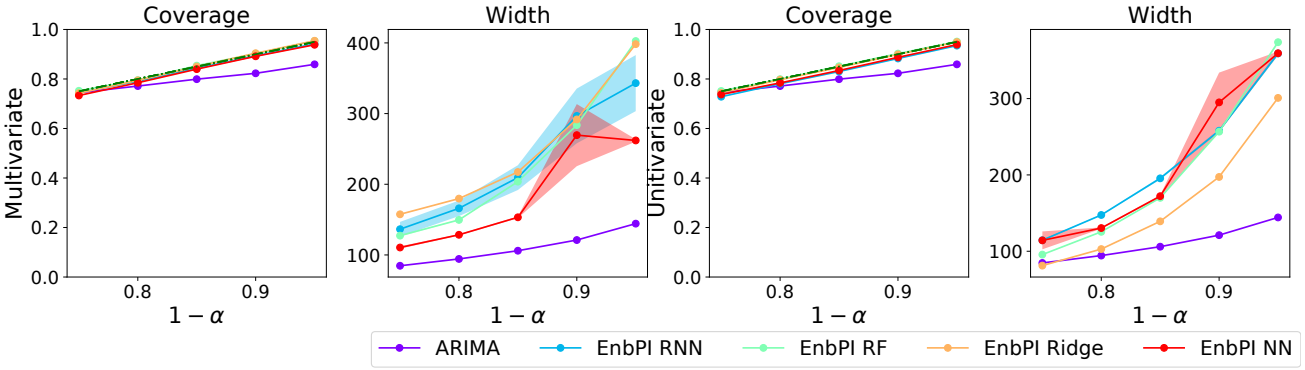


Figure 1. Solar—Atlanta: average coverage and width vs. $1-\alpha$ target coverage by `EnbPI` under different regression algorithms and by ARIMA. Five equally spaced $1-\alpha \in [0.75, 0.95]$ are chosen. The green dash-dotted line at 0.9 represents the target coverage.

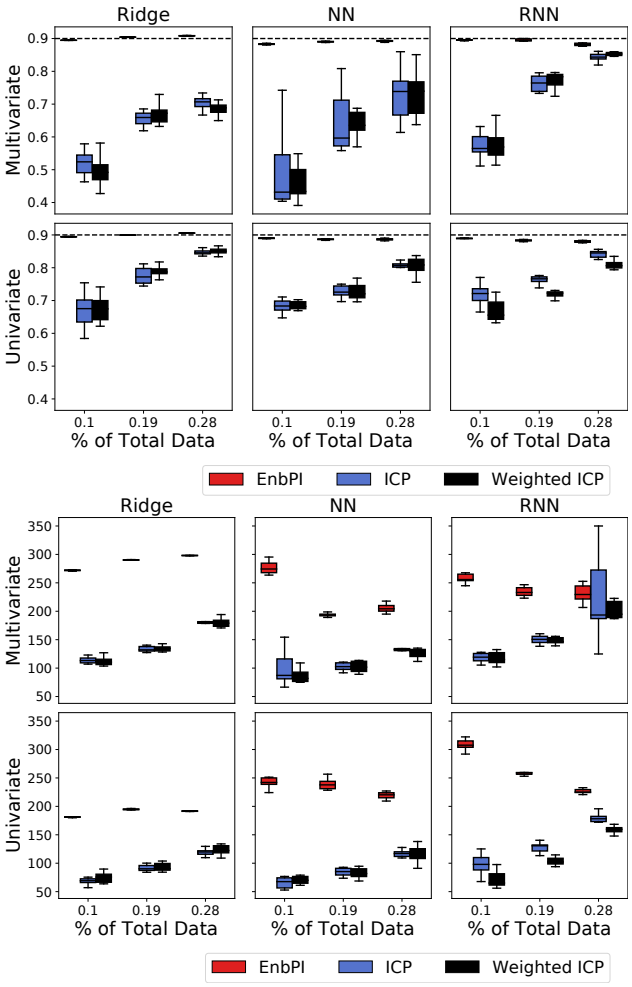


Figure 2. Solar—Atlanta: boxplots of average coverage (top) and width (bottom) by `EnbPI`, ICP, and WeightedICP, whose training data vary as a percentage of total data (x-axis). Each box contains results from 10 independent trials. The black dash dotted line at 0.9 indicates target coverage.

of missing data. The result without missing data is very similar to Figure 3 and is in Section 8.2 of the Appendix. The hourly training data come from the first 92 days of observation (January-March), with multivariate X_t features. For clarity, we only show the result of `EnbPI` under one trial, as results hardly vary across trials, and selectively show coverage at three hours from 10AM—2PM and at three other hours. We do so because conditional coverage from 10AM—2PM is much poorer than the rest. Several things are noticeable. Firstly, despite not being shown in the figures, marginal coverage over all hours is always 90%, regardless of the presence of missing data. Secondly, there is almost no conditional coverage difference when missing data are present, so that `EnbPI` is robust under a modest amount of missing data. Lastly, poor conditional coverage during 10AM—2PM likely happens because of two things: firstly, radiation near noon is much higher and significantly different from the rest; secondly, there are possible *change points* near summertime (e.g., around August), as the training data come from winter time (e.g., January—March). Nevertheless, we show in the appendix (Section 8.2) that by applying `EnbPI` only on data during 10AM—2PM (so $s = 5$), we can ensure valid conditional coverage at all these hours. We will also show additional results when new data are not available to `EnbPI` (i.e., $s = \infty$).[3] In general, we think `EnbPI` has the potential to reach conditionally valid coverage in the sense of 4 and aim to analyze this theoretically in the future.

5.3. Supervised Anomaly detection

Consider a supervised credit card fraud detection task on Kaggle, where $y_t \in \{0, 1\}$. The task is to identify anomalous transactions at each time step. Challenges arise since the data is highly imbalanced (only 0.172% of 284,807 total observations are anomalies), features are only given as prin-

[3]For datasets with fixed sizes, $s = \infty$ is replaced by the length of the test data.

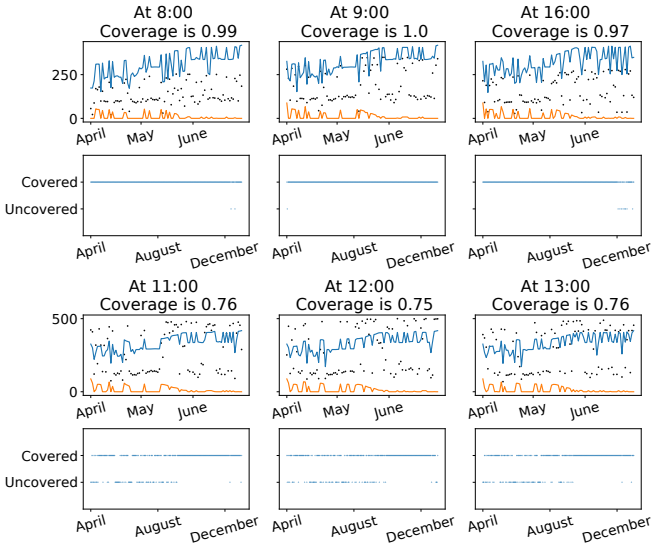


Figure 3. Solar—Atlanta, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December).

cipal components, and decisions must be made sequentially.

Figure 4 compares ECAD, a modified version of EnbPI that wraps around *binary classification algorithms*, against 8 other anomaly detectors, four of which are unsupervised (e.g., IForest, PCA, OCSVM, and HBOS) and the other four are supervised (e.g., MLPClassifier, GBoosting, KNN, SVC). It is clear that ECAD consistently obtains the highest F_1 scores. Its F_1 score also varies little over different training sizes. Therefore, ECAD can be used for detecting anomalies in time-series with a small number of training data. See Section 8.5 in the Appendix for the formal problem setup, ECAD algorithm, data and competing methods.

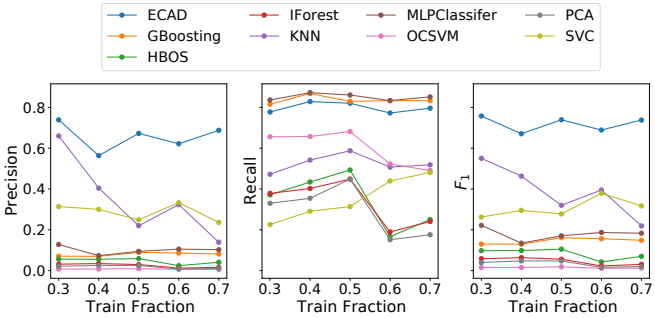


Figure 4. Kaggle Data: Precision, Recall, and F_1 scores vs. different amounts of training data (as percentages of total data) for different detectors.

6. Conclusion

In this paper, we present a predictive inference method for dynamic time-series. Theoretically, its intervals are approximately marginally valid without assuming data exchangeability. Computationally, EnbPI is an efficient ensemble-based wrapper for many regression algorithms, including deep neural networks. Empirically, it is versatile on a wide range of time-series, including network data and data with missing entries, and maintains validity when traditional methods fail. Furthermore, it can be used for unsupervised and supervised sequential anomaly detection.

Future work includes several possible directions. Methodologically, we aim to (1) adapt EnbPI for classification problems, especially those in computer vision (Angelopoulos et al., 2020; Romano et al., 2020); (2) connect EnbPI more closely with other applications, such as anomaly detection (Ishimtsev et al., 2017) and sequential change-point detection (Volkhonskiy et al., 2017). Theoretically, we want to (1) closely analyze how LOO ensemble predictors in EnbPI can better satisfy Assumption 2 on estimator consistency than non-ensemble ones, so as to provide tighter bounds in Theorem 1; (2) provide theoretical guarantee for conditional coverage, as in (Barber et al., 2019a; Izbicki et al., 2020); (3) bound deviation of width between estimated prediction intervals and oracle ones, as in (Lei et al., 2018) for the i.i.d. case.

Acknowledgement

The work is partially supported by NSF CAREER CCF-1650913, CMMI-2015787, DMS-1938106, and DMS-1830210.

References

Aggarwal, C. C. (2015). Outlier analysis. In *Data mining*, pages 237–263. Springer.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. (2020). Uncertainty sets for image classifiers using conformal prediction. *ArXiv*, abs/2009.14193.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2019a). The limits of distribution-free conditional predictive inference. *arXiv preprint arXiv:1903.04684*.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2019b). Predictive inference with the jackknife+.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., et al. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732.

Bobkov, S. G., Götze, F., et al. (2010). Concentration of empirical distribution functions with applications to non-iid models. *Bernoulli*, 16(4):1385–1414.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brockwell, P. J., Davis, R. A., and Fienberg, S. E. (1991). Time series: theory and methods: theory and methods. *Springer Science & Business Media*.

Candanedo, L. M., Feldheim, V., and Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81–97.

Cauchois, M., Gupta, S., Ali, A., and Duchi, J. C. (2020). Robust validation: Confident predictions even when distributions shift.

Chen, X. and White, H. (1999). Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2):682–691.

Chernozhukov, V., Wüthrich, K., and Yinchu, Z. (2018). Exact and robust conformal inference methods for predictive machine learning with dependent data. volume 75 of *Proceedings of Machine Learning Research*, pages 732–749. PMLR.

Chernozhukov, V., Wuthrich, K., and Zhu, Y. (2020). An exact and robust conformal inference method for counterfactual and synthetic controls. *arXiv preprint arXiv:1712.09089*.

Cochran, J., Denholm, P., Speer, B., and Miller, M. (2015). Grid integration and the carrying capacity of the us grid to incorporate variable renewable energy. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Doukhan, P. (2012). Mixing: properties and examples. *Springer Science & Business Media*, 85.

Goldstein, M. and Dengel, A. (2012). Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63.

Hesse, C. (1990). Rates of convergence for the empirical distribution function and the empirical characteristic function of a broad class of linear processes. *Journal of Multivariate Analysis*, 35(2):186 – 202.

Ishimtsev, V., Bernstein, A., Burnaev, E., and Nazarov, I. (2017). Conformal k-NN anomaly detector for univariate data streams. In Gammerman, A., Vovk, V., Luo, Z., and Papadopoulos, H., editors, *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, volume 60 of *Proceedings of Machine Learning Research*, pages 213–227, Stockholm, Sweden. PMLR.

Izbicki, R., Shimizu, G., and Stern, R. (2020). Flexible distribution-free conditional predictive bands using density estimators. volume 108 of *Proceedings of Machine Learning Research*, pages 3068–3077, Online. PMLR.

Kim, B., Xu, C., and Barber, R. F. (2020). Predictive inference is free with the jackknife+-after-bootstrap.

Kivaranovic, D., Johnson, K. D., and Leeb, H. (2020). Adaptive, distribution-free prediction intervals for deep networks. volume 108 of *Proceedings of Machine Learning Research*, pages 4346–4356, Online. PMLR.

Kosorok, M. R. (2007). *Introduction to empirical processes and semiparametric inference*. Springer Science & Business Media.

Lathuilière, S., Mesejo, P., Alameda-Pineda, X., and Horaud, R. (2019). A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence*.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39.

Lucas, D., Kwok, C. Y., Cameron-Smith, P., Graven, H., Bergmann, D., Guilderson, T., Weiss, R., and Keeling, R. (2015). Designing optimal greenhouse gas observing networks that consider performance and cost. *Geoscientific Instrumentation, Methods and Data Systems*, 4(1):121.

Papadopoulos, H., Vovk, V., and Gammerman, A. (2007). Conformal prediction with neural networks. In *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, volume 2, pages 388–395.

Rio, E. (2017). *Asymptotic theory of weakly dependent random processes*, volume 80. Springer.

Romano, Y., Patterson, E., and Candes, E. (2019). Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, pages 3543–3553.

Romano, Y., Sesia, M., and Candès, E. (2020). Classification with valid and adaptive coverage. *arXiv: Methodology*.

Rosenfeld, N., Mansour, Y., and Yom-Tov, E. (2018). Discriminative learning of prediction intervals. volume 84 of *Proceedings of Machine Learning Research*, pages 347–355, Playa Blanca, Lanzarote, Canary Islands. PMLR.

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

Tibshirani, R. J., Barber, R. F., Candes, E., and Ramdas, A. (2019). Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, pages 2530–2540.

Volkhonskiy, D., Burnaev, E., Nouretdinov, I., Gammerman, A., and Vovk, V. (2017). Inductive conformal martingales for change-point detection. In *Conformal and Probabilistic Prediction and Applications*, pages 132–153. PMLR.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Zeni, G., Fontana, M., and Vantini, S. (2020). Conformal prediction: a unified review of theory and new challenges. *arXiv preprint arXiv:2005.07972*.

Zhang, S., Guo, B., Dong, A., He, J., Xu, Z., and Chen, S. X. (2017). Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457.

Appendix: Conformal prediction interval for dynamic time-series

7. Additional Derivations

We first present proofs of Lemma 1 and 2. In particular, proof of Lemma 1 relies on the assumption that the error process $\{\epsilon_i\}_{i\geq 1}$ is stationary and strongly mixing. As a consequence, we can extend the result in Lemma 1 by modifying Assumption 1. More precisely, Corollary 1 presents the overall bound on $|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha|$ where the error process is assumed to be independent. Corollary 2 presents the bound where $\epsilon_i = \sum_{j=1}^{\infty} \delta_j z_{i-j}$, which captures a broad class of stationary linear processes. Corollary 3 presents the bound where ϵ_i are generated by random symmetric matrices $\frac{1}{\sqrt{n}}\psi_{jk}, 1 \leq j, k \leq i$ and the density of ϵ_i satisfies a logarithmic Sobolev inequality. We restate the assumptions below and prove Lemma 1 and 2 afterwards.

7.1. Proof of Main Lemmas

Assumption 1 (Stationary and strongly mixing error process)**.** *Assume $\{\epsilon_t\}_{t\geq 1}$ are stationary and strongly mixing, with sum of mixing coefficients bounded by M. Their common CDF F also satisfies a Lipschitz condition with constant $L > 0$.*

Lemma 1. *Suppose Assumption 1 hold. Define $C_1 := (M/2)^{1/3}$. Then, for any training size T, there is an event A_T in the probability space of $\{\epsilon_t\}_{t=1}^{T}$, such that conditional on the event A_T,*

$$\sup_x |\tilde{F}(x) - F(x)| \leq C_1(\log T/T)^{1/3}.$$

Moreover

$$P(A_T^C) \leq C_1(\log T/T)^{1/3}.$$

Proof. Based on Lemma 3 below, we let $x_T := \left(\frac{1+4M}{T}(3 + \frac{\log T}{2\log 2})\right)^{1/3} \approx \left(\frac{M\log T}{2T}\right)^{1/3}$ and see that

$$\mathbb{P}\left(\sup_x |\tilde{F}(x) - F(x)| \leq \left(\frac{M\log T}{2T}\right)^{1/3}\right) \geq 1 - \left(\frac{M\log T}{T}\right)^{1/3}.$$

Thus, define the event A_T on which $\sup_x |\tilde{F}(x) - F(x)| \leq \left(\frac{M\log T}{2T}\right)^{1/3}$, whereby we have

$$\begin{aligned}
&\sup_x |\tilde{F}(x) - F(x)| \Big| A_T \leq C_1(\log T/T)^{1/3} \\
&P(A_T^C) \leq C_1(\log T/T)^{1/3}.
\end{aligned}$$

Furthermore, note that

$$\begin{aligned}
\sup_x |\tilde{F}(x) - F(x)| &\overset{(i)}{=} \mathbb{P}(F(\epsilon_{T+1}) \leq \sup_x |\tilde{F}(x) - F(x)|) \\
&= P([F(\epsilon_{T+1}) \leq \sup_x |\tilde{F}(x) - F(x)|] \cap A_T) + P([F(\epsilon_{T+1}) \leq \sup_x |\tilde{F}(x) - F(x)|] \cap A_T^C) \\
&\leq P(F(\epsilon_{T+1}) \leq \sup_x |\tilde{F}(x) - F(x)| \Big| A_T) + P(A_T^C) \\
&\overset{(ii)}{=} \sup_x |\tilde{F}(x) - F(x)| \Big| A_T + P(A_T^C),
\end{aligned}$$

where (i) holds because $F(\epsilon_{T+1}) \sim \text{Unif}[0,1]$ and $\sup_x |\tilde{F}(x) - F(x)| \in [0,1]$, and (ii) holds because $F(\epsilon_{T+1}) \sim \text{Unif}[0,1]$ and the distribution of $F(\epsilon_{T+1})$ is unaffected by the bound on $\sup_x |\tilde{F}(x) - F(x)|$ conditioning on A_T. Thus, we can provide a deterministic bound on $\sup_x |\tilde{F}(x) - F(x)|$ for any T. □

Assumption 2 (Estimation Quality)**.** *There exists a real sequence $\{\delta_T\}_{T\geq 1}$ that converges to zero such that*

$$\sum_{t=1}^{T}(\hat{\epsilon}_t - \epsilon_t)^2/T \leq \delta_T^2.$$

Lemma 2. *Assume Assumption 1 and 2 hold. Define $C_2 := L + 1$. With the same C_1 in Lemma 1, we have on the event A that*

$$\sup_x |\hat{F}(x) - \tilde{F}(x)| \leq C_2\delta_T^{2/3} + 2\sup_x |\tilde{F}(x) - F(x)|.$$

Proof. Let $S := \{i \in [T] : |\hat{\epsilon}_i - \epsilon_i| \geq \delta_T^{2/3}\}$. It follows that

$$|S|\delta_T^{4/3} \leq \sum_{i=1}^{T}(\hat{\epsilon}_i - \epsilon_i)^2 \leq T\delta_T^2,$$

where the second inequality follows by the first condition in Assumption 2. As a result, $|S| \leq T\delta_T^{2/3}$ and we see that for any $x \in \mathbb{R}$,

$$\begin{aligned}
|\hat{F}(x) - \tilde{F}(x)| &\leq T^{-1}\sum_i |\mathbf{1}\{\hat{\epsilon}_i \leq x\} - \mathbf{1}\{\epsilon_i \leq x\}| \\
&\overset{(i)}{\leq} T^{-1}(|S| + \sum_{i\notin S} |\mathbf{1}\{\hat{\epsilon}_i \leq x\} - \mathbf{1}\{\epsilon_i \leq x\}|) \\
&\overset{(ii)}{\leq} T^{-1}(|S| + \sum_{i\notin S} \mathbf{1}\{|\epsilon_i - x| \leq \delta_T^{2/3}\}) \\
&\leq \delta_T^{2/3} + \mathbb{P}(|\epsilon_{T+1} - x| \leq \delta_T^{2/3}) + \\
&\quad \sup_x |T^{-1}\sum_{i\notin S} \mathbf{1}\{|\epsilon_i - x| \leq \delta_T^{2/3}\} - \mathbb{P}(|\epsilon_{T+1} - x| \leq \delta_T^{2/3})| \\
&\overset{(iii)}{=} \delta_T^{2/3} + [F(x + \delta_T^{2/3}) - F(x - \delta_T^{2/3})] + \\
&\quad \sup_x |[\tilde{F}(x + \delta_T^{2/3}) - \tilde{F}(x - \delta_T^{2/3})] - [F(x + \delta_T^{2/3}) - F(x - \delta_T^{2/3})]| \\
&\leq (L+1)\delta_T^{2/3} + 2\sup_x |\tilde{F}(x) - F(x)|. \qquad (5)
\end{aligned}$$

We remark that (i) follows as we analyze $i \in S$ and $i \notin S$ separately, (ii) follows since $|\mathbf{1}\{a \leq x\} - \mathbf{1}\{b \leq x\}| \leq \mathbf{1}\{|b - x| \leq |a - b|\}$ for any constant a, b and univariate x and $|\hat{\epsilon}_i - \epsilon_i| \leq \delta_T^{2/3}$ for $i \notin S$, and (iii) follows since we assume $\{\epsilon_t\}_{t\geq 1}$ have common cdf F.

□

Theorem 1 (Approximately uniform p-value)**.** *Let $C_1 = (M/2)^{1/3}, C_2 = L + 1$. For any training size T and $\alpha \in (0, 1)$, the empirical p-value $\hat{p}_{T+1}$ obeys:*

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \leq 12C_1\left(\log T/T\right)^{1/3} + 2C_2\delta_T^{2/3}$$

Proof. Recall the following definitions:

- $\hat{p}_{T+1} := T^{-1}\sum_i \mathbf{1}\{\hat{\epsilon}_i > \hat{\epsilon}_{T+1}\}$, which is the empirical p-value defined using *estimated* residuals.
- $\tilde{p}_{T+1} := T^{-1}\sum_i \mathbf{1}\{\epsilon_i > \epsilon_{T+1}\}$, which is the empirical p-value using *actual* residuals.
- $\tilde{F}(x) := T^{-1}\sum_{i=1}^{T} \mathbf{1}\{\epsilon_i \leq x\}$, so that $\tilde{F}(\epsilon_{T+1}) = 1 - \tilde{p}_{T+1}$. Equivalently define $\hat{F}(x)$ for $\hat{p}_{T+1}$.

As a consequence, the following are equivalent:

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| = |\mathbb{P}(\hat{F}(\hat{\epsilon}_{T+1}) \geq 1 - \alpha) - \mathbb{P}(F(\epsilon_{T+1}) \geq 1 - \alpha)| \qquad (6)$$

We can rewrite the right hand side of (6) as follows:

$$
\begin{aligned}
&|\mathbb{P}(\hat{F}(\hat{\epsilon}_{T+1}) \geq 1-\alpha) - \mathbb{P}(F(\epsilon_{T+1}) \geq 1-\alpha)| \\
&\leq \mathbb{E}|\mathbf{1}\{1-\hat{F}(\hat{\epsilon}_{T+1}) \leq \alpha\} - \mathbf{1}\{1-F(\epsilon_{T+1}) \leq \alpha\}| \\
&\overset{(i)}{\leq} \mathbb{P}(|F(\epsilon_{T+1}) - (1-\alpha)| \leq |\hat{F}(\hat{\epsilon}_{T+1}) - F(\epsilon_{T+1})|) \\
&\overset{(ii)}{=} 2|\hat{F}(\hat{\epsilon}_{T+1}) - F(\epsilon_{T+1})| \\
&\leq 2|\hat{F}(\hat{\epsilon}_{T+1}) - \tilde{F}(\epsilon_{T+1})| + 2|\tilde{F}(\epsilon_{T+1}) - F(\epsilon_{T+1})| \\
&\leq 2\sup_x |\hat{F}(x) - \tilde{F}(x)| + 2\sup_x |\tilde{F}(x) - F(x)|,
\end{aligned}
\tag{7}
$$

where inequality (i) follows since $|\mathbf{1}\{a \leq x\} - \mathbf{1}\{b \leq x\}| \leq \mathbf{1}\{|b-x| \leq |a-b|\}$ for any constant a, b and univariate x and $\mathbb{E}[\mathbf{1}\{A\}] = \mathbb{P}(A)$, and (ii) follows because the distribution of $F(\epsilon_{T+1})$ is Unif$[0,1]$.

Since Lemma 1 says

$$
\begin{aligned}
&\sup_x |\tilde{F}(x) - F(x)| \leq \sup_x |\tilde{F}(x) - F(x)| \Big| A_T + P(A_T^C), \\
&\sup_x |\tilde{F}(x) - F(x)| \Big| A_T \leq C_1(\log T/T)^{1/3}, \\
&P(A_T^C) \leq C_1(\log T/T)^{1/3}
\end{aligned}
$$

and Lemma 2 says

$$
\sup_x |\hat{F}(x) - \tilde{F}(x)| \leq C_2\delta_T^{2/3} + 2\sup_x |\tilde{F}(x) - F(x)|,
$$

we have that

$$
2\sup_x |\hat{F}(x) - \tilde{F}(x)| + 2\sup_x |\tilde{F}(x) - F(x)| \leq 12C_1\left(\log T/T\right)^{1/3} + 2C_2\delta_T^{2/3}
$$

□

Lemma 3. *Suppose Assumption 1 holds. Then for any sequence x_T converging to zero,*

$$
\mathbb{P}(\sup_x |\tilde{F}(x) - F(x)| \leq x_T) \geq 1 - \frac{1+4M}{Tx_T^2}(3 + \frac{\log T}{2\log 2})
$$

Proof. Define $v_T(x) := \sqrt{T}(\tilde{F}(x) - F(x))$. Then, Corollary 7.1 in (Rio, 2017) shows that

$$
\mathbb{E}(\sup_x |v_T(x)|^2) \leq (1+4M)(3 + \frac{\log T}{2\log 2}).
$$

Therefore, Markov Inequality shows that

$$
\begin{aligned}
\mathbb{P}(\sup_z |\tilde{F}(x) - F(x)| \geq x_T) &\leq \frac{\mathbb{E}(\sup_x |v_T(x)|^2/T)}{x_T^2} \\
&\leq \frac{1+4M}{Tx_T^2}(3 + \frac{\log T}{2\log 2}).
\end{aligned}
$$

□

7.2. Corollaries Under Other Error Assumptions

Upon scrutinizing the proof of Lemma 3 above, we notice that the crux to bounding $\sup_x |\tilde{F}(x) - F(x)|$ is to find appropriate sequences x_T and $g(x_T)$, both of which converge to zero, such that

$$
\mathbb{P}(\sup_x |\tilde{F}(x) - F(x)| > x_T) \leq g(x_T).
$$

The optimal rate of decay then reduces to finding x_T such that $x_T = g(X_T)$ and we can define A_T on which $\sup_x |\tilde{F}(x) - F(x)| \leq x_T$. Therefore, without modifying the underlying proof technique, we can yield different decay factors under different assumptions on the error process, as shown below.

Corollary 1. *Suppose that $\{\epsilon_i\}_{i=1}^{T+1}$ are independent and identically distributed according to cdf F, with F having Lipschitz constant L and that Assumption 2 holds, then for any $\alpha \in [0,1]$,*

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \leq 12 \left(\log(16T)/T\right)^{1/2} + (2L+2)\delta_T^{2/3}.$$

Proof. When the error process is iid, the famous Dvoretzky–Kiefer–Wolfowitz inequality (Kosorok, 2007, p.210) implies that

$$\mathbb{P}(\sup_x |\tilde{F}(x) - F(x)| > x_T) \leq 2e^{-2Tx_T^2}.$$

Thus, by picking $x_T = \frac{\sqrt{W(16T)}}{2\sqrt{T}}$, where $W(T)$ is the Lambert W function that satisfies $W(T)e^{W(T)} = T$, we see that $x_T \leq \left(\frac{\log(16T)}{T}\right)^{1/2}$. Therefore, following the proof strategy of Theorem 1, we have

$$\begin{aligned} |\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| &\leq 12x_T + (2L+2)\delta_T^{2/3} \\ &\leq 12\left(\frac{\log(16T)}{T}\right)^{1/2} + (2L+2)\delta_T^{2/3}. \end{aligned}$$

□

Corollary 2. *Suppose that $\{\epsilon_i\}_{i=1}^{T+1}$ satisfy $\epsilon_i = \sum_{j=1}^{\infty} \delta_j z_{i-j}$ for each i and are identically distributed according to cdf F, with F having Lipschitz constant L. Assume that z_{i-j} are iid with finite first absolute moment and δ_j are bounded in absolute value by some function g such that $\sum_{i=1}^{\infty} ig(i)$ converges. Lastly, if Assumption 2 holds, then for any $\alpha \in [0,1]$ and some constant C,*

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \leq 12C\left(\log T/\sqrt{T}\right) + (2L+2)\delta_T^{2/3}.$$

Proof. The assumptions on ϵ_i were first imposed by (Hesse, 1990), who stated that $\sup_x |\tilde{F}(x) - F(x)| = O(\log T/\sqrt{T})$ (see Hesse, 1990, Theorem 3). This guarantee yields the desired result as we follow the proof strategy of Theorem 1. □

Corollary 3. *Suppose $\{\epsilon_i\}_{i=1}^{T+1}$ satisfy that for each i, ϵ_i are generated by random symmetric matrices $\frac{1}{\sqrt{n}}\psi_{jk}, 1 \leq j,k \leq i$ and identically distributed according to cdf F, with F having Lipschitz constant L. Assume that F satisfies a logarithmic Sobolev inequality with constant σ^2, which states that for all bounded smooth g,*

$$Ent_F(g^2) \leq 2\sigma^2 \int |\nabla g|^2 dF,$$

where $Ent_F(g^2) = \int g \log g dF - \int g dF \log(\int g dF)$ denotes the entropy of $g \geq 0$ under F.

Furthermore, if Assumption 2 holds, then for any $\alpha \in [0,1]$,

$$|\mathbb{P}(\hat{p}_{T+1} \leq \alpha) - \alpha| \leq 12C' \left(\log(cT)/T\right)^{1/3} + (2L+2)\delta_T^{2/3},$$

where $C' = (L\sigma/\sqrt{a})^{2/3}, c = 12a/(L\sigma)^2$ for some positive absolute constant a.

Proof. Under these assumptions of ϵ_i, (Bobkov et al., 2010) proved that

$$\mathbb{P}(\sup_x |\tilde{F}(x) - F(x)| > x_T^2) \leq 4e^{-aTx_T^3/(M\sigma)^2}.$$

Thus, by picking $x_T = C'\left(\frac{W(cT)}{T}\right)^{1/3}$ for C' and c defined above and $W(T)$ being the Lambert W function, we see that $x_t \leq C'\left(\frac{\log(cT)}{T}\right)^{1/3}$. The result then follows as we follow the proof strategy of Theorem. 1 □

8. Additional Experiments

We provide a quick overview of how this section is organized: In Section 8.1, we first describe and visualize the renewable energy time-series. We then clarify input parameters to $\mathcal{A}$ in the `EnbPI` algorithm. In Section 8.2, we briefly compare `EnbPI`against J+aB (Barber et al., 2019b) and mainly present multi-step ahead inference results on Atlanta solar radiation data when $s = 5$ and $s = \infty$. In Section 8.3, we provide experimental results on the network California solar data and Austin wind data, similar to those on Atlanta solar data. In Section 8.3.1, we first show the interval validity of `EnbPI`. In Section 8.3.2, we then apply `EnbPI` on the more challenging multi-step ahead inference task, with and without the presence of missing data. In Section 8.4, we apply `EnbPI` on other datasets, such as greenhouse gas emission data, air pollution data, and appliances energy data. We observe that `EnbPI` never loses marginal validity and often produces shorter intervals than competing methods. We do not study multi-step ahead inference on these dataset as we primarily aim to demonstrate the applicability of `EnbPI`. In Section 8.5, we first presents the `ECAD` algorithm, which performs supervised anomaly detection as a modification of `EnbPI`, and then provide details of the 8 other competing anomaly detection methods.

8.1. Data Visualization and Regression Algorithm Parameters

Data Description and Raw Data Plot. The solar dataset is available at `https://nsrdb.nrel.gov/`. The 9 cities we chose are Fremont, Milpitas, Mountain View, North San Jose, Palo Alto, Redwood City, San Mateo, Santa Clara, Sunnyvale. The wind dataset is publically available at `https://github.com/Duvey314/austin-green-energy-predictor`. Figure 5 shows the raw data plot of the first 2000 data points from the three renewable energy datasets. We can see periodic fluctuations of different magnitude, which actually persist throughout the whole dataset. These fluctuations indicate changing mean and variance, with certain seasonal patterns that leads to clear non-stationary in the data.

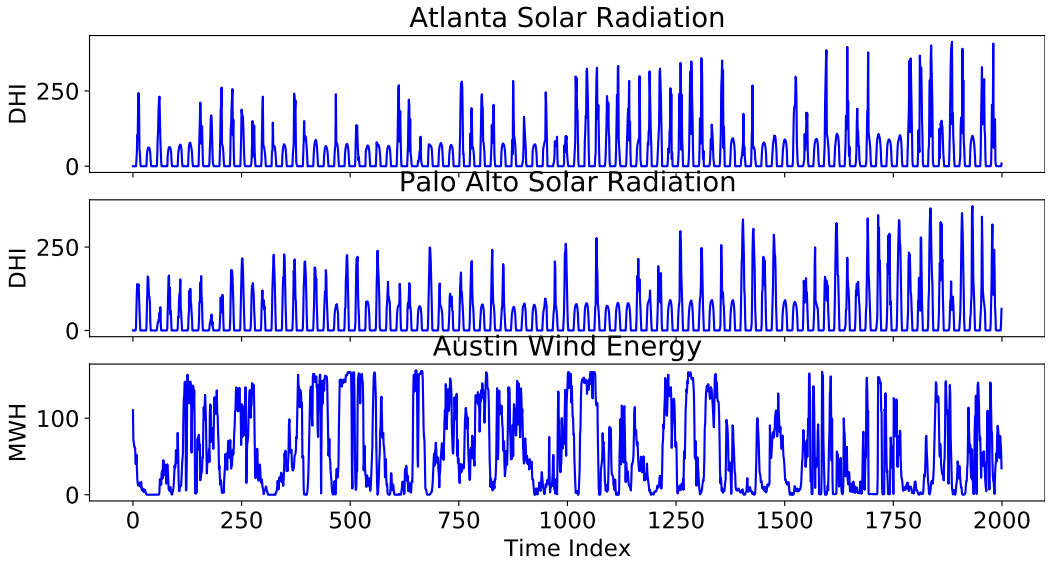


Figure 5. Plot of first 2000 response data points for the three renewable energy datasets.

Input parameters to $\mathcal{A}$. Below are the parameter specifications for the four baseline models $\mathcal{A}$, unless otherwise specified:

- For ridge, the penalty parameter α is chosen with generalized cross-validation over ten uniformly spaced grid points between 0.0001 to 10 (the package default α is 1). Higher α means more robust regularization.
- For RF, we build ten trees under the mean-squared-error (MSE) criterion. We restrict the maximum search depth of each tree as 2 for faster training. We only allow each tree to split into features rather than samples, so that combining RFs trained on subsets of the training data is reasonable for `EnbPI`.
- For NN, we add three hidden layers, each having 100 hidden nodes, and apply 20% dropout after the second hidden layer to avoid overfitting. We use the Relu activation between hidden layers. The optimizer is Adam with a fixed

learning rate of 5×10^{-4} under the MSE loss. Batch size is fixed at 100 with a maximum of 1000 epochs. We also use early stopping if there is no improvement in training error after ten epochs.

- For RNN, we add two hidden LSTM layers, followed by a dense output layer. Each LSTM layer has 100 hidden neurons, so the output from the first hidden layer is fed into the second hidden layer. We use the Tanh activation function for these two hidden layers and the Relu activation function for the dense layer. The optimizer is Adam with a fixed learning rate of 5×10^{-4} under the MSE loss. The batch size is fixed at 100 with a maximum of 100 epochs. We use early stopping if there is no improvement in training error after 10 epochs.

8.2. Additional Results on Solar—Atlanta

Comparison with J+aB. Figure 6 makes it clear that unlike the other two methods (i.e. Split conformal used in (Chernozhukov et al., 2018; 2020) and J+aB in (Chernozhukov et al., 2018)), `EnbPI` always retains marginally valid coverage under any regression model $\mathcal{A}$ we considered.

Multi-step ahead inference when $s = 14$. Figure 7, in comparison to Figure 3, shows the conditional coverage plot without missing data. Although we expect `EnbPI` performs better when missing data do not exist, the conditional coverage in these Figures are almost the same at each hour. Thus, `EnbPI` is robust under modest amount of missing data.

Multi-step ahead inference when $s = 5$. As observed in Figure 3, we demonstrate how to restore conditionally valid coverage at hours near noon (e.g. 10AM—2PM). We do so by fitting `EnbPI` only on the first three months' of data coming from these hours and make prediction on the rest nine months. As shown in Figure 8, it is clear that regardless of the choice of $\mathcal{A}$ or whether missing data are present, conditional coverage is maintained at 90% level at any of these hours.

Multi-step ahead inference when $s = \infty$. We train on the same set of data as above. Since change points are present in the data (i.e. data near summer have very different radiation levels from the training data), we expect `EnbPI` to perform less well if it does not slide. Indeed, Figure 9 shows poor conditional coverage under ridge and RF, even if we train on the same set of data as in Figure 8 and further assume *no missing data exist*. Nevertheless, we have shown how to restore conditionally valid coverage by picking a small and reasonable s, as in Figure 8 above ($s = 5$).

Remark 4 (Choices of X_t and regression model $\mathcal{A}$)**.** *As the choice of x_t makes no clear difference on interval validity, we only present results from using multivariate x_t (i.e., exogenous time-series such as humidity, temperature, wind speed, etc.), so we need not impute missing entries for $s > 1$ in the case of using univariate x_t. On the other hand, we primarily show results under ridge or RF regression, since these models are more interpretable than NN and RNN but results are similar.*

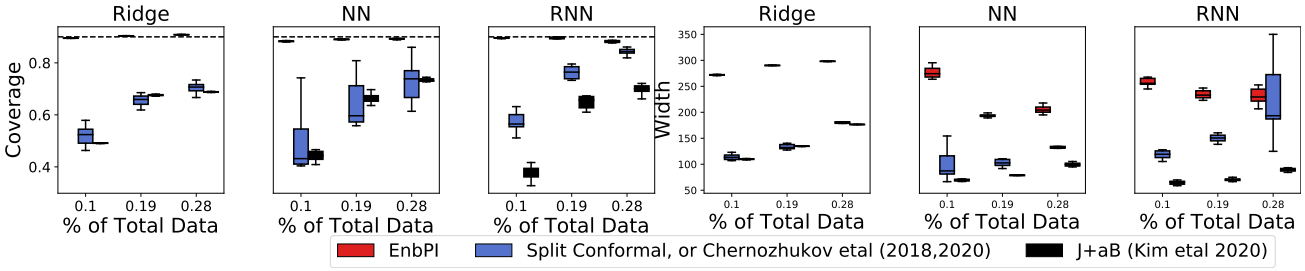


Figure 6. The setup is the same as Figure 2 in main text. Unlike the others, `EnbPI` almost always retains marginally valid coverage under any regression model $\mathcal{A}$ we considered.

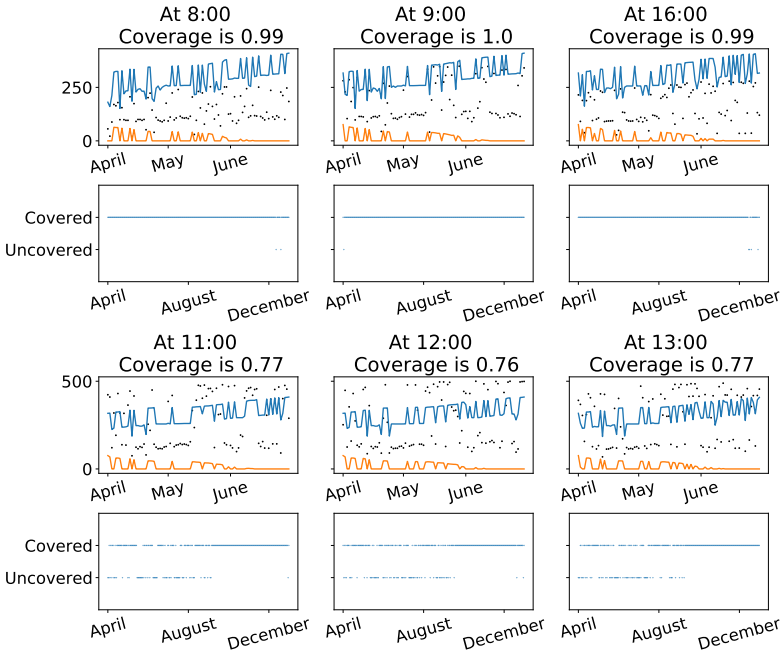
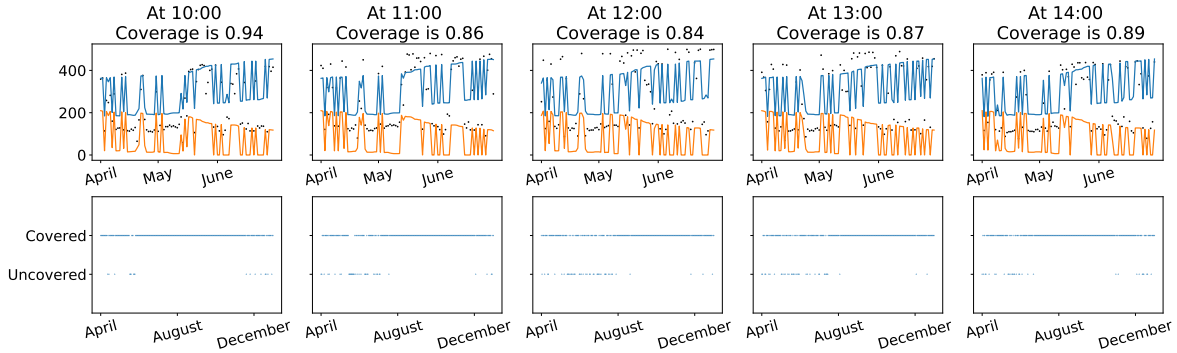
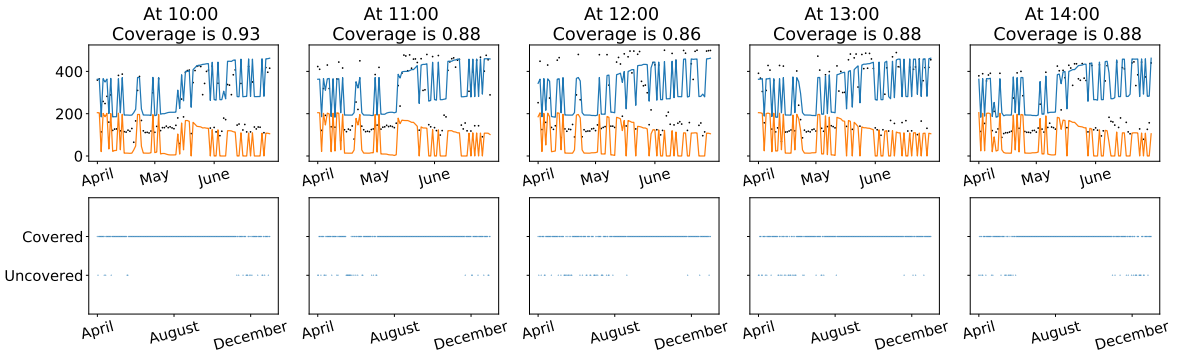


Figure 7. Solar—Atlanta, multi-step ahead prediction without missing data: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December).

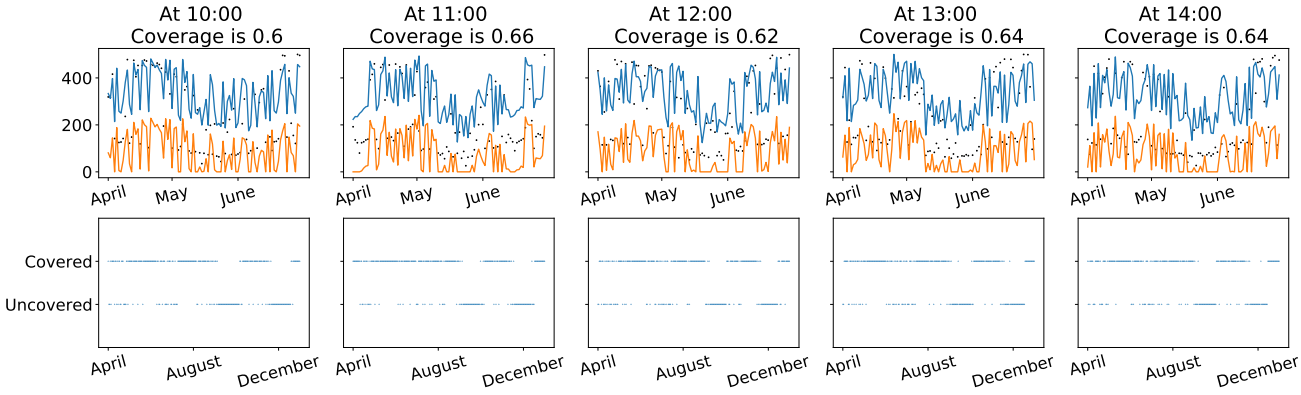


(a) $s = 5$: `EnbPI` under RF without missing data

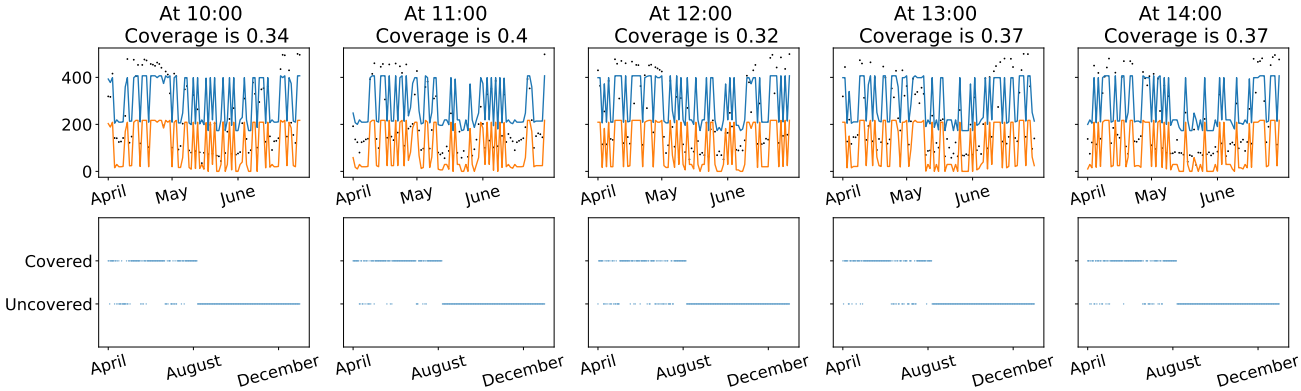


(b) $s = 5$: `EnbPI` under RF with missing data

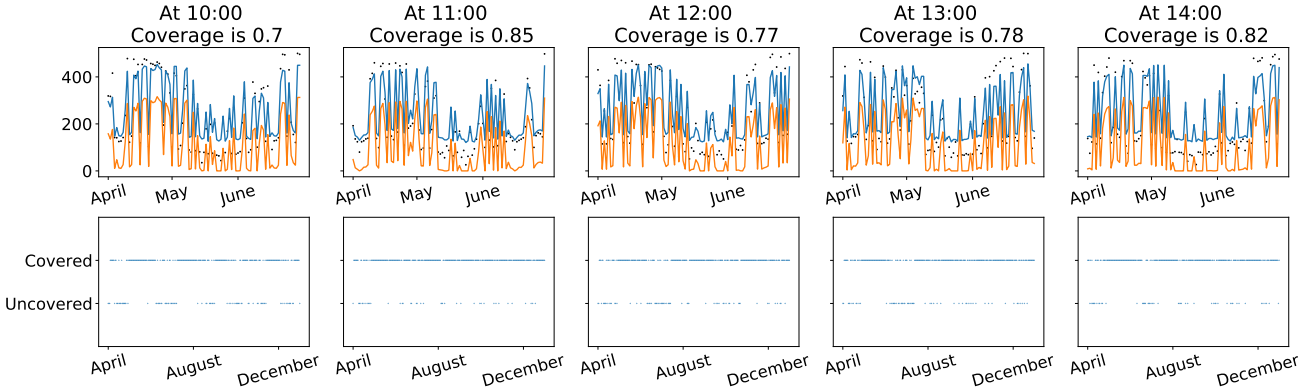
Figure 8. Solar—Atlanta, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December).



(a) $s = \infty$: `EnbPI` under ridge without missing data



(b) $s = \infty$: `EnbPI` under RF without missing data



(c) $s = \infty$: `EnbPI` under NN without missing data

Figure 9. Solar—Atlanta, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December).

8.3. Results on Solar—California and Wind—Austin

We note that in general, `EnbPI` on California solar data and on Austin wind data generates results very similar to those on Atlanta solar data. Therefore, we do not provide separate analyses of individual figures but highlight the overall pattern and differences in each part below.

8.3.1. Interval Validity of EnbPI

On California solar data: The whole California data constitute a network. Thus, the major difference from using the Atlanta solar data is that x_t^k at each city k is defined to include spatial-temporal information from other Californian cities. In general, results on different Californian cities look very similar to each other so that we only provide plots on the Palo Alto solar data.

Figure 10 shows coverage/width vs. $1-\alpha$ line plots as Figure 1. We note that `EnbPI` is always valid at different $1-\alpha$ levels regardless of $\mathcal{A}$ being used. However, it is no longer the best prediction inference method since ARIMA intervals are also valid but shorter. Nevertheless, since ARIMA can greatly lose coverage (see Figure 1 on the Atlanta solar data), `EnbPI` outperforms ARIMA in terms of stability and applicability. On the other hand, Figure 11 shows the boxplots as Figure 2. It is clear that `EnbPI` is still almost always valid regardless of the number of training data whereas ICP and WeightedICP can greatly lose coverage at times. Lastly, we summarize the performance of ARIMA, `EnbPI`, ICP, and WeightedICP under different regression functions on all Californian cities. Details are in Table 2 for ridge regression. We only show results for ridge because results under different $\mathcal{A}$ are similar. We can see from the table that ARIMA yields the shortest intervals among all methods. On the other hand, Winkler score[4] by `EnbPI` using ridge regression is often the smallest so that it reaches a better balance between validity and efficiency. Meanwhile, ICP and WeightedICP can greatly lose coverage so that they should not be used for dynamic time-series data.

On the other hand, because `EnbPI` performs very similarly on the wind data, we will only apply it on the more challenging multi-step ahead and missing data inference tasks in section 8.3.2.

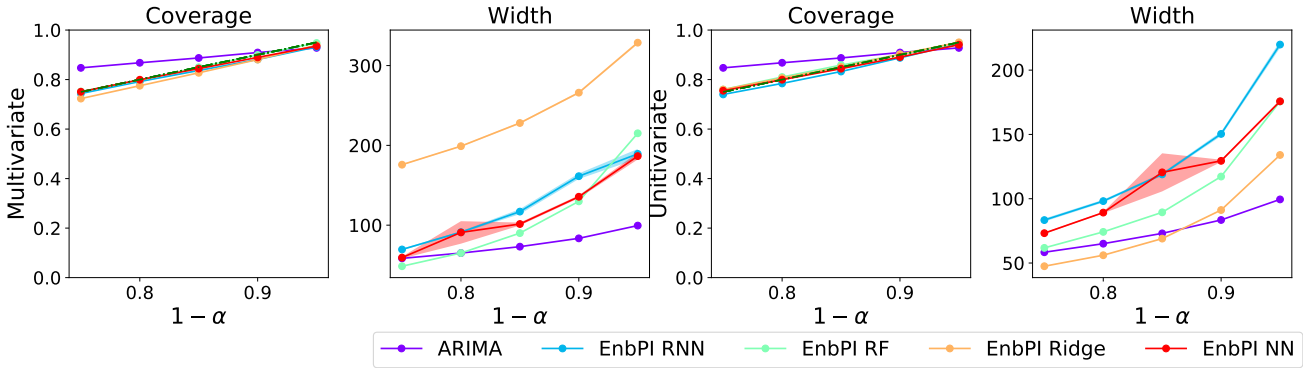


Figure 10. Solar—Palo Alto: average coverage and width vs. $1-\alpha$ target coverage by `EnbPI` under different regression algorithms and by ARIMA. Five equally spaced $1-\alpha \in [0.75, 0.95]$ are chosen. The green dash-dotted line at 0.9 represents the target coverage.

[4]Let the upper and lower end of the prediction interval at time t under level α be $L_t(\alpha), U_t(\alpha)$, so width is $W_t(\alpha) = U_t(\alpha) - L_t(\alpha)$. Then, Winkler score (WS) is:

$$(WS)_t = \begin{cases} W_t(\alpha), & \text{if } L_t(\alpha) \le y_t \le U_t(\alpha) \\ W_t(\alpha) + 2 \cdot \frac{L_t(\alpha) - y_t}{\alpha}, & \text{if } y_t < L_t(\alpha) \\ W_t(\alpha) + 2 \cdot \frac{y_t - U_t(\alpha)}{\alpha}, & \text{if } y_t > U_t(\alpha) \end{cases}$$

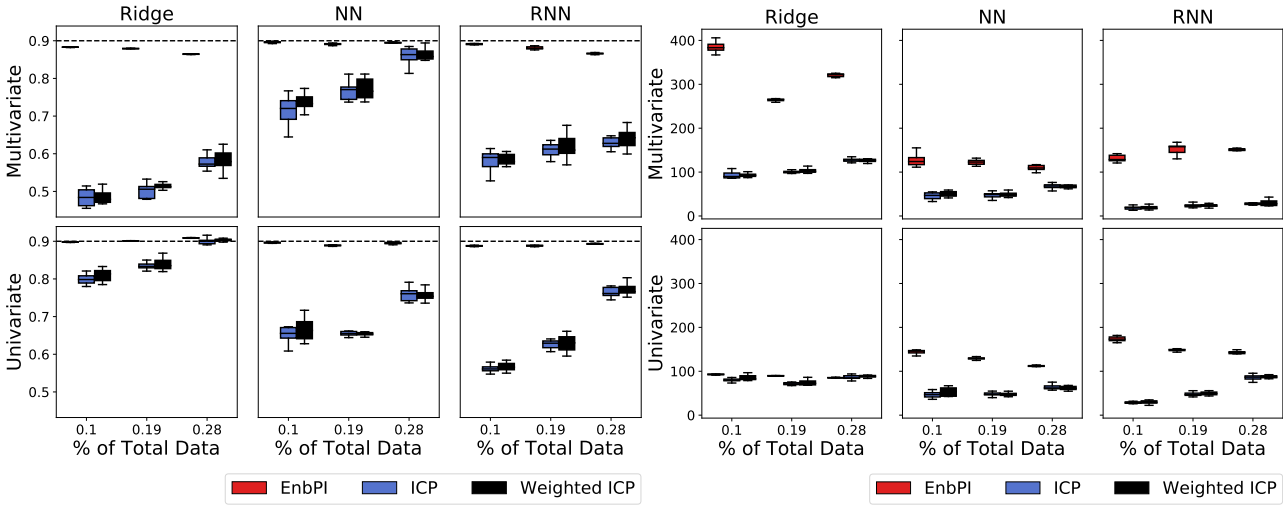


Figure 11. Solar—Palo Alto: boxplots of average coverage (top) and width (bottom) by `EnbPI`, ICP, and WeightedICP, whose training data vary as a percentage of total data (x-axis). Each box contains results from 10 independent trials. The black dash dotted line at 0.9 indicates target coverage.

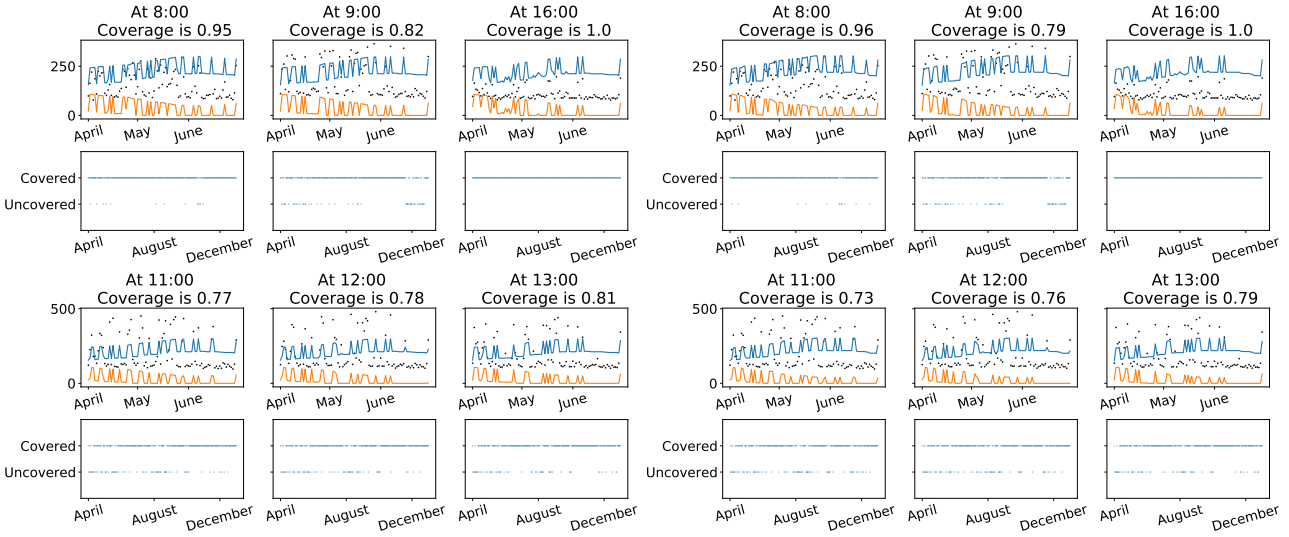
Table 2. Ridge: under validity constraint, the smallest widths and Winkler scores among all methods are in bold.

		Multivariate			**Univariate**		
Dataset	Method	Coverage	Width	Winkler Score	Coverage	Width	Winkler Score
Fremont	ARIMA	0.91	**89.47**	1.16e+06			
	Ensemble	0.88	260.30	2.53e+06	0.90	94.77	**1.11e+06**
	ICP	0.57	107.56	3.84e+06	0.85	78.80	1.28e+06
	WeightedICP	0.56	107.92	3.96e+06	0.84	75.27	1.29e+06
Milpitas	ARIMA	0.90	**83.60**	1.19e+06			
	Ensemble	0.88	262.30	2.54e+06	0.90	89.84	**1.08e+06**
	ICP	0.57	105.63	3.93e+06	0.85	76.43	1.22e+06
	WeightedICP	0.56	105.90	4.05e+06	0.85	74.62	1.25e+06
Mountain View	ARIMA	0.90	**78.54**	**1.07e+06**			
	Ensemble	0.88	271.78	2.61e+06	0.90	88.85	1.07e+06
	ICP	0.54	107.39	4.15e+06	0.86	78.28	1.21e+06
	WeightedICP	0.54	107.21	4.30e+06	0.85	74.00	1.23e+06
North San Jose	ARIMA	0.90	**84.71**	1.14e+06			
	Ensemble	0.88	265.11	2.59e+06	0.90	93.36	**1.08e+06**
	ICP	0.57	106.46	3.94e+06	0.85	77.99	1.20e+06
	WeightedICP	0.57	107.56	4.03e+06	0.85	76.56	1.23e+06
Palo Alto	ARIMA	0.91	**83.35**	**1.08e+06**			
	Ensemble	0.88	266.54	2.56e+06	0.90	91.20	1.10e+06
	ICP	0.54	107.13	4.09e+06	0.85	77.96	1.25e+06
	WeightedICP	0.53	106.38	4.20e+06	0.85	74.21	1.26e+06
Redwood City	ARIMA	0.91	**84.67**	1.11e+06			
	Ensemble	0.88	268.88	2.59e+06	0.90	90.65	**1.09e+06**
	ICP	0.55	106.46	4.07e+06	0.87	85.17	1.25e+06
	WeightedICP	0.55	105.88	4.12e+06	0.86	80.69	1.27e+06
San Mateo	ARIMA	0.89	**88.33**	1.17e+06			
	Ensemble	0.88	258.66	2.49e+06	0.90	95.56	**1.14e+06**
	ICP	0.57	109.63	3.70e+06	0.86	84.41	1.33e+06
	WeightedICP	0.57	108.32	3.83e+06	0.85	82.85	1.31e+06
Santa Clara	ARIMA	0.91	**85.38**	1.11e+06			
	Ensemble	0.89	257.84	2.47e+06	0.90	88.21	**1.06e+06**
	ICP	0.58	108.26	3.76e+06	0.85	75.61	1.20e+06
	WeightedICP	0.58	110.61	3.82e+06	0.85	75.22	1.22e+06
Sunnyvale	ARIMA	0.91	**84.44**	**1.06e+06**			
	Ensemble	0.88	275.88	2.64e+06	0.90	88.85	1.08e+06
	ICP	0.55	109.82	4.16e+06	0.86	79.31	1.22e+06
	WeightedICP	0.55	110.32	4.25e+06	0.86	77.86	1.24e+06

8.3.2. MULTI-STEP AHEAD PREDICTIVE INFERENCE

On California solar data: Figure 12 shows *conditional* coverage of `EnbPI` under RF at certain hours of the day, with and without the presence of missing data. We let $s = 15$ since sensors have 15 hourly non-zero recordings in any day and keep other parameter settings the same as those in Figure 3 (Solar—Atlanta). and the figure look very similar to that one, where conditional coverage by `EnbPI` is still valid on all hours except 10AM—2PM. On the other hand, Figure 13 presents results by fitting only on data between 10AM—2PM (similar to Figure 8), where `EnbPI` reaches conditional coverage at all these hours.

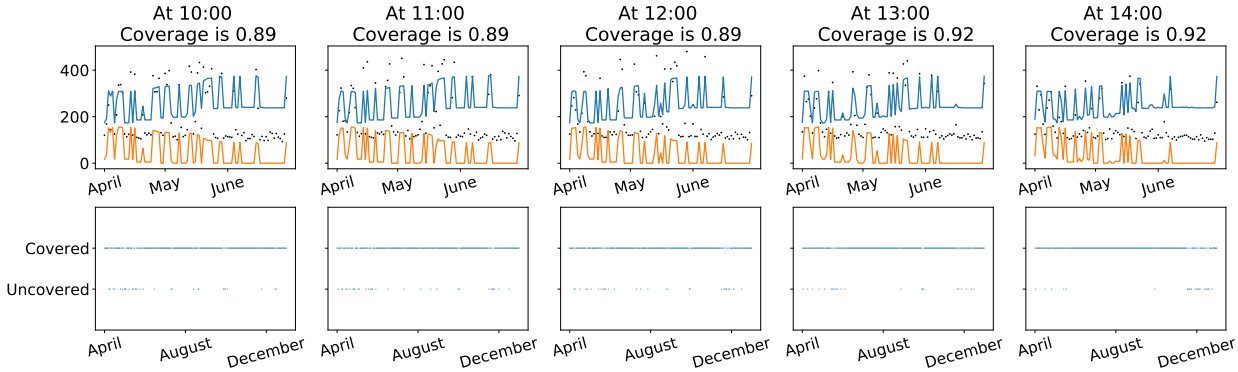
On Wind solar: Figure 14 shows conditional coverage of multi-step ahead inference of `EnbPI` under RF, with and without missing data. No ambient information is available as features in this data so we can only use past history of the wind power as response (i.e., x_t is the history of y_t). Note, one difference from earlier solar results is that we do not choose $s = 5$, but only train `EnbPI` on the whole 24 hourly data (e.g., $s = 24$). We do so because conditional coverage on hours different from 10AM—2PM may also not be valid, so that fitting `EnbPI` on data from 10AM—2PM ($s = 5$) is not very helpful. Nevertheless, `EnbPI` still maintains valid approximately conditional coverage at most hours, even under the presence of missing data. To restore valid coverage at certain hours, we suggest applying `EnbPI` on subgroups of hourly data separately, as we did for solar—Atlanta (Figure 13) and picked $s = 5$ instead.



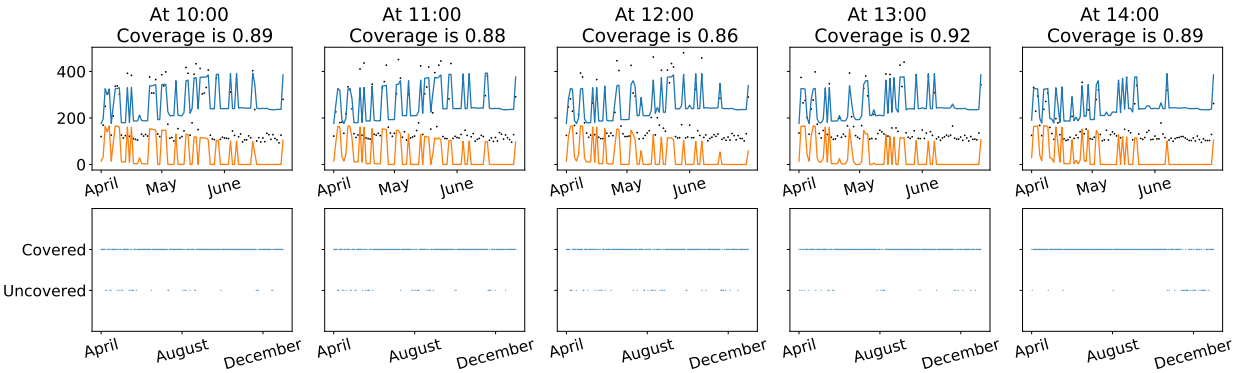
(a) $s = 15$: `EnbPI` under RF without missing data

(b) $s = 15$: `EnbPI` under RF with missing data

Figure 12. Solar—Palo Alto, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December). We selectively show coverage at three hours from 10AM—2PM and at three other hours with observations, because conditional coverage from 10AM—2PM is much poorer than the rest.

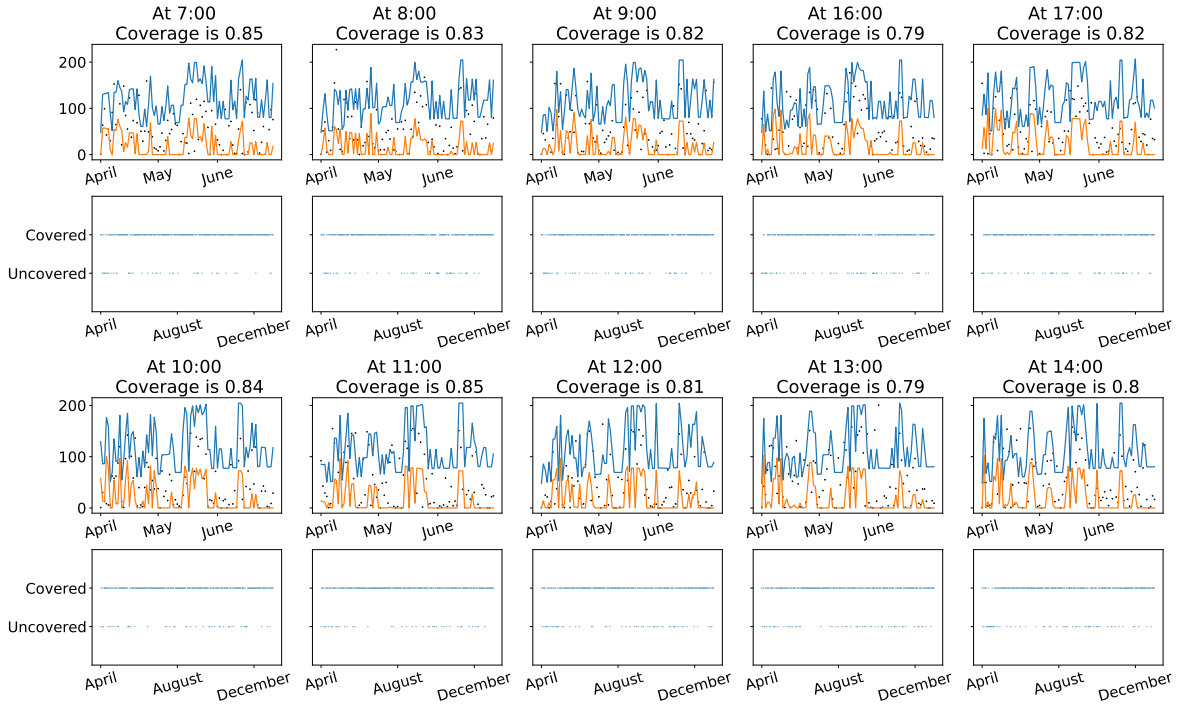


(a) $s = 5$: `EnbPI` under RF without missing data

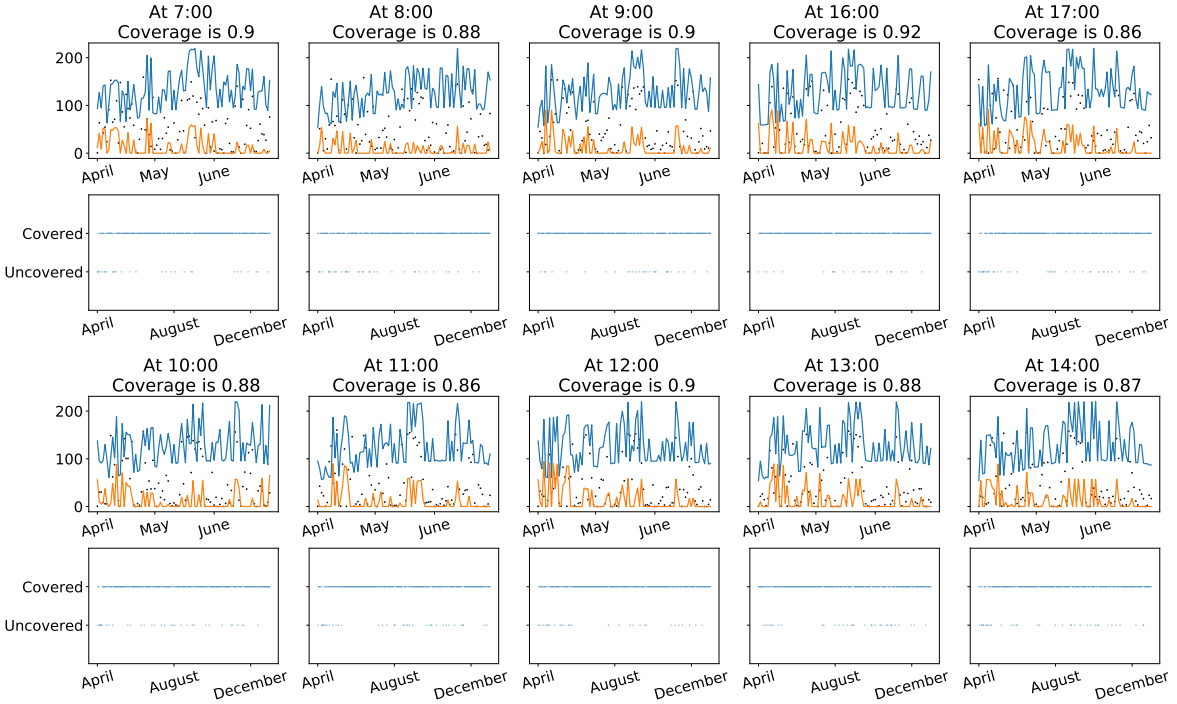


(b) $s = 5$: `EnbPI` under RF with missing data

Figure 13. Solar—Palo Alto, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December).



(a) $s = 24$: `EnbPI` under RF without missing data



(b) $s = 24$: `EnbPI` under RF with missing data

Figure 14. Wind—Austin, multi-step ahead prediction: We plot the upper and lower prediction intervals in blue and orange on top of the actual data for three months (April-June), and examine whether intervals fail to cover throughout the test period (April-December). We show all results near noon and selectively show results at four other hours (7-9AM and 4-5PM). We see no obvious differences between the behavior of `EnbPI` under missing data.

8.4. Results on Datasets in Other Domains

Description. We describe the additional three datasets being used, which are greenhouse gas emission data, air pollution data, and appliances energy data. The first dataset contains Greenhouse Gas observation (Greenhouse) (Lucas et al., 2015) from 5.10 till 7.31, 2010, with four samples every day and 6 hours apart between data points. The goal is to find the optimal weights for the 15 observation series to match the synthetic control series. The second dataset contains appliances energy usage (Appliances) (Candanedo et al., 2017). Consecutive data points are 10 minutes apart for about 4.5 months. We can use 27 different humidity and temperature indicators to predict the appliances' energy use in Wh. The third dataset on Beijing air quality (Beijing air) (Zhang et al., 2017) contains air pollutants data from 12 nationally-controlled air-quality monitoring sites. The data is from 3.1, 2013 to 2.28, 2017. The goal is to predict PM2.5 air pollutant levels using 10 different air pollutants and meteorological variables. We use the data from the Tiantan district.

Results. We first show additional average coverage and width vs. $1-\alpha$ line plots as in Figure 1 (Solar—Atlanta). Then, we present grouped boxplots using both multivariate and univariate X_t as in Figure 2 (Solar—Atlanta). We do not study multi-step ahead inference on these dataset as we primarily aim to demonstrate the applicability of `EnbPI`.

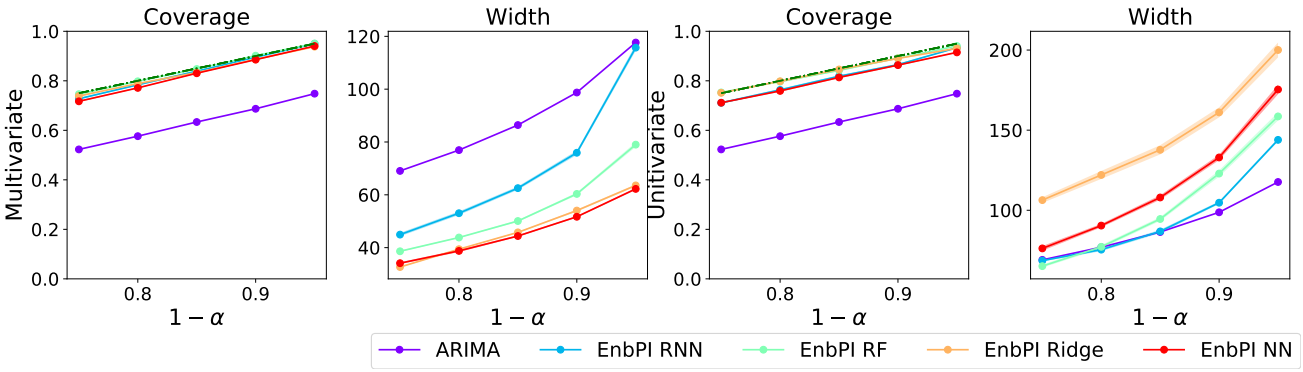
(1) Observations from the other coverage/width vs. $1-\alpha$ plots:

- Figure 15 (a) (Greenhouse): `EnbPI` never loses coverage under any $\mathcal{A}$ and intervals are the shortest under NN on the multivariate version. In contrast, ARIMA can significantly lose coverage, although its intervals are shorter than other univariate ones by `EnbPI`.
- Figure 15 (b) (Appliances Energy): ARIMA no longer loses coverage, but its intervals are much wider than `EnbPI` under any $\mathcal{A}$ on the univariate version, so that ARIMA intervals are not efficient.
- Figure 15 (c) (Beijing Air): ARIMA also does not lose coverage. However, its intervals are much wider than `EnbPI` under ridge and RF, so that ARIMA intervals are not efficient.

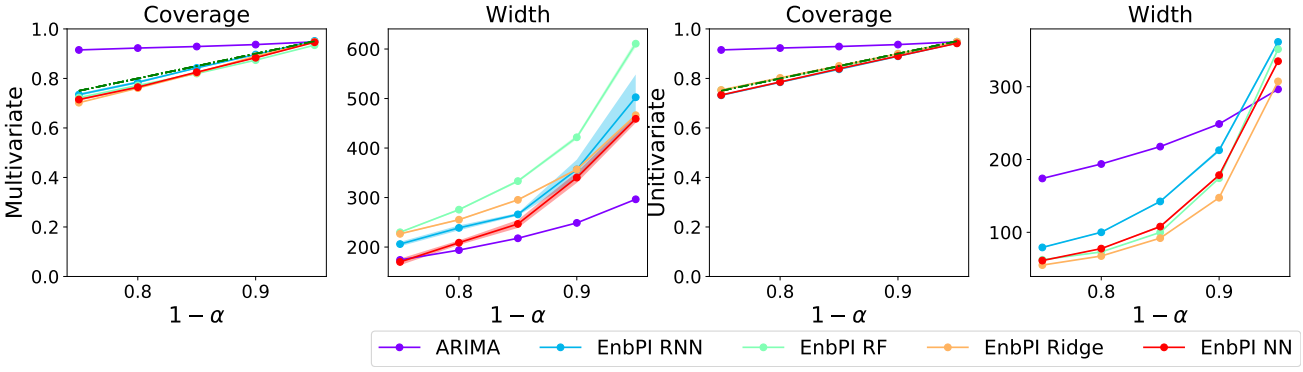
(2) Observations from the other grouped boxplots:

- Figure 16 (a) (Greenhouse): `EnbPI` almost always maintain valid coverage, whereas ICP and WeightedICP can significantly lose coverage significantly (see NN on univariate). Overall, `EnbPI` coverage and widths results show much less variance than the other ones.
- Figure 16 (b) (Appliances Energy) reveals similar patterns. In particular, ICP and WeightedICP can significantly lose coverage significantly (see ridge on multivariate). Moreover, ICP and WeightedICP also have much higher variance than `EnbPI` (see RNN on multivariate). Overall, we notice that intervals on univariate versions are much shorter than those on multivariate versions, likely because the past history of energy use is more predictive of future energy use than the exogenous variables such as humidity and temperature of the ambient space (e.g. kitchen, bathroom, living room, etc.).
- Figure 16 (c) (Beijing Air) shows very similar results as Figure 16 (b) (Appliances Energy), so that we omit its discussion.

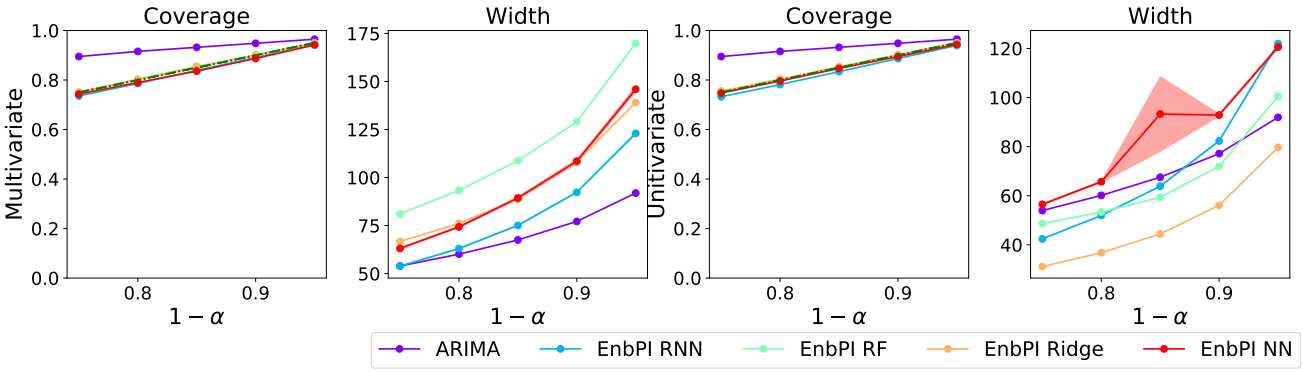
In general, we think `EnbPI` is stable across different combinations of regression algorithms and datasets. Since other CP methods such as ICP and WeightedICP can severely lose coverage, we advocate the use of `EnbPI` for time-series predictive inference. Regarding interval width, using the history of the response (univariate version) to predict its future values tends to yield shorter intervals.



(a) Greenhouse Gas

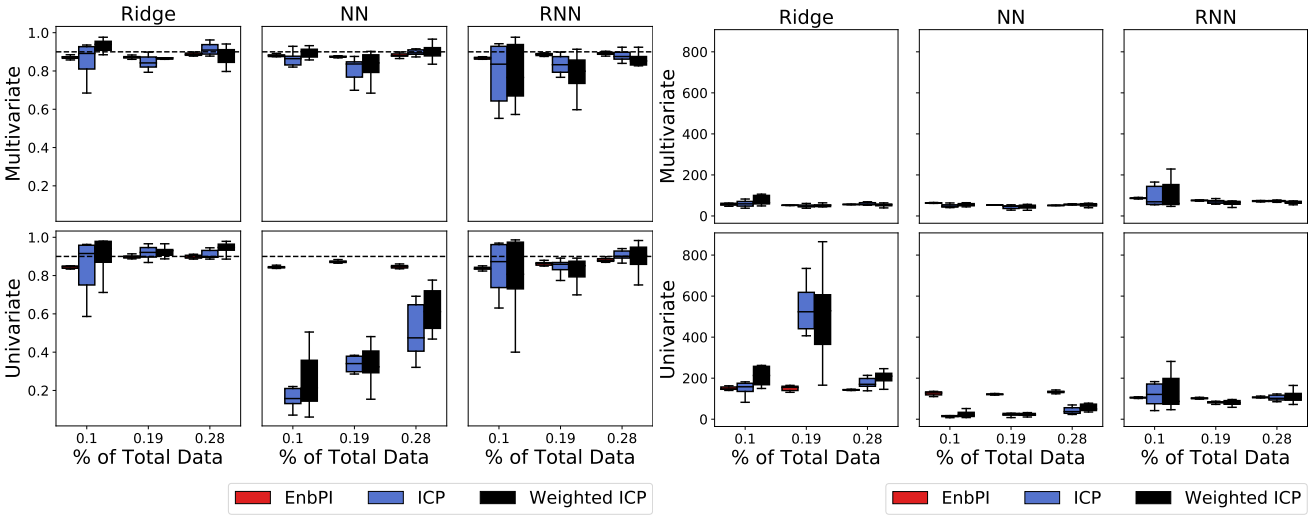


(b) Appliances Energy

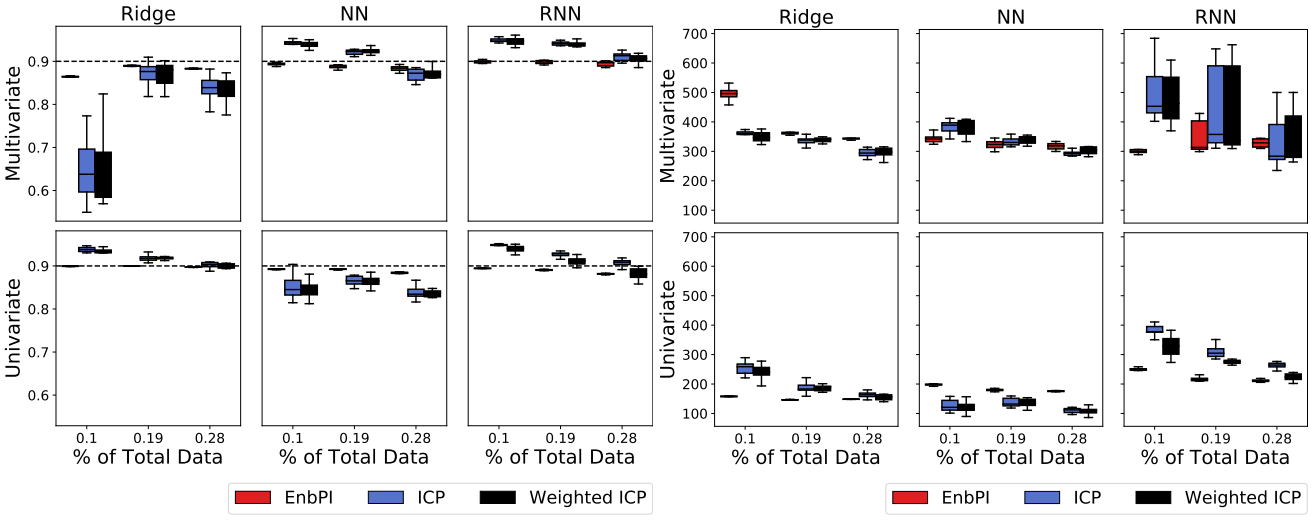


(c) Beijing Air

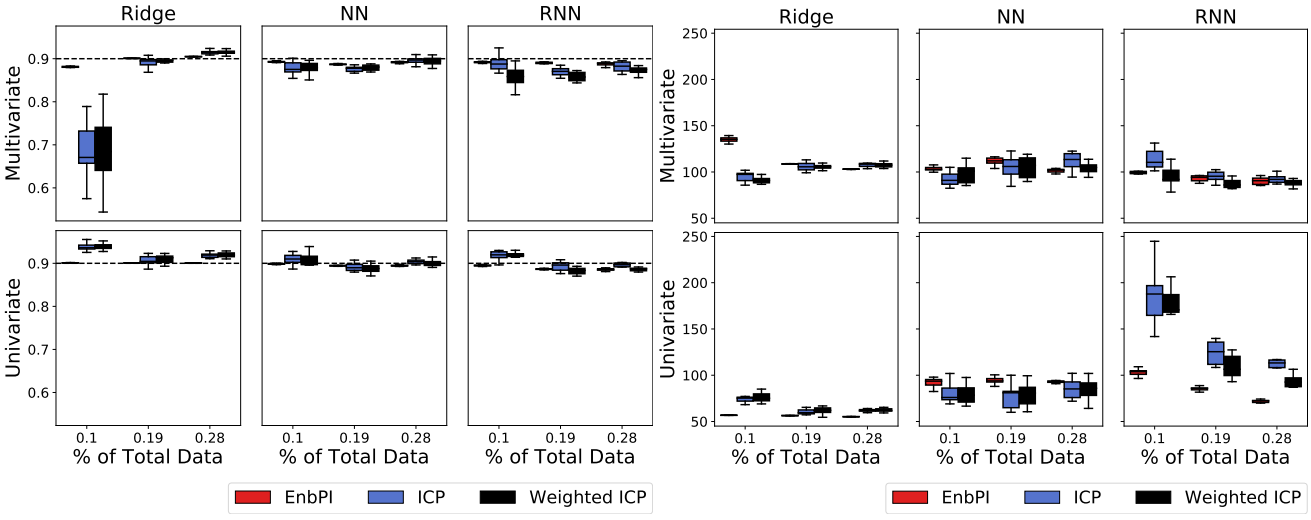
Figure 15. Other three datasets: average coverage and width vs. $1-\alpha$ target coverage by `EnbPI` under different regression algorithms and by ARIMA. Five equally spaced $1-\alpha \in [0.75, 0.95]$ are chosen. The green dash-dotted line at 0.9 represents the target coverage.



(a) Greenhouse Gas



(b) Appliances Energy Gas



(c) Beijing Air

Figure 16. Other three datasets: boxplots of average coverage (top) and width (bottom) by `EnbPI`, ICP, and WeightedICP, whose training data vary as a percentage of total data (x-axis). Each box contains results from 10 independent trials. The black dash dotted line at 0.9 indicates target coverage.

8.5. Details on Supervised Anomaly Detection

Problem Setup. In contrast to the regression model where $f : \mathbb{R}^d \to \mathbb{R}, Y_t = f(X_t) + \epsilon_t$, we instead assume for the anomaly detection task that $f : \mathbb{R}^d \to [0, 1]$,

$$Y_t = \mathbf{1}(f(X_t) > 0.5), \tag{8}$$

so that d is still the dimension of feature X_t but $f(X_t)$ is required to output an probability that quantifies how anomalous Y_t is. In addition, Y_t is equally likely to be anomalous or normal *a priori*.

Detection Algorithm. We now present Algorithm 2 named `ECAD`, which takes in data $\{(x_t, y_t)\}_{t=1}^T$ from (8) and performs anomaly detection on $y_t, t = T + 1, T + 2, \ldots$ It is very similar to `EnbPI` with two main differences:

- Firstly, the residual $\hat{\epsilon}_{-t}^{\phi}$ at time t is no longer the prediction residual but the difference between predicted anomalous probability (i.e., $\hat{\mathbb{P}}^{\phi}(y_t = 1)$) and normal probabilities (i.e., $\hat{\mathbb{P}}^{\phi}(y_t = 0) = 1 - \hat{\mathbb{P}}^{\phi}(y_t = 1)$). This modification is necessary because y_t is given as a binary observation without its actual anomalous probability (i.e., $\mathbb{P}(y_t = 1)$).
- Secondly, we append $\tilde{\epsilon}_{-i}^{\phi} := \mathbf{1}(y_t = 1)(\hat{\mathbb{P}}^{\phi}(y_t = 1) - \hat{\mathbb{P}}^{\phi}(y_t = 0))$, not just $\hat{\epsilon}_{-i}^{\phi} := \hat{\mathbb{P}}^{\phi}(y_t = 1) - \hat{\mathbb{P}}^{\phi}(y_t = 0)$ to the list of past residuals. The reason is more subtle but doing so helps increase precision: Suppose $y_t = 0$ but the ensemble classification model makes the slightly erroneous prediction that $\hat{\mathbb{P}}^{\phi}(y_t = 1) = \hat{\mathbb{P}}^{\phi}(y_t = 0) + \varepsilon_t$ for a small possible factor ε_t. Including $\hat{\epsilon}_{-i}^{\phi}$ negatively affects *future detection*: if for some $t_1 > t$, this prediction error occurs again with $\varepsilon_{t_1} < \varepsilon_t$, then y_{t_1} is more likely a false positive. On the other hand, including this correction factor $\mathbf{1}(y_t = 1)$ reduces the number of future false positives, as $\hat{\mathbb{P}}^{\phi}(y_{t_1} = 1)$ needs to be at least 0.5 and large enough (comparing to previous T predictions) in order to be detected as an anomaly.

Algorithm 2 Supervised Sequential Ensemble Conformal Anomaly Detector (`ECAD`)

Require: Training data $\{(x_i, y_i)\}_{i=1}^T$, classification algorithm $\mathcal{A}$, decision threshold α, aggregation function ϕ, number of bootstrap models B, the batch size s, and test data $\{(x_t, y_t)\}_{t=T+1}^{T+T_1}$, with y_t revealed only after the batch of s detections with t in the batch are made.

Ensure: Detection Predictions $\{\hat{y}_t\}_{t=T+1}^{T+T_1}$

for $b = 1, \ldots, B$ **do**
  Sample with replacement an index set $S_b = (i_1, \ldots, i_T)$ from indices $(1, \ldots, T)$.
  Compute $\hat{f}^b = \mathcal{A}(\{(x_i, y_i) \mid i \in S_b\})$.
end for
Initialize $\boldsymbol{\epsilon} = \{\}$
for $i = 1, \ldots, T$ **do**
  $\tilde{\epsilon}_i^{\phi} = \mathbf{1}(y_i = 1)\phi(\{2\hat{f}^b(x_i) - 1 \mid i \notin S_b\}_{b=1}^B)$,
  assuming $\hat{f}^b(x_i) = \hat{P}^b(y_i = 1)$ so that $2\hat{f}^b(x_i) - 1 = \hat{P}^b(y_i = 1) - \hat{P}^b(y_i = 0)$
  $\boldsymbol{\epsilon} = \boldsymbol{\epsilon} \cup \{\tilde{\epsilon}_i^{\phi}\}$
end for
for $t = T + 1, \ldots, T + T_1$ **do**
  $\hat{\epsilon}_t^{\phi} = \phi(\{2\hat{f}^b(x_t) - 1\}_{b=1}^B)$
  $\tau_t^{\phi} = (1 - \alpha)$ quantile of $\boldsymbol{\epsilon}$.
  Predict $\hat{y}_t = \mathbf{1}(\hat{\epsilon}_t^{\phi} \geq \tau_t^{\phi})$
  if $t - T = 0 \bmod s$ **then**
    for $i = t - s, \ldots, t - 1$ **do**
      Compute $\tilde{\epsilon}_i^{\phi} = \mathbf{1}(y_i = 1)\hat{\epsilon}_i^{\phi}$
      $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon} - \{\tilde{\epsilon}_1^{\phi}\}) \cup \{\tilde{\epsilon}_i^{\phi}\}$ and reset index of $\boldsymbol{\epsilon}$.
    end for
  end if
end for

Data and Competing method. Data comes from `https://www.kaggle.com/mlg-ulb/creditcardfraud`. Since the class of anomalies vs. normal data are highly imbalanced, we downsample the normal data to reduce its training size up to 5 times the number of anomaly data. The other 8 competing anomaly detection methods are described as follows:

Four unsupervised methods: All the unsupervised methods are implemented in the `pyod` library in Python. We consider IForest, PCA,OCSVM, and HBOS and descriptions below mostly come from the package description with minor changes:

- The IsolationForest (IForest) "isolates" observations x_t by randomly selecting a feature of x_t and then randomly selecting a split value between the maximum and minimum values of the selected feature. See (Liu et al., 2012).
- In the Principal Component Analysis (PCA) for anomaly detection, covariance matrix of the data is first decomposed to orthogonal vectors, which are eigenvectors. Then, outlier scores are obtained as the sum of the projected distance of a sample on all eigenvectors. See (Aggarwal, 2015).
- The one-class support vector machine (OCSVM) is a wrapper of scikit-learn one-class SVM Class with more functionalities. See `https://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection` for detailed descriptions.
- The Histogram-based Outlier Detection (HBOS) assumes feature independence in x_t and calculates the degree of outlyingness by building histograms. See (Goldstein and Dengel, 2012).

Four supervised methods: All the supervised methods are taken as binary classification methods from the `sklearn` package in Python. We take descriptions of methods from the package and specify the following parameters for each method

- The Gradient Boosting Classifier (GBoosting) builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. We build 100 estimators, pick a learning rate of 1, and let maximum depth be 1.
- The Multi-layer Perceptron classifier (MLPClassifier) optimizes the log-loss function. We use LBFGS for optimization, let l_2 penalty α be 1e-5, and pick two hidden layers with 5 neurons in the first and 2 in the second.
- The k-nearest neighbor (KNN) algorithm is specified with $k = 20$ and weights="distance", so that closer neighbors of a query point will have a greater influence than neighbors which are further away.
- The support vector classification (SVC) uses all the default settings except with gamma="auto", which uses 1 / # features as the kernel coefficient.