Conformal Time-series Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

Current approaches for (multi-horizon) time-series forecasting using recurrent neural networks (RNNs) focus on issuing point estimates, which are insufficient for informing decision-making in critical application domains wherein uncertainty estimates are also required. Existing methods for uncertainty quantification in RNN-based time-series forecasts are limited as they may require significant alterations to the underlying model architecture, may be computationally complex, may be difficult to calibrate, may incur high sample complexity, and may not provide theoretical guarantees on the frequentist coverage of the issued uncertainty intervals. In this paper, we extend the inductive conformal prediction framework to the time-series forecasting setup, and propose a lightweight, Bonferroni-corrected uncertainty estimation procedure to address the above limitations—with minimal exchangeability assumptions, our approach provides uncertainty intervals with theoretical guarantees on frequentist coverage for *any* multi-horizon forecast predictor and *any* dataset. We demonstrate the effectiveness of our approach by comparing it against existing baselines on a variety of synthetic and real-world datasets.

1 Introduction

Time-series forecasting tasks are central to a broad range of application domains, including stock price predictions [1, 2], service demand forecasting [3, 4], and medical prognoses [5–7]. Recurrent neural networks (RNNs) and its variants (e.g., LSTM, GRU, etc.) constitute an instrumental class of models that are most commonly used to carry out time-series forecasting tasks [8, 9]. These models, however, are usually used to issue *point* predictions—i.e., singular estimates of the future values of a time-series. In many high-stakes applications—such as finance and medicine—these are not enough; we also need estimates of uncertainty in our forecasts that would be utilized for accurate risk assessment when using a model's forecasts to inform decisions [10]—e.g., clinical practitioners need to make treatment decisions accounting for all potential scenarios, where less likely scenarios may have graver consequences and require more care compared to the most likely scenarios [11, 12].

While various methods for uncertainty estimation in standard feed-forward neural networks have been recently proposed [13–15], equivalent methods for RNN-based time-series models are still under-explored. Existing solutions include Bayesian recurrent neural networks [16–18], quantile regression layers [3, 19], latent variable modeling with deep state-space architectures [6, 20], and post-hoc uncertainty estimates using bootstrapping or jackknife procedures [21, 22]. Each of these solutions has its own limitations: Bayesian models may be difficult to calibrate, bootstrapping methods scale poorly for RNNs with large number of parameters, and quantile predictors may "overfit" their uncertainty estimates. All existing methods share two major drawbacks: (1) they require major modifications to the underlying model architecture, and (2) they provide no theoretical guarantees on frequentist coverage. Here, we leverage the *conformal prediction* (CP) framework, originally developed in [23], to develop an uncertainty estimation procedure for time-series forecasts that addresses the limitations above.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

How is our method different? The CP framework has been originally developed for classification and regression problems with exchangeable training data points [23, 24]. Our work extends this framework to the time-series forecasting setup, which comes with its own set of challenges. First, unlike regression problems, different data points in a time-series dataset cannot be "compared" directly since different time-series may have different lengths, different observation frequencies, or asynchronous observation of different time-series samples. (Comparison between training points is a key step in the CP framework [24].) Second, a time-series may be *non-stationary*, i.e., its temporal dynamics may depend on the timestamps of the different observations—consequently, the uncertainty in a multi-horizon time-series forecast will vary across the different prediction horizons.

Our method uses *inductive* conformal prediction (ICP) to construct upper and lower bounds on the forecasts issued sequentially by an RNN model for the future values of a time-series at multiple prediction horizons. Figure 1 illustrates our forecasting setup—here, we show a snapshot for a 5-step forecast made at time step 2 of the series (solid red dots are past observations, shaded dots are future observations, and dashed lines are the RNN forecasts). The output of our procedure is a sequence of intervals that should cover the (yet to be observed) time-series values with high probability. Note that the width of the uncertainty intervals varies with each prediction horizon; typically, higher levels of uncertainty (i.e., wider intervals) would be assigned to horizons farther from the prediction time.

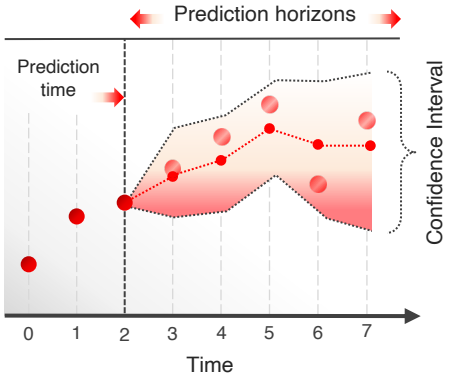

Figure 1: **Depiction of our forecasting setup.**

The key idea behind our method is that the time-series forecasting problem can be viewed as a sequential multi-target regression problem, with the input feature at each time step being the state of the RNN model, and the prediction targets being the prediction horizons at that time step. In Figure 1, the input feature is the hidden state of the RNN at time step 2, and the prediction targets are the time-series values at steps 3 to 7. Similar to the multiple hypothesis testing setup, we use the Bonferroni correction scheme to adjust the empirical distribution of the nonconformity scores in order to ensure that frequentist coverage holds *jointly* for all prediction horizons. Our proposed method addresses the limitations of previous methods mentioned above: it is computationally efficient, does not require any modifications to the underlying predictive model, and provides theoretical coverage guarantees for any multi-horizon forecast predictor and any dataset with minimal exchangeability assumptions.

Throughout this paper, we focus on RNNs as the base models for time-series forecasting—we call our RNN-based conformal forecasting architectures Conformal Recurrent Neural Networks (CoRNN). Before introducing our method in Section 3, we first discuss the related literature in the next Section.

2 Related Work

Most previous work in the area of uncertainty quantification in deep learning focuses on feed-forward neural network models. Much less work has been done on uncertainty estimation for time-series models. In what follows, we discuss previous methods developed for uncertainty estimation for RNNs, which we also summarise in Table 1.

Table 1: Overview of the most popular RNN-based time-series uncertainty forecasting methods.

Method	**Paradigm**	**Architecture**	**Time-series observations**	**Coverage guarantees**
Monte Carlo dropout [25]	Bayesian (approx.)	Built-in	Multiple	None
Multi-quantile [3]	N/A	Built-in	Multiple	None
Blockwise jackknife [21]	Frequentist	Post-hoc	Multiple	$1 - 2\alpha$
EnbPI [26]	Frequentist	Ensemble	Single	$1 - \alpha$
CoRNN (proposed)	Frequentist	Post-hoc	Multiple	$1 - \alpha$

Bayesian RNNs [16–18] place prior distributions on the model parameters and evaluate their posteriors given the time-series observations; the posteriors of model parameters are propagated to

model outputs to reflect uncertainty in predictions. However, this approach requires changes in the underlying model architecture, and exact Bayesian inference in this setup is intractable. Monte Carlo sampling and variational inference can be used to conduct approximate inference [25], but calibrating the priors for such models can be challenging [21].

Quantile RNNs [19, 3] use prediction intervals in the output layer and are trained using quantile loss rather than maximum likelihood. This class of models can be viewed as a deep neural network extension of quantile regression [27], but it is limited in that the distribution is learnt by optimisation (gradient descent) methods and is not mathematically rigorous. By having no mathematical constraints on quantiles, it is possible to encounter problems such as *quantile crossing*, where less extreme quantiles give wider intervals compared to the more extreme quantiles. In addition, this approach is still at risk of *overfitting the quantiles* themselves—the model may be confident about its predictions of the quantiles due to the lack of knowledge about the particular region of the object space (epistemic uncertainty), and be therefore confident in its (wrong) estimates of uncertainty intervals.

Deep state-space models [20, 6] are based on a combination of principled mathematical state space models that extract key time-series components (such as trend, seasonality, etc.), and the RNNs that give the ability to train models on the entire time-series dataset rather than fitting a model to each example separately. However, these models often submit to certain structural assumptions about the time-series dynamics (e.g., Markovianity) and are tailored to particular applications.

Frequentist uncertainty estimation models [21, 22] are based on jackknife resampling of the datasets and estimation of changes of the parameters in retrained models (e.g. using influence functions). While these models are post-hoc and provide frequentist coverage guarantees, their key limitation is time and space complexity—for P parameters of the underlying RNN recurrent neural network, the exact procedure involves estimation of the inverse Hessian matrix taking $O(P^3)$ time. While approximations are available, these may affect the correctness of results. Since parameters of all perturbations have to be evaluated for every example (even during the testing stage), their use does not scale beyond small datasets.

Our proposed method builds on conformal prediction, aiming to determine the levels of confidence for *any* predictive model using past experience [23, 24]. For a given confidence or coverage level α and a learning algorithm producing a point estimate $\hat{y}$, the CP algorithm returns a *prediction region* $\Gamma^{1-\alpha}$ (a set of labels in case of classification, an interval in case of regression) such that, under the minimal assumptions of exchangeability and examples coming from the same distribution, it is guaranteed to contain the true value y with probability of at least α. While it has been initially developed for an *online* prediction setting of accumulating datasets, it can also work in batch setting using a separate calibration set (an approach called *inductive* conformal prediction [28, 29]).

To our knowledge, there is a single model drawing on some ideas from CP and applying them for time-series forecasting (EnbPI; [26]); however, it is limited to the analysis of a single time-series, which is not applicable to the setting where a database of a multiple independent time-series is available (see Table 1 for details). Direct application of CP to time-series data is challenging for two reasons. First, CP assumes the *exchangeability* of examples, so that every permutation of the examples has the same probability of being generated and that they come from the same distribution. Of course, if the observations are assumed to be the steps within a single time-series, the exchangeability assumption is broken due to temporal dependencies. Second, when the CP algorithm is applied to a single time-series, it cannot use information from the remaining dataset $\{y_{1:t}^{(i)}\}_{i=1}^{N}$, where N is the total number of observed time-series.

3 Conformal Recurrent Neural Networks (CoRNN)

In this Section, we introduce the conformal RNN (CoRNN) model. We start off by formalizing the multi-horizon time-series forecasting problem in Section 3.1, and providing the necessary background on conformal prediction in Section 3.2. We introduce the details of CoRNN in Section 3.3.

3.1 Multi-horizon time-series forecasting

Let $y_{n:m} = (y_n, \ldots, y_m)$ be a time-series of multidimensional observations that start at time step n and end at time step m. In the time-series forecasting problem, our goal is to predict the future values

$$\hat{y}_{(t+1):(t+H)} = (\hat{y}_{t+1}, \ldots, \hat{y}_{t+H}) \in \mathbb{R}^H, \tag{1}$$

where H is the number of steps to be predicted (the *prediction horizon*), given the history of observed values $y_{1:t} = (y_1, \ldots, y_t)$. For critical applications, we are interested in the *uncertainty* associated with the forecast—for each time step h in the prediction horizon, we would like to obtain prediction intervals of the form $[\hat{y}^L_{t+h}, \hat{y}^U_{t+h}],\ h \in \{1, \ldots, H\}$, so that the ground truth value y_{t+h} is contained in the interval with a sufficiently high probability.

In this paper, our goal is to achieve *joint coverage* of the multi-horizon forecast, such that the ground-truth values of the *entire time-series trajectory* are contained within the predicted intervals with a pre-defined coverage probability of (α), i.e.,

$$\mathbb{P}(\big(y_{t+h} \in [\hat{y}^L_{t+h}, \hat{y}^U_{t+h}],\ \forall h \in \{1, \ldots, H\}\big) \geq \alpha. \tag{2}$$

3.2 Conformal prediction (CP) framework

Given a set of observations $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{l}$ and a new example $\mathbf{x}^{(l+1)}$, conformal prediction [23] returns a prediction region (a prediction set for classification and prediction interval for regression) $\Gamma^{1-\alpha}$ such that the true label $y^{(l+1)}$ is contained within $\Gamma^{1-\alpha}$ with probability at least α. The CP framework applies to any underlying predictive model and any dataset satisfying the assumptions of exchangeability (where the exchangeability condition dictates that any order of observations $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{l}$ is equally probable).

Estimating prediction intervals with CP. In the standard transductive setting of CP, the regions are estimated as follows [24]. For a given label y, we consider an extended dataset $\mathcal{D}^y_{\text{ext}} = \mathcal{D} \cup \{(\mathbf{x}^{(l+1)}, y\}$ and estimate the *nonconformity score*, which represents how unusual is the new potential example compared to the observed dataset. The nonconformity scores can be obtained using any real-valued function (including a random number generator)—while the choice of the function will not affect the *validity* of the intervals (they will all satisfy the required coverage), it can affect their *efficiency* (the resulting intervals may be unnecessarily wide). A standard method for estimating the nonconformity scores is to train an *underlying algorithm* M on the extended dataset $\mathcal{D}^y_{\text{ext}}$, where the label y is assumed to be the true label, and compute a distance measure between the prediction and the true label. In the regression setup, for the reasons that will become clear in the next Section, we choose the absolute difference as the error; for each example $i \in \{1, \ldots, l+1\}$:

$$\hat{y}^{(i)} = M(\mathbf{x}^{(i)}),\ R_i = |y^{(i)} - \hat{y}^{(i)}|. \tag{3}$$

These nonconformity scores are compared across all the training examples in the extended dataset $\mathcal{D}^y_{\text{ext}}$ to derive the p-value $p_{(\mathbf{x}^{(l+1)}, y)}$ as follows:

$$p_{(\mathbf{x}^{(l+1)}, y)} = \frac{|\{i = 1, \ldots, l+1 : R_i \geq R_{l+1}\}|}{l+1}, \tag{4}$$

and the prediction interval is given by

$$\Gamma^{\alpha} = \{y : p_{(\mathbf{x}^{(l+1)}, y)} > \alpha\}. \tag{5}$$

3.3 CoRNN: Inductive conformal prediction for multi-horizon RNNs

Transductive CP is not computationally feasible as the model cannot be retrained for every possible $y \in \mathbb{R}$, especially for a complex underlying model like an RNN. In this Section we review the *inductive* conformal prediction (ICP) setting, which only requires training the underlying model once while maintaining the validity of the predictions. In addition, we extend the ICP framework to handle the *multi-horizon* forecasting setup, which is necessary for obtaining valid multi-horizon forecast intervals, as the standard CP only considers one-dimensional labels.

In the ICP setup [30, 31], the training set is split into the proper training set of size n and a *calibration set* of size m: $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{cal}}$. The true training set is used to train the underlying predictive model M, and the calibration set is used to obtain the nonconformity scores for the new example $\mathbf{x}^{(l+1)}$ and a possible label y:

$$R_i = |y^{(n+i)} - M(\mathbf{x}^{(n+i)})|, \quad \forall i \in 1, \ldots, m, \tag{6}$$

with corresponding p-values, $p_{(\mathbf{x}_{n+1}, y)}$, computed as follows:

$$p_{(\mathbf{x}_{n+1}, y)} := \frac{|\{i = n+1, \ldots, n+m, l+1 : R_i \geq R_{l+1}\}|}{n+1}. \tag{7}$$

While in regression all potential values $y \in \mathbb{R}$ cannot be tested for their nonconformity, we observe that the choice of the nonconformity measure, $|y - \hat{y}|$, allows us to still derive the set of values of y that will satisfy Equation (5): in particular, we compute a *critical nonconformity score* $\hat{\varepsilon}$ that is at the $(1 - (1-\alpha)n/(n+1))$-th quantile of all calibration nonconformity scores (with $n/(n+1)$ term correcting for the finite sample). Our prediction intervals are thus given by:

$$\Gamma^{1-\alpha}(\mathbf{x}^{(l+1)}) = [\,\hat{y}^{(l+1)} - \hat{\varepsilon}, \hat{y}^{(l+1)} + \hat{\varepsilon}\,], \quad \hat{y}^{(l+1)} = M(\mathbf{x}^{(l+1)}), \tag{8}$$

ICP for time-series forecasting. We now apply the ICP framework to the time-series forecasting problem, and extend it to multi-horizon predictions. Let $\mathcal{D}$ be the set of exchangeable observations from the same underlying distribution of the form $(y_{1:T}, y_{T+1:T+H})$, where $y_{1:T}$ is the time-series consisting of T observed steps, and $y_{T+1:T+H}$ is the H-step forecast. Note that we deviate from the standard ICP framework as the label y is now a H-dimensional value. Let M be an RNN with direct, as opposed to recursive (or iterative) forecast outputs, such that at each time step t the model predicts all values of the H-dimensional target $y_{t+1:t+H}$ directly from a single embedding. We motivate our choice of the direct strategy by it being more robust to error accumulation [32, 3], and the predictions being independent conditional on the internal time-series embedding stored in M—this will allow us to maintain the theoretical guarantees for joint prediction interval coverage. Now we replace the single-dimensional nonconformity score defined earlier by its H-dimensional counterpart,

$$R_i = \left[|y^{(i)}_{t+1} - \hat{y}^{(i)}_{t+1}|, \ldots, |y^{(i)}_{t+H} - \hat{y}^{(i)}_{t+H}|\right]^\top \tag{9}$$

where $\left[\hat{y}^{(i)}_{t+1}, \ldots, \hat{y}^{(i)}_{t+H}\right]^\top = M(y^{(i)}_{1:t})$, and obtain H p-values $p_{(y_{1:t}, y_{t+1})}, \ldots, p_{(y_{1:t}, y_{t+H})}$ following Equation (7). Since we produce H predictions that are conditionally independent given the internal state of the underlying model, we can apply Bonferroni correction of the obtained p-values by dividing the desired coverage α by H, so that the critical nonconformity scores $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_H$ are now obtained using the $(1 - (1-\alpha)n/(H(n+1)))$-th quantiles of the corresponding nonconformity score distributions, obtaining a set of prediction intervals $\Gamma^{1-\alpha}_1(y^{(l+1)}_{(1:t)}), \ldots, \Gamma^{1-\alpha}_H(y^{(l+1)}_{(1:t)})$, where

$$\Gamma^{1-\alpha}_h\left(y^{(l+1)}_{(1:t)}\right) = \left[\hat{y}^{(l+1)}_{t+h} - \hat{\varepsilon}_h, \hat{y}^{(l+1)}_{t+h} + \hat{\varepsilon}_h\right] \quad \forall h \in \{1, \ldots, H\}. \tag{10}$$

The resulting model is an RNN that issues a point prediction for its forecast along with a prediction region that covers the forecast trajectory. The entire procedure for constructing prediction intervals in CoRNN is illustrated in Figure 2 and summarized in Algorithm 1.

Finally, we show the theoretical motivations behind our approach via the following Theorem, which provides the joint coverage guarantees for intervals obtained from Algorithm 1.

Theorem 1. *For the target coverage $\alpha \in (0, 1)$ and the dataset of exchangeable observations from the same underlying distribution, the intervals obtained by the conformal forecasting algorithm satisfy*

$$\mathbb{P}\left(\forall h \in \{1, \ldots, H\}.\, y_{t+h} \in [\hat{y}_{t+h} - \hat{\varepsilon}_h, \hat{y}_{t+h} + \hat{\varepsilon}_h]\right) \geq \alpha$$

□

The proof follows from conditional validity of ICP in Vovk [33] and Boole's inequality. The full statement and detailed proof is provided in the Appendix.

4 Experiments

In this section we showcase the performance of the conformal RNN (CoRNN) against three baselines: the frequentist blockwise jackknife RNN (BJ-RNN) [21], multi-quantile RNN (MQ-RNN) Wen et al. [3], and Monte Carlo dropout-based RNN (DP-RNN) [25]. We chose these baselines as the most popular and representative examples of the different paradigms for uncertainty estimation (frequentist, quantile and Bayesian for BJ-RNN, MQ-RNN and DP-RNN respectively). All architectures use LSTM as the underlying recurrent neural network, and are adapted to produce direct multi-horizon forecasts. We first present the qualitative performance of CoRNNs on synthetic data with controlled properties. Due to the inability of BJ-RNNs to scale to larger real datasets, we use smaller synthetic datasets to provide an in-depth comparison of CoRNNs and BJ-RNNs. Finally, we compare the performance of CoRNNs against the other two baselines on three real-world medical datasets.

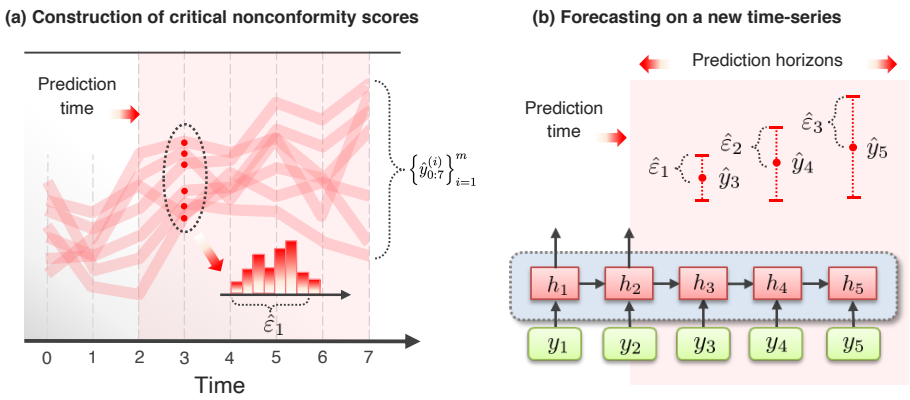

Figure 2: **Illustration for our uncertainty estimation procedure**. (a) The calibration set is used to obtain the empirical distribution of nonconformity scores $\hat{\boldsymbol{\varepsilon}}_h$, and its appropriate quantile is selected depending on the desired target coverage level. (b) Critical nonconformity scores are used to obtain the prediction interval.

Algorithm 1 Conformal recurrent neural network (CoRNN)

Input: A trained model M producing H-step forecasts,
calibration dataset $\mathcal{D}_{\text{cal}} = \left\{(y^{(i)}_{1:t}, y^{(i)}_{t+1:t+H})\right\}_{i=1}^{m}$,
target coverage α, the size of the training set n.
Output: Critical nonconformity scores $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_H$.

Initialize $\boldsymbol{\varepsilon}_1 = \{\}, \ldots, \boldsymbol{\varepsilon}_H = \{\}$.
for $i = 1$ **to** m **do**
  $\hat{y}^{(i)}_{t+1:t+H} \leftarrow M(y^{(i)}_{1:t})$.
  for $h = 1$ **to** H **do**
    $\boldsymbol{\varepsilon}_h \leftarrow \boldsymbol{\varepsilon}_h \cup \{|\hat{y}^{(i)}_{t+h} - y^{(i)}_{t+h}|\}$.
  end for
end for
for $h = 1$ **to** H **do**
  `// Bonferroni and finite sample correction`
  $\hat{\varepsilon}_h \leftarrow (1 - \frac{(1-\alpha)n}{H(n+1)})$-th quantile of $\boldsymbol{\varepsilon}_h$.
end for
return $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_H$.

For a new time-series example $y^*_{1:t}$:
  $\hat{y}^*_{t+1:t+H} \leftarrow M(y^*_{1:t})$.
  return intervals $\hat{y}^*_{t+1} \pm \hat{\varepsilon}_1, \ldots \hat{y}^*_{t+H} \pm \hat{\varepsilon}_H$.

4.1 Synthetic data

We first generate the synthetic time-series consisting of three components: the autoregressive process determining the trend of the time-series, the seasonal process representing periodic fluctuations in the time-series values, and the noise process representing the inherent uncertainty of the dataset. For a time-series of length T, this is expressed mathematically as:

$$y_t = \sum_{k=0}^{t} a^k \cdot x_k + \gamma_t + \epsilon_t, \forall k \in \{1, \ldots, T\} \tag{11}$$

where $x_t \sim \mathcal{N}(\mu_x, \sigma_x^2)$, $a = 0.9$ is the memory parameter, γ_t is the stochastic seasonal process representing periodic fluctuations and $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ is the noise process. The periodic component is defined following the quasi-random walk model in Durbin and Koopman [34] (Equations (3.7)

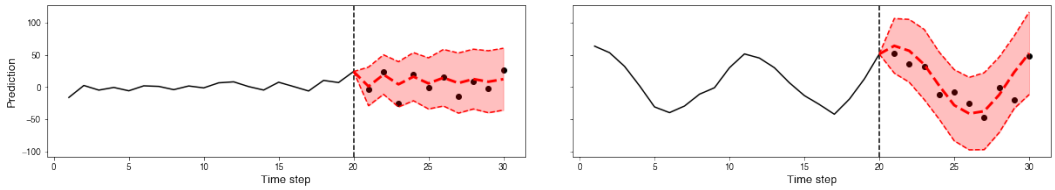

Figure 3: Example prediction intervals for the CoRNN model for datasets with periodicities 2 and 10.

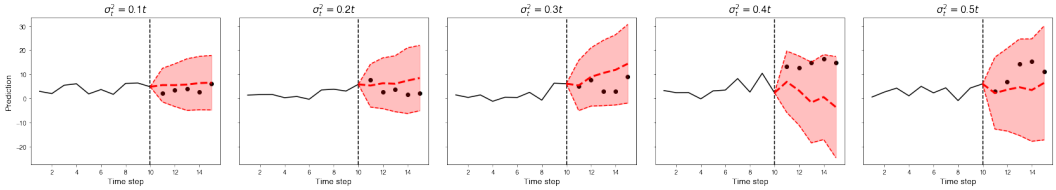

Figure 4: Example prediction intervals for the CoRNN model in the time-dependent noise setting.

and (3.8)): we define $\gamma_t = \sum_{j=1}^{\lfloor s/2 \rfloor} \gamma_{jt}$ where $\gamma_{j,t+1} = \gamma_{jt} \cos \lambda_j + \gamma^*_{jt} \sin \lambda_j + w_{jt}$ and $\gamma^*_{j,t+1} = -\gamma_{jt} \sin \lambda_j + \gamma^*_{jt} \cos \lambda_j + w^*_{jt}$, s is the period length, $\lambda_j = 2\pi/s$ and $w_{jt}, w^*_{jt} \sim \mathcal{N}(0, u)$ for some amplitude u.[1]. For the noise component ϵ_t, we consider five time-dependent noise variance profiles, $\sigma_t^2 = 0.1tn$ and five static noise variance profiles $\sigma_t^2 = 0.1n$, for $n = \{1, \ldots, 5\}$.

4.2 Results

We first demonstrate the qualitative performance of the CoRNN architecture on two synthetic datasets containing seasonal components of different periodicities $s = 2$ and $s = 10$ (Figure 3), and show the qualitative changes in prediction interval widths as time-dependent variance increases (Figure 4). Despite the larger errors in the higher-frequency series in Figure 3, CP intervals cover the ground truth in both cases. Prediction interval (PI) widths also increase moving further away from the prediction step and with increasing dataset variance, reflecting increasing uncertainty.

Next, we train CoRNN and BJ-RNN models on 1000 training sequences (and 1000 calibration sequences for CoRNN) for the two noise variance profiles discussed in Section 4.1. We aim to forecast prediction intervals for H future values $y_{T+1:T+H}$ for a default coverage rate of 90% ($\alpha = 0.9$). In these experiments, $T = 10$ and $H = 5$: we select short time-series examples due to the computational limitations of the baseline BJ-RNN model. The RNN hyperparameters for the networks underlying the two post-hoc uncertainty estimation models are fixed in order to ensure fair comparison, and largely follow those provided in previous work [21]. These are detailed in the Appendix along with the time-series model parameters. Where possible (i.e. for CoRNN models), we repeat the experiments five times with a new randomly generated dataset, reporting variation in empirical joint coverage over the different dataset realisations.

Table 2 compares the uncertainty intervals of the two models. Both models tend to empirically surpass the target joint coverage of 90% ($\alpha = 0.9$) in both static and time-dependent noise settings. In case of CoRNNs, we note that the variations in empirical joint coverage are expected due to limited dataset sizes determining the critical quantiles, but they will be close to the theoretically proven result (as discussed further in Vovk [33]). We observe that CoRNN adapts to the properties of the underlying time-series distribution: when the noise is static (and the time-series more predictable), CoRNN PI widths remain monotonic while maintaining the coverage close to the desired 90% level. For the time-dependent noise profile, uncertainty in data increases with further time steps and larger base variance, so PI widths also on average increase monotonically for different n, in order to maintain the desired coverage level. While the intervals for the BJ-RNN model generally provide higher (or even perfect) coverage, they do so in sacrificing the efficiency of intervals, which is undesirable if we prefer narrow intervals that are as close to target coverage as possible. In addition, BJ-RNNs take prohibitively long to compute (the reason for which they will also be excluded from the comparisons on real data);

[1]For implementation details, see also https://www.statsmodels.org/devel/examples/notebooks/generated/statespace_seasonal.html

Table 2: Performance comparison of CoRNNs and BJ-RNNs on autoregressive series with static or time-dependent noise profiles. Where possible, empirical joint coverages are aggregated over repeated trials with randomly generated datasets. The prediction intervals are aggregated over the prediction horizon of a single trial.

Noise mode		**Empirical joint coverage**		**PI lengths**	
		CoRNN	BJ-RNN	CoRNN	BJ-RNN
Static $\sigma_t^2 = 0.1n$	$n = 1$	$93.0 \pm 1.3\%$	100%	17.33 ± 4.04	98.45 ± 25.95
	$n = 2$	$94.8 \pm 0.9\%$	100%	18.73 ± 5.09	32.53 ± 2.92
	$n = 3$	$92.8 \pm 1.3\%$	100%	16.99 ± 3.81	35.82 ± 1.59
	$n = 4$	$93.7 \pm 1.9\%$	100%	18.77 ± 5.10	33.83 ± 2.49
	$n = 5$	$93.6 \pm 1.1\%$	100%	18.47 ± 4.14	51.23 ± 3.21
Time-dependent $\sigma_t^2 = 0.1tn$	$n = 1$	$93.5 \pm 1.1\%$	99.4%	20.07 ± 4.89	27.09 ± 1.16
	$n = 2$	$92.9 \pm 1.4\%$	100.0%	22.98 ± 3.88	104.85 ± 5.68
	$n = 3$	$92.1 \pm 1.9\%$	100.0%	27.90 ± 4.74	36.45 ± 1.25
	$n = 4$	$89.6 \pm 2.3\%$	97.0%	32.56 ± 6.68	33.24 ± 2.32
	$n = 5$	$90.8 \pm 1.2\%$	99.4%	37.97 ± 5.96	51.45 ± 5.37

conversely, the ICP procedure only requires running the trained RNN model on a calibration set, at which point adding uncertainty intervals to a prediction takes constant time. Therefore the CoRNN model is competitive in PI width/coverage trade-off, and outperforms BJ-RNNs in computational efficiency.

Finally, the experiments on the data with controlled properties provide insight on the trade-offs between the desired coverage rate and how far into the future can the predictions be reliably made. For a dataset with a time-dependent noise profile $\sigma_t^2 = 0.1t$, we fix a prediction interval and for each coverage level α compute the furthest horizon H at which the coverage can be maintained. As Figure 4 illustrates, low target coverage levels allow us to make valid predictions far into the future, and ideal coverage levels can only be achieved with horizons near the prediction point. We additionally observe that the underlying recurrent neural network model M has little effect on the overall trend.

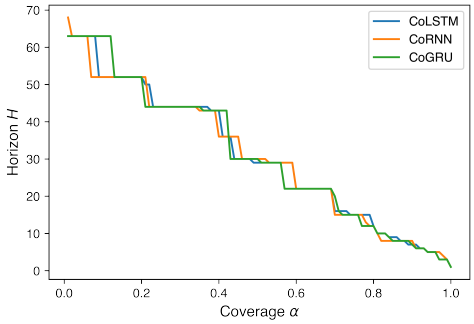

Figure 4: The trade-off between the target coverage rate and prediction horizon for a fixed prediction interval width.

4.3 Experiments on real data

We now demonstrate the effectiveness of our procedure by forecasting real-world time-series. We train the proposed CoRNN architecture as well as the MQ-RNN and DP-RNN baselines on three datasets summarised in Table 3. For the first task, we use the data from the Medical Information Mart for Intensive Care (MIMIC-III) [35] dataset, where we forecast daily observations of white blood cell counts of varying lengths. For the second task, we use the electroencephalography (EEG) dataset from the UCI machine learning repository [36], where we forecast the trajectories of downsampled EEG signals obtained from healthy subjects exposed to three types of visual stimuli. For the final task, we forecast the daily COVID-19 cases within the United Kingdom local authority districts. All datasets are publicly available and the medical data is anonymised. We selected these datasets to represent a variety of scenarios of real time-series: the datasets have different orders of magnitude of available training instances (from hundreds to tens of thousands), have varying observation sequence lengths, have different stationarity properties (e.g. the COVID-19 time-series are synchronous—each time step representing the same point in time—the others are not), have different noise profiles (e.g. EEG signal data will inherently have higher frequencies than MIMIC-III white blood cell count data), have different target prediction horizons. Full details on preprocessing are provided in the Appendix.

Performance of the models is summarised in Table 4. We first note that the underlying LSTM model of the proposed CoRNN architecture had fewer training instances available while having the same hyperparameters as the competing baselines—some of the training instances had to be

Table 3: Dataset properties. The number in parentheses under the training example column indicates how many examples were used for training when a calibration set was required, such as in the case of CoRNNs.

Dataset	# Training sequences	Window length T	Prediction horizon H
MIMIC-III [35]	3823 (2000)	[3, 47]	2
EEG [36]	19200 (15360)	40	10
COVID-19 [37]	300 (200)	100	50

Table 4: Uncertainty forecasting model performance on three real-world datasets. Coverage refers to joint coverage (higher is better), and is averaged over the random splits of the dataset and training seeds. Prediction interval lengths (lower is better) are averaged over the prediction horizons of a single dataset realisation.

	MIMIC-III		**EEG**		**COVID-19**	
Model	Coverage	CI/PI lengths	Coverage	CI/PI lengths	Coverage	CI/PI lengths
DP-RNN	$39.0 \pm 5.4\%$	3.71 ± 0.00	$8.3 \pm 10.3\%$	7.91 ± 0.55	$0.0 \pm 0.0\%$	51.46 ± 24.02
MQ-RNN	$87.6 \pm 2.7\%$	17.33 ± 0.31	$49.1 \pm 3.0\%$	21.42 ± 2.27	$14.0 \pm 7.1\%$	122.06 ± 57.98
CoRNN	$94.0 \pm 1.2\%$	22.12 ± 0.02	$96.4 \pm 1.0\%$	56.28 ± 12.69	$89.0 \pm 5.1\%$	536.88 ± 417.38

used for the conformal calibration procedure. However, CoRNN obtains the highest coverage for all datasets, and is the only one to empirically achieve the target joint coverage rates. While this seems to disproportionately affect the efficiency of CoRNN intervals, we see that the predictions are indeed reliable across the range of scenarios, unlike the baseline models which seem to have competitive coverage with better efficiency in some settings (e.g. MIMIC-III) yet revert to unreliable predictions in less certain scenarios (e.g. COVID-19). In other words, CoRNN adapts its PI widths to reliably match the required target coverage, increasing PI width for unpredictable datasets. Table 5 additionally shows the details on *independent* coverage rates for different prediction horizons (rather than *joint* coverage of the entire trajectory): we observe that CoRNN outperforms the baselines on prediction of both individual and joint coverage levels.

Table 5: Example ranges of independent coverage and MAEs averaged over the horizon steps.

	MIMIC-III		**EEG**		**COVID-19**	
Model	Coverage range	MAE	Coverage range	MAE	Coverage range	MAE
DP-RNN	[44.6%, 48.8%]	2.07 ± 2.22	[35.9%, 54.8%]	5.60 ± 6.29	[22.5%, 67.5%]	28.90 ± 41.53
MQ-RNN	[88.8%, 90.0%]	4.87 ± 5.16	[85.3%, 90.2%]	5.50 ± 6.26	[80.0%, 93.8%]	34.71 ± 43.87
CoRNN	[94.2%, 94.4%]	4.49 ± 5.41	[98.3%, 99.6%]	5.22 ± 5.56	[93.8%, 100.0%]	32.32 ± 46.38

Table 6: Bonferroni-corrected and uncorrected empirical coverages of the CoRNN model.

	MIMIC-III		**EEG**		**COVID-19**	
Model	Joint	Independent	Joint	Independent	Joint	Independent
CoRNN	94.0%	[94.2%, 94.4%]	96.4%	[98.3%, 99.6%]	89.0%	[93.8%, 100.0%]
Uncorrected	87.8%	[89.0%, 89.8%]	58.0%	[85.8%, 90.6%]	57.5%	[77.5%, 98.8%]

Finally, we briefly explore the importance of and motivation behind the Bonferroni correction for the CoRNN model. Table 6 shows that uncorrected calibration scores generally lead to poor joint coverage yet similar independent coverage (as the Bonferroni procedure corrects for multiple comparisons across the horizon, without affecting individual coverage rates). We again emphasise that the estimates in this case are noisy due to small testing and empirical calibration set sizes, especially for noisy COVID-19 data.

5 Conclusion

In this paper, we extended the ICP framework to the multi-horizon time-series forecasting problem, providing a lightweight algorithm with theoretical guarantees for frequentist coverage. Future work would focus on increasing the overall efficiency of prediction intervals by reducing their width, which would involve making them more adaptive to individual observations.

References

[1] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1643–1647. IEEE, 2017.

[2] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[3] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[4] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE, 2017.

[5] Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7494–7504, 2018.

[6] Ahmed M Alaa and Mihaela van der Schaar. Attentive state-space modeling of disease progression. In *Advances in Neural Information Processing Systems*, pages 11334–11344, 2019.

[7] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604, 2017.

[8] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[9] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, 2013.

[10] Christian Gollier. *The economics of risk and uncertainty*. Edward Elgar Publishing Limited, 2018.

[11] Michael W Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M Dai. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 204–213, 2020.

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[13] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *arXiv preprint arXiv:1902.02476*, 2019.

[14] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

[15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413, 2017.

[16] Meire Fortunato, Charles Blundell, and Oriol Vinyals. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*, 2017.

[17] Jen-Tzung Chien and Yuan-Chu Ku. Bayesian recurrent neural network for language modeling. *IEEE transactions on neural networks and learning systems*, 27(2):361–374, 2015.

[18] Derrick T Mirikitani and Nikolay Nikolaev. Recursive bayesian recurrent neural networks for time-series modeling. *IEEE Transactions on Neural Networks*, 21(2):262–274, 2009.

[19] Jan Gasthaus, Konstantinos Benidis, Yuyang Wang, Syama Sundar Rangapuram, David Salinas, Valentin Flunkert, and Tim Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd international conference on artificial intelligence and statistics*, pages 1901–1910. PMLR, 2019.

[20] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31:7785–7794, 2018.

[21] Ahmed Alaa and Mihaela van der Schaar. Frequentist uncertainty in recurrent neural networks via blockwise influence functions. In *International Conference on Machine Learning*, pages 175–190. PMLR, 2020.

[22] Ahmed Alaa and Mihaela Van Der Schaar. Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. In *International Conference on Machine Learning*, pages 165–174. PMLR, 2020.

[23] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[24] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

[25] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[26] Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. *International Conference on Machine Learning*, 2021.

[27] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4):143–156, 2001.

[28] Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. Uncertainty sets for image classifiers using conformal prediction. *International Conference on Learning Representations*, 2021.

[29] Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. In *Tools in artificial intelligence*. Citeseer, 2008.

[30] Harris Papadopoulos and Haris Haralambous. Reliable prediction intervals with regression neural networks. *Neural Networks*, 24(8):842–851, 2011.

[31] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, pages 345–356. Springer, 2002.

[32] Souhaib Ben Taieb and Amir F Atiya. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(1):62–76, 2015.

[33] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

[34] J Durbin and SJ Koopman. Linear state space models. *Time Series Analysis by State Space Methods*, pages 43–75, 2012.

[35] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[36] Catherine Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/~ mlearn/MLRepository. html*, 1998.

[37] Coronavirus (COVID-19) in the UK. https://coronavirus.data.gov.uk/, 2021. Accessed: 2021-05-25.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? Yes
   (b) Did you describe the limitations of your work? Yes
   (c) Did you discuss any potential negative societal impacts of your work? N/A: we believe there is no direct and obvious negative societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? Yes
   (b) Did you include complete proofs of all theoretical results? Yes
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, where that was possible considering the computational resources.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? N/A: the experiments were carried out on a single standard machine.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? Yes
   (b) Did you mention the license of the assets? Yes
   (c) Did you include any new assets either in the supplemental material or as a URL? Yes
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? Yes
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? Yes
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? N/A
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? N/A
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? N/A