Evaluation of conformal-based probabilistic forecasting methods for short-term wind speed forecasting

Simon Althoff SI5084AL-S@STUDENT.LU.SE
Lund University, Sweden and Algorithma AB, Sweden
Johan Hallberg Szabadváry JOHAN.HALLBERG@ALGORITHMA.SE
Algorithma AB, Sweden
Jonathan Anderson JONATHAN.ANDERSON@ALGORITHMA.SE
Algorithma AB, Sweden
Lars Carlsson LARS.CARLSSON@JU.SE
Dept. Computing, Jönköping University, Sweden and Centre for Reliable Machine Learning, University of London, UK and Algorithma AB, Sweden

Editor: Harris Papadopoulos, Khuong An Nguyen, Henrik Boström and Lars Carlsson

Abstract

We apply Conformal Predictive Distribution Systems (CPDS) and a non-exchangeable version of the traditional Conformal Prediction (NECP) method to short-term wind speed forecasting to generate probabilistic forecasts. These are compared to the more traditional Quantile Regression Forest (QRF) method. A short-term forecast is available from a few hours before the forecasted time period and is only extended a couple days into the future. The methods are supplied ensemble forecasts as input and additionally the Conformal methods are supplied with post-processed point forecasts for generating the probability distributions. In the NECP case we propose a method of producing probability distributions by creating sequentially larger prediction intervals. The methods are compared through a teaching schedule, to mimic a real-world setting. For each model update in the teaching schedule a grid-search approach is applied to select each method's optimal hyperparameters, respectively.

The methods are tested out of the box with tweaks to few hyperparameters. We also introduce a normalized nonconformity score and use it with the conformal method that handles data that violates the exchangeability assumption. The resulting probability distributions are compared to actual wind measurements through Continuous Ranked Probability Scores (CRPS) as well as their validity and efficiency of certain prediction intervals. Our results suggest that the conformal based methods, with the pre-trained underlying model, produce slightly more conservative but more efficient probability distributions than QRF at a lower computational cost. We further propose how the conformal-based methods could be improved for the application to real-world scenarios.

Keywords: conformal predictive distribution systems · quantile regression forests · exchangeable · non exchangeable · short-term wind forecast · normalized nonconformity

1. Introduction

Wind prediction has become a hot topic the last couple of years. There are many applications where good wind forecasts are interesting. Perhaps most notably, as the world is transitioning to fossil free energy, wind power production is playing a larger role in our energy supplies. Being able to reliably predict wind speed on the short-term is paramount for both energy trading markets and for planning other types of energy production to ensure stability of the grid. Forecasts have historically mostly been deterministic, but as the demand for wind forecasts has increased so has the demand for research into producing prediction intervals and distributions, especially in wind power production (Bazionis et al. (2022)). With the introduction of ensemble forecasts, where several simulations of the atmosphere are run simultaneously with slightly differing conditions, methods efficiently using these to produce probabilistic forecasts have risen in popularity.

Conformal predictions (Vovk et al. (2022)) is a relatively new method which is used to produce set predictions from machine learning models. The application of Conformal predictions with extensions to weather and power forecasting is, to the best of our knowledge, a relatively unexplored subject of research apart from the application to power markets (Kath and Ziel (2021)). Our work should be viewed as an introduction to how Conformal Prediction methods can be used in the area of probabilistic wind forecasting. In section 2 we will briefly describe Conformal Predictions (CP) and Conformal Predictive Distribution Systems (CPDS) (Vovk et al. (2022)) and cover basic theory. In the same section we will also explain how CP can be extended to the non-exchangeable setting, we will cover Quantile Regression Forest (QRF), which is used as comparison method to the previously mentioned methods and we will explain how these methods will be evaluated. Section 3 will describe how we use these methods on a data set of measurements and forecasts, the results of which we present in section 4. Finally, in section 5 we discuss the results and suggest how we could move forward with research on this subject.

2. Preliminaries

2.1. Conformal Prediction and CPDS

In a machine learning setting it is natural to ask, given a point prediction from some model, how well we can trust the prediction. Conformal prediction is a simple and efficient way of producing valid set predictions, a sort of quantified uncertainty analysis. Using already viewed data points we can construct sets for predictions. Based on what we have viewed previously, these sets can be constructed in such a way that they will contain the actual value with some probability. In this study, we will try to follow the notation in (Vovk et al. (2022)) as far as possible. We stick to the regression type problem and for an object $x_n \in \mathbf{X}$, we would like to predict either; the prediction range, $\Gamma^\epsilon \in \mathbb{R}$, where ϵ is a significance level corresponding to the fraction of errors we are prepared to tolerate, or the probability distribution. In both the CP and the CPDS setting we are given a training set of $n-1$ examples where the training examples are denoted by $z_i = (x_i, y_i), \quad i = 1, \dots, n-1$.

It all starts with the choice of a (non-)conformity function, A, that maps an object x together with all training examples to a real number, a conformity score,

$$\alpha_i = A(z_1, \dots, z_n).$$

We remark that the most obvious setup for A is that we use both training and prediction of a machine-learning model as part of the application of A. The validity, the fraction of errors we can tolerate compared to the actual errors made, of the predictions is not dependent on the choice of conformity score, however the efficiency, the size of the prediction range, might improve if it is chosen well. The idea behind the conformity score is to quantify how the test object, assuming a label for the object x_n, conforms to the training examples. This is called a *conformal transducer* and it creates a *p-value*

$$p_n^y = \frac{|\{i = 1, \dots n-1 \mid \alpha_i < \alpha_n\}|}{n}.$$

The conformal transducer can in turn be used to form a range prediction

$$\Gamma^\epsilon = \{\ p_n^y > \epsilon\}\,.$$

We choose different y, without knowing y_n, in the range of values in which y can lie such that the ratio of conformity scores less than α_n is ϵ. Usually the conformal prediction is built around some underlying predictive model. As an example assume we have a machine learning model that predicts y_i from x_i by producing $\hat{y}_i$. We could then construct a conformity score where

$$\alpha_i = |y_i - \hat{y}_i|,$$

which would be dependent on all other examples since that is what we have trained the model on. The model would however need to be retrained for each data point to exclude itself from the prediction. This can be computationally heavy which is why it is common to construct a *proper training set*, to train the underlying model, and a *calibration set* which will produce the needed conformity scores. This is commonly called an inductive algorithm. The conformal prediction algorithm will, under the assumption that the data is exchangeable, produce valid prediction intervals (Vovk et al. (2022)).

In certain applications one would prefer to get entire probability distributions instead of set predictions. One of the advantages of conformal transducers is that it is possible to use them to create a distribution for y. These are called Conformal Predictive Systems that produce Conformal Predictive Distributions, or Conformal Predictive Distribution Systems (CPDS) to simplify things. Instead of using a conformal transducer to create a range prediction, we use the conformal transducer to create a distribution, Π, by arranging the p-values in to a distribution function. Certain criteria need to be fulfilled:

1. $\Pi(z_1, \dots, z_{n-1}, (x_n, y))$ is an increasing function of $y \in \mathbb{R}$
2. $\lim_{y\to-\infty} \Pi(z_1, \dots, z_{n-1}, (x_n, y)) = 0$
3. $\lim_{y\to\infty} \Pi(z_1, \dots, z_{n-1}, (x_n, y)) = 1$

There are several ways in which a distribution can be formed, for further details see (Vovk et al. (2022)). These distributions are valid in the sense that the conformal transducer create p-values that are uniformly distributed on $[0, 1]$, under the same assumptions as earlier, while the efficiency will depend on the underlying model and conformity score (Vovk (2020)). To reduce computational complexity an inductive-conformal method is often used as to not have to recompute all the conformity scores for each possible value of y, similar to the CP case.

2.2. Non-exchangeable CP

Though exchangeability is slightly relaxed compared to the classical i.i.d. assumption, it is still quite idealised compared to the real world. Exchangeability means in rough terms that a set of data points has the same conditional likelihood no matter the ordering of the points, e.g. we are just as likely to get $\{x_1, x_2, x_3\}$ as $\{x_3, x_1, x_2\}$ or any other permutation when we collect the data. For exchangeability to hold it is also assumed that the underlying model supplying the point predictions treat data points symmetrically. Real world processes can seldom be assumed to be strictly exchangeable and wind, which for instance might have a seasonable component, would be no different. This is why, in recent years there have emerged several techniques for performing conformal prediction on non-exchangeable data. One such technique is where weights $\omega_i \in [0, 1]$ are assigned to each example with higher weights to examples that are "trusted" more (Barber et al. (2023)). We call this Non-Exchangeable Conformal Prediction (NECP). In a time-series setting with potential distribution drift it would be wise to assign larger weights to recent examples since they will likely have a more similar distribution as the point to predict. To produce a prediction interval for y_n we first normalize the weights according to

$$\begin{aligned}\tilde{\omega}_i &= \frac{\omega_i}{\omega_1 + \cdots + \omega_{n-1} + 1}, \quad i = 1, \ldots, n-1,\\ \tilde{\omega}_n &= \frac{1}{\omega_1 + \cdots + \omega_{n-1} + 1}.\end{aligned}$$

The resulting interval $\hat{C}$, given a prediction $\hat{y}_n$ from an underlying model is then

$$\hat{C}_n = \hat{y}_n \pm \mathbf{Q}_{1-\epsilon}\left(\sum_{i=1}^{n-1} \tilde{\omega}_i \cdot \delta_{R_i} + \tilde{\omega}_n \cdot \delta_{+\infty}\right)$$

where $\mathbf{Q}_{1-\epsilon}$ represents the $1-\epsilon$ quantile of the expression within parentheses, δ is the Dirac delta function and $R_i = |y_i - \hat{y}_i|$. This is called the Nonexchangeable Split Conformal method in (Barber et al. (2023)), note that this is an inductive approach, since it assumes a pre-trained underlying model. This method has a theoretical coverage result which is described by

$$\mathbb{P}\{y_n \in \hat{C}_n\} \geq 1 - \epsilon - \sum_{i=1}^{n-1} \tilde{\omega}_i \cdot d_{TV}(R_{splitCP}(Z), R_{splitCP}(Z^i)) \tag{1}$$

$$\mathbb{P}\{y_n \in \hat{C}_n\} < 1 - \epsilon + \tilde{\omega}_n + \sum_{i=1}^{n-1} \tilde{\omega}_i \cdot d_{TV}(R_{splitCP}(Z), R_{splitCP}(Z^i)) \tag{2}$$

where $d_{TV}(R_{splitCP}(Z), R_{splitCP}(Z^i))$ is the total variation distance in distributions between the residuals $(R_{splitCP}(Z))_i = |y_i - \hat{y}_i|$ and the residuals $R_{splitCP}(Z^i)$ where the ith point has been replaced by the nth point. Condition (2) holds when, for the pre-trained model, the residuals $R_1, \ldots, R_n$ are distinct, with probability 1. If the exchangeability assumption holds, all ω_i should be put to 1, giving the standard CP method.

2.3. Quantile regression forest

The Quantile Regression Forest (QRF) (Meinshausen (2006)) is a generalization of the Random Forest regression method. It is a non-parametric method for predicting conditional quantiles. The random forest algorithm grows a number of decision trees to perform the regression. Each decision tree will split the predictor variables along the height of the tree to gather "similar" predictors in a leaf where the mean of the labels associated with the predictors is stored. The random forest bags the training data for each tree and randomly selects a subset of predictor variables at each split node. The predicted mean of a new point from the forest is the averaged output of all the trees, which is the mean stored in the leaf associated with the predictor of the new point. The mean is produced by assigning weights to all observations in the training set. Passing the object $x_n \in \mathbf{X}$ to the trees, each tree will then have a corresponding leaf, $l(x_n)$, associated with the object, the leaf representing some subset $\mathbf{X}_{l(x_n)} \subseteq \mathbf{X}$. Going through all examples $i = 1, \ldots, n-1$ in the training set we set the weight $\omega_i(x_n)$ to

$$\omega_i(x_n) = \frac{\mathbf{1}_{\{x_i \in \mathbf{X}_{l(x_n)}\}}}{|\{j : x_j \in \mathbf{X}_{l(x_n)}\}|}$$

i.e. examples associated with other leafs gets weight 0 and the ones associated with the same get weights which in total sum to 1. This procedure is performed for all the trees in the forest. The final weight $\bar{\omega}_i$ for each example is the averaged weight of that example over all trees. The final output is then

$$\hat{y}_n = \sum_{i=1}^{n-1} \bar{\omega}_i(x_n) y_i.$$

The QRF deviates from the Random Forest by storing all labels associated with each leaf in each tree, instead of just the mean. From that we can construct the estimated conditional CDF from

$$\hat{F}(y|X = x_n) = \sum_{i=1}^{n-1} \bar{\omega}_i(x_n) \mathbf{1}_{\{y_i \leq y\}}.$$

From this, conditional quantiles can be produced and in extension, a numerical representation of the estimated conditional distribution. Note that we have not used the exact mathematical notation in this segment, for the full presentation of the theory we refer to the source. The QRF method has been used successfully in calibrating wind speed and surface temperature from ensemble forecasts by Taillardat et al. (2016). It was shown to outperform the parametric Ensemble Model Output Statistics method, which can be considered baseline in ensemble post-processing.

2.4. Continuous ranked probability score

Evaluating probabilistic forecasts is not as straightforward as point predictions. Usually one or several scoring rules are employed. Common ones are the logarithmic score, continuous ranked probability score (CRPS) and variogram score (Bjerregård et al. (2021)). Since in this article we work with univariate prediction, and we get the estimated CDF from the

Data type	Longitude	Latitude
Measurement	58.0937	11.3312
Ensemble	58.101800	11.309300
Deterministic Forecast	58.099915	11.327288

Table 1: Coordinates of the different data-types

models, we will only consider CRPS (Matheson and Winkler (1976)) due to computational efficiency. The score is calculated through the following formula

$$CRPS(F, y) = \int_{-\infty}^{y} F(x)^2 dx + \int_{y}^{\infty} (1 - F(x))^2 dx \tag{3}$$

where F is the estimated CDF and y is the true value of a prediction. Scores closer to 0 are considered better. It is easy to see that the optimal predicted distribution is the step function in y. If we are 100% confident in the point-prediction, and it turns out to be right, we get the best possible score. To get the score over a set of predictions we take the mean CRPS of all predictions.

3. Method

All the code referenced in this part together with the data can be found on the Github[1] page for this article with the results in the supplied Jupyter Notebook[2]

3.1. Data

The data[3] used in the analysis is a little over a year's worth of measurements, ensemble forecasts and deterministic forecasts from January 2, 2022 until January 23, 2023. All measurements are made at noon and forecasts are produced 24 hours in advance. The coordinates of each data-type can be seen in Table 1.

3.1.1. Measurements

The gathered data was based around the choice of weather station where we could gather wind-speed measurements for surface wind. For this reason, we picked the station on Måseskär[4], a small island outside the west coast of Sweden, in the hopes that the local topography would affect the wind minimally. The weather station is operated by the Swedish Meteorological and Hydrological Institute (SMHI) so the data was subsequently gathered from their website [5].

[1] https://github.com/salthoff/wind-prediction-conformal

[2] https://nbviewer.org/github/salthoff/wind-prediction-conformal/blob/master/model_compare.ipynb

[3] Gathered under CC BY 4.0 license https://creativecommons.org/licenses/by/4.0/

[4] https://goo.gl/maps/rxTZtzNKs4BvCwGB6

[5] https://www.smhi.se/data/meteorologi/vind

Data type	Variable	Unit
Measurement	`wind speed`	m/s
Ensemble	`x_wind_10m`	m/s
	`y_wind_10m`	m/s
Deterministic	`wind_speed_10m`	m/s

Table 2: Data type with corresponding MET Norway and SMHI variable names and units.

3.1.2. Ensemble

The input that generates the probabilistic forecasts is the ensemble forecasts. In this case they come from the MetCoOp Ensemble Prediction System (MEPS) (Müller et al. (2017)) which is a numerical ensemble forecasting model for the Nordic region. It is a collaboration between the meteorological institutes of Sweden, Finland, Norway and Estonia. It is developed from the AROME model which was produced by météo-France. The ensemble system has 30 members meaning each forecast time should contain 30 different forecasts. New forecasts are produced every 6 hours and contain forecasts for each hour up to 61 hours into the future. The system produces forecasts over a grid which has a horizontal resolution of about 2.5 km. The grid spans the Scandinavian countries, Finland and the Baltic states as well as the Baltic Sea and parts of the North Sea and Atlantic Ocean. We gathered the data from one of the closest grid-points to the Måseskär weather station, the coordinates of which can be viewed in Table 1. The data was fetched from the Norweigan Meteorological Institute (MET Norway) using their thredds server[6].

3.1.3. Deterministic Forecasts

Before being delivered to the end consumer, the ensemble forecasts are post-processed to become deterministic. MET Norway supplies these forecasts as well through their thredds service. They are also down-scaled to a finer grid with about a 1 km resolution. Subsequently another closer point to the weather station was used the coordinates of which is also displayed in Table 1. The forecasts gathered here are the ones that are supplied to the service yr.no. These datapoints are effectively output from an underlying machine-learning model, which we utilize in our NECP and CPDS methods.

3.1.4. Data handling and cleaning

The raw data contains a lot of unnecessary variables as well as `NaN` values and thus the data has to be processed. What variables to include in a post-processing method is a topic for intense research. Here only the most basic variables were used for simplicity. The variables of each datatype together with units are presented in Table 2. Wind speed in this regard represents the mean wind speed at 10 meters over surface during 10 minutes. The measurements and deterministic forecasts have one value per time point, meaning if one of these are corrupted then that renders all data for that time useless. Thus we remove all such data in the import process. The ensemble data contains 30 ensemble members and

[6] https://thredds.met.no/thredds/metno.html

two variables, resulting in 60 values per time point. If the ratio of corrupted values is below 75% we bootstrap the missing values from the others of the same variable. If the ratio is above that threshold the time-point was however discarded. Bootstrapping is done at import stage which might result in slightly differing results between runs of the analysis.

3.2. Models

In the following section we will describe how each method was implemented. Since the application of CP and CPDS methods to forecast post processing is quite novel we keep modifications to the algorithms to a minimum and the QRF will be applied in a similar fashion. If a basic versions of these methods can compare to a basic version of the baseline, the QRF in this case, that would suggest that it is a valid technique to keep researching. The Conformal methods use some input, in this case the ensemble, together with the underlying point forecast to produce the distribution around it. The QRF uses the ensemble as input and produces a distribution directly from this data without an additional deterministic forecast. This is all visualized in Figure 1.

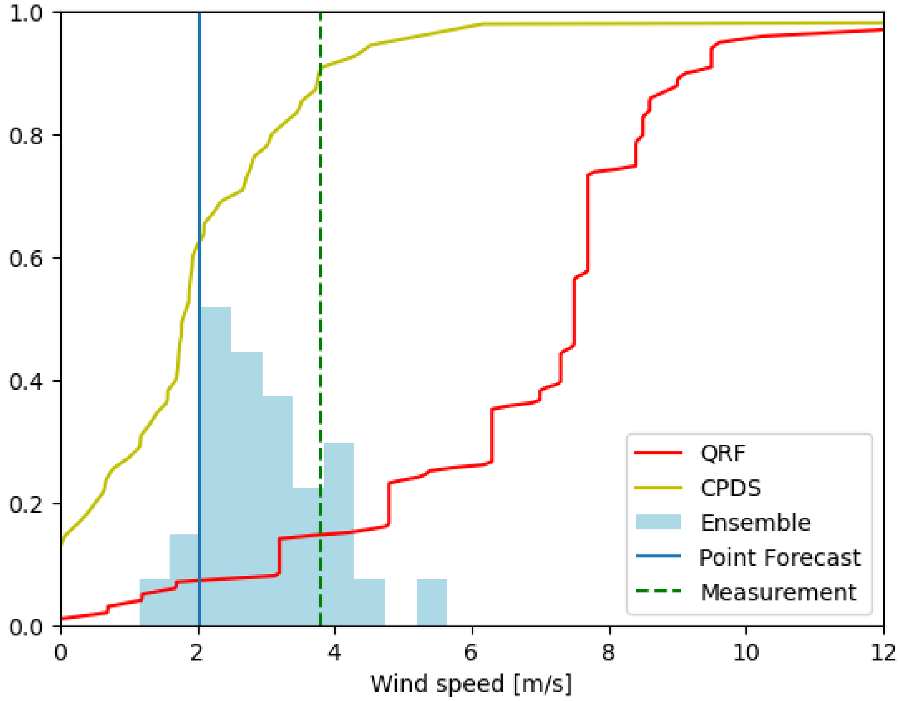

Figure 1: This illustrates data at one time point. These are the inputs, in form of resultant of the ensemble and the point forecast, the measurement and example CDFs from QRF and CPDS, respectively. Note that the vertical axis represents cumulative probability for the two CDFs and relative frequency for the histogram.

3.2.1. CPDS

For the CPDS model we used the python library Crepes (Boström (2022)). Here we use the normalized version which employs a k-nearest neighbors technique for determining the difficulty of a certain prediction. Here k is kept at the default value of 5. Since the underlying data might not be exchangeable we added a data window option. This allows the system to limit the calibration set to the past m training examples. The default is otherwise to include all training examples. When performing predictions the complete distribution is produced. All values below 0 are truncated since that is the minimum value possible. We do not expect the wind speed to be above 100 m/s thus that is added to the end of the distribution and if 0 is not present that is added to the beginning. Lastly the distribution is linearly interpolated to a length of 200 to keep it consistent throughout observations.

3.2.2. NECP

An NECP model typically only produces range predictions. To get distributions we sequentially add on larger prediction intervals. Each lower interval bound is added to the beginning of the distribution and the upper bound is analogously added to the end. This is done until final confidence level is reached. We remark that this constructs symmetric distributions around the point prediction. We have no theoretical guarantees that this will maintain validity for the distributions other than that the intervals around the point forecast has the coverage in (1) and (2). Since there might be a seasonality to wind, we also employ a weighting scheme with exponential decay in time. Given training examples z_i, $i = 1, \dots, n-1$, note that the order is chronological, we assign weights according to

$$\omega_i = \lambda^{n-i}, \lambda \in [0, 1]$$

where λ is some forgetting factor, used as a model hyperparameter. Note that if $\lambda = 1$ we get the standard conformal predictor.

Further we also employ a version of this algorithm with a custom conformity score. We will call this algorithm NECP Normalized (NECP-N). The validity guarantees are not bound by the choice of the conformity score so to demonstrate the flexibility of the system we produce this second one. This nonconformity score is influenced by the normalized nonconformity score in (Papadopoulus et al. (2008)) and the idea is to add information from each example that would describe the difficulty in predicting it's label. We proceed to define our nonconformity score as

$$\alpha_i = |y_i - \hat{y}_i|(1 + \beta^T \hat{\sigma}[\mathbf{x_i}]),$$

where β is a hyperparameter and $\hat{\sigma}[\mathbf{x_i}]$ is the estimated standard deviation of each variable in the input, *i.e.* the two wind components from each of the 30 ensemble forecast models. In this case $\hat{\sigma}[\mathbf{x_i}]$ is a vector of size 2×1 which requires β to be of the same size. To then get the prediction intervals for confidence $1 - \epsilon$, we arrive at an expression for the prediction range

$$
\begin{aligned}
\hat{C}_n = \{y \in \mathbf{Y} : A(\{z_1, \ldots, z_{n-1}\}, z_n) \leq \mathbf{Q}_{1-\epsilon}\left(\sum_{i=1}^{n-1} \tilde{\omega}_i \cdot \delta_{\alpha_i} + \tilde{\omega}_n \cdot \delta_{+\infty}\right)\} &\implies \\
|y_i - \hat{y}_i|(1 + \beta^T \hat{\sigma}[\mathbf{x_i}]) \leq \mathbf{Q}_{1-\epsilon}\left(\sum_{i=1}^{n-1} \tilde{\omega}_i \cdot \delta_{\alpha_i} + \tilde{\omega}_n \cdot \delta_{+\infty}\right) &\implies \\
|y_i - \hat{y}_i| \leq \frac{1}{(1 + \beta^T \hat{\sigma}[\mathbf{x_i}])}\left(\mathbf{Q}_{1-\epsilon}\left(\sum_{i=1}^{n-1} \tilde{\omega}_i \cdot \delta_{\alpha_i} + \tilde{\omega}_n \cdot \delta_{+\infty}\right)\right) &\implies \\
\hat{C}_n = \hat{y}_i \pm \frac{1}{(1 + \beta^T \hat{\sigma}[\mathbf{x_i}])}\left(\mathbf{Q}_{1-\epsilon}\left(\sum_{i=1}^{n-1} \tilde{\omega}_i \cdot \delta_{\alpha_i} + \tilde{\omega}_n \cdot \delta_{+\infty}\right)\right)&.
\end{aligned}
$$

Again, just as the CPDS case, we truncate parts below 0, add 100 (and potentially 0) and linearly interpolate to get a length of 200.

3.2.3. QRF

For the QRF we use the python library by Zillow available on Github [7]. Much like the the case for CPDS, to adjust for potential distribution drift, we implement a data window parameter. The other parameter is the number of trees of the forest. We let the library do the interpolation by requesting evenly spaced quantiles from $Q_{1/200}$ to $Q_{1-1/200}$ and add on the minimum and maximum values to the start and end. However, if necessary, we truncate the distribution as before and interpolate again ourselves. Note that we do not need to pass any deterministic forecast here, the QRF predicts directly from the ensemble.

3.3. Teaching schedule and evaluation

To emulate as closely as possible what a real-world scenario would look like, we use a teaching schedule to determine which of our different methods performs the best according to the CRPS metric, described below in 3.5. All examples are, as stated earlier, sorted in chronological order. Through this schedule we train, where applicable, and calibrate models for a set of hyperparameters. To determine optimal hyperparameters, at each time point, we do model selection which will be defined below. All examples recorded before the date will be used for training and model selection, and the example at the prediction date will be used for testing. The prediction date is advanced in time until it reaches the end of our dataset. This means that we will have one CRPS metric calculated for all examples predicted and for each method, respectively. The final evaluation is done by computing the average of the CRPS metric for each method.

3.4. Model selection

As mentioned earlier, each method has hyperparameters and to select a configuration of the hyperparameters we will mimic a real world scenario as closely as possible. Some of the methods are computationally costly so we will use two versions

[7] https://github.com/zillow/quantile-forest

3.4.1. Base model selection

This selection is used for NECP and NECP-N. This is the ideal way to select hyperparameters as it handles the training examples from our teaching schedule in the same way as the teaching schedule handles all examples. Given a prediction date, we calibrate the models on all preceding dates before making the prediction. This prediciton date is advanced until it reaches current prediction date of the teaching schedule. All different configurations of the hyperparameters is used to build models . The resulting CRPS metrics are averaged and the best configuration is chosen for the next prediction in the teaching schedule.

3.4.2. Block model selection

The reason for using a block model selection is that CPDS and QRF are more computationally costly and this will reduce the model-selection time. Similar to the base model selection, all examples prior to the prediction date are used as training examples. Here, we split the current training examples in to 5 equal size, chronological in time subsets. Each subset is held out, in turn, for calculating the CRPS metric and the remainder of the training examples are used for training. If some of the training examples lay in the future compared to the held out subset, then those examples are prepended to the other training examples.

3.5. Evaluation metric

The predicted distributions come in the form of a vector of quantiles from the minimum value of 0 to the maximum of 100. We estimate the CDF from this empirically

$$\hat{F}(x) = \sum_{i=0}^{m-1} \frac{1}{m} \cdot \mathbf{1}_{\{Q_{i/m} \leq x\}} \tag{4}$$

where $Q_{i/m}$ is the i/m quantile and m is the number of quantiles. To then get the CRPS we insert (4) and the actual wind measurement into (3) and integrate using the trapezoidal rule. We also evaluate the validity of certain intervals. Since the predictions give us the quantiles, we can just calculate the ratio of points that fall within their respective $Q_{0.05}$ and $Q_{0.95}$ to get the validity of the 90% prediction interval.

4. Results

The hyperparameter configurations used in the teaching schedule for all the systems are presented in Table 3. Additionally the ratio of each configuration to make the final predictions is also displayed. The initial data split used is January 2, 2022 to March 1, 2022 for training and March 2, 2022 to January 23, 2023 for testing. The data sets then contain 55 and 314 examples, respectively.

The measured results, besides the mean CRPS, are the validity and width of 0.9 and 0.5 prediction intervals, respectively. The width can be interpreted as an efficiency metric. These results for each model type are presented in Table 4. An important note is that since the quantiles are not located exactly on the desired ones, we choose the closest ones and thus we get a small bias in theoretical validity. We should then have 0.8995 and 0.4975

CPDS	Window length	Used in test
1	all	0.445
2	200	0.131
3	100	0.287
4	50	0.134

NECP	Forgetting factor	Used in test
1	1	1
2	0.995	0
3	0.99	0
4	0.98	0
5	0.97	0

NECP-N	Forgetting factor	β	Used in test
1	1	$[0.00, 0.00]$	1
2	1	$[0.05, 0.05]$	0
3	1	$[0.10, 0.10]$	0
4	0.99	$[0.00, 0.00]$	0
5	0.99	$[0.05, 0.05]$	0
6	0.99	$[0.10, 0.10]$	0

QRF	Number of trees	Window Length	Used in test
1	200	100	0.172
2	100	all	0.003
3	200	all	0.825

Table 3: Sets of hyperparameters used in the teaching schedule for each of the models as well as the ratio of how much each were used in the final result. Due to round-off the ratios do not all add to 1.

as target, respectively. We can observe that NECP and NECP-N have basically the same value of the CRPS metric. Also, CPDS and QRF only differ about 0.0004 in error from optimal validity in the 0.9 case. The execution times[8] for each model's teaching schedule is presented in Table 5.

5. Discussion

One fundamental assumption we make in this study is that the underlying forecasting model is not trained on the data we use. If it was it would violate the conditions for inductive

[8] The tests were performed on a 2020 MacBook air with M1 processor using a single core.

Model type	CRPS	0.9 validity	0.9 mean width	0.5 validity	0.5 mean width
CPDS	0.8926	0.9172	7.4997	0.5255	2.1089
NECP	**0.8649**	0.9299	**6.0870**	0.5064	**2.0621**
NECP-N	**0.8649**	0.9299	**6.0870**	0.5064	**2.0621**
QRF	0.9207	**0.8822**	6.9348	**0.5000**	2.3975

Table 4: Overall results from each model. Note that the actual target intervals are 0.8995 and 0.4975 instead of 0.9 and 0.5. The best result in each category is written in bold.

Model	CPDS	NECP	NECP-N	QRF
Time (min:s)	03:07	08:26	11:09	41:16

Table 5: Execution times for each teaching schedule.

conformal-based methods. However, with an external model it is not possible to perform for example a full conformal-based method since that requires retraining of the model for all data points. If the underlying model is trained on the data we use, that might have a negative impact on the validity of the predicted distributions.

It is clear from the results in Table 5 that the conformal-based methods are computationally faster than the QRF. However, the conformal-based methods have instant access to the results of the underlying forecasting model. In the real world this would take time to produce, while the QRF can work directly on the ensemble. Simultaneously this is one of the strengths of conformal-based methods. In many cases we might already have a good deterministic model which we would like to supplement with probabilistic predictions. The conformal-based methods then provide that very efficiently, of course with the caveat that some data has to be sacrificed for calibration.

We supply the raw ensemble output of the wind components to the QRF for post processing. This should not be seen as an optimal use of the QRF algorithm. We can see that Taillardat et al. (2016) supply a lot more variables to the QRF and generally supply different quantiles rather than the entire ensemble. Thus we can assume that the QRF results can be improved with further development. However, likewise there are improvements to be made to the conformal-based methods as well. The more important question is perhaps how much can the conformal-based methods be improved?

The first thing to look at, to improve the conformal-based methods is if it is possible to improve the underlying deterministic model. Assuming that this is optimal there are other things to consider. One such thing could be the implementation of Mondrian conformal predictive distributions (Boström et al. (2021)) which is based on sorting the calibration data into different categories. A Mondrian approach might produce more distinct conditional distributions and categories could be created based on different wind directions. We also believe that there still is a potential for improvement regarding the conformity score in this case. Adding information from different variables and ensuring that these can have a greater impact than what we get in NECP-N might be a good start. However, NECP-N still introduces flexibility and allows the normalization to be completely removed by setting $\beta = 0$. We also note that the forgetting factor in NECP is never used in our study. Our conclusion here is similar to that of the normalization; it is better to allow for it as it might be useful when studying other types of datasets.

The results in Table 4 do not directly give any conclusive information about which model is the best in this case. The QRF method seems to produce the intervals with closest to target validity. It does this with generally wider intervals than the conformal-based methods, the only exception being the CPDS in the 0.9 case. However, validity of the conformal-based methods is also always slightly conservative, which might be desirable in many cases. The reason why the conformal-based methods have higher validity and generally narrower intervals might be that they base their prediction on a well trained deterministic forecast. If the QRF was set to predict the error of the deterministic forecast, it might compete better with the conformal-based methods in this regard. However then the QRF would loose the competitive edge of not having to wait for a point predicting, meaning the conformal-based methods would edge out in computational efficiency. Looking at the CRPS, we can determine that the NECP methods outperform both CPDS and QRF by a solid margin. The CPDS model is also better in this regard than the QRF but due to the random nature of the QRF we might see differing results between runs. So while CPDS performs better in this case, with some margin, the gap might be closed over several test runs.

5.1. Conclusions

While we cannot draw any solid conclusions about the usefulness of the methods from data during a small time period and in a single location, we can say something about what these initial results tell us. Conformal-based methods provide a simple and fast way of supplementing wind-speed forecasts with a probabilistic prediction. The out-of-box versions of the conformal-based methods can compete with the QRF method both in terms of validity and efficiency when supplied with the raw ensemble. Further research should be put into improving the conformal-based methods to see how they hold up to more optimized versions of the QRF and other popular ensemble post-processing techniques.

References

Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability, 2023.

Ioannis Bazionis, Panagiotis Karafotis, and Pavlos Georgilakis. A review of short-term wind power probabilistic forecasting and a taxonomy focused on input data. *IET Renewable Power Generation*, 16, 01 2022. doi: 10.1049/rpg2.12330.

Mathias Blicher Bjerregård, Jan Kloppenborg Møller, and Henrik Madsen. An introduction to multivariate probabilistic forecast evaluation. *Energy and AI*, 4:100058, 2021. ISSN 2666-5468. doi: https://doi.org/10.1016/j.egyai.2021.100058. URL https://www.sciencedirect.com/science/article/pii/S2666546821000124.

Henrik Boström. crepes: a python package for generating conformal regressors and predictive systems. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction and Applications*, volume 179 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Henrik Boström, Ulf Johansson, and Tuwe Löfström. Mondrian conformal predictive distributions. In Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin, and Khuong An Nguyen, editors, *Proceedings of the Tenth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 152 of *Proceedings of Machine Learning Research*, pages 24–38. PMLR, 08–10 Sep 2021. URL https://proceedings.mlr.press/v152/bostrom21a.html.

Christopher Kath and Florian Ziel. Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, 37 (2):777–799, apr 2021. doi: 10.1016/j.ijforecast.2020.09.006.

James E. Matheson and Robert L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2629907.

Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(35):983–999, 2006. URL http://jmlr.org/papers/v7/meinshausen06a.html.

Malte Müller, Mariken Homleid, Karl-Ivar Ivarsson, Morten Køltzow, Magnus Lindskog, Knut Midtbø, Ulf Andrae, Trygve Aspelien, Lars Berggren, Dag Bjørge, Per Dahlgren, Jørn Kristiansen, Roger Randriamampianina, Martin Ridal, and Ole Vignes. Arome - metcoop : A nordic convective scale operational weather prediction model. *Weather and Forecasting*, 32, 01 2017. doi: 10.1175/WAF-D-16-0099.1.

Harris Papadopoulus, Alex Gammerman, and Volodya Vovk. Normalized nonconformity measures for regression conformal prediction. volume 152 of *Proceedings of AIA 2008*, pages 64–69. ACTA Press, Feb 2008.

Maxime Taillardat, Oliver Mestre, Michaël Zamo, and Philippe Naveau. Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Monthly Weather Review*, 144:2375–2393, 2016. doi: 10.1175/MWR-D-15-0260.1.

Vladimir Vovk. Conformal predictive distributions: an approach to nonparametric fiducial prediction. On-line Compression Modelling Project Working paper 30, aug 2020. URL http://alrw.net/articles/30.pdf.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, New York, 2022. ISBN 0-387-00152-2.