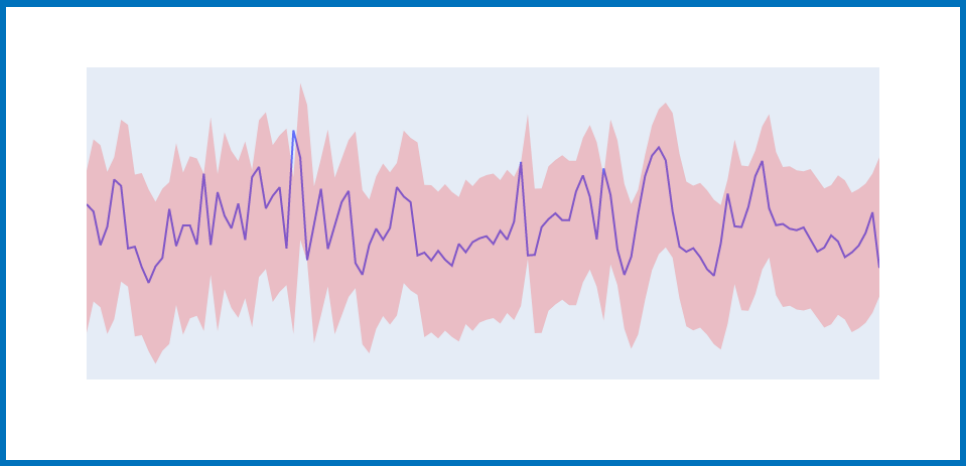
Benchmarking conformal prediction methods for time series regression



Derck W.E. Prinzhorn

Layout: typeset by the author using LaTeX.
Cover illustration: Derck Prinzhorn

Benchmarking conformal prediction methods for time series regression

Derck W.E. Prinzhorn
13058207

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 900
1098 XH Amsterdam

Supervisor
Alexander R. Timans MSc

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 900
1098 XH Amsterdam

Semester 2, 2022-2023

Abstract

As machine learning is becoming increasingly prevalent, the need for uncertainty quantification has grown substantially. One way of quantifying uncertainty is conformal prediction, designed to achieve valid marginal coverage. One of the challenges for conformal prediction is possible distribution shift over time when applied to time series, requiring adaptive conformal methods. This thesis presents an empirical experiment in which recently proposed conformal prediction methods that are theoretically suited for application to time series are benchmarked. These methods are applied to various base model predictions, constructing conformal prediction intervals. After evaluating uncertainty quality, the results demonstrate that Ensemble Batch Prediction Intervals (EnbPI) generally outperforms other adaptive conformal prediction methods and provides more narrow and informative prediction intervals.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Alexander Timans. His precise direction and support paved the way to overcome obstacles that emerged throughout my project. His consistent availability and pinpoint feedback are profoundly appreciated.

Contents

1 Introduction

Today's machine learning (ML) models are used in various settings, including some with high risk (Hupont, Micheli, Delipetrev, Gómez, & Garrido, 2023). Misguided applications of ML in areas such as medical diagnostics (Karmakar, Chatterjee, Das, & Mandal, 2023) and autonomous driving can have severe consequences, including accidents and injury (Siddiqui & Albergotti, 2022). In order to deal with model failures, models should have the ability to quantify the uncertainty of predictions. Ideally, predictions would be exact. However, achieving this precision is highly unlikely. Nevertheless, uncertainty quantification can be achieved using methods by which one can be confident that the true value is contained within the prediction region with a certain probability. These models could still fail, but provide information to improve the decision-making process.

One way of quantifying such uncertainty is conformal prediction. This is done by creating statistically rigorous prediction sets for classification and intervals for regression to quantify uncertainty (Figure 1.1). An interval is statistically rigorous when it is a valid and reliable measure of uncertainty. They are valid as well as distribution-free (Angelopoulos & Bates, 2021) due to non-asymptotic guarantees without any distributional assumptions. This is useful, since these assumptions may be difficult to justify.

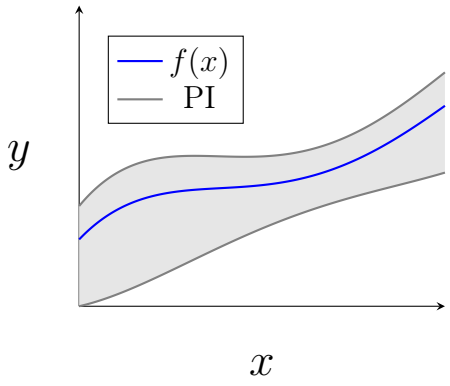


Figure 1.1: Uncertainty quantification using a prediction interval (PI) for a 1-D exemplary regression setting.

As a result of the theoretical guarantee of conformal methods, it is possible to define a user-chosen error rate α, resulting in a prediction interval with a marginal probability of $1 - \alpha$ to contain the correct value. This validity could also be ob-

tained by constructing an extremely large prediction interval, but this would be uninformative. Therefore, the conformal method should not only be valid but also efficient (Vovk, Gammerman, & Shafer, 2005), constructing small sets for classification and small intervals for regression. Conformal prediction can be considered a post hoc wrapping function around any uncertainty-capable model. Thus, it can be used with any heuristic notion of uncertainty (e.g. quantile predictions), converting it to a rigorous one (Angelopoulos & Bates, 2021), as shown in Figure 1.2.

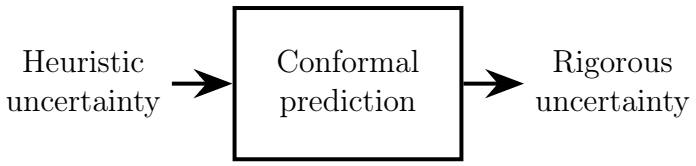


Figure 1.2: Conformal prediction process.

Whereas current conformal prediction methods work as expected in non-time series regression and classification problems, in time series the guaranteed probability of $1 - \alpha$ no longer holds due to possible distribution shift (section 2.4). This is problematic for various applications of conformal inference in a time series setting, such as climate temperature prediction and financial trading. Recently, several conformal methods have been proposed to account for this distribution shift.

In this thesis, these proposed conformal prediction methods will be examined for four different time series regression datasets. The proposed methods provide theoretical conformal coverage guarantees, which remain to be thoroughly examined empirically. Therefore the focus of this thesis is on their implementation. Recently proposed conformal prediction implementations for time series that will be examined in this work are: Adaptive Conformal Inference (Gibbs & Candes, 2021), Ensemble Batch Prediction Intervals (Xu & Xie, 2021) and Ensemble Conformalized Quantile Regression (Jensen, Bianchi, & Anfinsen, 2022) and their respective performances on various real-life datasets will be evaluated. Consequently, this work is guided by the following two research questions:

1. To what extent do proposed conformal methods designed for time series settings, provide practical solutions for diverse, real-life time series, and are their theoretical coverage guarantees supported by valid empirical coverage?

2. How do the proposed conformal methods designed to address time series compare to each other in terms of predictive performance and uncertainty quantification quality on the basis of different learning models for real-life time series problems?

2 | Background

This chapter delves into key topics that form the basis of this thesis. Firstly, quantile regression, a technique that enables predictions at different quantiles, is described. This is followed by an introduction to time series and its inherent components. The final part of this chapter focuses on the key concept of this thesis: conformal prediction. Its application is discussed, particularly in time series settings, to provide a comprehensive understanding of its role in this research.

2.1 Quantile regression

In classical regression, a model predicts the mean value of a variable. Such models could also predict a median, which is the 50% quantile. Quantile regression is a method designed to estimate different specific conditional quantile functions (Koenker & Bassett, 1978). In contrast to a cumulative distribution function (CDF), which maps probability p values from quantile values q, a quantile function is the inverse of the CDF and does the opposite: it gives q values as a function of the p values (Hao & Naiman, 2007). For example, when forecasting the quantile at 80^{th} to be the value 50, the predictive variable is forecasted to have a probability of 80% of being below the predicted value of 50. To calculate the α^{th} conditional quantile function, we find the greatest lower bound where the cumulative density function is greater than or equal to α (Figure 2.1). This can be written formally as[1]:

$$q_\alpha(x) := \inf\{y \in \mathbb{R} : F(y \mid X = x) \geq \alpha\}, \tag{2.1}$$

where $F(\cdot)$ is defined as the CDF of the data (Romano, Patterson, & Candes, 2019).

[1]See Appendix A for mathematical naming conventions.

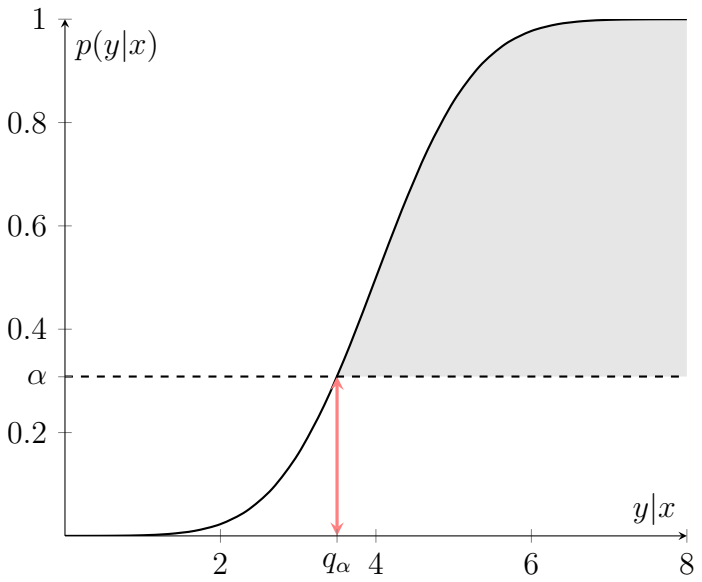


Figure 2.1: Determining the α^{th} conditional quantile from a CDF. The region highlighted in grey signifies where $F(y \mid X = x) \geq \alpha$.

In this thesis, quantile regression is used to fit one or more quantiles (Figure 2.2), which can be used as point predictors individually, or in the case of predicting multiple quantiles can be used to construct uncertainty regions such as prediction intervals by considering all values between two quantiles. In order to approximately obtain a symmetrical PI with coverage of $1-\alpha$, where α is a design choice, the two quantiles are fixed such that $\alpha_{\text{low}} = \alpha/2$ and $\alpha_{\text{upper}} = 1 - \alpha/2$. After determining the quantile functions, a prediction interval $\hat{\mathcal{C}}(x) = [q_{\alpha_{\text{low}}}(x), q_{\alpha_{\text{up}}}(x)]$ is obtained.

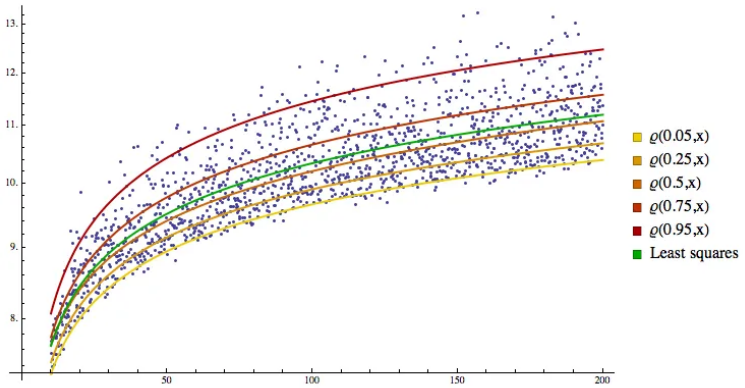


Figure 2.2: Quantile regression and least squares on an exemplary regression setting (Dye, 2020).

2.2 Time series

A time series consists of four major components, trends, seasonality, cycles and random variations (Figure 2.3). A trend is a persistent long-term change in the mean of a series, which is the slowest-moving part of a series (Hamilton, 1994). A simple method of recognizing the trend is using a moving average. This method computes an average in a sliding window. Seasonality is present when there is a regular periodic change in the mean of a series (Hyndman & Athanasopoulos, 2021). To model those seasons, Fourier transforms are commonly used. Cycles are typically encountered in systems that affect themselves. Whereas trends and seasons are time-dependent, cycles are not necessarily (Jose, 2022). For traditional time series, models such as ARIMA, autoregressive integrated moving average, are used to predict future values based on past values (Hyndman & Athanasopoulos, 2021).

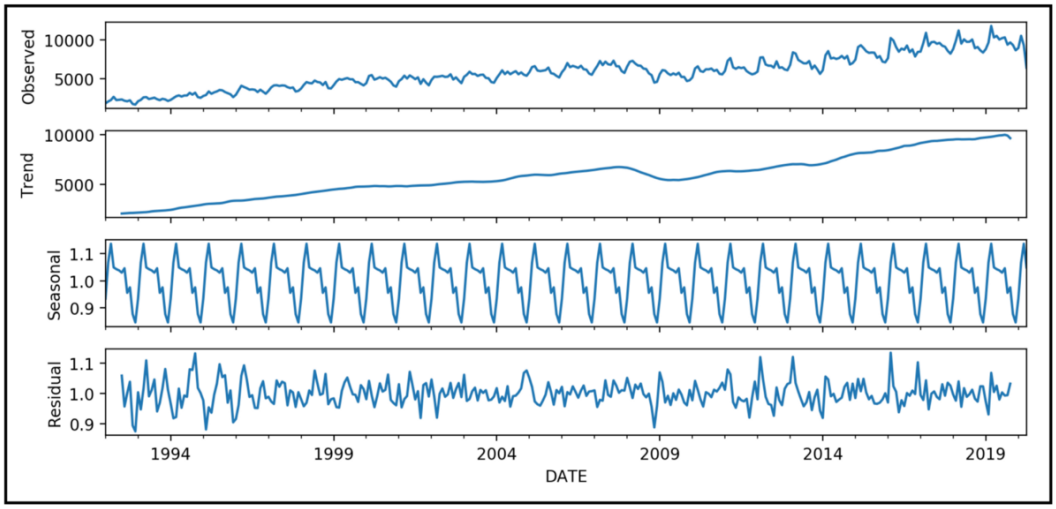


Figure 2.3: Decomposition of a time series (Date, 2021).

Stationary series require the mean of the series, as well as the covariance and variance, not to be a function of time (Shumway & Stoffer, 2011). In the case of variance, this stationary property would be called homoscedasticity. Likewise, the non-stationary property is called heteroscedasticity.

2.3 Conformal prediction

Conformal prediction methods construct prediction sets, which are intervals with a lower and upper bound in regression, to quantify the uncertainty of a model.

Conformal prediction guarantees that for any $(X_1, Y_1), \ldots, (X_n, Y_n)$ used as calibration data, with a probability of at least $1 - \alpha$, and under the assumption of *exchangeability* (section 2.4) of $\{(X_1, Y_1), \ldots, (X_n, Y_n), (X_{test}, Y_{test})\}$ for some test sample (X_{test}, Y_{test}), the ground truth value Y_{test} is covered by the prediction set $\mathcal{C}(X_{test}) \subset \mathcal{Y}$, i.e. we have that:

$$\mathbb{P}(Y_{test} \in \mathcal{C}(X_{test})) \geq 1 - \alpha \tag{2.2}$$

This validity property is called marginal coverage, because the probability in Equation 2.2 is marginal, i.e. across all test and calibration data on average (Angelopoulos & Bates, 2021). There is another type of coverage, called conditional coverage which is ideally the goal. Conditional coverage implies that the coverage is guaranteed across each test sample in the data. However, this is not achievable in finite samples. Formally, it can be written as:

$$\mathbb{P}[Y_{test} \in \mathcal{C}(X_{test}) \mid X_{test}] \geq 1 - \alpha \tag{2.3}$$

The conformalization process consists of four steps (Figure 2.4). Firstly, a labeled dataset is split into a training, calibration and test set. Then a model is fitted on the training data, after which predictions are made for the calibration dataset. These predictions are used to calibrate the prediction interval by the use of conformity scores, which are determined by the errors of the prediction. Finally, when the model makes predictions on the test data, a specific conformal quantile is taken from the conformity scores, which is used to construct the prediction interval around test predictions, as can be seen in Equation 3.12.

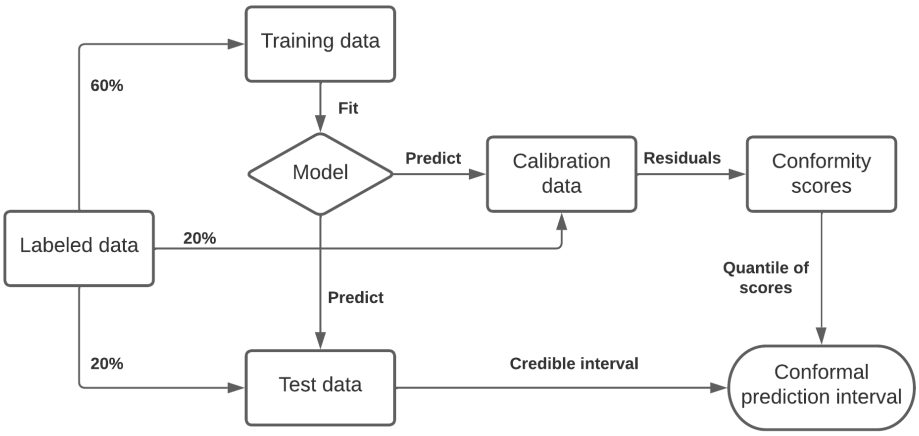


Figure 2.4: The process of conformalization visualised as a flow diagram.

2.4 Conformal prediction for time series

Although conformalized quantile regression is a possible approach for regular regression problems, it is not a realistic approach for time series problems due to the assumption of *exchangeability*. A sequence of random variables is *exchangeable* when its joint probability distribution is a symmetric function of its arguments (O'Neill, 2009). Intuitively, this means variables in a sequence can be reordered without altering the joint probability distribution (e.g. $P(x_1, x_2, x_3) = P(x_3, x_1, x_2)$). This assumption does not hold, since there is a possible shift in distribution in time.

To account for this distribution shift, various methods have been proposed. Among those methods are Adaptive Conformal Inference (ACI), Ensemble Batch Prediction Intervals (EnbPI) and Ensemble Conformalized Quantile Regression (EnCQR). Technically, ACI does rely on the classical exchangeability assumption, similar to non-adaptive conformal prediction methods. In contrast to CQR, it conformalizes by modelling the distribution shift by learning an additional parameter. EnbPI does not rely on exchangeability, but replaces this assumption with mild assumptions on the error process and the estimation quality of regressors. EnCQR does not rely on exchangeability either and is similar to EnbPI, since it also uses ensemble learners and updated conformity scores to conformalize. These methods will be explained in more detail in the next chapter.

3 | Methods

This chapter begins with an outline of the baseline model. It then delves into the regression algorithms used for quantile prediction and concludes with an exploration of the proposed conformal methods.

3.1 Baseline

ARIMA is used as a baseline model to benchmark the other models (Hyndman & Athanasopoulos, 2021). The full non-seasonal ARIMA(p, d, q) (see Table 3.1 for the parameters), can be written as:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{3.1}$$

where y'_t is the differenced time series with AR parameter ϕ_p and ϵ_t is white noise with MA parameter θ_q.

p = order of the autoregressive part
d = degree of first differencing involved
q = order of the moving average part

Table 3.1: ARIMA parameters

In addition to the mean prediction, ARIMA uses the standard deviation of the residuals $\hat{\sigma}$ to construct prediction intervals $[\mu - z\hat{\sigma}, \mu + z\hat{\sigma}]$, where $z = 1.64$ for a 90% prediction interval (Hyndman & Athanasopoulos, 2021). This value for z is only a valid symmetric quantile under normality assumption, i.e. this prediction interval construction assumes a Gaussian and is not distribution-free. The conformal methods will be benchmarked to this as well. Par example, a simple model is ARIMA$(0, 0, q)$, which can be written as:

$$y_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{3.2}$$

During forecasting the estimated variance using the model above can then be written as:

$$\hat{\sigma}_h = \hat{\sigma}^2 \left[1 + \sum_{i=1}^{h-1} \hat{\theta}^2 \right], \quad \text{for } h = 2, 3, \ldots, \tag{3.3}$$

where h is the forecasting horizon, $\hat{\theta}_i = 0$ for $i > q$ and a 90% prediction interval is given by $\hat{y}_{T+h|T} \pm 1.64\sqrt{\hat{\sigma}_h}$.

3.2 Regression algorithms

To estimate conditional quantiles, various regression algorithms are used. This section details the regression algorithms used to estimate conditional quantiles. It starts with the linear model, followed by outlines of the random forest and neural network.

3.2.1 Quantile Linear Regression

A linear model is used as the simplest base model. It gives linear predictions $\hat{y}(w, X) = Xw$ for a quantile q, for which weights w are found by minimizing the following function:

$$\min_w \frac{1}{n} \sum_i PB_q(y_i - X_i w) + \alpha ||w||_1, \tag{3.4}$$

where α is the parameter controlling the L1 regularization penalty and where $PB_q(\cdot)$ is the quantile loss function (Pedregosa et al., 2011). This is equivalent to:

$$PB_q(y, \hat{y}_q) = \begin{cases} q(y - \hat{y}_q) & \text{if } y > \hat{y}_q \\ (1-q)(\hat{y}_q - y) & \text{otherwise} \end{cases} \tag{3.5}$$

The pinball loss, visualised in Figure 3.1, or quantile loss function, is larger when the difference between the quantile forecast $\hat{y}_q$ and the target y is larger (Romano et al., 2019). In contrast to L1 loss, quantile loss penalizes overprediction and underprediction differently.

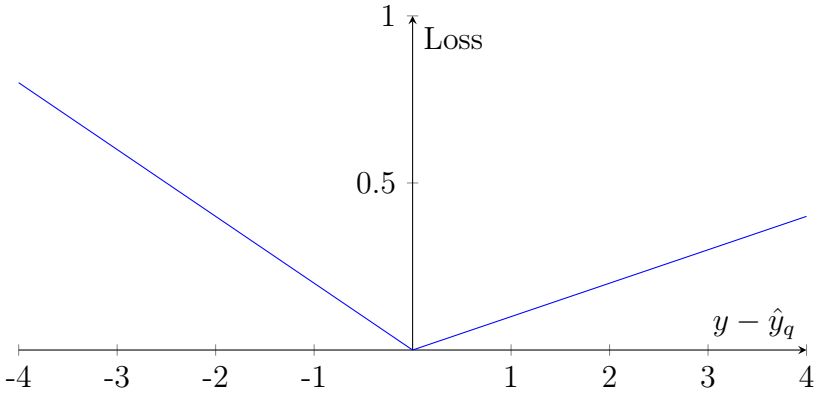


Figure 3.1: Visualisation of the pinball loss function in Equation 3.5.

3.2.2 Quantile Random Forest

Secondly, a quantile regression forest is used as a tree-based base model. Classic random forest regressors estimate the conditional mean by a weighted sum over all observations Y. Unlike these classic random forests, quantile regression forests estimate $F(Y = y|x) = q$, i.e. the conditional CDF of the target values being equal to the quantile, by a weighted mean over the observations of $\mathbb{1}_{\{Y \leq y\}}$ (Meinshausen, 2006).

$$\hat{F}(y|X = x) = \sum_{i=1}^{n} w_i(x)\ \mathbb{1}_{\{Y_i \leq y\}} \tag{3.6}$$

The weight w_i (Equation 3.7) corresponds to the fraction of samples in the same leaf, if the observation y_i is in the same leaf as the new observation x falls into. If not, the weight is zero (Meinshausen, 2006).

$$w_i(x) = \frac{1}{T} \sum_{t=1}^{T} \frac{\mathbb{1}(y_i \in L(x))}{\sum_{j=1}^{N} \mathbb{1}(y_j \in L(x))}, \tag{3.7}$$

where $t \in T$ represents every tree in the forest and $L(x)$ denotes the leaf that x falls into.

3.2.3 Quantile Neural Network

As the most complex model, a quantile regression neural network (QRNN) is used (Pfreundschuh, 2020). The most simple neural network (Goodfellow, Bengio, & Courville, 2016), computes its predictions by an affine transformation, on top of which an activation g can be used:

$$y = g(\mathbf{W}^\top \mathbf{x} + \mathbf{b}) \tag{3.8}$$

A more complex variation of this, a deep neural network, uses multiple hidden layers (Goodfellow et al., 2016). These layers are arranged in a chain structure, with each layer being a function of the layer that preceded it, each computed as:

$$h^{(l+1)} = g(\mathbf{W}^{(l+1)\top} h^{(l)} + \mathbf{b}^{(l+1)}) \tag{3.9}$$

The weights of the neural network layers are typically optimized by minimizing a loss function, such as Mean Squared Error (MSE) or Mean Absolute Error (MAE), which represents the error between predictions and the ground truth values. However, for quantile regression neural networks to effectively learn a selection of quantiles, the quantile loss function is used. The total loss is computed by the sum of the quantile loss for each quantile. This approach enables the QRNN to predict the

quantiles of the distribution $p(y|x)$. In the most extreme case, it can predict all 99 quantiles in parallel and get a CDF, as shown in Figure 3.2. Instead of predicting all these quantiles, only two or three quantiles are predicted to construct a PI, as these are the upper and lower bounds and a median.

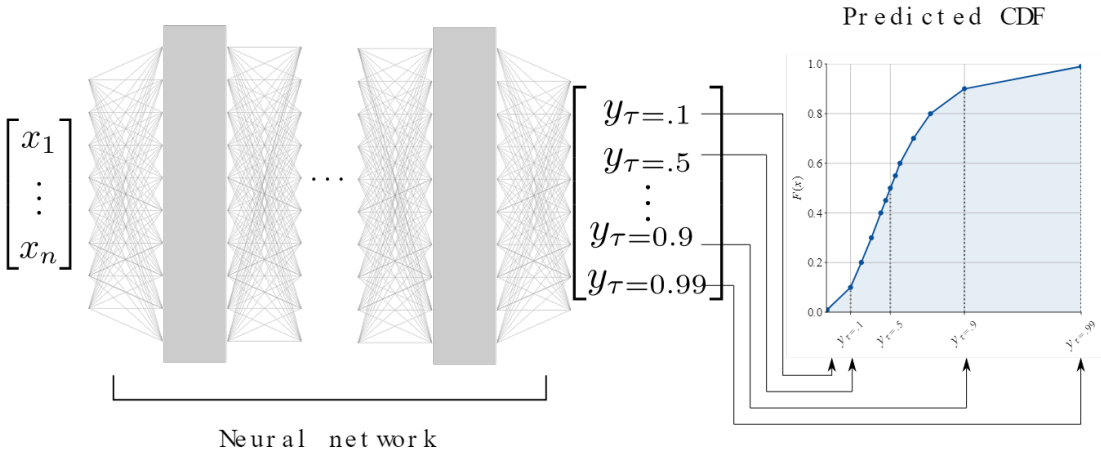


Figure 3.2: A quantile regression neural network that predicts a sequence of quantile functions y_τ for $\tau \in [0, 1]$ as a CDF (Pfreundschuh, 2020).

3.3 Conformal methods

This section provides a description of the four conformal methods used in this thesis. It begins with an explanation of the non-adaptive conformal prediction method CQR, followed by outlines of the adaptive methods: ACI, EnbPI and EnCQR.

3.3.1 CQR

The predicted conditional quantiles are only approximately accurate and in finite samples they may deviate from the true conditional quantile. This is why conformalization is useful, since it guarantees finite-sample coverage (Romano et al., 2019). Conformalized Quantile Regression splits the training data into a separate training and calibration set, $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively. Subsequently, two conditional quantile functions are fitted by any quantile regression model $\hat{f}$ on the training set:

$$\{\hat{q}_{\alpha_{\text{low}}}, \hat{q}_{\alpha_{\text{up}}}\} \leftarrow \hat{f}(\{(X_i, Y_i) : i \in \mathcal{I}_1\}) \tag{3.10}$$

Next, the conformity scores S_i are computed, which are used to quantify the error of the estimated prediction interval $\hat{\mathcal{C}} = [\hat{q}_{\alpha_{\text{low}}}(x), \hat{q}_{\alpha_{\text{up}}}(x)]$, by taking the maximum

of the residuals between the quantiles and the true value.

$$S_i := \max\{\hat{q}_{\alpha_{\text{lo}}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{\text{up}}}(X_i)\}, \text{ for each } i \in \mathcal{I}_2 \tag{3.11}$$

For a new datapoint (X_{test}, Y_{test}) the conformal prediction interval is then constructed by computing a specified quantile of the conformity scores corresponding to a chosen confidence level $1 - \alpha$, and extending the initial quantile prediction interval with that quantile. Formally, this is defined as:

$$\mathcal{C}(X_{test}) = [\hat{q}_{\alpha_{\text{low}}}(X_{test}) - Q_{1-\alpha}(S, \mathcal{I}_2), \hat{q}_{\alpha_{\text{up}}}(X_{test}) + Q_{1-\alpha}(S, \mathcal{I}_2)], \tag{3.12}$$

where $Q_{1-\alpha}(S, \mathcal{I}_2)$ is the $(1-\alpha)(1+1/|\mathcal{I}_2|)$-th empirical quantile of $\{S_i : i \in \mathcal{I}_2\}$.

3.3.2 ACI

Adaptive Conformal Inference is based on the concept of achieving approximate or exact marginal coverage by iteratively calibrating the miscoverage rate α of the prediction interval (Gibbs & Candes, 2021). ACI uses an online update to increase or decrease the estimate of the miscoverage rate α, based on the empirical miscoverage rate. If the prediction intervals were historically over-covering, α_t is increased, aiming to construct a more narrow interval. On the other hand, under-coverage causes α_t to decrease, thus aiming for a higher probability of including the ground truth value. This results in larger prediction intervals. ACI uses a binary variable err_t, which is equal to 1 if the prediction interval is under-covering and 0 otherwise. Using a step size parameter $\gamma > 0$, the online update can be applied by tuning α using the previous miscoverage frequency α_t. The online update can be written as follows:

$$\alpha_{t+1} = \alpha_t + \gamma(\alpha - \text{err}_t) \tag{3.13}$$

Another way would be to create a sequence of increasing weights and assign them to previous errors and tune α using the following weighted error:

$$\alpha_{t+1} = \alpha_t + \gamma \left(\alpha - \sum_{s=1}^{t} w_s \text{err}_s \right), \tag{3.14}$$

where $\{w_s\}_{1 \leq s \leq t} \subseteq [0, 1]$ is a sequence of increasing weights with $\sum_{s=1}^{t} w_s = 1$.

3.3.3 EnbPI

Ensemble Batch Prediction Intervals uses bootstrap ensemble estimators to construct sequential prediction intervals (Xu & Xie, 2021). It constructs a batch of prediction intervals at once, sequentially for every time step. The algorithm consists of 3 main phases:

1. **Boostrapped ensemble estimators**. B bootstrap models are trained on a subset of the data which is sampled with replacement. A quantile point predictor $\hat{f}^b$ is fitted for every $b \in$ B.

2. **Compute conformity scores**. All models create predictions on the samples not included in the data S_b on which the models are trained (leave-one-out predictions). All model predictions are then aggregated using an aggregation function ϕ:

$$\hat{f}^{\phi}_{-i}(x_i) = \phi(\{\hat{f}^b(x_i) \mid i \notin S_b\}) \tag{3.15}$$

   For all those aggregated predictions the predictions residuals are computed as $\hat{\epsilon}^{\phi}_i = |y_i - \hat{f}^{\phi}_{-i}(x_i)|$.

3. **Construct PI for test data and update residuals**. For every future time steps $t = T+1, \ldots, T+T_1$, we compute the $(1-\alpha)$-quantile of the aggregated predictions until the current time T and construct the PI by adding a margin with the size of the $(1-\alpha)$-quantile of the prediction residuals. For all these future time steps the set of prediction residuals is updated by removing the oldest residual score and adding the new residual.

3.3.4 EnCQR

Ensemble Conformalized Quantile Regression constructs prediction intervals that are distribution-free and approximately marginally valid (Jensen et al., 2022). EnCQR uses an ensemble estimator, removing the requirement of exchangeability, and consists of three main steps:

1. **LOO ensemble learners**. Independent subsets are constructed on which ensemble learners are trained. For each observation i, all ensemble learners trained on subsets not including i are aggregated to make leave-one-out (LOO) predictions for i.

2. **Compute conformity scores**. The conformity scores are computed corresponding to the CQR methodology. Whereas conformity scores are computed as the maximum of the residuals in regular CQR (Equation 3.11), this can be extended by computing these conformity scores asymmetrically between the aggregated LOO predictions and the training labels, i.e:

$$E_{lo_i} = \hat{q}_{\alpha_{low}}(x_i) - y_i, \quad E_{hi_i} = y_i - \hat{q}_{\alpha_{hi}}(x_i) \tag{3.16}$$

3. **Construct PI for test data and update residuals**. These ensemble learners predict the observations in the test set, producing a set of B quantile functions for both PI boundaries. Subsequently, by aggregating the estimated quantile functions, the definite boundaries of the prediction interval are obtained as in Equation 3.17.

$$\begin{aligned} \hat{q}_{\alpha_{lo}}(x_i) &= \phi(\{\hat{q}_{\alpha_{lo}}(x_i)\}_{b=1}^{B}) \\ \hat{q}_{\alpha_{hi}}(x_i) &= \phi(\{\hat{q}_{\alpha_{hi}}(x_i)\}_{b=1}^{B}) \end{aligned} \tag{3.17}$$

Finally, these boundaries are conformalized by taking the $(1-\alpha)^{th}$ quantile of the conformity scores. For every s new observations, s being the window size, the conformity scores are updated by replacing the oldest s scores.

4 Experiments

In this chapter, the conducted experiments are described. It begins with an overview of the selected data, followed by an overview of the specific metrics used to evaluate both the base model predictions and the conformal prediction intervals. The chapter concludes with the experimental setup, describing feature engineering and implementation details. This includes elaboration on different quantile regression base models and conformal methods for the main experiment and ablation studies.

4.1 Data

4.1.1 Synthetic data

We strictly follow the synthetic data generation process used by recent works (Zaffran, Feron, Goude, Josse, & Dieuleveut, 2022). Aiming to create non-stationary time series (Figure 4.1), the data is generated according to:

$$Y_t = 10\sin(\pi X_{t,1} X_{t,2}) + 20(X_{t,3} - 0.5)^2 + 10X_{t,4} + 5X_{t,5} + 0X_{t,6} + \epsilon_t \qquad (4.1)$$

The X-data, which represents explanatory variables, is sampled from a uniform distribution and the noise is generated from an ARMA(1,1) process, i.e. $\epsilon_{t+1} = \phi\epsilon_t + \xi_{t=1} + \theta\xi_t$.

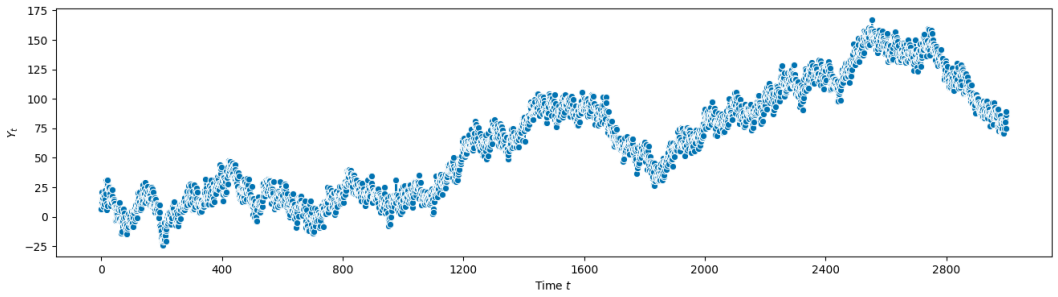


Figure 4.1: Generated synthetic time series.

4.1.2 Real-life data

It is especially relevant to examine the performance of proposed conformal methods on real-life data, since this is an empirical research study. We retrieve time series

datasets from previous research, Euro Stoxx, and Kaggle, and specifically select four datasets. These are Google stock data, power consumption data, stock index data and climate (temperature) data.

Google stock

The stock market is the first realistic time series setting chosen, where conformal prediction could be useful. As can be seen in Figure 4.2, the Google stock time series is certainly non-stationary and therefore this dataset is relevant for the experiment. It is retrieved from Kaggle[1] and contains around 3000 samples of daily stock data from 2006 to 2017. The time series has a heteroscedastic, strong upward trend, but no significant seasonality.

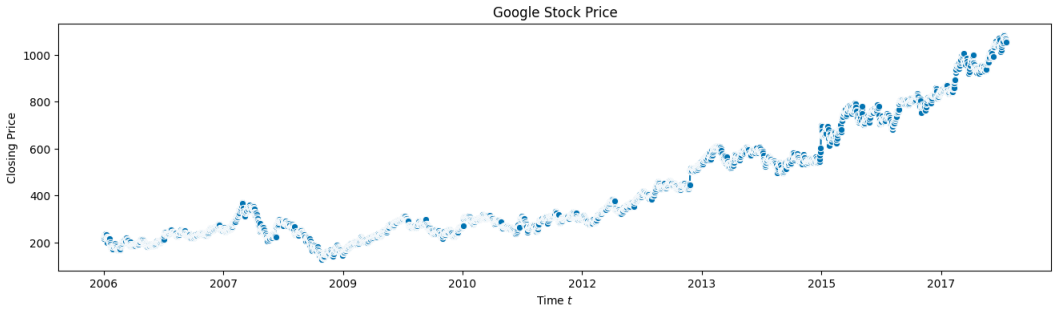


Figure 4.2: Google stock time series.

Euro Stoxx

Whereas stock options are known to be volatile, index options are expected to be more stable due to a diverse portfolio of companies. Euro Stoxx is a stock index based on the 50 companies with the highest market share in the Eurozone. The time series (Figure 4.3) is retrieved from Yahoo finance[2] and contains around 4000 samples of daily stock index data from 2007 to 2023. This time series is highly volatile and heteroscedastic.

[1] `https://www.kaggle.com/datasets/szrlee/stock-time-series-20050101-to-20171231`

[2] `https://finance.yahoo.com/quote/%5ESTOXX50E/`

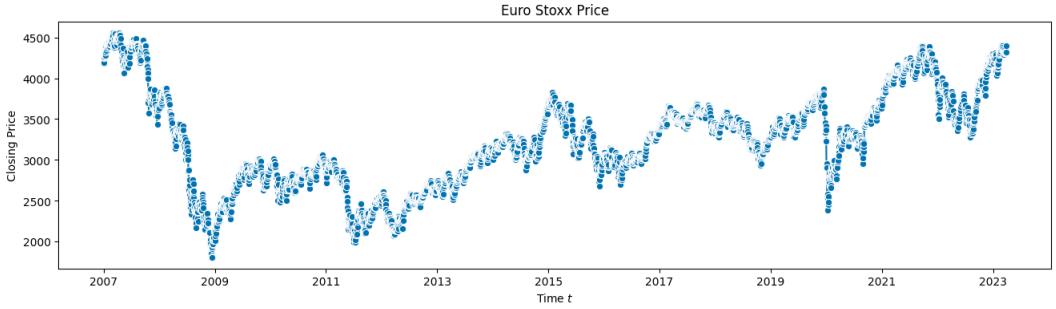


Figure 4.3: Euro Stoxx index time series.

Hackberry wind power

Wind energy generation is a very volatile process, resulting in strong variability in power generation. The data is from the Hackberry Wind Project from Texas and is retrieved from the GitHub repository[3] of Xu and Xie (2021). It contains around 14000 samples of hourly data from 2019 to 2020. The time series is extremely volatile and heteroscedastic, as can be seen in Figure 4.4.

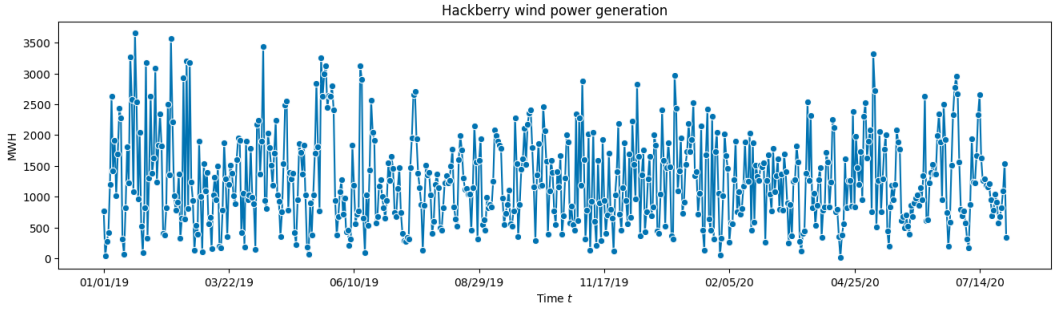


Figure 4.4: Hackberry wind power time series.

Delhi temperature

Temperature data is commonly very seasonal in nature. The time series (Figure 4.5) is retrieved from Kaggle[4] and contains around 1500 samples of daily temperature data from the city of Delhi in India, from 2013 to 2017. The time series has significant seasonality, but no strong trend.

[3] `https://github.com/hamrel-cxu/EnbPI`

[4] `https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data`

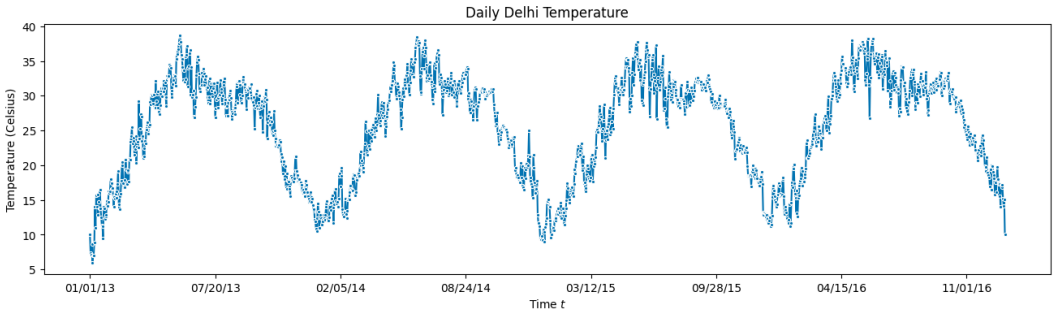


Figure 4.5: Delhi temperature time series.

4.2 Evaluation metrics

In order to check uncertainty quality and make sure that base model predictive performance is stable, base model predictions are evaluated with regards to predictive performance, i.e. point predictions. Furthermore, related to conformal uncertainty quantification, validity and efficiency can be evaluated for a prediction interval by measuring the corresponding coverage and width respectively.

4.2.1 Evaluating prediction performance

RMSE

Root mean squared error (RMSE) is a metric to quantify the prediction error of the base model. In contrast to the mean squared error (MSE), the root is measured in the same unit as the response variable, making it more interpretable. It is equivalent to

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

4.2.2 Evaluating uncertainty quality

PICP

PI coverage probability is a metric to quantify the coverage of the PIs.

$$\text{PICP} = \frac{1}{n}\sum_{i=1}^{n} c_i, \quad c_i = \begin{cases} 1, & y_i \in [L_i, U_i] \\ 0, & y_i \notin [L_i, U_i] \end{cases} \tag{4.3}$$

where L_i and U_i are the lower and upper bound of the prediction interval respectively.

PIAW

The PI average width is a metric to quantify the width of the PIs. It is equivalent to

$$\text{PIAW} = \frac{1}{n}\sum_{i=1}^{n}(U_i - L_i) \tag{4.4}$$

CWC

An optimal PI has a PICP close to the preferred coverage and a minimal prediction interval size. The coverage width-based criterion is a metric that minimizes PINAW and penalizes under and over-coverage. A higher CWC score corresponds to a better result. Note that in this thesis $\eta = 30$, strictly following Jensen et al. (2022).

$$\text{CWC} = (1 - \text{PINAW})e^{-\eta(\text{PICP}-(1-\alpha))^2}, \tag{4.5}$$

where PINAW is the PI normalized average width. This is PIAW normalized by the range of the actual data. Values greater than 1 indicate that a PI is wider than the range of the data. It is equivalent to

$$\text{PINAW} = \frac{1}{nR}\sum_{i=1}^{n}(U_i - L_i), \quad R = y_{\max} - y_{\min} \tag{4.6}$$

4.3 Experimental setup

In this section, the design choices and configurations of the main experiment are described, as well as the setup of the ablation experiments. To create conformal predictions, a model must be trained on training data. After this, the model must create predictions on the calibration set to generate a set of conformity scores, which are used to construct a prediction interval on the test set. During the experiments, 60% of the data has been used for training, 20% of the data has been used for calibration and 20% of the data has been used for testing. All results are generated using three different random seeds (seeds 100, 300, 500), which are averaged to a final result.

4.3.1 Feature engineering

Lagged features

During the experiments, a sequential setting is used. Different features can be constructed for time series. In this thesis, to create forecasts sequentially, a one-lag feature y_{t-1} has been constructed as the only feature for y_t. Using this one-lag

feature, models can be trained to predict the next data point based on the most recent observation (Figure 4.6).

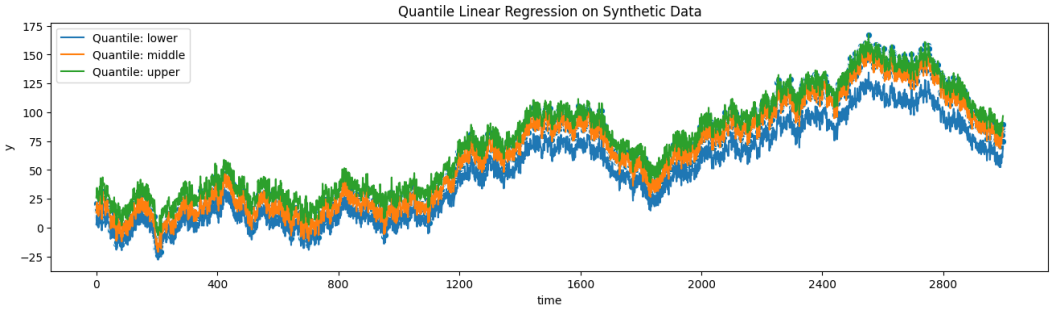


Figure 4.6: Temporal lag y_{t-1} used as a feature for forecasting y_t.

Differenced features

Random Forests cannot extrapolate outside of the range of previously observed training data on its domain. In time series regression, this results in flatlining predictions when the data stretches beyond the domain of the training data. Differencing as shown in Figure 4.7 is a standard time series technique to remove non-stationarity.

Figure 4.7: An exemplary once-differenced time series.

Once differencing is applied by computing the difference between the current value and the previous value:

$$\Delta y_t = y_t - y_{t-1} \tag{4.7}$$

This results in a possibility for the random forest to be fitted usefully. Since the setting is sequential, predicted differences are based on one-lag differenced y values. These predictions can be transformed back to estimate quantile functions by adding the difference to the previous ground truth:

$$\hat{y}_{t+1} = \hat{\Delta} y_{t+1} + y_t \tag{4.8}$$

4.3.2 Establishing the baseline

The baseline ARIMA was implemented using the `ForecasterSarimax` class from `skforecast` (Amat Rodrigo & Escobar Ortiz, 2023) which wraps around the `ARIMA` model from `pmdarima` (Smith et al., 2017). The forecaster fits the `ARIMA` model on the training data. After training, for every sequential forecasting step, the `predict_interval` function is used. This function expects α as a parameter, which is 0.1, to construct a confidence interval of $1 - \alpha$. Additionally, it expects a forecasting horizon, which is 1. Finally, after every prediction, the ground truth is concatenated to the training data and the model is refitted.

4.3.3 Implementation of base models

The quantile regressor from machine learning library `sklearn` is used as a linear model (Pedregosa et al., 2011). It needs a target quantile as a parameter, after which it can be fitted on training data. To predict multiple quantiles at once, multiple instances of the quantile regressors are combined into one model.

Additionally the library `quantile-forest` is used to implement the random forest (Johnson, 2022). The random forest expects a list of quantiles to predict as a parameter. Furthermore, a maximum depth is provided to the model, to prevent the random forest from overfitting. The random forest is trained on differenced data. Therefore the model predicts a difference for every time step and adds this to the previous ground truth.

Lastly, the quantile neural network from `quantnn` is used (Pfreundschuh, 2020). It expects a list of quantiles to predict as a parameter. Furthermore, the input dimension is defined as 1, since there is a single measurement to predict. The model uses 4 hidden layers with 256 neurons and ReLU activation functions.

4.3.4 Implementation of conformal methods

CQR

For the implementation of CQR the Python library `aws-fortuna` is used (Detommaso et al., 2023). This is an accessible library with clear documentation that shows that the library codebase follows the methodologies proposed by Romano et al. (2019). Firstly, the validation data from the training, validation and test split is defined as calibration data. The algorithm then iterates through all test samples. For every iteration, it gets the prediction intervals from the calibration and test data and creates a conformal prediction interval by taking the conformal quantile from the calibration conformity scores and adding this quantile value to the upper and lower bounds of the test data. Finally, the real-time test values are concatenated to the calibration data, so the conformity scores are extended by recent values.

ACI

As the first adaptive conformal method, ACI is implemented using the Python library `aws-fortuna` (Detommaso et al., 2023). This is an accessible library with clear documentation that shows that the library codebase follows the methodologies proposed by Gibbs and Candes (2021). The implementation is very similar to CQR. However, the miscoverage rate parameter is updated by an additional error update step. To increase the experimental simplicity, the non-weighted update is implemented.

EnbPI

The second adaptive conformal method, EnbPI, is implemented using the Python library `aws-fortuna` (Detommaso et al., 2023). This is an accessible library with clear documentation that shows that the library codebase follows the methodologies proposed by Xu and Xie (2021). Firstly, the data is bootstrapped by randomly sampling indices from the data. Subsequently, batches of training data are used to fit a bootstrap model, which predicts values on the data it is not trained on due to bootstrapping. These predictions are aggregated and together construct the conformity scores. By taking the quantile of the conformity scores and adding this as a margin to the test predictions, the conformal prediction interval is constructed.

EnCQR

Finally, code from the paper repository is used to implement EnCQR (Jensen et al., 2022). The code follows the methodologies proposed by Jensen et al. (2022) and is maintained by one of the authors. The implementation needs training data to create multiple ensemble models which make LOO predictions. These are aggregated by taking the mean, after which conformity scores are computed using

asymmetric residuals. The PI is then constructed making predictions for every test sample and aggregating these predictions. Then the quantiles of the residuals are added to the prediction intervals, to construct conformal prediction intervals. Finally, the set of conformity scores is updated, by replacing the oldest conformity score with the most recent one.

4.3.5 Sequential prediction

All models forecast sequentially, by using the previous ground truth as a feature to predict during the next time step. The linear model and quantile neural network can do this by using (Y_{lag}, Y) as training data (X, Y). However, this method cannot be used for the random forest, since the prediction in the next time step cannot be extrapolated outside of the training data. To fix this, both the target column and lagged feature are differenced, resulting in more stationary data. The random forest is trained on $(Y_{lagged\ diff}, Y_{diff})$ as (X, Y). To create sequential forecasts, at every time step, a predicted differenced value is added to the ground truth from the previous time step, as in Equation 4.8.

4.3.6 Conformalizing model predictions

To evaluate the conformal methods, the different base models and conformal prediction methods are combined. The linear model, random forest and neural network are used to estimate quantile functions. During the experiments, the desired coverage rate is fixed at 90% in line with many conformal papers (Romano et al., 2019; Gibbs & Candes, 2021; Xu & Xie, 2021; Jensen et al., 2022). Those quantile functions are then used to make predictions on the calibration set and compute conformity scores. For CQR and ACI, historical predictions can be picked, on which conformalization can be applied as a wrapper using these predictions as uncertainty estimates.

In contrast to CQR and ACI, the ensemble methods EnbPI and EnCQR, initialize multiple instances of the base model. The ensemble models create predictions inside the conformal procedure after which all ensemble predictions are aggregated into a single estimate. Finally, the conformity scores can be computed using the residuals.

4.3.7 Results visualisation

In Figure 4.8 a toy plot is presented to show how to interpret the main results. On the x-axis, the PICP score is represented, which is a measurement of how much coverage the model achieved. On the y-axis, PIAW is represented, which is

a measurement of how large the width of the prediction interval was. There are three unique marker shapes, corresponding to the three base models. Additionally, there are four different colours, representing every conformal method. The black dashed line is at 90% coverage, which is the desired coverage. Additionally, the red circle highlights the best-performing combination of prediction model and conformal method according to the selected uncertainty estimation quality metrics. This is the overall best combination, compared across all combinations of models and methods. Ideally, the method is located near the bottom of the plot and in proximity to the black dashed line, from the right-hand side.

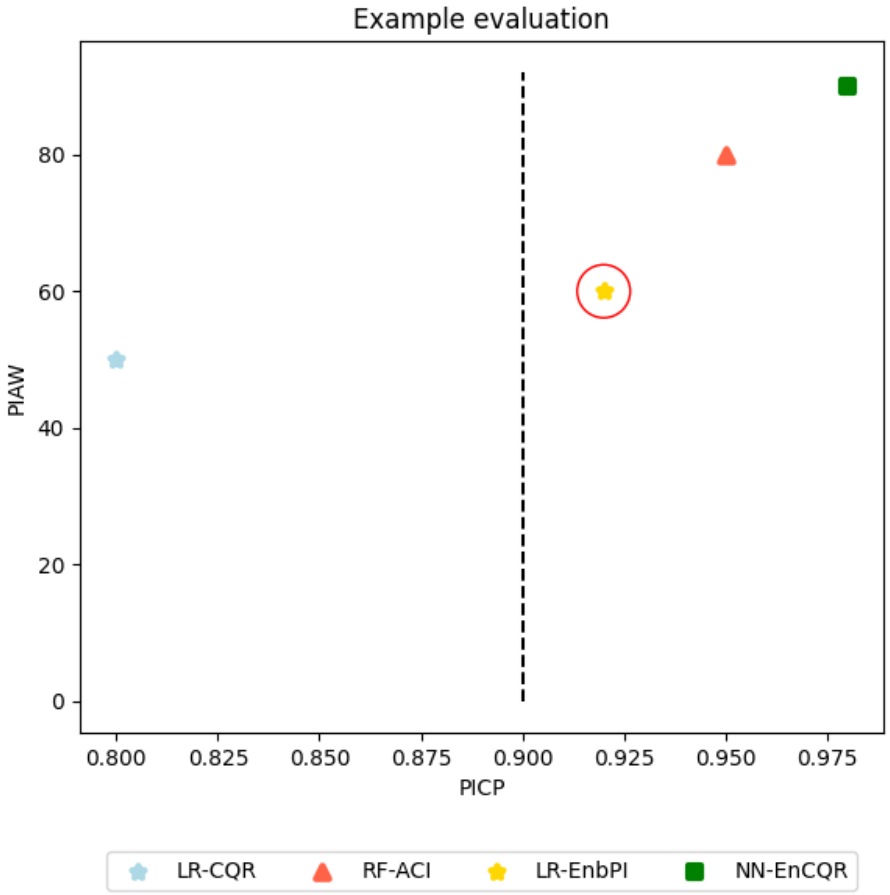


Figure 4.8: An example of the visualisation plot for the main results.

As an extension of the main experiment, a series of ablation experiments were conducted to explore the impact of different configurations on the construction of conformal prediction intervals.

4.3.8 ACI parameter optimization

As a first experiment, the ACI parameter is optimized to be closer to the preferred coverage. In the proposal paper of Gibbs and Candes (2021), parameter $\gamma = 0.005$ was used, which could be different for other methods. The experiment can be described as follows:

- **Data split.** Since the optimization should be independent of the test set, the parameter γ should not be evaluated on the actual test set. Therefore the concatenation of training and calibration data is used to create a new training, calibration and test set for optimization. The training set is used to fit the base models, the validation set is used to calibrate the conformal methods and the test set to evaluate the coverage.
- **Base model predictions.** For all three base models, predictions are made on the new test set. Those can be evaluated using grid search optimization.
- **Grid search.** Between the range [0,1] with step size 0.002 there are 500 values in the parameter grid to search for. The grid search is performed for every base model.

4.3.9 Quantile tuning

In the second experiment, an analysis of different quantile functions was conducted to examine the impact of base model predictions on the construction of conformal prediction intervals. Conformal prediction operates with a guarantee embedded within the computation of the conformity scores. Consequently, the prediction intervals, which correspond to the estimated quantile functions, can be updated while preserving this guarantee as these are nominal quantiles and not conformal quantiles. A range of upper and lower quantiles are examined in this analysis. The lower bound quantiles varied from 10 to 40, while the upper bound quantiles ranged from 60 to 90, with both bounds adjusted in increments of 5.

4.3.10 Multi-step forecasting

In the third experiment, the initial sequential setup is replaced by a multi-step forecasting setup. At first, this modified approach makes a prediction based on the previous ground truth. For the next k steps, the forecasts are made based on the last predicted value. Thus, the lagged feature is not based on the ground truth but based on predictions. After k steps, the process is repeated and another prediction based on the ground truth is made.

The average accuracy is expected to decrease as a consequence of the multi-step forecasting setup, due to error accumulation over multiple forecasting steps caused by centering future step predictions around predicted values. Consequently, quantile estimates would exhibit higher errors, resulting in wider conformal prediction intervals.

5 | Results

This chapter presents an analysis of the outcomes obtained from the main experiment as well as the conducted ablation experiments. Furthermore, the insights from the figures are described. A complete overview of the main results and ablation results can be found in Appendix B and Appendix C respectively.

5.1 Main experiment

Base models

As can be seen in Table B.2, the baseline ARIMA produces the smallest error for the synthetic and temperature dataset. The RMSE scores of the neural network are higher on all real-life datasets. On the power dataset, the relative difference between the neural network and the other base models is significantly larger. On the Google and Euro Stoxx datasets, base model scores are similar, except for the neural network.

Prediction intervals

It is illustrated in Figure 5.1 that on the synthetic dataset, 90% coverage is achieved for all methods except for CQR on linear regression (LR). The highlighted best-performing method is CQR on random forest (RF). Although the method has similar interval width, it is in greater proximity to the 90% marker than the cluster of EnbPI models. Notably, ACI and EnCQR achieve close to 100% coverage, having strictly wider intervals than the CQR and EnbPI methods. Furthermore, EnCQR produces significantly wider intervals than ACI, which are scattered across base models as well. As demonstrated in Table B.1, baseline ARIMA shows better performance than all conformal methods in terms of uncertainty quality for the synthetic and temperature data.

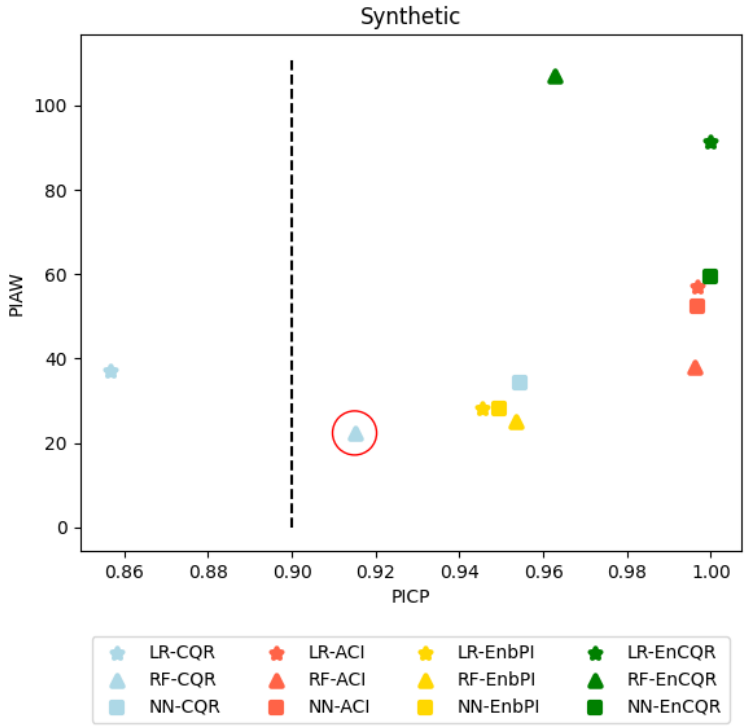


Figure 5.1: The main results on the synthetic data.

Figure 5.2 demonstrates that on the temperature dataset, all methods achieve 90% coverage, except for CQR on RF. The best-performing method is CQR on LR. Although having a similar width, this method is slightly closer to the 90% marker than the EnbPI methods, which are clustered between 0.92 and 0.93. ACI and EnCQR achieve close to 100% coverage, having prediction intervals around three and five times wider respectively than the CQR and EnbPI methods.

The observations on the power dataset (Figure 5.2) are very different, on which strict and significant under-coverage by the EnCQR method is demonstrated, even though 2 out of 3 models construct prediction intervals with similar widths compared to the other conformal methods. The best-performing method is CQR on LR. Additionally, the linear regressors consistently produce smaller intervals compared to the other base models. Furthermore, there is variation in performance across other conformal methods on the right of the 90% marker.

The observations on the Google dataset (Figure 5.2) show significant under-coverage for the CQR and EnbPI methods. However, CQR on the neural network shows the best performance across all methods. ACI and EnCQR achieve close to 100% coverage, producing prediction intervals around four and five times wider respectively.

On the Euro Stoxx dataset (Figure 5.2), CQR strictly under-covers. EnbPI on linear regression performs best here. However, the random forest and neural network show performance that is very similar. ACI and EnCQR achieve close to 100% coverage. EnCQR produces significantly wider intervals than ACI, which PIAW scores are scattered across base models as well.

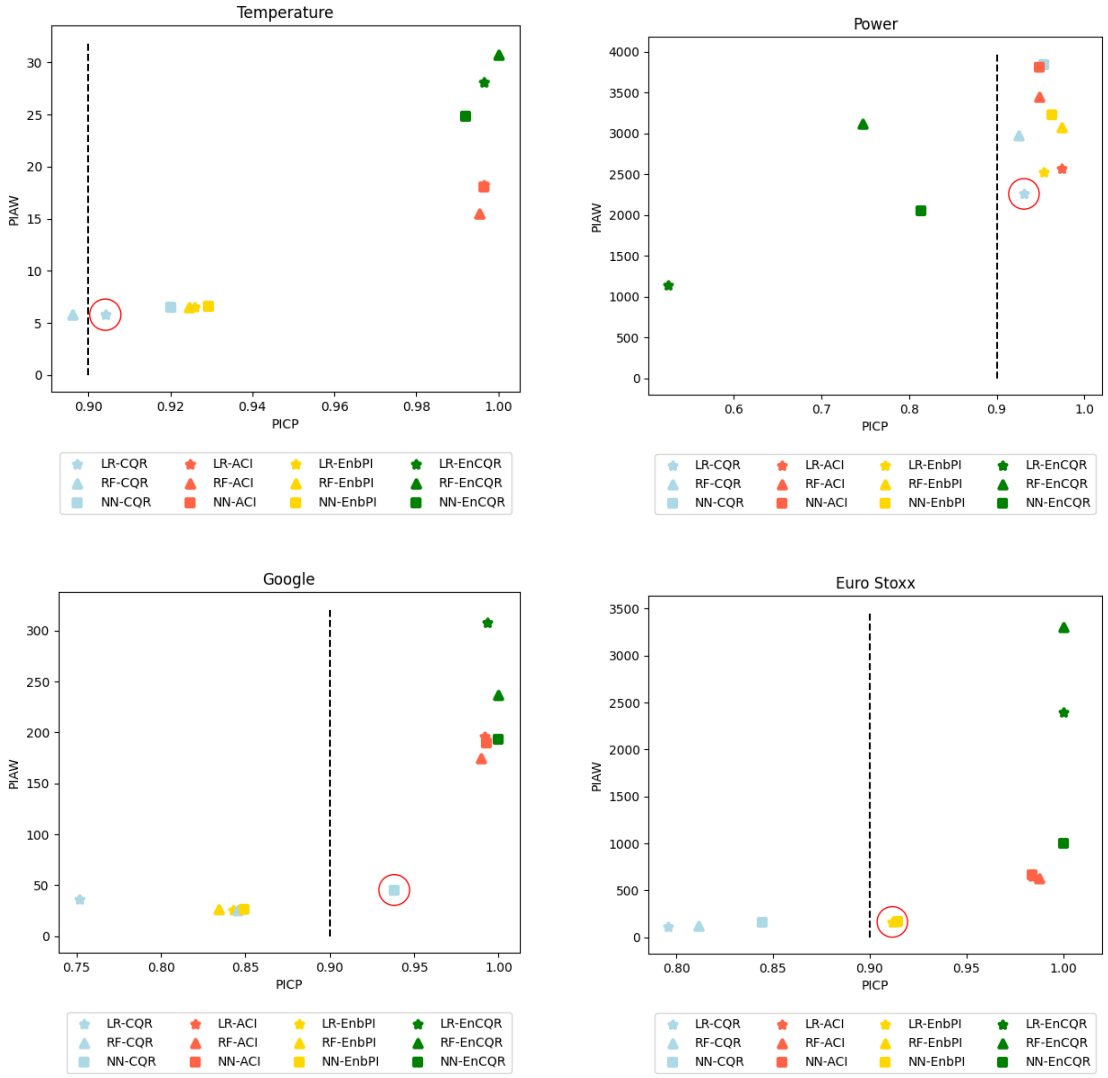


Figure 5.2: The main results on the real-life data.

The overall results in Figure 5.1 and Figure 5.2 illustrate that ACI achieves $95^{+}\%$

coverage for all datasets. EnCQR also achieves this for all datasets except for the power datasets. Thus, both ACI and EnCQR exhibit a high level of over-coverage. Additionally both ACI and EnCQR exhibit wider intervals compared to the CQR and EnbPI methods. EnCQR demonstrates an exceptionally wide interval, often constructing intervals up to five times the size of intervals constructed by the other approaches.

5.2 Ablation experiments

In this section, the results of the ablation experiments used to examine possible improvements for ACI are described.

5.2.1 Ablation experiment 1: ACI parameter optimization

As demonstrated in the upper two plots of Figure 5.3 and additionally in Figure C.1, as the value of γ increases for the synthetic data, the ACI coverage becomes closer to the desired level of 90%. Specifically, for the linear regression and random forest models, the optimal γ is observed to be close to 1. When γ is close to 1 implies a strong update based on the previous error, making the method more adaptive to observed distribution shifts, also inducing greater volatility in the value of α_t (Gibbs & Candes, 2021). Furthermore, the prediction interval width also decreases, as the value of γ increases. This effect is observed on the synthetic dataset, whereas on the Google dataset, the neural network is the only model showing a similar effect (Figure C.2). The complete results can be found in Appendix C.

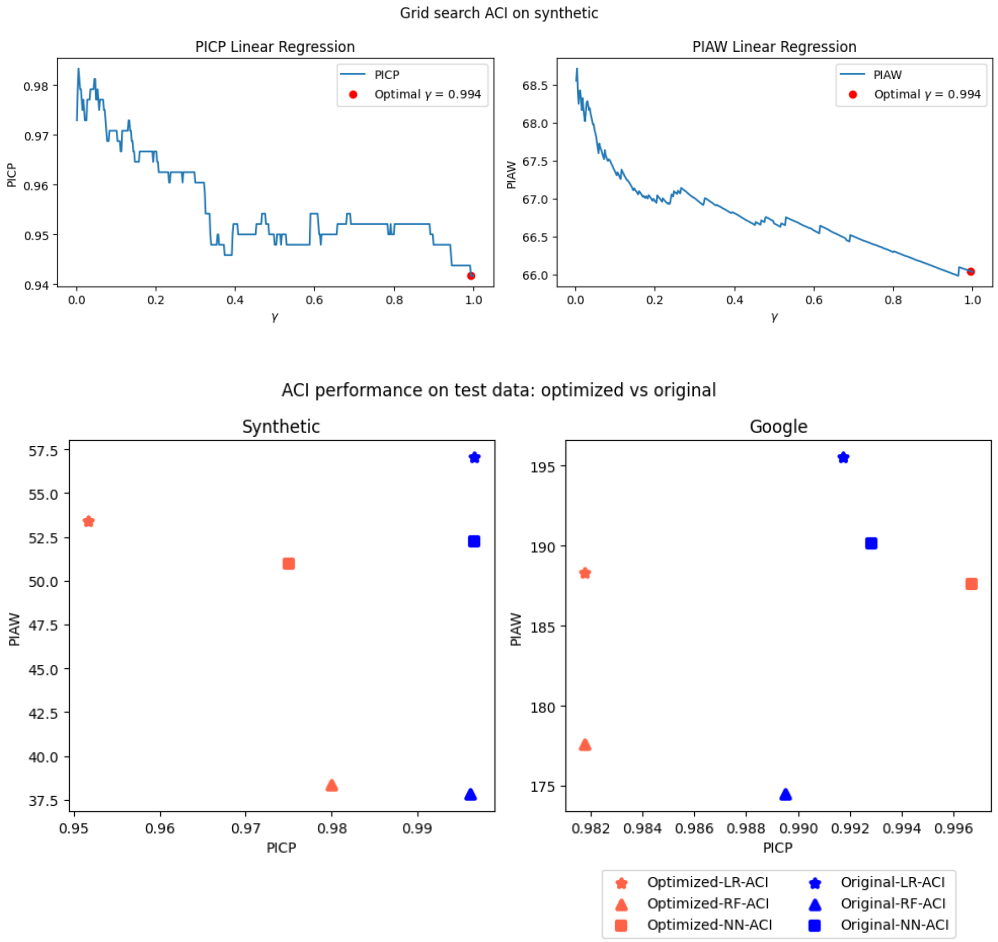


Figure 5.3: A visualisation of the ACI grid search and the performance of ACI on the test set afterwards.

5.2.2 Ablation experiment 2: Quantile tuning

Figure 5.4 shows that as the quantile interval size increases, in general, the prediction interval width increases as well for the linear regressor and neural network. For the random forest, however, the PI size decreases. The neural network shows a significant decrease in width for the large quantile interval. For all models, there are minimal differences in coverage across quantile intervals.

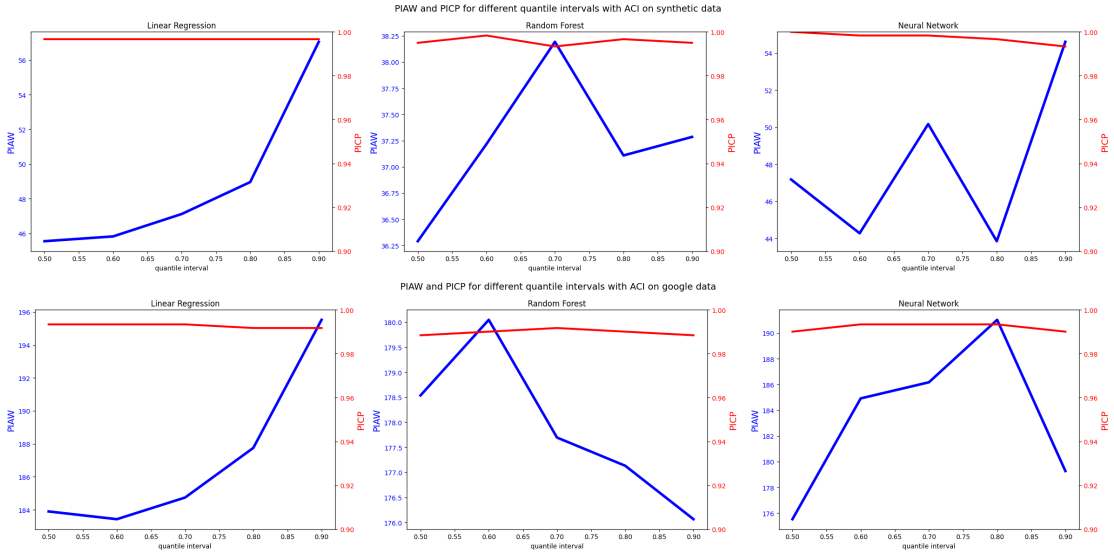


Figure 5.4: Quantile tuning on synthetic and Google data.

5.2.3 Ablation experiment 3: Multi-step forecasting

Figure 5.5 demonstrates that a larger step in multi-step forecasting causes the error of the prediction to increase, as well as the prediction interval width. However, for ACI the PI width is not consistently increasing.

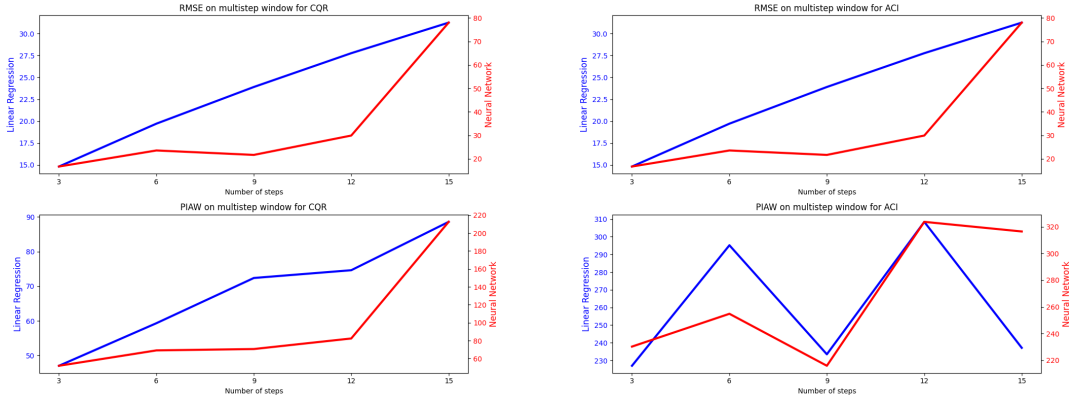


Figure 5.5: RMSE and PIAW scores on Google data with different multi-step prediction step sizes.

6 | Discussion

In this chapter, an interpretation and analysis of the obtained results are presented, to address the research questions based on the empirical experiments conducted. The research objectives of this thesis involved benchmarking of recently proposed conformal methods designed for time series forecasting. Their effectiveness in achieving empirical coverage has been evaluated and compared in terms of performance, coverage, mean prediction interval (PI) width and CWC.

The main experiment revealed that the ACI and EnCQR methods achieved the desired guaranteed coverage. Nevertheless, these methods are over-covering and constructing relatively large prediction intervals. Large prediction intervals are uninformative and over-coverage is not preferred. Through various ablation experiments, the impact of different factors on the interval width and coverage has been examined.

Specifically, the ACI parameter γ was optimized to be closer to the preferred coverage level of 90%. In general, this optimization led to a small reduction in over-coverage on the test data. This was not a consistent trend for the neural network, which also exhibited a small increase in over-coverage on the Google dataset. The γ close to 1 implies a strong update based on the previous error. This is somewhat intuitive, since there is coverage and over-coverage can be compensated for using this strong gamma. On the other hand, the prediction interval width would significantly increase when there is no coverage. In this case, the intuition behind the α and γ is lost. This work did not investigate the behaviour of α_t when a γ close to 1 is used. Since this would provide more intuition behind the workings of ACI, examining this is justified.

The second ablation experiment in which different quantiles were predicted by the base models, revealed a decrease in prediction interval width for smaller predicted quantile intervals. This outcome aligns with expectations, given that the quantile prediction does not impact coverage due to the conformalization occurring afterwards. Lastly, the sequential setup was replaced by a multi-step setup, which resulted in a significant increase in both RMSE and prediction interval average width. This is as expected, since a larger step size results in higher errors that accumulate over multiple forecasting steps.

The cause behind ACI's substantial over-coverage and wide prediction intervals is not clear. The ablation experiments did not mitigate the issues with ACI.

The parameter optimization did not have a significant impact on mitigating over-coverage and wide prediction intervals. As expected, the experiments with predicted quantile intervals and prediction step sizes indicated that predictions closer to the ground truth result in smaller intervals and suggest that conformalization does not depend on those.

In terms of coverage and mean prediction interval width, both CQR and EnbPI have demonstrated superior performance compared to ACI and EnCQR. Both CQR and EnbPI achieve coverage close to the 90% target and maintain a relatively small prediction interval. CQR generally outperformed EnbPI in terms of coverage and PIAW by a slight margin. On the other hand, for the Euro Stoxx dataset, CQR exhibited under-coverage, whereas EnbPI did not. Contrarily, EnbPI provided the desired coverage for all datasets except for the Google dataset, where it shows slight under-coverage. This discrepancy might be attributed to the less stable nature of the Google time series in the test set, combined with a strong upward trend towards the end of the series. Other factors such as the number of ensembles and data splits could also contribute to this under-coverage. Even though EnbPI and EnCQR are similar ensemble methods, EnbPI outperformed EnCQR. EnbPI constructs ensemble methods using bootstrapping and EnCQR does not, which could be a possible explanation for this observation. Although CQR outperformed the other conformal methods, it lacks a theoretical conformal guarantee. Hence, the conformal method that demonstrated the best performance, while preserving this theoretical guarantee, is EnbPI.

Considering the empirical nature of the experiments conducted in this thesis, there is a potential bias introduced by the choice of data. Even though data from different sources with varying distributions have been collected to counter this, there remains an incentive to examine CQR's exceptional performance. Furthermore, the sequential setting used in the experiments, causes the predictions to be close to the ground truth. For a sequential setting, the data may not fluctuate enough to need adaptive conformal methods. Finally, this thesis used a 90% coverage objective. Adaptive conformal methods may perform differently at higher coverage objectives. Hence, exploring the performance of these conformal methods at different coverage rates is justified.

7 | Conclusion

In conclusion, this thesis investigated the effectiveness and performance of recently proposed adaptive conformal methods. These conformal methods were applied to various base model predictions to construct conformal prediction intervals, after which the conformal prediction intervals were evaluated on uncertainty quality. Our main experiments suggest that Conformalized Quantile Regression stands out as the best-performing method, based on empirical evaluation. However, CQR lacks a theoretical guarantee under scenarios of potential distribution shift, as observed in time series data. Among all the adaptive methods explored in this thesis, Ensemble Batch Prediction Intervals consistently demonstrated the best overall performance. For future research, it is recommended to explore the application of these adaptive conformal methods on additional datasets to further validate their effectiveness and generalizability. Furthermore, investigating alternative prediction setups, while remaining theoretically valid, and varying coverage objectives would contribute to a more comprehensive understanding of the adaptive methods' capabilities in time series. Ultimately, by empirically investigating adaptive conformal methods, this work not only contributes to the collective understanding of such methods, but also promotes better decision-making through uncertainty quantification and improved reliability of predictions.

References

Amat Rodrigo, J., & Escobar Ortiz, J. (2023, 5). *skforecast.*

Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *CoRR, abs/2107.07511*. Retrieved from https://arxiv.org/abs/2107.07511

Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2023). *Conformal prediction beyond exchangeability.*

Date, S. (2021, Sep). *How to isolate trend, seasonality and noise from a time series.* Retrieved 2023-06-25, from https://timeseriesreasoning.com/contents/time-series-decomposition/?ref=timescale.com

Detommaso, G., Gasparin, A., Donini, M., Seeger, M., Wilson, A. G., & Archambeau, C. (2023). Fortuna: A library for uncertainty quantification in deep learning. *arXiv preprint arXiv:2302.04019*.

Dye, S. (2020, Feb). *Quantile regression.* Towards Data Science. Retrieved 2023-06-25, from https://towardsdatascience.com/quantile-regression-ff2343c4a03

Gibbs, I., & Candes, E. (2021). Adaptive conformal inference under distribution shift. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 1660–1672). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2021/file/0d441de75945e5acbc865406fc9a2559-Paper.pdf

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press. (http://www.deeplearningbook.org)

Hamilton, J. D. (1994). *Time series analysis.* Princeton University Press. Retrieved from https://www.worldcat.org/title/time-series-analysis/oclc/1194970663&referer=brief_results

Hao, L., & Naiman, D. (2007). *Quantile regression.* doi: 10.4135/9781412985550

Hupont, I., Micheli, M., Delipetrev, B., Gómez, E., & Garrido, J. S. (2023). Documenting high-risk ai: A european regulatory perspective. *Computer*, *56*(5), 18-27. doi: 10.1109/MC.2023.3235712

Hyndman, R., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). Australia: OTexts.

Jensen, V., Bianchi, F. M., & Anfinsen, S. N. (2022). Ensemble conformalized quantile regression for probabilistic time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 1-12. doi: 10.1109/TNNLS.2022.3217694

Johnson, R. (2022). Retrieved 2023-05-01, from https://github.com/zillow/quantile-forest

Jose, J. (2022, 08). Introduction to time series analysis and its applications.

Karmakar, R., Chatterjee, S., Das, A., & Mandal, A. (2023, 05). Bcpuml: Breast cancer prediction using machine learning approach—a performance analysis. *SN Computer Science*, *4*. doi: 10.1007/s42979-023-01825-x

Koenker, R., & Bassett, G. (1978). Regression quantiles. *Econometrica*, *46*(1), 33–50. Retrieved 2023-05-09, from http://www.jstor.org/stable/1913643

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., & Király, F. J. (2019). *sktime: A unified interface for machine learning with time series.*

Löning, M., Király, F., Bagnall, T., Middlehurst, M., Ganesh, S., Oastler, G., ... rice, B. (2022, September). *sktime/sktime: v0.13.4.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.7117735 doi: 10.5281/zenodo.7117735

Meinshausen, N. (2006, 06). Quantile regression forests. *Journal of Machine Learning Research*, *7*, 983-999.

O'Neill, B. (2009). Exchangeability, correlation, and bayes' effect. *International Statistical Review*, *77*(2), 241-250. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2008.00059.x doi: https://doi.org/10.1111/j.1751-5823.2008.00059.x

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pfreundschuh, S. (2020). *Quantile regression neural networks.* https://github.com/simonpf/quantnn/blob/main/docs/source/qrnn.rst. GitHub.

Romano, Y., Patterson, E., & Candes, E. (2019). Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf

Shumway, R., & Stoffer, D. (2011). *Time series analysis and its applications with r examples* (Vol. 9). doi: 10.1007/978-1-4419-7865-3

Siddiqui, F., & Albergotti, R. (2022, Feb). *"full self-driving" clips show owners of teslas fighting for control, and experts see deep flaws.* WP Company. Retrieved 2023-06-25, from https://www.washingtonpost.com/technology/2022/02/10/video-tesla-full-self-driving-beta/

Smith, T. G., et al. (2017). *pmdarima: Arima estimators for Python.* Retrieved from http://www.alkaline-ml.com/pmdarima

Vovk, V., Gammerman, A., & Shafer, G. (2005, 01). Algorithmic learning in a

random world.. doi: 10.1007/b106715

Xu, C., & Xie, Y. (2021, 18–24 Jul). Conformal prediction interval for dynamic time-series. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (Vol. 139, pp. 11559–11569). PMLR. Retrieved from `https://proceedings.mlr.press/v139/xu21h.html`

Xu, C., & Xie, Y. (2023). *Sequential predictive conformal inference for time series.*

Zaffran, M., Feron, O., Goude, Y., Josse, J., & Dieuleveut, A. (2022, 17–23 Jul). Adaptive conformal predictions for time series. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (Vol. 162, pp. 25834–25866). PMLR. Retrieved from `https://proceedings.mlr.press/v162/zaffran22a.html`

Appendices

A | Naming conventions

Table A.1: Naming conventions

Symbol	Description
$\mathbb{P}$	The probability function
$\inf S$	The infimum of some set S
$\mathbb{R}$	The real numbers
$\mathbb{1}_S$	The indicator function for some set S
$\mathbb{E}$	The expected value
$\hat{\mathcal{C}}$	The non-conformalized prediction interval
$\mathcal{C}$	The conformal prediction interval
$F(\cdot)$	The CDF of the data
α	The miscoverage rate

B | Complete evaluation results

Table B.1: PICP, PIAW, CWC scores for CQR, ACI, EnbPI, EnCQR for different base models and different datasets. Marked bold is the overall best performer and underlined is the overall best performing conformal method for time series.

		ARIMA			Linear model			Random forest			Neural network		
	Model	PICP	PIAW	CWC	PICP	PIAW	CWC	PICP	PIAW	CWC	PICP	PIAW	CWC
synthetic	CQR				0.857	36.973	0.584	0.915	22.338	0.764	0.954	34.176	0.592
	ACI	**0.915**	**19.583**	**0.792**	0.997	57.049	0.310	0.996	37.815	0.462	0.997	52.266	0.347
	EnbPI				0.945	28.207	0.666	0.953	24.956	0.681	0.949	28.181	0.659
	EnCQR				1.000	91.289	0.042	0.963	106.966	−0.093	1.000	59.392	0.286
temperature	CQR				0.904	5.781	0.795	0.896	5.801	0.794	0.920	6.491	0.760
	ACI	**0.901**	**5.289**	**0.813**	0.997	18.177	0.270	0.995	15.530	0.343	0.997	18.053	0.273
	EnbPI				0.926	6.526	0.754	0.925	6.527	0.755	0.929	6.559	0.748
	EnCQR				0.997	28.073	0.05	1.000	30.699	−0.064	0.992	24.838	0.096
power	CQR				**0.931**	**2258.008**	**0.308**	0.925	2977.389	0.096	0.954	3846.110	−0.150
	ACI	0.940	2393.434	0.263	0.974	2564.369	0.190	0.948	3448.569	−0.0415	0.948	3803.984	−0.140
	EnbPI				0.954	2528.840	0.215	0.974	3070.833	0.060	0.963	3225.429	0.021
	EnCQR				0.526	1134.019	0.010	0.747	3115.942	0.025	0.813	2046.489	0.286
google	CQR				0.752	36.398	0.477	0.845	25.820	0.864	**0.938**	**45.344**	**0.864**
	ACI	0.786	21.606	0.648	0.992	195.522	0.455	0.990	174.469	0.496	0.993	190.114	0.462
	EnbPI				0.843	25.758	0.857	0.834	26.208	0.830	0.849	26.030	0.875
	EnCQR				0.993	307.685	0.269	1.000	236.790	0.370	1.000	193.547	0.437
Euro Stoxx	CQR				0.796	114.689	0.682	0.811	124.586	0.741	0.844	156.660	0.841
	ACI	0.869	141.374	0.904	0.984	644.673	0.552	0.988	630.999	0.547	0.984	665.611	0.543
	EnbPI				**0.912**	**163.831**	**0.915**	0.913	166.372	0.913	0.915	168.458	0.911
	EnCQR				1.000	2393.405	−0.136	1.000	3302.720	−0.469	1.000	995.958	0.376

For the power dataset, both EnbPI and EnCQR are underlined. EnCQR has a higher CWC score, but does not achieve 90% coverage, whereas EnbPI does achieve 90%, but has a lower CWC score. This is slightly ambiguous and therefore both are underlined.

Table B.2: RMSE scores for different base models and different datasets.

Dataset	ARIMA	Linear model	Random forest	Neural network
synthetic	5.860	8.046	6.416	7.039
temperature	1.606	1.686	1.698	1.716
power	663.340	636.302	711.003	802.298
google	9.864	9.990	9.841	11.615
Euro Stoxx	49.230	49.048	48.859	52.886

C | Ablation experiment 1: ACI parameter optimization

In Figure C.1 and Figure C.2 it is illustrated that in general higher values for γ result in coverage closer to the desired level of 90% and more narrow prediction intervals. The only exception to this is linear regression on the google data (Figure C.2), for which $\gamma = 0.002$ is an outlier.

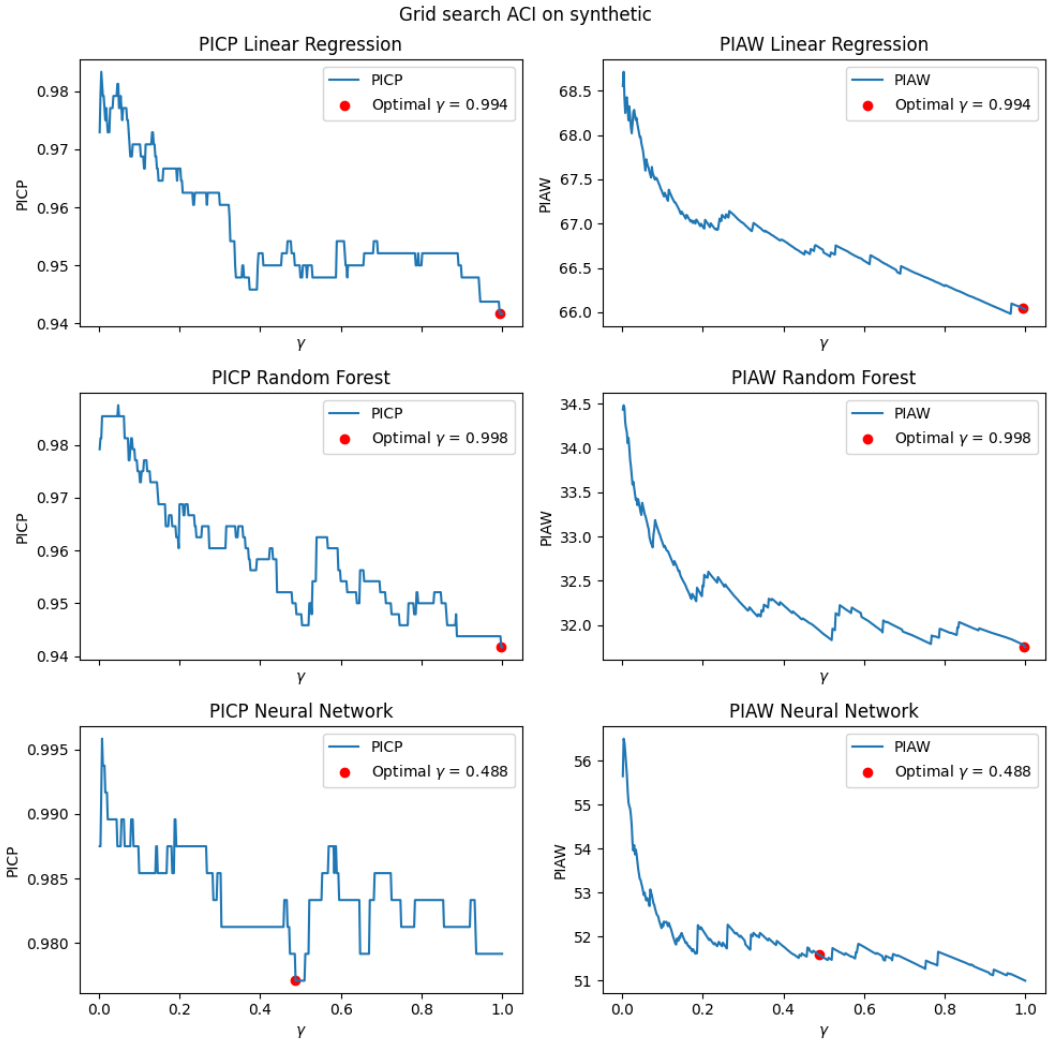


Figure C.1: ACI parameter optimization on synthetic data

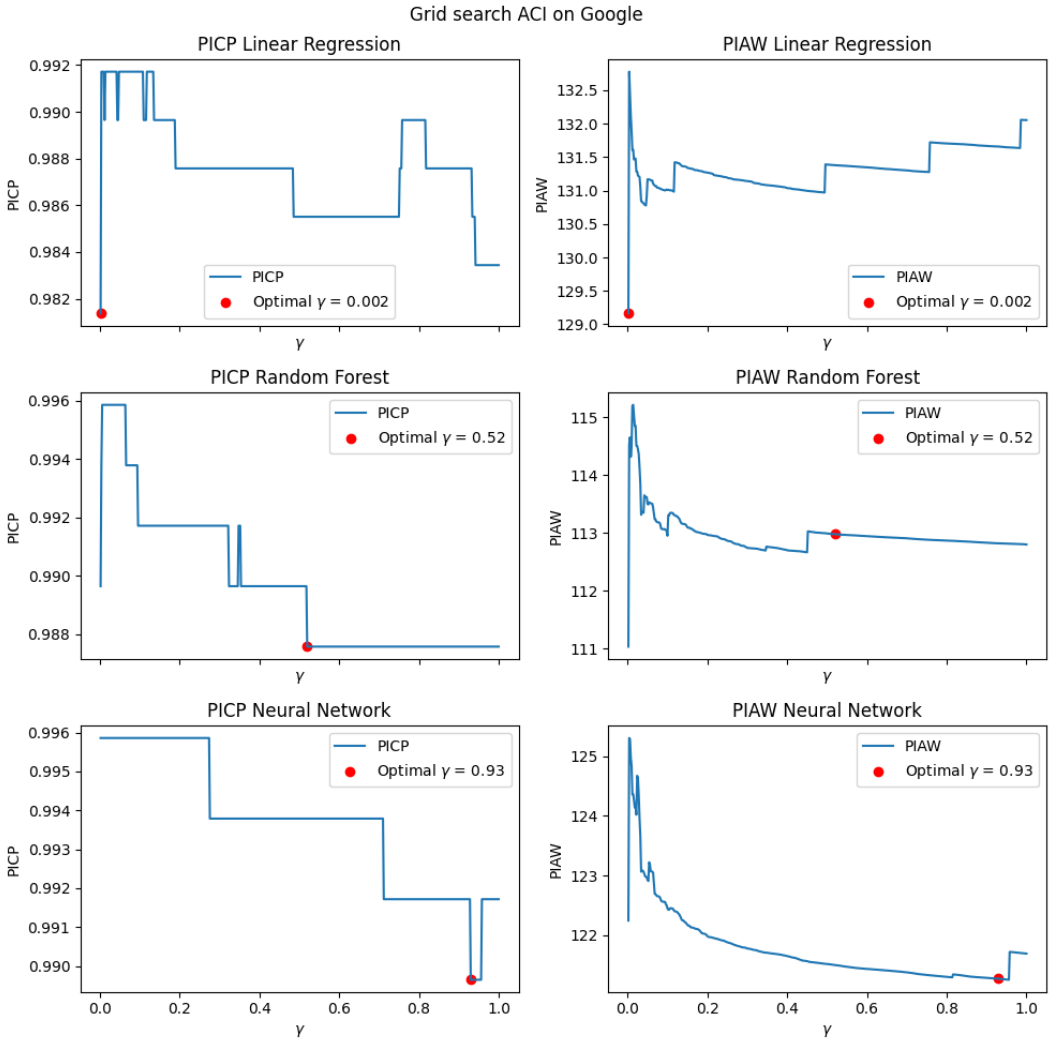


Figure C.2: ACI parameter optimization on google data